Datalogisk Institut
KØBENHAVNS UNIVERSITET

Proceedings fra den 14. Danske Konference i Mønstergenkendelse og Billedanalyse

Eds. Søren I. Olsen

Technical Report no. 2005/06
ISSN: 0107-8283

DIKU
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Indholdsfortegnelse:

Real-time Illumination Estimation from Image Sequences

Claus B. Madsen, Moritz Störring,
Tommy Jensen, Mikkel S. Andersen, and Morten F. Christensen
Laboratory of Computer Vision and Media Technology
Aalborg University, Aalborg, Denmark
cbm/mst@cvmt.aau.dk
www.cospe.dk

Abstract

Knowledge about the illumination conditions in a real world scene has many applications among them Augmented Reality which aims at placing virtual objects in the real world. An important factor for convincing augmentations is to use the illumination of the real world when rendering the virtual objects so they are shaded consistently and cast consistent shadows.

This paper proposes two approaches to continuously estimate the illumination conditions in a static outdoor scene based on images from a single viewpoint of that scene while using the scene itself as light probe. Thus, no additional calibration objects are required. Experimental results show that the proposed illumination estimation is sufficient for Augmented Reality applications.

1 Introduction

Images are formed as a result of light interacting with surfaces. The radiation emitted by a light source hits a material's surface under a certain angle where it is then reflected, absorbed, and transmitted depending on the material's properties. The reflected light may hit other objects causing interreflections, and one object may occlude another object's reflections or a light source resulting in shadowing. When images are synthesized using computer graphics techniques it is important to have good models of these interactions in order to achieve realism. Similarly, when the images are real images acquired with some form of camera it is paramount to understand how the image was formed in order to analyze it using computer vision techniques. Generally, three different elements come together in forming images: 1) the 3D geometry of the scene, 2) the reflectance properties of the surfaces in the scene, and 3) the illumination conditions in the scene. Given a model of all three elements it is possible both to render synthetic images and to design robust computer vision techniques for analyzing images of the scene.

The Laboratory of Computer Vision and Media Technology at Aalborg University, Denmark (CVMT/AAU) has recently initiated a research project (CoSPE: Computer Vision-Based Scene Parameter Estimation) which lies on the border between computer vision and computer graphics. The project focuses on estimating the reflectance properties and the illumination conditions in scenes based on images. For more information about the project please visit the project's web-site www.cospe.dk

In this paper we present some initial results of this research, namely two approaches to the same problem: to continuously estimate the illumination conditions in a static scene based on a sequence of images from a single viewpoint. So far, the most commonly used approach to scene illumination measurement/estimation has been the so-called light probe, which is a reflective sphere placed in the scene and photographed with a camera to get an omni-directional measurement of light, [5, 8, 10]. None of the approaches presented in this paper require any special purpose radiometric calibration objects to be present in the scene. In fact one could say we are proposing techniques that allow the scene to act as its own light probe.

Real-time, continuous estimation of scene illumination conditions is really important for Augmented Reality (AR) systems. Figure 1 shows an example of an AR system where a virtual object has been rendered into a real scene. The virtual object is rendered with illumination conditions corresponding to the illumination condition that are estimated for the real scene, so the virtual object is shaded consistently with the scene, and it also casts a consistent shadow on surfaces in the real scene.

The application scenario we are targeting is a system, be it an AR or a vision system, which needs to continuously update its internal model of the illumination in an outdoor scenario. Consider for example a computer screen mounted on a pole at an archaeological site allowing visitors to view the real scene (filmed with a video camera) augmented with visualization of virtual 3D buildings that no longer exist. For such a system the illumination conditions constantly change due to the passing of time causing the sun to travel across the sky, clouds causing partial or complete blockage of the direct light from the sun, and changing the illumination from the sky.

The approaches presented in this paper can estimate the intensities and color of the direct sunlight and of the

Figure 1: Two images taken at different times (approximately one hour apart) on a sunny day with partial cloud cover causing constant changes in the illumination conditions. With one of the methods proposed in this paper we have automatically estimated the current illumination conditions and used this illumination estimate to render a virtual sculpture into the scene.

indirect skylight. Additionally one method is able to estimate the direction of the sunlight relative to the scene, whereas the other technique assumes that the system knows the direction of the sunlight using date, time and position information. The latter approach is more robust for cloudy conditions, whereas the former is more readily applicable to a scene as there is less positional and orientational calibration to carry out.

Both approaches involve an "off-line" photometric calibration phase where the reflectances (albedo) of diffuse surfaces in the scene are estimated. After the once-only reflectance calibration the approaches enable continuous "on-line" illumination estimation.

This paper is organized as follows. In section 2 we give an overview of related work. Section 3 then lists the assumptions behind the presented techniques, and presents the illumination model used by both approaches (both approaches estimate the values of parameters in this model). In section 4 we then present the approach which assumes availability of sunlight direction information, whereas the approach which also estimates sunlight direction is presented in section 5. Conclusions are given in section 6.

2 State-of-the-art

Estimating scene illumination conditions from images is the dual problem of estimating surface reflectance properties, because the image represents light reflected off surfaces, and this reflection is governed by the illumination and the reflectances. Therefore illumination estimation cannot be performed without knowledge of surface reflectance. This is the reason all related work is based on placing some kind of special purpose object with a priori known reflectance properties in the scene. For continuously operating AR or vision systems performing illumination estimation it is not a viable approach to be forced to have calibration objects in the scene. Therefore we have developed and tested two approaches to estimate dynamic illumination conditions based on the surfaces naturally present in the scene. Subsequently we briefly describe some of the most closely related work. Recent surveys on illumination estimation may be found in [9, 13].

One group of related work has a somewhat different focus, namely that of estimating scene reflectances. Yu and Malik proposed estimation of pseudo BRDFs[1] for outdoor scenes, [17]. The approach requires multiple images of the outdoor scene taken from different viewpoints and under differing illumination conditions. The goal is to be able to re-render the scene under arbitrary novel illumination conditions. Knowledge of scene illumination is obtained by combining a parameterized outdoor skylight model with light probe images.

Yu and Debevec proposed an inverse global illumination rendering approach, [16]. By using multiple images of all surfaces and a complete 3D model of an entire indoor scenario, and by using knowledge of the illumination conditions they demonstrate that it is possible to estimate glossy BRDFs for all surfaces. Knowledge of illumination conditions is obtained by manually measuring the positions and emittances of all light sources.

Loscos and Drettakis proposed a system for interactive re-lighting of indoor scenarios, [11]. Using a single image of the scene, combined with a complete 3D model of the entire room, and knowledge of the original illumination conditions they are able to re-render the scene under arbitrary novel illumination conditions. The knowledge of the original scene illumination is obtained by manual measurement of the positions and emittances of light sources.

Boivin and Gagalowicz proposes an iterative global

[1]BRDF: Bi-directional Reflectance Distribution Function is defined as the ratio of the reflected radiation to the incident radiation on a surface.

illumination approach to estimating surface reflectance parameters, [3, 4]. This work is also based on a single image of an indoor scene, and assumes that the positions and emittances of the light sources are measured manually.

Masselus and Dutre proposed an approach to image-based modeling of surface reflectances with the aim of being able to re-render under novel illumination conditions, [12]. By acquiring multiple images from a single viewpoint of a scene illuminated with a manually moved single light source they were able to model the reflectance field for re-lighting. The location of the moving light source is computed for each image by a triangulation technique based on the shading of four diffuse spheres present at known locations in the scene.

Sato and Sato proposed a technique for estimation of complex illumination environments, [15]. The technique requires that a known object is casting shadows on a surface in the scene, and the reflectance of the shadow receiver must be known. If this information is not available the method requires an image of the scene without the shadow casting object. In this case the method cannot be applied to scenes with changing illumination conditions.

Kanbara and Yokoya designed an approach to automatic, real-time estimation of scene lighting for augmented reality, [10]. The approach involves placing a reflective sphere which is always in the camera's field of view. The dynamic scene illumination conditions are estimated from the environment's reflection in this special purpose sphere.

Using reflective spheres has for several years been the standard approach to acquiring omni-directional knowledge of scene illumination. The approach has been pioneered by Debevec and taken up by several other for various purposes, including real-time AR systems [7, 5, 6, 8]. The problem with using this approach for continuously operating systems is that it requires high resolution images of the reflective sphere, which has to be placed in the scene.

As seen from the above brief review the standard approaches to determining scene illumination conditions are either to manually measure the light sources, or to photograph a reflective sphere placed in the scene. As stated our goal is to investigate whether images of surfaces naturally present in the scene can be used for estimating illumination, i.e., to determine if changing illumination can be detected from a video sequence.

3 Background

This work is based on a number of assumptions, which we will list together here. First of all our approaches are targeted at daytime outdoor scenarios, allowing us to assume that the illumination conditions are in effect completely governed by light from a directional source (the sun) and light from the sky hemisphere. In addition we assume that the imaged scene is static, that a complete 3D model of the scene is available, and that the camera is internally and externally calibrated, such that each pixel corresponds to a ray that can be traced to a unique 3D point in the scene.

Additionally the presented techniques assume that the scenes contain diffusely reflecting surfaces and that different normal directions are represented by these diffuse surfaces. We use the approach that the 3D model of the scene is manually annotated with information about which surfaces can be considered diffuse reflectors. As described in section 1 the techniques involve a reflectance calibration phase, and it is assumed that surface reflectances do not change after this calibration phase. This means that precipitation is not allowed, i.e., it is not allowed to rain or snow after reflectance calibration.

Both presented techniques are based on an assumption that the Phong Illumination Model can be used as a reasonable approximation to outdoor illumination conditions, and finally one of the techniques further assumes that the direction of the sun light is known at all times, computed automatically based on knowledge of date, time, and the camera's position in latitude and longitude.

In order to estimate the illumination conditions of a scene from 2D images of that scene a model of the image formation process is needed that describes the interactions between light and surfaces. Such a model requires the reflectance properties of the surfaces in the scene as well as a 3D description of the scene. Given a sufficient number of surfaces of different orientations it is then possible to set up a system of equations and solve for the variables describing the illumination conditions.

The remainder of this section describes an illumination model and the acquisition of reflectance properties. As stated we assume that the scene can be measured and modeled manually, using for example 3D Studio Max or similar 3D modeling software.

3.1 Phong Illumination Model

The Phong Illumination Model [14] is a local illumination model that is often used in computer graphics because it is fast to compute and gives reasonably realistic results although it is largely an empirical model. It is called a local illumination model because interreflections between surfaces are not considered. Interreflections – also known as global illumination effects – are approximated with an ambient term that allows for a global control of brightness in a scene. Besides the ambient term the Phong Model is composed of two reflection components that are due to direct illumination on a surface: a diffuse and a specular term. Diffuse reflections scatter light equally in all directions, i.e., the intensity at a point on a surface does not depend on the viewing direction. The diffuse reflections are modeled with Lambert's cosine law which states that the reflected light is proportional to the cosine of the an-

gle between the surface normal and the incident light θ_i. The specular reflections depend on both the incident angle θ_i and the viewing angle θ_r, and may be modeled as proportional to the cosine of the angle α, see figure 2.

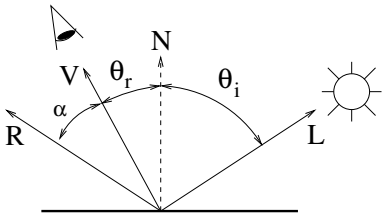


Figure 2: Image formation components of the Phong Illumination Model. L is a unit vector in the direction to the light source, N is the unit surface normal, V is the viewing direction, and R is the mirror-like reflected light.

The illumination of an outdoor scene may be modeled by one direct light source, the sun, and ambient light representing the skylight. The reflected light I_r approximated with Phong's Illumination Model is then given by the following equation:

$$I_{r,l} = k_{a,l} \cdot I_{a,l} + I_{i,l} \cdot (k_{d,l} \cdot \cos(\theta_i) + k_{s,l} \cdot \cos(\alpha)^m) \quad (1)$$

where k_a, k_d, and k_s are the reflection coefficients for the ambient, diffuse, and specular components, respectively, also called albedos which are described in the next subsection. I_a is the ambient illumination, I_i the direct light source, and m is a factor controlling the shininess of the surface. To handle color there is a separate equation for red, green, and blue, therefore the subscript $l \in \{R, G, B\}$.

In the following we assume pure diffuse surfaces and that $k_a = k_d$. Equation 1 then becomes:

$$I_{r,l} = k_{d,l} \cdot (I_{a,l} + I_{i,l} \cdot \cos(\theta_i)) \quad (2)$$

or using the vectors from figure 2:

$$I_{r,l} = k_{d,l} \cdot (I_{a,l} + I_{i,l} \cdot (\vec{N} \bullet \vec{L})) \quad (3)$$

3.2 Reflectance Properties

The reflectance properties of the surfaces are needed when using the scene as light probe. In the Phong Illumination Model (eq. 3) the reflectance properties are modeled with the scalar k_d. This is often called the *albedo* which is the reflectivity of a surface, or in other words the ratio of radiation reflected to the amount incident upon it. The reflected radiation may be expressed by the radiometric term *radiance* L_e which is the power leaving a surface per unit solid angle[2] and per unit surface area. The radiance can be measured using an image of a scene.

[2]The solid angle is the angle that, seen from the center of a sphere, includes a given area on the surface of that sphere. The value of the solid angle is numerically equal to the size of that area divided by the square of the radius of the sphere. It is measured in steradians [sr].

The radiometric term describing the received power per unit area, i.e., the power falling onto a surface, is the Irradiance E_e. For pure diffuse surfaces the albedo is then:

$$k_d = \frac{L_e}{E_e} \quad (4)$$

While it is rather easy to obtain the radiance from a scene using an image, the irrandiance requires knowledge of a 3D model of the scene and the light sources. One way to calculate the irrandiances for every pixel is then to synthesize (render) an image using the 3D model and setting all surface albedos to one.

4 Illumination Estimation under known Sun Position

In this first approach we take the illumination model presented in the previous paragraph and use it to model the measured pixel intensities from an image of the scene. If it is assumed that the system continuously can compute the unit direction vector to the light source relative to the scene coordinate system, then we arrive at a set of equations, one for each color channel for each pixel. These equations are linear in the ambient and the direct light, $I_{a,l}$ and $I_{i,l}$, respectively.

4.1 Approach

Let subscript j refer to the jth 3D point in the scene. Some points will in fact be in shadow and not receive direct light from the sun. Let S_j be a boolean parameter of value 1 if the jth point is in direct light, and 0 if it is in shadow. Furthermore, let C_j be a real number between 0 and 1, with the value of 1 if the jth point receives light from the entire hemi-spherical sky, and 0 if the sky is completely occluded seen from the jth point. The reflected light from the jth point can then be written as (the scene is small compared to the distance to the sun, so the unit direction vector to the sun is the same for all points in the scene):

$$I_{r,j,l} = k_{d,j,l} \cdot (C_j \cdot I_{a,l} + S_j \cdot I_{i,l} \cdot (\vec{N}_j \bullet \vec{L})) \quad (5)$$

The ambient occlusion factor, C_j, can be computed a priori for all points in the scene. Given knowledge of the sun's position the shadow masking parameter S_j can be computed at run-time for all points in the scene. From the offline reflectance calibration we know the albedos, $k_{d,j,l}$, of all diffusely reflecting points. The surface normal for all points, N_j is known from the 3D model of the scene. The direction vector to the sun, L, can be computed given: 1) the date, 2) the time, 3) the Earth position in latitude and longitude of the scene origin, 4) and the direction of North in the scene, [1]. The only unknowns in eq. 5 are the 6 parameters for ambient and direct light, $I_{a,l}$ and $I_{i,l}$.

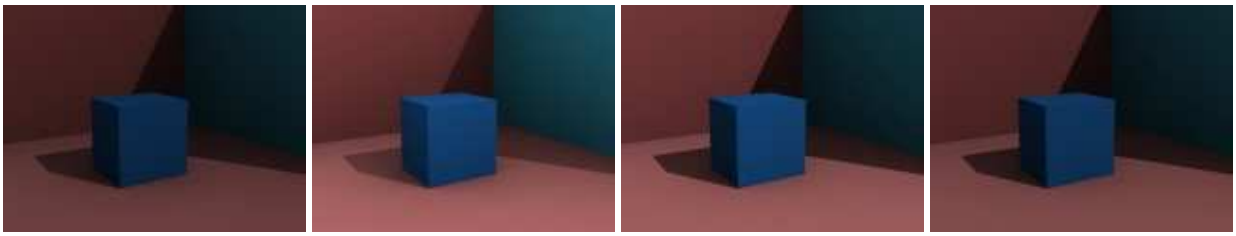


Figure 3: Frames 0, 29, 59 and 89 from synthetic test sequence with known illumination changes. The direction of the direct light source is not changing but the ambient and source emittances both change over the sequence.

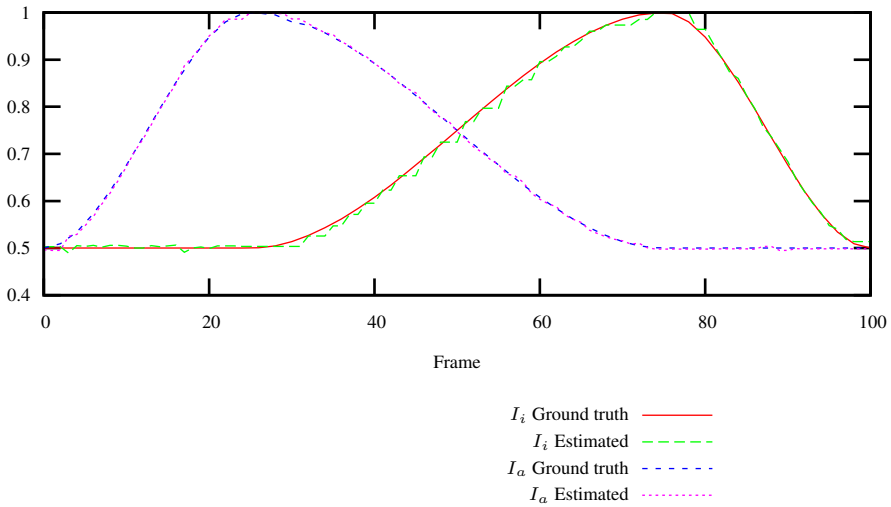


Figure 4: Comparison of estimated and true values for direct and ambient radiances for the synthetic test sequence shown in figure 3. All values are normalized to a maximum of 1.

Using the assumption that we know what surfaces in the scene can be considered diffuse and that the camera is calibrated to the scene we can also find those pixels that correspond to diffuse surfaces. The RGB pixel values of such a pixel is denoted $P_{j,l}$. If the camera is radiometrically linear the measured pixel values are some camera constant K times the reflected radiance from the corresponding scene point:

$$P_{j,l} = K \cdot I_{r,j,l} \tag{6}$$

Thus, by picking pixels from the image we can set up a system of linear equations in $I_{a,l}$ and $I_{d,l}$ of the form:

$$1/K \cdot P_{j,l} = k_{d,j,l} \cdot (C_j \cdot I_{a,l} + S_j \cdot I_{i,l} \cdot (\vec{N}_j \bullet \vec{L})) \tag{7}$$

In our implementation of this framework we at random select on the order of a few hundred pixels evenly distributed across the image (among those pixels that correspond to diffuse surfaces). It is important that the pixel population represents both areas in shadow (only ambient light) and in direct light (both ambient and direct light). The camera scene radiance to pixel value scaling factor K is of course unknown, but a system of equation of the form of eq. 7 allows us to estimate scene illumination up to a scaling factor.

4.2 Experiments and Results

The presented framework has been tested extensively on both synthetic and real images. Figure 3 shows a few frames from a synthetically generated sequence where a simple scene has been rendered with known ambient and direct intensities. Correspondingly, figure 4.1 shows the estimated intensities. As seen synthetic data results in near perfect estimations. The same scene has been tested with generating a sequence where a yellow ball is falling into the scene and bouncing out again in order to test how the illumination estimation procedure reacts to dynamic objects in the scene, thus violating the static scene assumption. The estimation results from this scenario is not shown, but due to the extraction of a large number of sample points across the entire image the illumination estimation is very stable and only in minor degree affected by the dynamic object.

To test the approach on real data a 2 hour time-lapse sequence has been acquired with one frame every 20 seconds. Figure 5 shows select frames from the sequence. Naturally, we do not have ground truth data for the illumination conditions in this real scene, but figures 4.2 and 4.2 show the estimated ambient and direct light. The estimated illumination has been verified

Figure 5: Frames 1, 72, 134, 201, and 259 from real test sequence covering approximately 2 hours of moving sun and changing cloud cover.

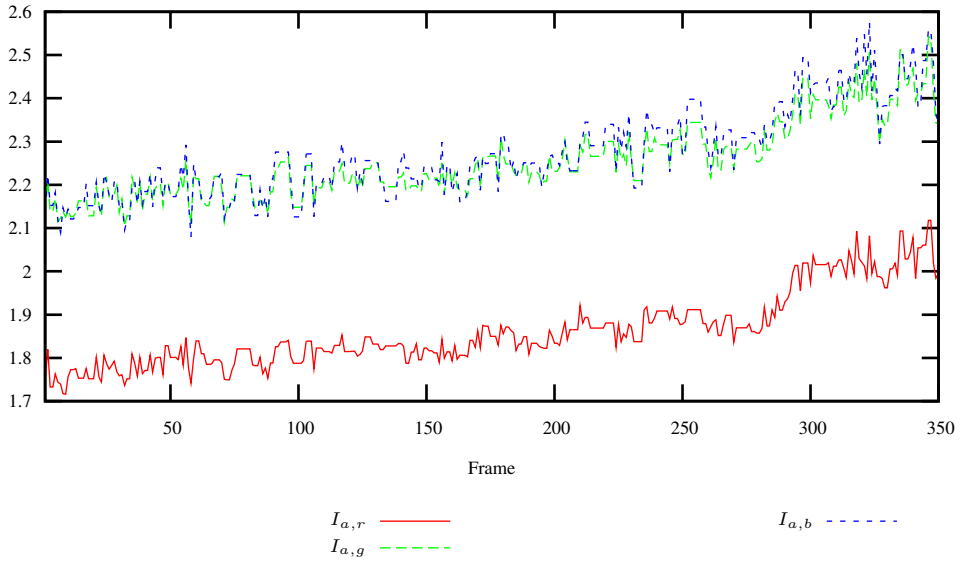


Figure 6: Estimated values for ambient radiance for the real test sequence shown in figure 5. Notice that the ambient light in the scene has a relatively low red (R) component as is expected for a sky with only partial cloud cover.

quantatively by taking the known albedos of surfaces in the scene, illuminating these surfaces with the estimated light and comparing these values with the real sequence pixel value for the same surface. These tests (not shown) demonstrate that the estimated illumination follows the real scene illumination quite accurately apart from a tendency to over-estimate the red component of the direct light with approximately 10%. This may be caused by mis-estimating the albedo of the dominant red brick wall due to an in-accurate determation of the illumination conditions at time the image was acquired for albedo estimation.

In addition to quantitative tests on real data qualitative tests of the estimated illumination has been evaluated by rendering virtual objects into the scenes and visually judging the quality of the virtual shading. Especially for sequences with very dynamic lighting conditions it is clearly seen that the estimated light results in consistent shading of virtual objects. Two frames from such a test were shown in figure 1.

The current implementation of this estimation technique can run the estimation at about 10 frames per second and is thus easily able to respond to the illumination condition changes an outdoor AR system would experience.

5 Illumination Estimation under unknown Sun Position

This section describes the estimation of the illumination conditions including the sun direction from an image of a scene given a 3D model of the scene and the albedos of the surfaces in the scene, and assuming that the illumination model in equation 2 can be used to approximate outdoor illumination conditions.

5.1 Approach

The illumination conditions may be estimated by setting up a sufficient number of equation 3, and solving this non-linear system of equations for the unknowns $\vec{L}$, $I_{a,l}$, and $I_{d,l}$ ($l \in \{R, G, B\}$). Thus, there are nine unknowns. Assuming that the distance r to the sun is known the estimation of $\vec{L}$ reduces to the two angles (azimuth, φ, and zenith, θ). In order to solve a non-linear system of equations one may formulate it as a least squares problem and then use a numerical optimization. Let $f_i(x)$ be the minimization function that should converge to zero. The minimization function for the Phong Illumination Model is given in equation 8:

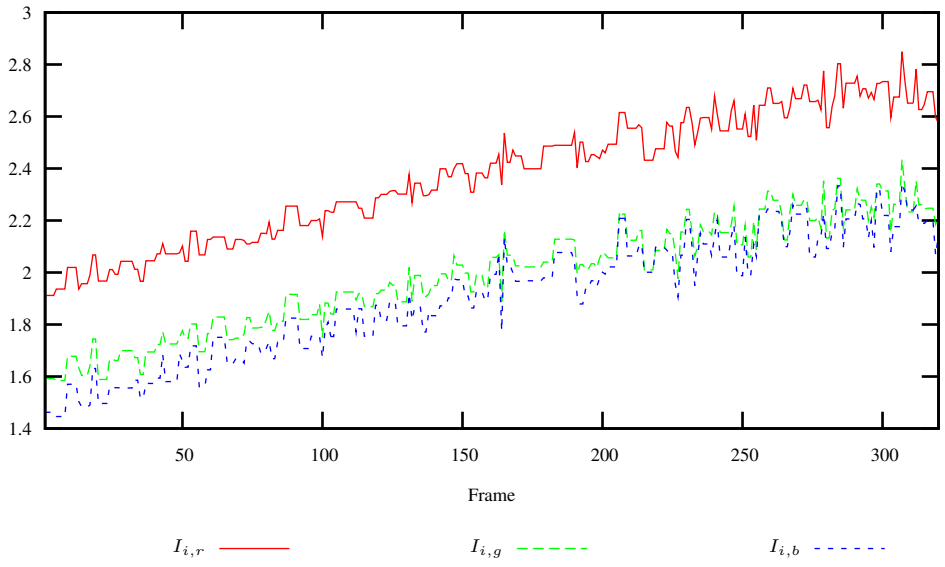


Figure 7: Estimated values for direct radiance for the real test sequence shown in figure 5.

$$f_{j,l}(x) = k_{d,j,l}(I_{a,j,l} + I_{d,j,l}\overline{\cos\theta_j}) - I_{r,j,l} = 0 \quad (8)$$

where

$$\overline{\cos\theta}_j = \begin{cases} 0 & , \quad \theta > \pi/2 \\ \dfrac{\vec{N}_j \bullet \vec{L}}{\|\vec{N}_j\| \cdot \|\vec{L}\|} & , \quad 0 \leq \theta \leq \pi/2 \end{cases} \quad (9)$$

and

$$\vec{L} = \begin{pmatrix} r \cdot \cos\varphi \cdot \sin\theta \\ r \cdot \sin\varphi \cdot \sin\theta \\ r \cdot \cos\theta \end{pmatrix} \quad (10)$$

5.2 Experiments and Results

The evaluation of computer vision methods using real image data is often difficult due to the lack of ground truth. In this work the estimation of a 3D model and of the albedos introduces an error which makes the evaluation of the illumination estimation inaccurate. Therefore the main evaluation was done using synthetic image data that were generated from a 3D scene description including light positions and object reflectances. They were rendered using a ray-tracer (Radiance [2]) that generates radiometric correct images including global illumination effects. Radiance supports all kinds of light sources among them a daylight model to create realistic illumination in outdoor scenes with sun and skylight. Figure 8 shows an example of a rendered image that was used for evaluation.

All estimations were done using MATLAB's *lsqnonlin* for solving non-linear least squares problems.

Using the daylight model images were rendered for illumination conditions from sunrise to sunset. The estimation errors are shown in figure 9. In most of the estimations the error lies around a few degrees $[0.6-2.5°]$ except from three estimations, which are at 9, 9:30 and 17 o'clock. With a look at the image of these time periods it can be seen that some of the measurement pixels were occluded resulting in no direct illumination.

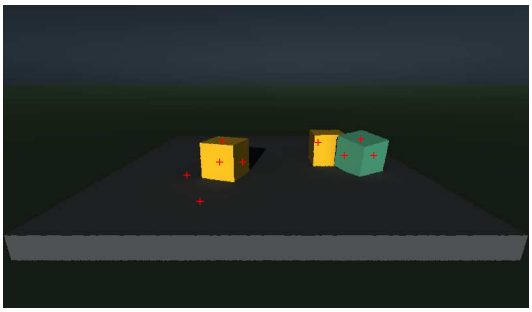


Figure 8: Synthesized image using the daylight model.

These estimation tests have been extended with sun positions across the entire hemisphere. The azimuth ranges $[0 \rightarrow 2\pi]$ and the zenith ranges $[0 \rightarrow \frac{1}{2}\pi]$. All in all 339 images have been rendered in Radiance. The scene illustrated in figure 8 has also been used in this experiment.

The error in these experiments are given as the total angle between the estimated and actual angles. Total error means the two actual angles (azimuth φ and zenith θ) seen in relation to the two estimated, which has been calculated from equation 11.

$$\Delta E = \arccos\left(\frac{\vec{a} \bullet \vec{e}}{|\vec{a}||\vec{e}|}\right) \quad (11)$$

The two vectors for actual ($\vec{a}$) and estimated ($\vec{e}$) are calculated from equation 12, where $r = 1$ since it is only the angular difference that is of interest.

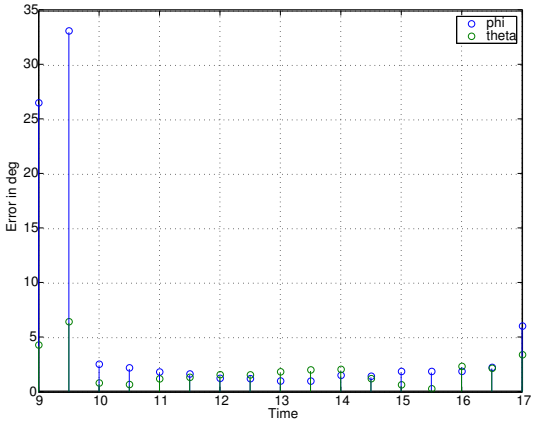


Figure 9: Estimation error in degrees as a function of the time for azimuth (φ) and zenith (θ) respectively.

$$\vec{a} = \begin{pmatrix} r \cdot \cos\varphi \cdot \sin\theta \\ r \cdot \sin\varphi \cdot \sin\theta \\ r \cdot \cos\theta \end{pmatrix} \tag{12}$$

The total angular error for all azimuth and zenith angles of the sun is shown in figure 10 where the error is indicated by the height of the small circles. The azimuth angle is given as the angle in the plan, and the zenith is zero in the center of the plot and increasing with distance to the center.

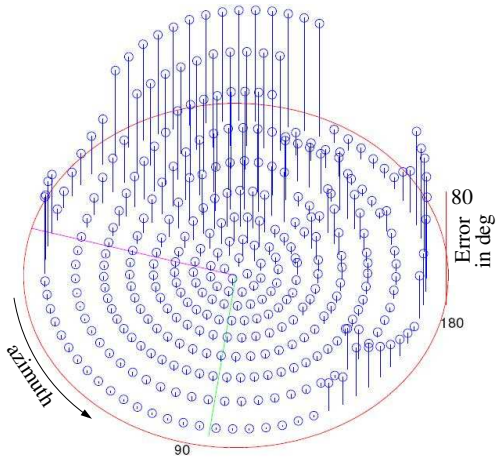


Figure 10: Total angular error over all azimuth and zenith angles.

The camera viewing direction is indicated by the green line around the an azimuth angle of $90°$. It can be seen that the error is rather low when the illumination direction is close to the camera direction, whereas the error increased significantly (up to $80°$) when the illumination is opposite to the camera. This is due to the reduced number of directly illuminated surfaces.

Besides simulated image data we tested the method on real images of building bricks. These bricks have rather diffuse reflectance properties. Figure 11 shows an image of a real scene that was used for illumination estimation, and figure 12 show a part of that scene with a virtual shadow that was simulated using the estimated illumination direction.

Figure 11: Real image used for illumination estimation.

Figure 12: Real scene showing both, the real shadow casted by the brick and a virtual shadow casted by the brick. The virtual shadow is rendered using the estimated sun position.

Figure 13 show an example application where the estimated illumination direction was used to augment the scene with four virtual vases.

6 Conclusions

In this paper two methods were proposed and tested to estimate the illumination conditions of a real outdoor scene while using the scene itself as a light probe, i.e., no additional light probe has to be placed into the scene. The methods work on single view images and require a 3D model of the scene as well as the reflectance properties of the surfaces present in the scene. The preliminary results show their applicability to Augmented Reality.

In future work we aim at combining several complementary methods in order to achieve more robust illumination estimation. Furthermore, we will look into possibilities to reduce the offline calibration, e.g., using reflectance models for common everyday outdoor objects. These objects may be recognized by some automatic object recognition and be used as light probes.

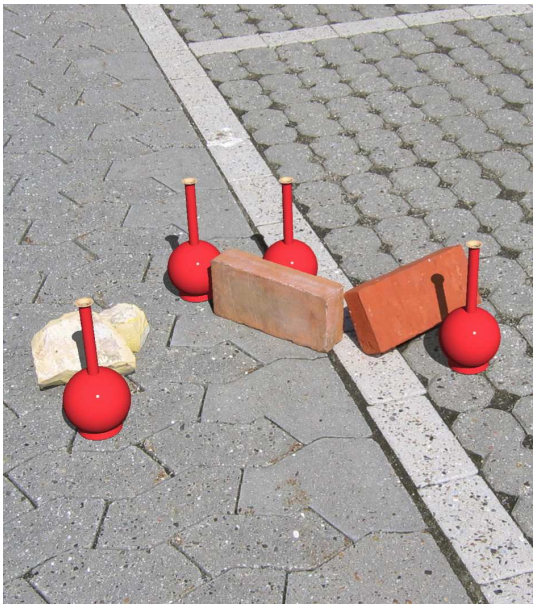


Figure 13: Real scene augmented with virtual vases.

Acknowledgments

This research is funded in part by the BENOGO project under the European Commission IST program (IST-2001-39184), and in part by the CoSPE project (26-04-0171) under the Danish Research Agency. This support is gratefully acknowledged.

References

[1] Computing planetary positions – a tutorial with worked examples. http://www.stjarnhimlen.se/comp/tutorial.html.

[2] Radiance synthetic imaging system homepage. http://radsite.lbl.gov/radiance/HOME.html.

[3] Samuel Boivin and André Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 107–116, August 2001.

[4] Samuel Boivin and André Gagalowicz. Inverse rendering from a single image. In *IS&T's First Europ. Conf. on Color in Graphics, Images and Vision*, pages 268–277, Poitiers, France, April 2002.

[5] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings: SIGGRAPH 1998, Orlando, Florida, USA*, July 1998.

[6] P. Debevec. Tutorial: Image-based lighting. *IEEE Computer Graphics and Applications*, pages 26 – 34, March/April 2002.

[7] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings: SIGGRAPH 1997, Los Angeles, CA, USA*, August 1997.

[8] S. Gibson, J. Cook, T. Howard, and R. Hubbold. Rapic shadow generation in real-world lighting environments. In *Proceedings: EuroGraphics Symposium on Rendering, Leuwen, Belgium*, June 2003.

[9] K. Jacobs and C. Loscos. State of the art report on classification of illumination methods for mixed reality. In *EUROGRAPHICS*, Grenoble, France, September 2004.

[10] M. Kanbara and N. Yokoya. Real-time estimation of light source environment for photorealistic augmented reality. In *Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, United Kingdom*, pages 911–914, August 2004.

[11] Celine Loscos, George Drettakis, and Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualisation and Computer Graphics*, 6(3), July 2000.

[12] V. Masselus, P. Dutre, and F. Anrys. The free-form light stage. In *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 247–256, Pisa, Italy, June 2002.

[13] G. Patow and X. Pueyo. A survey of inverse rendering problems. *COMPUTER GRAPHICS forum*, 22(4):663–687, 2003.

[14] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[15] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(3):290–300, March 2003.

[16] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series*, pages 215–224, Los Angeles, California, USA, August 1999. Addison Wesley Longman.

[17] Yizhou Yu and Jitendra Malik. Recovering photometric properties of architectural scenes from photographs. In *Proc. of SIGGRAPH 98*, pages 207–217, Orlando, Florida, USA, July 1998.

Detection of Cast Shadows in Surveillance Applications

Søren Gylling Erbou[1], Helge B.D. Sørensen[2], Bjarne Stage[3]

[1] Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Kgs. Lyngby.
sge@imm.dtu.dk

[2] Ørsted•DTU, Technical University of Denmark, 2800 Kgs. Lyngby.
hbs@oersted.dtu.dk

[3] Danish Defence Research Establishment, Ryvangs Allé 1, 2100 Kbh. Ø.
bs@ddre.dk

Abstract

Cast shadows from moving objects reduce the general ability of robust classification and tracking of these objects, in outdoor surveillance applications. A method for segmentation of cast shadows is proposed, combining statistical features with a new similarity feature, derived from a physics-based model. The new method is compared to a reference method, and found to improve performance significantly, based on a test set of real-world examples.

1 Introduction

The introduction of digital video cameras, and recent advances in computer technology, make it possible to apply (semi-)automated processing steps to reduce the amount of data presented to an operator in a surveillance application. This way the amount of trivial tasks are reduced, and the operator can focus on a correct and immediate interpretation of the activities in a scene.

The Danish Defence Research Establishment (DDRE) is currently focusing part of it's research on implementing a system for automated video surveillance. The main objectives of the DDRE are to gain general knowledge in this area, and eventually implement an automated surveillance application that is capable of detecting, tracking and classifying moving objects of interest.

At this point the DDRE has carried out some initial studies in testing and implementing parts of the W^4-system [4] for automated video surveillance. The W^4-system effectively detects moving objects, tracks them through simple occlusions (blocking of the view), classifies them and performs an analysis of their behavior. One limitation of W^4 is that the tracking, classification and analysis of objects fails when large parts of the moving objects are actually cast shadows.

Distinguishing between cast shadows and self shadows is crucial for the further analysis of moving objects in a surveillance application. Self shadows occur when parts of an object are not illuminated directly, but only by diffuse lighting. Cast shadows occur when the shadow of an object is cast onto background areas, cf. figure 1. The latter are a

major concern in today's automated surveillance systems because they make shape-based classification of objects very difficult.

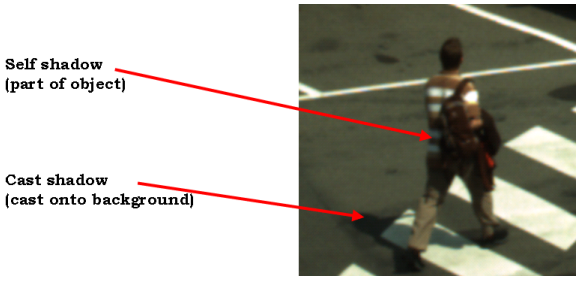


Figure 1: *Types of shadows. Self shadow is shadow on the object itself, a person in this case. Cast shadow is the shadow cast onto the background.*

In [9] Prati *et al.* give a comparative evaluation of the most important methods up until 2001. They conclude that the more general situations a system is designed to handle, the less assumptions should be made, and if the scene is noisy, a statistical approach is preferable to a deterministic model. In [5], Hsieh *et al.* focus on removing cast shadows from pedestrians using a statistical model combined with spatial assumptions. Only situations with pedestrians in an upright posture are handled and the cast shadows are assumed to touch their feet. Javed *et al.* [6] make no spatial assumptions of posture or composition prior to a statistical modelling of shadows, based on a correlation of the derivatives for regions of similar pixels.

In [7] Nadimi *et al.* apply a number of steps in a physics-based shadow detection algorithm. No spatial assumptions are made, but other assumptions makes it less suitable for some types of weather. Furthermore several threshold dependent parameters should be optimized. Finlayson *et al.* [3] use a physics-based approach to derive an illumination invariant, therefore shadow free, gray-scale image of an RGB image. From this image the original RGB image, without shadows, is derived. Finlayson's approach is aimed at shadow elimination in general in images obtained with a color calibrated standard digital camera [2],[3].

The rest of this paper consists of three sections, in section 2 existing methods for shadow handling are described in more detail, leading to a new combined method for segmentation of cast shadows. In section 3 the experimental results are presented, and section 4 is the conclusion.

2 Methods

The statistical approach suggested by Javed *et al.* [6] is implemented as a reference, because it makes no spatial assumptions and has the least number parameters to tune. The physics-based method suggested by Finlayson *et al.* is elegant, but not previously applied in surveillance applications. The new similarity feature proposed in this work is based on the ideas og Finlayson *et al.* Combining Javed's method with the new similarity feature, a new approach for handling cast shadows in surveillance applications is suggested.

2.1 Statistical Approach

Javed *et al.* [6] use a statistical approach for segmenting foreground pixels darker than a reference image (pixel-candidates) into cast shadow, self shadow and object pixels darker than the background. A K-means approximation of the EM-algorithm is used to perform unsupervised color segmentation of the pixel candidates. Each pixel candidate is assigned to one of the K existing Gaussian distributions if the Mahalanobis distance is below a certain threshold. If above this threshold a new distribution is added with it's mean equal to the pixel value. All distributions are assumed to have the same fixed covariance matrix $\Sigma = \sigma^2\mathbf{I}$, where σ^2 is a fixed variance of the colors and $\mathbf{I}$ is the identity matrix. After a pixel candidate is assigned to a distribution, the distribution mean is updated as follows:

$$\mu_{n+1} = \mu_n + \frac{1}{n+1}(x_{n+1} - \mu_n), \tag{1}$$

where x is the color vector of the pixel and μ_n is the mean of the Gaussian before the n+1th pixel is added to the distribution. Using a connected component analysis the spatially disconnected segments are divided into multiple connected segments. Smaller segments are then merged with the largest neighboring segment using region merging. Then each segment is assumed to belong to one of the three classes, cast shadow, self shadow or part of the object darker than the background image. To determine which of the segments are cast shadows, the textures of the segments are compared to the texture of the corresponding background regions. Because the illumination in a cast shadow can be very different from the background the gradient direction is used:

$$\theta = \arctan\frac{f_y}{f_x}, \tag{2}$$

where θ is the gradient direction and f_y and f_x are the vertical and horizontal derivatives respectively. If the correlation is more than a certain threshold, the region is considered a cast shadow. Otherwise it is either self shadow or dark part of the object. This method is considered as a state-of-the-art method in surveillance applications but still faces fundamental problems concerning some very context dependent parameters.

2.2 Physics-based Approach

The physics-based approach suggested by Finlayson *et al.* [3] derives an illumination invariant grayscale image from an RGB-image.

The color of a pixel in an image depends on the illumination, the surface reflection and the camera sensors. Denoting the spectral power distribution of the illumination $E(\lambda)$, the surface spectral reflection function $S(\lambda)$, and the camera sensor sensitivity functions $Q_k(\lambda)$ $(k = R, G, B)$, the RGB color ρ_k at a pixel can be described as an integral over the visible wavelengths λ:

$$\rho_k = \int E(\lambda)S(\lambda)Q_k(\lambda)d\lambda \quad , \quad k = \{R, G, B\}. \tag{3}$$

This description assumes no shading and distant lighting and camera placement. If the camera sensitivity functions $Q_k(\lambda)$ are furthermore assumed to be narrow-band, they can be modelled by Dirac delta functions $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, where q_k is the strength of the sensor. Substituting this into (3) reveals:

$$\rho_k = E(\lambda)S(\lambda)q_k \quad , \quad k = \{R, G, B\}. \tag{4}$$

Lighting is approximated using Planck's law:

$$E(\lambda, T) = Ic_1\lambda^{-5}\left(e^{\frac{c_2}{T\lambda}} - 1\right)^{-1}, \tag{5}$$

where I is the intensity of the incident light, T is the color temperature, and c_1 and c_2 are equal to $3.74183 \cdot 10^{-16} Wm^2$ and $1.4388 \cdot 10^{-2} Km$ respectively. Daylight is very near to the Planckian locus. The illumination temperature of the sun is in the range from $2500K$ to $10000K$ (red through white to blue). For the visible spectrum (400-700nm) the exponential term of (5) is somewhat larger than 1. This is Wien's approximation [6]:

$$E(\lambda, T) \simeq Ic_1\lambda^{-5}e^{-\frac{c_2}{T\lambda}}. \tag{6}$$

If the surface is Lambertian (perfectly diffuse reflection) shading can be modelled as the cosine of the angle between the incident light $\mathbf{a}$ and the surface normal $\mathbf{n}$. This reveals the following narrow-band sensor response equation:

$$\rho_k = (\mathbf{a} \cdot \mathbf{n})Ic_1\lambda^{-5}e^{-\frac{c_2}{T\lambda}}S(\lambda)q_k \quad , \quad k = \{R, G, B\}. \tag{7}$$

Defining band-ratio chromaticities r_k remove intensity and shading variables:

$$r_k = \frac{\rho_k}{\rho_G} \quad , \quad k = \{R, B\}. \tag{8}$$

Taking the natural logarithm (ln) of (8) isolates the temperature:

$$\begin{aligned} r'_k \equiv ln(r_k) &= ln(s_k/s_G) + (e_k - e_G)/T \quad , \quad k = \{R, B\}, && (9)\\ s_k &= \lambda^{-5}S(\lambda)q_k, && (10)\\ e_k &= -c_2/\lambda_k. && (11) \end{aligned}$$

For every pixel the vector $(r'_R, r'_B)^T$ is formed as a constant vector plus a vector $(e_R - e_G, e_B - e_G)^T$ times the inverse color temperature. As the color temperature changes, pixel values are constrained to a straight line in 2D log-chromaticity space, since (9) is the equation for a line. By projecting the 2D color into the direction orthogonal to the vector $(e_R - e_G, e_B - e_G)^T$, the pixel value only depends on the surface reflectance and not temperature hence illumination:

$$\begin{aligned} r'_R - \frac{e_R - e_G}{e_B - e_G}r'_B &= ln(s_R/s_G) - \frac{e_R - e_G}{e_B - e_G}ln(s_B/s_G),\\ &= f(s_R, s_G, s_B). && (12) \end{aligned}$$

Applying (12) to all pixels reveals the illumination invariant image $gs(x, y)$:

$$gs(x, y) = a_1 r'_R(x, y) + a_2 r'_B(x, y), \tag{13}$$

where the constant vector $a = (a_1, a_2)^T$ is orthogonal to $(e_R - e_G, e_B - e_G)^T$, determined by the camera sensitivity functions only (12)(11), and scaled to unit length:

$$\begin{aligned} a &= \frac{a'}{\|a'\|},\\ a' &= \begin{pmatrix} 1 \\ -\frac{e_R - e_G}{e_B - e_G} \end{pmatrix}. && (14) \end{aligned}$$

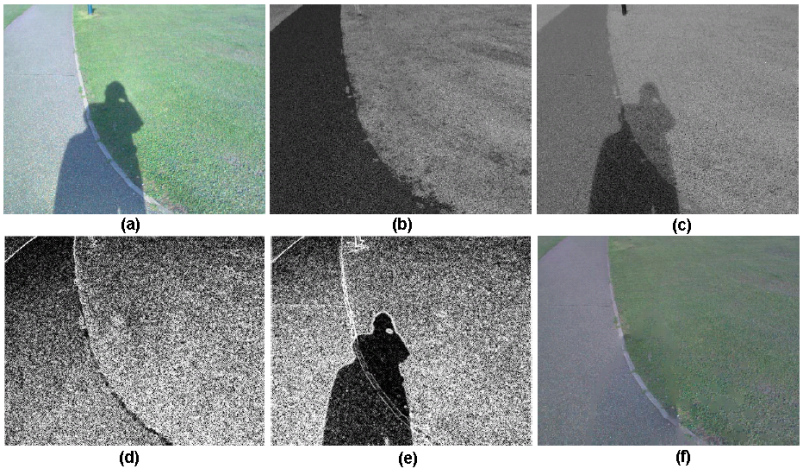


Figure 2: *Finlayson's approach to shadow removal [3]. (a): Original image. (b) Illumination invariant grayscale image. (c): Grayscale of original image. (d): Edge map for invariant image. (e): Edge map for non-invariant image. (f): Recovered shadow-free image.*

Figure 2(b) shows an example of an illumination invariant grayscale image, where edges due to shadows are not visible. Figure 2(a) and 2(c) show the original image, and the normal grayscale image.

If the sensor functions of the camera, and thereby λ_k of (11), are unknown, [2] and [3] outline a procedure for camera color calibration. The invariant direction is estimated by comparing a number of images taken during the day with changing illumination. Daylight is assumed to be Planckian with varying temperature. Each image contains different standard color patches from the Macbeth Color Chart.

The shadow edges are detected by comparing the gradient of each channel in the original log image, $\nabla\rho'(x,y)$, with the gradient of the illumination invariant image, $\nabla gs(x,y)$, cf. figure 2(d) and 2(e). The idea is that if the gradient in $\rho'(x,y)$ is high, while it is low in $gs(x,y)$, the edge is most likely to be a shadow edge. The following threshold function reveals a gradient image of the log response where gradients due to shadows are eliminated (set to zero):

$$S(\nabla\rho'(x,y),\nabla gs(x,y)) = \begin{cases} 0 & \text{if } \|\nabla\rho'(x,y)\| > \mathrm{t}_1 \\ & \text{and } \|\nabla gs(x,y)\| < \mathrm{t}_2 \\ \nabla\rho'(x,y) & \text{otherwise,} \end{cases} \tag{15}$$

where t_1 and t_2 are context dependent thresholds. By integrating S a log response image without shadows is recovered. This corresponds to solving the following Poisson equation:

$$\nabla^2 q'(x,y) = \nabla \cdot S(\nabla\rho'(x,y),\nabla gs(x,y)), \tag{16}$$

where ∇^2 is the Laplacian and q' is the log of the image without shadows. The gradient image of S equals the Laplacian of q' for each color band. Assuming Neumann boundary conditions ($\nabla q' = 0$ for boundary normals), q' can be solved uniquely up to an additive constant using the cosine transform [10]. When exponentiating q' to arrive at the shadow free image q the unknown constant becomes multiplicative. For the colors to appear "realistic" in each band, the mean of the top 5-percentile of pixels is mapped to maximum of the RGB image. In this way the unknown constants are fixed, and a shadow free image q is derived, cf. figure 2(f).

The major drawback of this method is reported to be defining the shadow edges. It turns out that using a robust edge detection algorithm (e.g. Canny or SUSAN [3]) and setting the thresholds are crucial factors. Furthermore a morphological opening is applied on the binary edge map to thicken the shadow edges and thereby improve the suppression of shadow gradients before the re-integration step.

Despite all of the assumptions and difficulties reported the method shows good results on the images shown in [2],[3]. It should be noted that the gradient images and thresholds are very context dependent. However, even when the method performs poorly it still attenuates the shadows. This is often the case for shadows with diffuse edges. Therefore the method is interesting in conjunction with surveillance tasks, where the artifacts introduced by the imperfect shadow edge detection and the re-integration are not crucial.

Due to assumptions in the model, and in the derivation of the shadow free RGB image, the method is far from perfect, but shadows are attenuated significantly. The method has not been applied in a surveillance application yet.

2.3 New Similarity Feature

It was found that the illumination invariant image is sensitive to the limited dynamic range in the video sequences of the camera used (8 bit) and to the spectral sensor functions of the camera not being delta functions. Because of this, determining edges due to shadows in a robust way becomes very difficult. Finlayson *et al.* also reports this to be the major drawback of the method [3].

Instead of only using the illumination-invariant image to determine edges due to shadows, other information should also be used. An important observation to make is that a foreground mask is available from the background model in a surveillance application. This can be used to eliminate artifacts from false shadow edges outside the foreground mask, and should be exploited in the detection of shadow edges.

A dilated version of the edges of the foreground mask is used to determine which gradients to suppress in the gradient image of the illumination invariant image, before reconstructing the "shadow-free" image. Figure 3(a) shows an image and a version of it, figure 3(b), that is reconstructed without suppressing any gradients. Therefore the two images are similar. Figure 3(c) shows the mask used for suppressing gradients, and figure 3(d) shows the corresponding reconstructed image.

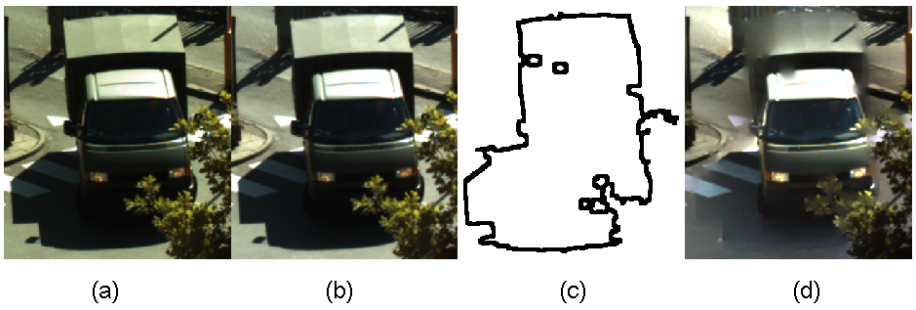


Figure 3: *Reconstruction of an image. (a): Original image. (b): Reconstructed image without suppressed gradients. (c): Suggested mask for suppressing gradients. (d): Reconstructed image with suppressed gradients.*

Both shadow and object gradients are suppressed, but figure 3(d) still clearly contains additional information that can be exploited in the segmentation of cast shadows.

The new similarity feature compares corresponding pixels of the reconstructed image and the background image, for every color segmented region:

$$CS = \frac{1}{\hat{\sigma}^2_{R,BG}(K-1)} \sum_{i=1}^{K} (R_i - BG_i)^2, \tag{17}$$

where CS is the similarity feature of a region, K is the number of pixels of the region times the three colorbands, R and BG are the intensity values of the i'th pixels in the reconstructed image and the background image, respectively. $\hat{\sigma}^2_{R,BG}$ is a variance normalization factor, which is the estimated variance between all pixels in a background image, BG, and all pixels in a reconstructed image, R, of a new frame containing no foreground objects.

Performing a variance normalization of CS makes it a relative measure of similarity that, ideally, only contains variation due to the region not being cast shadow, and not contains variation due to the experimental setup and the complex processing of the images. The estimate of the variance is based only on one sequence since it was difficult to obtain sequences, without foreground objects, that were static while an entire background model was estimated. It is therefore a rough estimate.

The CS measures a normalized mean value of squared differences between regions in the reconstructed foreground image, cf. figure 3(d), and corresponding regions in the background image. If the reconstructed image contains shadow regions along the border of the foreground mask, cf. figure 3(c), these shadow regions are attenuated in the reconstructed image, making them more similar to the background image. This is the key observation that the enhanced similarity feature, CS, is based on. Therefore a large value of CS corresponds to little similarity, which indicates that the region is part of the object. Small values of CS indicate high similarity, i.e. the region is then part of a cast shadow.

It is emphasized that CS only supplies useful information when the shadow edges are actually part of the edge of the foreground mask. In some cases it will not supply any additional information, e.g. when edges due to objects instead of shadows are suppressed. This will tend to smear neighboring background and object regions, for which reason it is suggested only to apply the CS in cases where the correlation threshold, described in 2.1, does not produce confident results. This corresponds to introducing a *reject class* for the correlation feature.

Figure 4 shows the suggested enhanced classification of color segmented regions. The

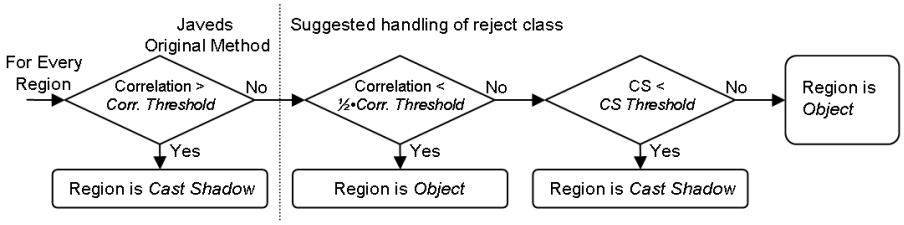


Figure 4: *Flowchart illustrating the enhanced classification of color regions. The enhanced similarity feature, (CS), classifies all regions that the correlation feature assign to a reject class (k·Corr. threshold < Correlation < Corr. threshold ⇒ reject class, 0<k<1).*

left part corresponds to the classification originally suggested by Javed, using a simple correlation threshold. The enhanced classification introduces a reject class if the correlation lies in an interval between k and 1 times the *Correlation threshold* introduced by Javed [6]. k should lie in the interval $[0; 1]$, and is empirically chosen to be 0.5 in this framework. If

the regions in the reject class have a CS larger than the *CS threshold* they are classified as object regions. Otherwise they are classified as cast shadow regions.

3 Data and Results

The camera used for data acquisition is a state-of-the-art industry digital video camera (SVS-204CFCL) with a resolution of 1024x768 pixels. The frame rate currently available is 20 fps., with a dynamic range of 8 bits, and with colors obtained through standard Bayer filtering. A typical scene for a surveillance application is chosen where the typical moving objects are vehicles, people and bicycles.

A kernel-based background model is used to segment foreground objects [1]. Only one frame of an object is used in the data set to avoid stochastic dependence between samples. 18 foreground objects are used in a manual optimization of model parameters and 72 foreground objects are used for validation and comparison of methods [1]. The main performance parameter used is the overall accuracy (AC), defined as the ratio of correctly classified pixels and the total number of pixels that are shadow candidates. True positives (TP) are defined as the proportion of correctly classified object pixels, and true negatives (TN) as the proportion of cast shadow pixels correctly classified.

A color calibration of the camera was performed to determine the the optimal angle of projection in the log-chromaticity space (39.4°). This angle corresponded well with the angle obtained from the spectral sensitivity functions of the camera.

As a reference Javed's statistical segmentation of shadow candidates is used. This is compared to the new method using the new similarity feature. In the optimization of model parameters of the two methods different values for the region merging criteria were found to be optimal. In the reference method more regions were merged into larger regions, making it hard to obtain a performance better than mediocre, because some regions contained both shadow- and object pixels and was classified as a whole. Due to the new similarity feature, the optimal merging parameter was found to produce more and therefore smaller regions to classify, making the method less susceptible to regions containing both types of pixels. Figure 5 compares the classification using the reference method and the enhanced method on the example of figure 3.

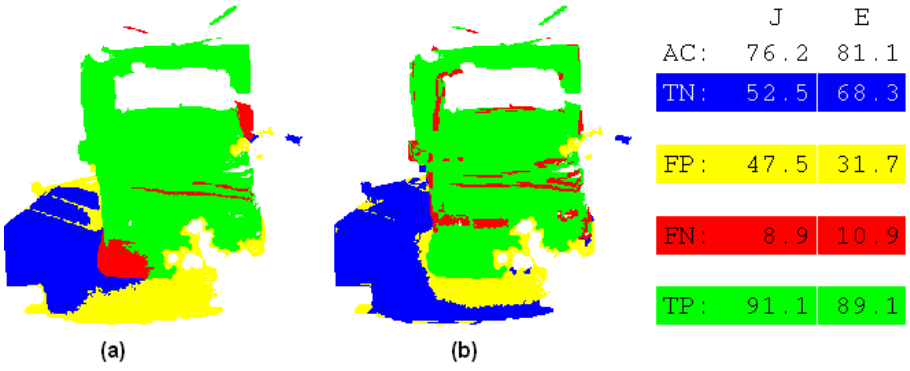


Figure 5: *Classification (%), AC=Accuracy, TN=True cast shadow pixels, FP=False object pixels, FN=False cast shadow pixels, TP=True object pixels. (a): Reference method (*J*). (b): Enhanced method (*E*) applying the new similarity feature.*

Table 1 shows the mean and std. of the absolute performance measures, based on the

test set, for the two methods.

Method	AC	TP	TN
Javed (J) - Mean (Std.) [%]	64.9 (17.8)	63.4 (30.0)	64.7 (33.4)
Enhanced (E) - Mean (Std.) [%]	69.2 (13.7)	69.7 (18.3)	66.0 (23.9)

Table 1: *Absolute performance of the two methods (*J *and* E*) based on the test set of 72 examples. Mean values and standard deviations are shown. AC=Accuracy, TP=True object pixels, TN=True cast shadow pixels.*

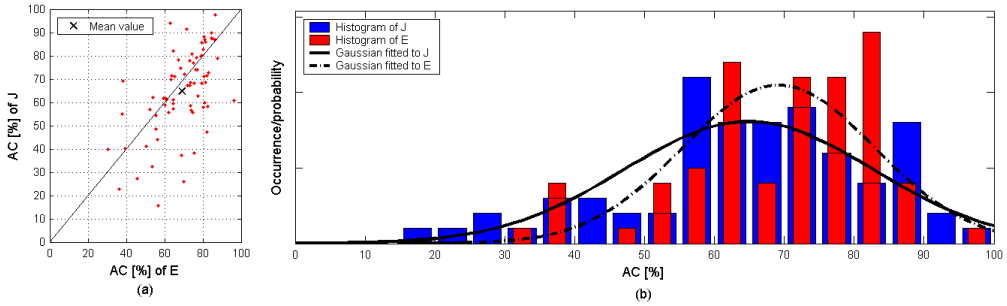


Figure 6: *Comparison of performance. (a): Accuracy of Javed's method (*J*), as a function of accuracy of enhanced method (*E*), based on the test set. (b): Histograms and fitted Gaussians of* J *and* E*, based on the test set.*

Figure 6 illustrates some of the results from table 1. There is a trend that examples with a higher AC in (E), are improved more than the examples with decreased AC, are decreased. This gives rise to the higher mean values, and indicates that fewer examples tend to have much better AC, while more examples tend to have slightly decreased AC.

A paired t-test is applied to determine if there is any significant difference, at a 5% level, in the mean values of the performance measures of the two methods. Table 2 shows the results.

Paired t-test, H_0: $\mu_E - \mu_J = 0$	AC	TP	TN
Difference in mean value ($E - J$)	1 (0.009)	1 (0.020)	0 (0.326)
Lower confidence bound [%]	1.31	1.28	−3.42

Table 2: *Statistical comparison of the absolute measures, AC=Accuracy, TP=True object pixels, TN=True cast shadow pixels. Row 1:* 0 *denote that the mean value cannot be rejected to be equal at a* 5% *level, and* 1 *that the difference of the means is significantly positive. p-values are shown in parentheses. Row 2: Lower confidence bounds for the differences in mean values for the absolute measures, at a* 95% *confidence level.*

0 denotes that the means cannot be rejected to be equal at a 5% level, and 1 that the difference of the means is significantly positive. The p-values are shown in parentheses. The conclusion to make from the test is that the new method (E) produces significantly better accuracy (AC) and is better at classifying object pixels correctly (TP), than the reference method J.

The lower confidence bounds of the difference in mean values, at a 95% confidence level, are shown in the second row of figure 2. They show that the difference in true mean values of the AC and TP for method E, are likely to be at least 1.3% above those of method J.

4 Conclusion

An enhanced method for shadow removal is suggested, based on a new similarity feature derived from a physics-based model. The new method significantly improves the mean accuracy at a 5% significance level, compared to the reference method.

The new similarity feature is only applied when the correlation feature of the reference method is uncertain, ensuring that the spatial assumption does not degrade performance, when compared to the reference method.

The final conclusion therefore is, that the suggested enhanced method for shadow removal, on average is better than the state-of-the-art method suggested by Javed. The enhanced method is also more robust, since it tends to improve the accuracy substantially, for examples where the reference method tends to fail completely.

Combining Javed's statistical-based method with some of the physics-based ideas of Finlayson, and a new similarity feature, therefore reveals a better and more robust algorithm for segmentation of cast shadows from moving objects.

The use of the illumination invariant image, as suggested by Finlayson, might be able to improve the performance even more, but requires a larger dynamic range than the 8 bits currently available with the present camera.

References

[1] Erbou, SG. "Segmentation of Cast Shadows from Moving Objects". M.Sc. Thesis, Ørsted•DTU, Technical University of Denmark, October 2004.

[2] Finlayson, GD., Hordley, SD. "Color Constancy at a Pixel". Journal of the Optical Society of America A, Vol.18 no. 2, pp.253-264, 2001.

[3] Finlayson, GD., Hordley, SD., Drew, MS. "Removing Shadows from Images". European Conference on Computer Vision (ECCV), part IV, p823-836, 2002.

[4] Haritaoglu, I., Harwood, D., Davis, LS. "W^4: Real-Time Surveillance of People and Their Activities". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.8, pp. 809-830, August 2000.

[5] Hsieh, JW., Hu, WF., Chang, CJ., Chen, YS. "Shadow elimination for effective moving object detection by Gaussian shadow modeling". Image and Vision Computing 21, pp.505-516, 2003.

[6] Javed, O., Shah, M. "Tracking And Object Classification For Automated Surveillance". European Conference on Computer Vision (ECCV), part IV, p343-357, 2002.

[7] Nadimi, S., Bhanu, B. "Moving Shadow Detection Using a Physics-based Approach". IEEE Proceedings of Pattern Recognition. Vol. 2, pp. 701-704, 2002.

[8] Park, S., Aggarwal, JK. "Segmentation and Tracking of Interacting Human Body Parts under Occlusion and Shadowing". IEEE Proceedings of the Workshop on Motion and Video Computing. pp. 105-111, 2002.

[9] Prati, A., Mikic, I., Trivedi, MM., Cucchiara, R. "Detecting Moving shadows: Algorithms and Evaluation". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 25, No. 7, pp. 918-923, 2003.

[10] Press, WH., Teukolsky, SA., Vetterling, WT., Flannery, BP. "Numerical Recipes in C: The Art of Scientific Computing". Cambridge University Press. 2nd ed. 1992.

Spectral unmixing for separation of reflection components

Kristian Kirk
kirk@cvmt.dk
Computer Vision and Media Technology Laboratory (CVMT)
Aalborg University (AAU)

July 1, 2005

Abstract

The light spectrum that is recorded by a pixel on an imaging sensor is often a mixture of several distinct source spectra. Such a mixture may be modeled as a linear combination of some basis spectra, and if the basis for the scene is known, each pixel may be decomposed into its original components using linear inversion. Since the maximum number of separable basis spectra is equal to the number of image bands, multispectral images offer advantages compared to RGB-images. It is shown how multispectral images may be used to separate various reflection components, including second-order scattering, based on the Dichromatic Reflection Model.

1 Introduction

The light spectrum that is recorded by a pixel on an imaging sensor is often a mixture of several distinct source spectra. The mechanisms behind the mixing may be many, and they may be divided into two categories:

1. **Mixed object surfaces**: A pixel may integrate light originating from different object surfaces, for several reasons:
    - A pixel has a certain spatial extent and therefore collects light from different optical paths.
    - No real optical system behaves as an ideal pinhole system. Light rays from one point in the scene will be dispersed over an area of the imaging sensor (often several pixels) and, conversely, each pixel receives light from different points in the scene. The major part of this optical dispersion is described by the point spread function (spatial domain) or the modulation transfer function (frequency domain) of the optics.
    - Motion blur: Due to the certain integration time required to collect enough light for the sensor elements, the camera as well as the objects in the scene may move during the exposure period.
2. **Mixed reflection components**:
    - The light reflected from a single light source by a point on an object surface is a mixture of specularly reflected light and light altered or colourized by the material of the object itself. These two reflection components may differ significantly in terms of intensity and spectral content.
    - Each point on an object surface is often illuminated by a mixture of different light sources. These may be true emittive (primary) light sources, as well as secondary sources appearing as a result of light scattering (reflection, transmission) during which the primary light sources have continued with changed direction, intensity and spectrum.

Automated image analysis relies on the ability to characterize a scene by interpreting such mixed signals. Sometimes, the effects of mixing are not significant, but in other situations they have to be taken into account. In order to make things easier, different efforts can sometimes be made before the image acquisition to control or reduce the mixing processes by controlling the light sources and the objects in the scene. More often, however, the scene can not be controlled and, furthermore, inherent effects of the imaging process, like the spatial extent of pixels and the optical blurring, can not be avoided. Thus, any account of these mixing effects must be taken in the subsequent image analysis.

The key ideas that are followed in this work are that 1) since all the mixing processes described mentioned above may be described as linear, it should be possible to use a linear basis for decomposing each pixel into some more original spectral constituents, and that 2) since the maximum number of linearly separable components is equal to the number of image

bands, multispectral images[1], should have more potential than RGB images.

2 Previous work

2.1 Several object surfaces

The initial inspiration for this work was the field of spectral mixture analysis, or spectral unmixing, for remote sensing (for a review, see [7]). Here, the goal is, given an observed spectrum from a physical mixture where a number of basic constituents are present in unknown quantities, to determine the so-called endmember spectra of the original constituents, and the quantities at which these constituents are present. The problem has been studied for many years in chemical spectroscopy for analyzing chemical mixtures, and in more recent years it has been important in remote sensing, where mixed pixels are more the rule than the exception. Unmixing algorithms in remote sensing operate on reflectance data, and normally use a linear mixing model, where the mixed spectrum of a pixel is assumed to be a linear combination of the original spectra present in the pixel, weighted by the quantities of their occurrences. Let there be L spectral measurements (bands) and M endmembers, then the linear mixing model is

$$s_i = \sum_{j=1}^{M} r_{ij} f_j + n_i, \ 1 \leq i \leq L,$$

or in, matrix notation,

$$\mathbf{s} = \mathbf{R}\mathbf{f} + \mathbf{n},$$

where $\mathbf{R}$ is an $L \times M$ basis matrix containing the endmember spectra $\mathbf{r}_j$, $\mathbf{f}$ is a vector of abundance fractions, and $\mathbf{n}$ is a vector of error terms. For a physically meaningful interpretation of the model, the abundance fractions f_j should be nonnegative and sum to one.

The linear mixing model is an intuitive and computationally attractive model accounting for the contribution of various object surfaces in the mixing process. However, it is only correct under the assumption that the surfaces are well-separated, uniform and non-interacting in terms of light. Thus, scattering (reflection and transmission) of light between surfaces are not accounted for, and neither is the local variation of geometric effects such as shading and specular reflection. A common way to handle the problems with shadows and shading is to augment the reflectance basis matrix with a dark "shade" endmember [11, 9].

[1] Throughout this paper, the term "multispectral" will be used in the meaning "more than three bands" (thus including the case of "many" bands which is often termed "hyperspectral").

2.2 Several reflection components

In physics-based colour image analysis, the standard reflection model is the Dichromatic Reflection Model [10], which describes the reflected light from a single point light source at a point on an object surface as a weighted mixture of so-called body reflection and surface reflection:

$$L(\theta, \lambda) = m_b(\theta) r_b(\lambda) E(\lambda) + m_s(\theta) r_s(\lambda) E(\lambda),$$

where θ is a vector of photometric angles (incident angle, exit angle, and phase angle) between the incident light and the observer, λ is the wavelength, E is the power spectral function of the illuminant, r_b and r_s are the body and surface reflection functions, and m_b and m_s are geometrical scaling factors which depend on the photometric angles. A commonly used special case (Type I) of the model is when the specular reflectance function does not depend on the wavelength, i.e., $r_s(\lambda) = r_s$ (Neutral Interface Reflection), which is valid for materials having high contents of water and oil.

Several methods have been proposed to separate surface reflection from body reflection [8, 3, 1, 12]. Some of these methods use several images (some using polarizing filters), and some of them use a single RGB colour image, mostly in somewhat restricted situations, for example, only one light source and no interreflections. Some works have also taken into account interreflections [6, 13, 3], but only in restricted cases.

2.3 Contribution of this work

In this study the idea is to take a combined approach to the problem of mixed surfaces and mixed reflection components using a linear mixing model with multispectral images. Linear basis models for separation of reflection components have been used before for RGB images [8, 12, 3], but since the maximum number of separable (independent) basis spectra equals the number of bands, only three basis components can be separated from RGB images. Therefore, an obvious hypothesis is that with multispectral images there is potential for handling less restricted scenarios, for example, several mixed illuminants and interreflections of higher order.

The remainder of the paper is structured as follows. In Section 3, a general linear mixture model will be proposed that allows mixed multiple scattering reflection components to any order. In Section 4, a case example is studied with an outdoor scene with

vegetation and soil, and some results for that case will be shown. Section 5 contains a discussion, and Section 6 gives a conclusion.

3 Linear mixture model for reflection components

The standard dichromatic reflection model describes the case of single-bounce reflections. In this section it is proposed how to include transmission, and it is shown how multiple scatterings may be included in a linear mixing model. Comments are also given on model requirements and inversion, and on interpretation of the mixing weights.

3.1 Scattering by transmission

Since scattering by transmission is produced by the same principles as scattering by body reflection, it will be regarded as a special case of dichromatic reflection, where the surface reflection is zero. Also, it will be assumed that the transmittance function is identical to the body reflectance function, and they will be jointly referred to as the "body spectrum", denoted $\mathbf{r}$. Thus, the term "scattering" will be used as a common term for light-matter interaction in any of the three forms: body reflection, transmission and surface reflection.

3.2 Dichromatic scattering of higher order

Let the spectrum $\mathbf{s}$ recorded at a pixel be a mixture originating from up to M different object surfaces (with body spectra $\mathbf{r}_i$), which may scatter light from up to N primary illuminants (with spectra $\mathbf{e}_j$). Assume that the reflectance function is the same for all points on surfaces of the same type, and that surface reflection preserves the shape of the illuminant spectrum (Neutral Interface Reflection). Then, from the dichromatic reflection model, all mixtures of scatterings up to order P can be modeled as

$$\mathbf{s} = \mathbf{B}^P \mathbf{w} + \mathbf{n},$$

where $\mathbf{B}^P$ is a mixing matrix whose columns are modulated spectra produced by dichromatic scatterings. For zeroth order scatterings (specular reflection, or direct illumination, from illuminants), the mixing matrix is

$$\mathbf{B}^0 = [\ \mathbf{e}_1\ \mathbf{e}_2\ \ldots\ \mathbf{e}_N\ ],$$

and for 1st order scatterings it is

$$\mathbf{B}^1 = [\, \mathbf{e}_1 \ldots \mathbf{e}_N\, \mathbf{e}_1\mathbf{r}_1 \ldots \mathbf{e}_1\mathbf{r}_M\ \ldots\ \mathbf{e}_N\mathbf{r}_1 \ldots \mathbf{e}_N\mathbf{r}_M\,].$$

Second-order scattering would include the components $\mathbf{e}_1\mathbf{r}_1\mathbf{r}_1$, $\mathbf{e}_1\mathbf{r}_1\mathbf{r}_2$, ... In general, an increased order of scattering can be included by adding new columns to the mixing matrix, where the new endmembers are constructed by multiplying the endmembers of the previous order by all the body spectra:

$$\mathbf{B}^{P+1} = [\ \mathbf{B}^P\ \mathbf{b}_i^P\mathbf{r}_j\ ].$$

Notice that, because of the neutral interface assumption, higher order scattering involving one or more surface reflections can not be distinguished from lower-order scattering with only body scattering.

Obviously, this method is a naive, exhaustive approach for constructing a linear basis. The number of columns in the mixing matrix increases quickly with M and N, and easily becomes high even for simple scenes. Therefore, it will normally be necessary to reduce the basis to make it practical for a particular problem. The case study in the next section will show an example of this.

3.3 Model requirements and inversion

An essential requirement is that the endmember spectra are linearly independent. Also, no body spectrum must be flat (grey), since it then will be inseparable from surface reflection. Furthermore, it is generally expectable that endmembers corresponding to higher-order scattering are less identifiable than lower-order scatterings due to a lower signal-to-noise ratio. Therefore, care must be taken when deciding which endmembers should be included in the model.

If there are exactly as many endmembers as measurements (spectral bands), the basis is full and a solution for the mixing weights might be found by inverting $\mathbf{B}$. If there are more endmembers than bands, the system is underconstrained, and no unique solution can be found. Often, there will be fewer endmembers than bands (undercomplete basis, over-constrained system), and a least-squares solution must be found. Also, it is normally necessary to constrain the solution to be within physically realistic bounds (non-negative weights, upper bounds on weights and their sum), which makes it necessary to solve for the weights using an iterative algorithm instead of a standard closed-form solution.

3.4 Interpretation of mixing weights

When a spectrum contains distributions from several surfaces, it is not straightforward to relate the weights to abundance fractions. This is because the geometrically dependent scaling factors (shading and specularity) as well as the intensity of the illuminants may vary between the surfaces. Some simplifying

assumptions will be necessary, as it is known from remote sensing.

4 Case study: Vegetation and soil

Colour image analysis of outdoor vegetation scenes is often complicated by the fact that leaves not only reflect, but also transmit a significant portion of the light they receive. This may confuse, for example, algorithms trying to distinguish plants from the background, since the background may be coloured by the light transmitted by the leaves above. Therefore, it would be useful to be able to detect and quantify the various scattering components in such scenes, also including specular reflections.

Interestingly, in [4], ground-based multispectral images of an agronomic scene were used for spectral mixture analysis with four reflection components as endmembers: Sunlit leaves, sunlit soil, shaded leaves, and shaded soil. Thus, no specular reflection was considered. Interreflection was only implicitly considered, since the endmembers were determined from analysis of manually selected image regions.

4.1 Model

A second-order scattering model will be presented for images of vegetation and soil. The model will be the same for all pixels; therefore, it is assumed that the spectra of the vegetation and the soil do not vary significantly. Also, only one illuminant spectrum will be used. This is an important assumption. In overcast outdoor scenes it is a reasonable approximation, but in sunny weather it must be expected to cause problems. In shadows, for example, the illumination contribution from the blue sky is more influent than in the sun.

The illuminant will be denoted $\mathbf{e}$, and the body reflectance of the soil will be denoted $\mathbf{r}_s$ (no transmittance is expected from the soil). It will be assumed that the transmittance spectrum of leaves is equal to the body reflectance, and they will be jointly denoted $\mathbf{r}_v$. The endmember spectra up to second order scattering will then be represented by the matrix

$$\mathbf{B}^2 = [\,\mathbf{e}\ \mathbf{e}\mathbf{r}_v\ \mathbf{e}\mathbf{r}_s\ \mathbf{e}\mathbf{r}_v\mathbf{r}_s\ \mathbf{e}\mathbf{r}_v\mathbf{r}_v\ \mathbf{e}\mathbf{r}_s\mathbf{r}_s\,]$$

4.2 Experiment

A multispectral image of some young leaves on a dry soil was taken with a monochrome camera fitted with an electronically tunable filter. The camera was of the model Retina EX from QImaging, and the filter was a liquid crystal tunable filter (LCTF) type, the model was VariSpec VIS from Cambridge Research & Instrumentation (CRI). 26 bands were used, from 470nm to 720nm, in 10nm intervals. For each band, expected dark offset was subtracted from the original digital number, and the result was then divided by the integration time and the band sensitivity. A scaled estimate $\hat{\mathbf{e}}$ of the illuminant spectrum was obtained from an image of a white calibration reference (Spectralon), and estimates of the leaf and soil spectra, $\hat{\mathbf{r}}_v$ and $\hat{\mathbf{r}}_s$, were measured with a spectrometer. A mixing matrix $\hat{\mathbf{B}}$ was then constructed as described above. The resulting endmember spectra are shown in Figure 1.

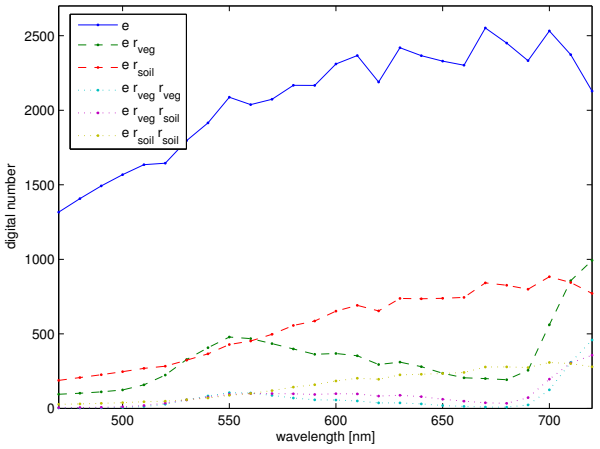


Figure 1: *Estimated endmember spectra up to second-order body scattering.*

The original image is shown as an RGB image in Figure 2. (An RGB image may be created from a multispectral image by integrating the spectrum measured at each pixel with some simulated RGB sensitivity functions. Gaussian sensitivity functions were used, with centres at 475, 550, and 625 nm, respectively, and standard deviations of 25 nm. The functions were scaled (white balanced) using the measured daylight spectrum $\hat{\mathbf{e}}$.)

For each pixel, a numerical inversion of the observed spectrum was performed, solving for the unknown weights of the reflection components. A least-squares solution $\hat{\mathbf{w}}$ was found, imposing some additional linear constraints on the weights: $0 \leq w_i \leq 1.5$, $1 \leq i \leq 6$, and $\sum_{i=1}^{6} w_i \leq 3$. The iterative optimization was done using the MATLAB function `lsqlin`. Typically, between 3 and 7 iterations were made, and for no pixel the number of iterations exceeded 16.

4.3 Results

The absolute values of the estimated weights $\hat{w}_i$ are not very illustrative, since they follow the intensity

Figure 2: *Original image (shown in RGB).*

of the signal. Therefore, relative weights are shown, which are defined as the fraction of the total sum of weights:

$$\hat{f}_i \equiv \frac{\hat{w}_i}{\sum_{j=1}^{6} \hat{w}_j}.$$

Weight fractions for all six reflection components are shown in Figure 3.

From $\hat{\mathbf{w}}$ it is possible to make a reconstruction $\tilde{\mathbf{s}}$ which contains only the reflection components of interest:

$$\tilde{\mathbf{s}} \equiv \hat{\mathbf{B}}\tilde{\mathbf{w}},$$

where $\tilde{w}_i = \hat{w}_i$ for all i corresponding to the components of interest, and $\tilde{w}_i = 0$ for all other i. For example, in Figure 4 are shown the individual contributions of surface reflection and second-order body scattering. Also, certain classes of reflections may be removed, as shown in Figure 5.

4.4 Comments on results

When assessing the results, it must be pointed out that some leaves were moving during the image acquisition[2] (especially the leaf in the bottom of the image and the one above it) and therefore show some unreliable spectra, especially on the edges.

Since no "ground truth" is available, an evaluation must be based on visual inspection. Generally, the estimated weight fractions seem believable. Perhaps the most questionable result is that the soil shows high fractions of surface reflection, since soil is not expected to be specular. There may be several explanations for the results:

- There are some grey stones, whose body reflection can not be distinguished from surface reflection.
- In shaded portions of the soil (shaded side of stones etc.), the used single-illuminant model is not good, since the bluish sky illumination is more prominent here. Therefore, in the lack of a $\mathbf{e}_{sky}\mathbf{r}_s$ endmember, the $\hat{\mathbf{e}}$ endmember may be used instead to account for the observed spectra.

Also, in general, saturated pixels (although they are few) may tend to be explained as surface reflections since their spectra are normally more flat than they should have been.

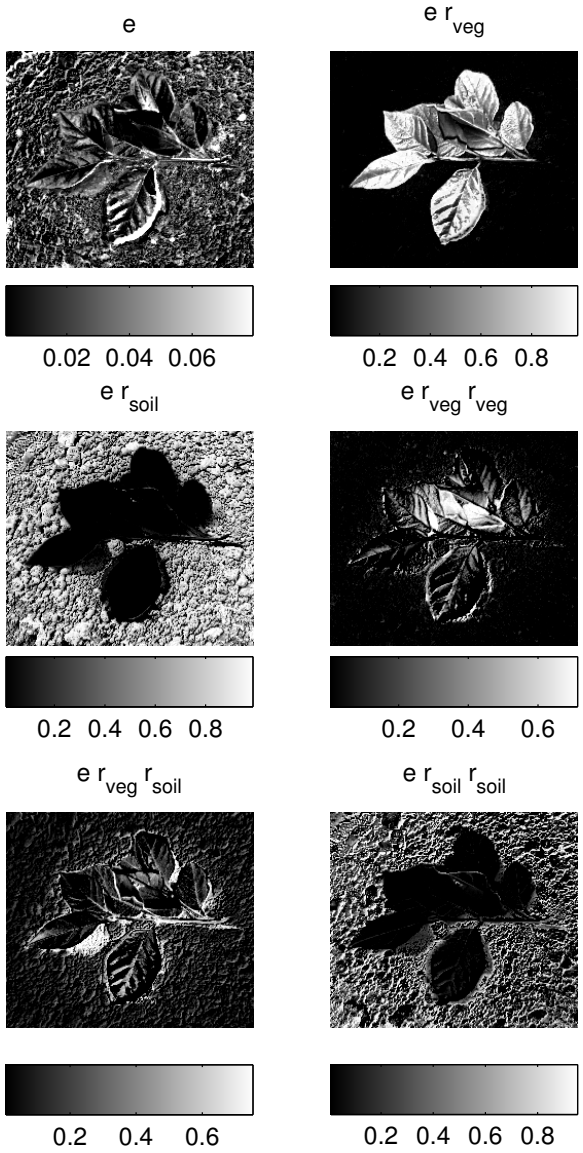


Figure 3: *Estimated weight fractions for all six reflection components. The fractions are mapped to grey scales, stretched for visualization purposes.*

[2]The total acquisition time was several seconds

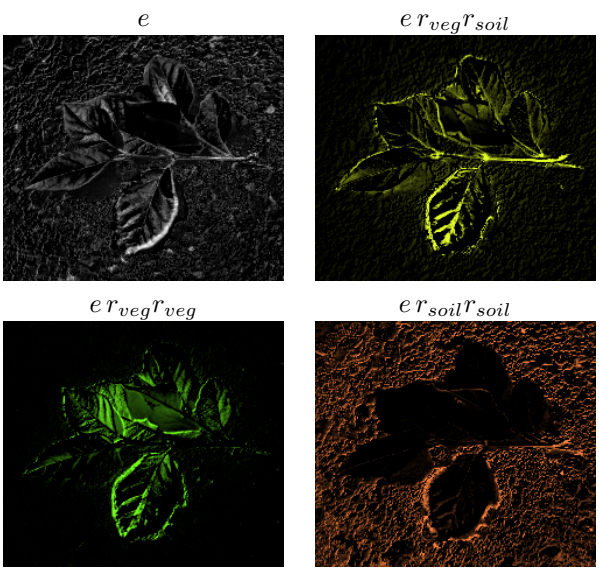


Figure 4: *Estimated contributions from surface reflection and second-order body scattering. (For visualization, the images are converted to RGB and intensity stretched.)*

Figure 5: *Image reconstructions using combinations of estimated reflection components. Top, specular reflections and second-order body scatterings. Bottom, only first-order body scatterings. (For visualization, the images are converted to RGB and intensity stretched.)*

5 Discussion

The experimental results indicate that multispectral imaging has the potential to separate multiple reflection components, including higher order scattering, with linear unmixing. Multiple scatterings is a nonlinear process in the resulting spectra, but the naive method proposed to construct and expand a linear basis from a few illuminant and material spectra proved useful.

Compared to other ways of representing imaging using linear bases (PCA, ICA, cosine transform, etc.), the proposed basis representation has some interesting properties. Instead of being "blindly" constructed from data out of pure statistical criteria (such as othogonality or independence of basis components), it is based on a physics-based model of the image formation, and therefore from the start provides a more "semantic" and physically meaningful basis representation than standard methods.

The obvious problem, of course, is how to determine the mixing basis. There are at least two aspects of this problem, 1) how many illuminant and material spectra, and how many orders of scatterings, to include in the model, and 2) to identify the shapes of the illuminant and material spectra. Since this is model-based approach, the answer must depend on the context. For an industrial scenario with well-controlled lighting and well-known homogenous objects, an appproriate model should be easily found. For an outdoor scene of vegetation and soil, a dynamic scene parameter estimation might be necessary, for example, classification of illumination conditions [2], estimation of light colour temperature(s), and estimation of soil and leaf spectra by matching image data with spectral libraries or *a priori* given models of mean and variation.

Instead of using the same basis for all pixels, a variable basis could be useful in at least two cases.

- When the variation of the material spectra is high, for example, in natural scenes.
- When within-pixel mixing of surfaces is not relevant (such as the image used in the experiment). The basis might then be reduced by excluding reflection components that can be ruled out by, for example, a rough pre-classification of the pixel.

In addition to the problem of determining the mixing model, an important numerical issue is, given a model, how to estimate the weights from an observed spectrum. Some questions are, how many bands are necessary, what is the dependence on the signal-to-noise ratio, what are the consequences of overlapping (correlated) bands, etc.. Also, some improvement on

the least-squares inversion might be achievable by weighting the observations according to their noise variances, which may be estimated from radiometric sensor calibration [5].

Finally, there is the question of how to use or interpret the estimated weights for the different reflection components. To begin with, for human observers, they may improve our image understanding and further our insight into the phenomena in play in a given scene. Also, as shown in the results, an obvious use might be to eliminate the reflection components that might confuse automated analysis algorithms. The idea that has initiated this study is to use the weights to perform within-pixel segmentation in the case of mixed pixels. This is not a straightforward task, however, and a formal study of the interplay between weights, abundance fractions, and geometrical scaling factors is still a subject to further research.

6 Conclusion

Based on the Dichromatic Reflection Model, a method has been proposed to construct a linear basis for separation of reflection components with scattering of any order. Experimental results showed that it is possible to unmix multiple reflections from multispectral image data. Possible issues for future research include how to automatically determine the mixing basis, and how to use the mixing weights for segmentation.

Acknowledgements

This work is mainly funded by the research project ACROSS, which is supported by the Danish Technical Research Council, the Danish Agricultural and Veterinary Research Council, and the Danish Ministry of Food, Agriculture and Fisheries. The work was made during a stay at Universitat Jaume I (UJI), Castellón, Spain, in spring 2005. Thanks to Filiberto Pla, UJI, and Hans Jørgen Andersen, AAU, for useful discussions, and to Arnoud Klaren and Cristina Ibáñez López, UJI, for helping with the experiment.

References

[1] Hans J. Andersen and Moritz Störring. Classifying Body and Surface Reflections Using Expectation Maximisation. In *The Digital Photography Conference (PICS 2003)*, pages 441–446, Rochester, New York, USA, May 2003.

[2] H.J. Andersen and E. Granum. Classifying Illumination Condition from Two Light Sources by Colour Histogram Assessment. *Journal of Optical Society of America A*, 17(4):667–676, April 2000.

[3] Ruzena Bajcsy, Sang Wook Lee, and Ales Leonardis. Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation. *International Journal of Computer Vision*, 17(3):241–272, 1996.

[4] Glenn J. Fitzgerald. Portable Hyperspectral Tunable Imaging System (PHyTIS) for Precision Agriculture. *Agronomy Journal*, 96:311–315, 2004.

[5] R. Healey, G.E. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[6] Y. Jang. Identification of interreflection in color images using a physics-based reflection model. In *Proceedings of 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 632–637, 1991.

[7] Nirmal Keshava and John F. Mustard. Spectral Unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, January 2002.

[8] S. Lin and Heung-Yeung Shum. Separation of diffuse and specular reflection in color images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages I–341–I–346, 2001.

[9] D.L. Schanzer. Comments on 'The least-squares mixing models to generate fraction images derived for remote sensing multispectral data' by Y.E. Shimabukuro and J.A. Smith. *IEEE Transactions on Geoscience and Remote Sensing*, 31(3):747, May 1993.

[10] S. A. Shafer. Using color to separate reflection components. *COLOR research and application*, 10(4):210–218, 1985.

[11] Y.E. Shimabukuro and J.A. Smith. The least-squares mixing models to generate fraction images derived from remote sensing multispectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 29(1):16–20, January 1991.

[12] Robby T. Tan and Katsushi Ikeuchi. Reflection Components Decomposition of Textured Surfaces using Linear Basis Functions. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, June 2005.

[13] T. Tjahjadi, D. Litwin, and Y.-H. Yang. A modified dichromatic reflection model for an analysis of interreflection. In *Proceedings, International Conference on Image Processing, 1995*, volume 2, pages 272–275, 1995.

Finding Key-Frame Motion Primitives in Human Body Gestures by Using a Density Measure

L. Reng, T.B. Moeslund, and E. Granum
Laboratory of Computer Vision and Media Technology
Aalborg University, Denmark
Email: reng@cvmt.dk

Abstract

In the last decade speech processing has been applied in commercially available products. One of the key reasons for its success is the identification and use of an underlying set of generic symbols (phonemes) constituting all speech. In this work we follow the same approach, but for the problem of human body gestures. That is, the topic of this paper is how to define a framework for automatically finding primitives for human body gestures. This is done by considering a gesture as a trajectory and then searching for points where the density of the training data is high. The trajectories are re-sampled to enable a direct comparison between the samples of each trajectory, and enable time invariant comparisons. This work demonstrates and tests the primitive's ability to reconstruct sampled trajectories. Promising test results are shown for samples from different test persons performing gestures from a small one armed gesture set.

1 Introduction

In the last decade speech synthesis and speech recognition have transferred from only being research topics into core technologies in commercially available products. One of the key reasons for this transfer is the identification and use of an underlying set of generic symbols constituting all speech, the phonemes. Phonemes are basically small sound samples that put together in the correct order can generate all the words in a particular language, for example English.

It is widely accepted that more than half of the information transmitted in a human-human interaction is done by other means than speech, and that the human body language is responsible for most of this information. Furthermore, for better human-computer interfaces to be build the computer might need to be equipped with the ability to understand the human body language [14]. Since automatic recognition of human body language is a desired ability research has been conducted in this area. Much of this research is based on defining a subset of the human body language, normally denoted "actions", and then building a classifier based on some kind of learning scheme applied to some training data. The result of the training is a sequence of values in some state-space for each action. The different learnt sequences are compared to the input data during run-time and a classification is carried out.

In some systems, however, a different approach is followed[1]. This approach is based on the idea that an action can be represented by a set of shorter (in terms of time duration) primitives. These primitives take different names such as movemes [4], atomic movements [5], activities [2], behaviors [11, 16], snippets [8], dynamic instants [15], states [3], and examplars [13].

Besides the different names used to describe the notion of motion primitives, the approaches also differ in another way, namely whether a primitive is dependent or independent on time. The approaches based on independence find their inspiration in key-frame animation. Key-frame animation is based on the idea that animating an articulated object in a time sequence is a matter of defining the configurations for a number of distinct frames (key-frames) and then interpolate all in-between frames using e.g., inverse kinematics. Mapping this concept to the problem of recognizing human body language converts the problem to a matter of recognizing a number of single configurations and ignoring all in-between configurations. This concept is sound but introduces a number of problems including the problem of defining which configurations (or key-frames) that best represent an action.

In the work by Rao *et al.* [15] the problem of recognizing dynamic hand gestures is addressed. They track a hand over time and hereby generate a trajectory in 3D space (x- and y-position, and time). They search the trajectory for significant changes, denoted dynamic instants, which are defined as instants with a high curvature. In the work by Jordi [7] the problem of finding key-frames for cyclic actions, like walking and running, is addressed. They capture

[1]These approaches are sometimes motivated directly by the notion of finding "phonemes" in the human body language.

the joint angles using an optical motion capture system and compactly represent a time sequence of such data using a point distribution model. Since the actions are cyclic they argue that the likelihood of a configuration being part of an action can be measured as the Mahalanobis distance to the mean. The key-frames are then defined as configurations where the Mahalanobis distance locally is maximum, i.e., key-frames are the least likely configurations!

The alternative to the key-frame approach is to represent the entire trajectory (one action), but doing so using a number of smaller sub-trajectories. That is, the entire trajectory through a state space is represented as opposed to only representing a number of single points. Several problems are associated with this approach, for example, how to define the length of the sub-trajectories. If too long then the primitives will not be generic. If too short the compactness of the representation is lost.

In the work by Howe *et al.* [8] the problem of capturing the 3D motion of a human using only one camera is addressed. The main body parts are tracked in 2D and compared to learned motion patterns in order to handle the inherent ambiguities when inferring 3D configurations from 2D data. The learned motion patterns are denoted ”snippets” and consist of 11 consecutive configurations. These are learned by grouping similar motion patterns in the training data. In the work by Bettinger *et al.* [1] the problem of modeling how the appearance of a face changes over time is addressed. They use an active appearance model to represent the shape and texture of a face, i.e., one point in their state-space corresponds to one instant of the shape and texture. They record and annotate a number of sequences containing facial changes. Each sequence corresponds to a trajectory in their state space. The states with the highest densities are found and used to divide the data into sub-trajectories. These sub-trajectories are modeled by Gaussian distributions each corresponding to a temporal primitive.

The different approaches found in the literature that uses the notion of motion primitives more or lees follow the structure below.

Temporal content Either only a single time instant define a primitive or a primitive is based on a consecutive number of temporal instants.

Motion capture In order to find the primitives the motion data needs to be captured. This could for example be done by an optical system or electromagnetic sensors.

Data representation What is measured by the motion capture system is normally the 3D position of the different body parts. These measurements are often represented used normalized angles. Furthermore, the velocity and acceleration might also be considered.

Preprocessing The captured data can have a very high dimensionality and can therefore be represented more compactly using, e.g., PCA. Furthermore, the data might be noisy and is therefore often filtered before further processing.

Primitives It needs to be decided how to define a primitive. Often this is done via a criteria function which local minima/maxia defines the primitives.

Application The chosen method needs to be evaluated. This can be with respect to the number of primitives versus the recognition rate, but it can also be a comparison between the original data and data synthesized using the primitives.

Our long term goal is to find a set of generic primitives that will enable us to describe all (meaningful) gestures conducted by the upper body of a human. Our approach is to investigate different data representations together with different criteria functions. We seek to find primitives for both recognition and synthesis, and evaluate the relationship between the two.

This particular paper presents the initial work towards our goal and the focus of the paper is to obtain experiences with all the topics listed above. Concretely we define a number of one-armed gestures and for each gesture we evaluate a method used to find primitives. The criteria function is based the density of a trajectory. We then use these primitives to reconstruct the complete gestures. Finally, the reconstructions are compared to reconstructions made without use of our density measure, and an optimized version of our approach.

The paper is structured as follows. In section 2 the gesture data and the applied motion capture technique are presented. In section 3 we describe how the data is normalized. In section 4 the concept behind the primitives is given. In section 5 we present the density measure used in the criteria function, and in section 6 we combine this with a distance measure and defined how the criteria function is evaluated in order to select the primitives. In section 7 the test results are presented and in section 8 a conclusion is given.

2 The Gesture Data

The gestures we are working with are inspired by the work of [12] where a set of hand gestures are defined. The gestures in [12] are primarily two-hand gestures, but we simplify the setup to one-hand gestures in order to minimize the complexity and focus on the primitives. Some of the gestures were exchanged with other more constructive ones. The final set of gestures are, as a result of this, all command gestures which can be conducted by the use of only one arm. The gestures are listed below.

Stop: Hand is moved up in front of the shoulder, and then forward (with a blocking attitude), and then lowered down.

Point forward: A stretched arm is raised to a horizontal position pointing forward, and then lowered down.

Point right: A stretched arm is raised to a horizontal position pointing right, and then lowered down.

Move closer: A stretched arm is raised to a horizontal position pointing forward while the palm is pointing upwards. The hand is then drawn to the chest, and lowered down.

Move away: Hand is moved up in front of the shoulder while elbow is lifted high, and the hand is then moved forward while pointing down. The arm is then lowered down.

Move right: Right hand is moved up in front of the left shoulder. the arm is then stretched while moved all the way to the right, and then lowered down.

Move left: Same movement as *Move right* but backwards.

Raise hand: Hand raised to a position high over the head, and then lowered down.

Each gesture is carried out a number of times by a number of different subjects, in order to have both data for interperson comparisons, and comparable data for each gesture by several different subjects.

The gestures are captured using a magnetic tracking system with four sensors: one at the wrist, one at the elbow, one at the shoulder, and one at the torso (for reference), as shown in figure 1. The hardware used is the Polhemus FastTrac [9] which gives a maximum sampling rate of $25Hz$, when using all four sensors.

In order to normalize the data and make it invariant to body size, all the collected 3-dimensional position data is converted to a time sequence of four Euler angles: three at the shoulder and one at the elbow. Besides normalizing the data, this transformation also decreases the dimensionality of the data from 12 to only 4 dimensions.

3 Normalizing the Data

In order to compare the different sequences they each need to be normalized. The goal is to normalize all the gesture trajectories so each position on a trajectory can be described by one variable t, where $t \in [0; 1]$.

The first step is to determine approximately where the gestures' endpoints are. In this experiment we have chosen to do so by defining a gesture set where all gestures are considered to both start and stop when the arm is hanging relaxed from the shoulder. A velocity threshold ensures that the small movements done between gestures is added to neither, and simplifies the separation of the individual gestures.

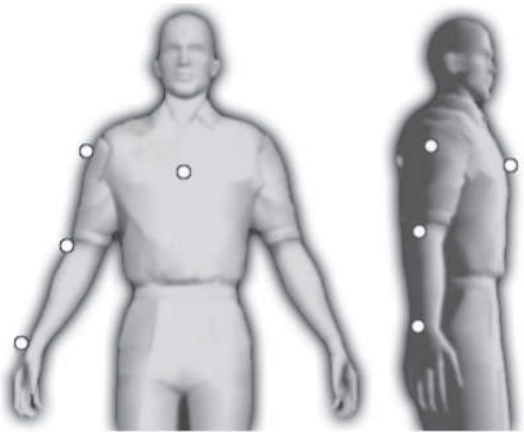


Figure 1: Placement of sensors. The figure is adapted from [10].

The trajectories are therefore homogeneously re-sampled in order to enable time invariant comparisons. This is done by interpolating each gesture, in the 4D Euler-space, by use of a standard cubic spline function. The time and velocity information is, however, still available from parameters in the new sample points, even though this is not used in this work. The homogeneously re-sampling allows for a calculation of the statistics for each gesture *and* at each sample point. Concretely, for each gesture we calculate the mean and covariance for each sample point, i.e., each instant of t. This gives the average trajectory for one gesture along with the uncertainties along the trajectory represented by a series of covariant matrices, see figure 2.

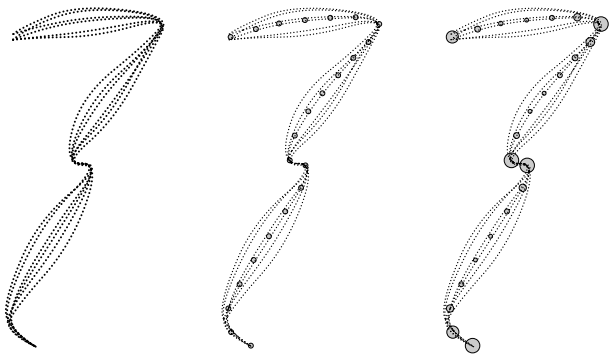


Figure 2: Six example trajectories for a fictive gesture. Left: Input after cubic spline interpolation. Middle: Input including the position of the mean points. Right: The sizes of the mean points indicate the density of the curves.

4 Defining Primitives of Human Gestures

This section gives an intuitive description of which criteria define a good primitive candidate. In order to find the primitives we apply the following reasoning. A primitive is a particular configuration of the arm, i.e., of the four Euler angles. For a configuration to qualify as a good primitive candidate the configuration must appear in all the training data, at approximately the same time. For such a configuration to exist, all the training data must vary very little at this point in space and time, which will result in a very high density of training trajectories at this position in space. The density of a particular configuration expresses how close the original sequences passed this configuration. The closer they passed the higher the density, corresponding to a good candidate. The logic behind this is very simple: At points on the reconstructed trajectory where all the training data have very little variance, we might also assume that future gestures of this kind will parse very close. It therefore makes good sense to compare an unknown trajectory to our known reconstructed trajectory, at exactly the points where all the training data trajectories laid closest, see figure 2. However, just selecting the *n* points with the highest density will result in very inefficient primitives. The point right next to a high density point is also likely to have a high density, and might therefore also be selected if density were the only criteria for the selection of primitives. One primitive is enough to direct the interpolated curve through an area, and also enough to act as control point when classifying unknown curves. So selecting more primitives at places where the trajectory already parses by will offer little to the reconstruction of the original trajectory. It is therefore also interesting to se how well each primitive can improve the reconstruction, even thou the benefits from the density measure is most visible in recognition.

In the next two sections we describe how we calculate the density measure, and how this is used to select our primitives.

5 Measuring the Density

In section 3 the points constituting each trajectory were normalized so that the trajectories for different test subjects can be compared. That is, each trajectory was re-sampled so that they each consist of the same amount of points which are aligned. We can therefore calculate the covariance matrix for each time instant.

The covariance matrices for each time instant express both how data are correlated but also how they are spread out with respect to the mean. The Mahalanobis distance expresses this relationship by defining a distance in terms of variances from a data point to the mean. It is defined as

$$r^2 = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad (1)$$

where $\mathbf{x}$ is a data point, $\boldsymbol{\mu}$ is the mean for this particular time instant, and $\mathbf{C}$ is the covariance matrix. If r is constant then equation 1 becomes a hyper ellipsoid in 4D space. The data points on its surface have the same variance-distance to the mean. The volume of a hyper ellipsoid with fixed Mahalanobis distance is a direct measure of the density of the data at this time instant. A big volume corresponds to a low density where the points are spread out, whereas a small volume corresponds to a high density as the same amount of data are located at a much smaller space. The volume of a hyper ellipsoid which is expressed as in equation 1 is given as [6]

$$V = \frac{\pi^2 \cdot r^4}{2} |\mathbf{C}|^{\frac{1}{2}} \quad (2)$$

where $|\mathbf{C}|$ is the determinant of the covariance matrix. We are not interested in the actual value of the volume but rather the relative volume with respect to the other time instants. Therefore equation 2 can be reduced to $V = |\mathbf{C}|^{\frac{1}{2}}$ and is illustrated in figure 2. Below we give an intuitive interpretation of this measure.

6 Selecting the Primitives

Above we have defined and presented a method for calculating the density measure, and are now ready to include this into one criteria function that can be evaluated in order to find the primitives. The criteria function will combine the density measure with the distance between the homogeneously re-sampled mean gesture trajectory (m) and a trajectory made by interpolating the endpoints and the first selected primitives, using a standard cubic spline function (c) for each of the four Euler angles. In order to make a direct comparison, both the mean gesture trajectory and the interpolated cubic spline trajectory were given the same amount of points. This enables a calculation of the *error*-distance (δ) between the curves for each point pair. If multiplying this error distance at each point with the density (V), we can get a distance measure much similar to the Mahalanobis.

Since the four angles might not have the same dynamic ranges and more freedom to optimize future parameters is desired, the criteria function (λ) is defined as a weighted sum of error measures (α_i) for each of the four Euler angles:

$$\lambda(t) = \omega_1\alpha_1(t) + \omega_2\alpha_2(t) + \omega_3\alpha_3(t) + \omega_4\alpha_4(t) \quad (3)$$

where the four weights $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$, and the error measure:

$$\alpha_i(t) = V_i(t) \cdot \delta_i(t)^2 \quad (4)$$

where:

$$\delta_i(t) = \sqrt{(m_i(t) - c_i(t))^2} \quad (5)$$

Given the criteria function in equation 3 we are now faced with the problem of finding the N best primitives for a given trajectory. The most dominant primitive, χ_1 is obviously defined as

$$\chi_1 = arg \max_t \ \lambda(t) \quad (6)$$

In order to find the second primitive, the first one is added to the cubic spline function (c), and the interpolated trajectory is then recalculated, so new error distance measures can be calculated, see figure 3. This procedure can be repeated until the sum of all (λ) falls below a given threshold, or the number of primitives reaches an upper threshold.

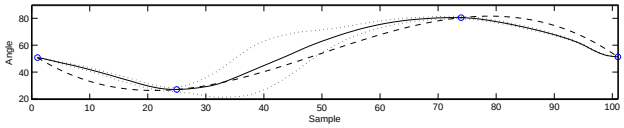


Figure 3: Calculating the error-distance for one angle. Solid: The mean gesture trajectory. Dashed: Interpolated cubic spline. Dotted: Variance of training data. Circles: Selected primitives and endpoints.

6.1 Optimizing the Primitive's Position

Placing the primitive where the density or error is largest might be a fairly good solution if the primitives are only to be used for recognition, but in respect to reconstruction that solution might be very far from optimal.

By doing a brute force recalculation of the interpolated trajectory by placing every primitive candidate in every possible position for each given number of primitives, an optimal solution should present it self for the given gesture, based on the reconstruction criteria. This method demands a very high amount of calculations and is therefore also very time consuming, and only valuable for the given data set.

Instead, tests were done with another much faster method. After each new primitive was selected by the rules described in the previous section, each selected primitive was tested in a position one step to each side along the mean gesture trajectory. Only if they could lower the total error sum, would they move to this position, and as long as just one primitive could be moved, all primitives were tested again. This method should bring the error sum to a local minimum, but not to a guarantied global minimum.

This method focuses solely on the primitives' ability to reconstruct the original trajectories, and might have an unwanted negative effect on the primitives' ability to recognize gestures, a problem that future tests might reveal. See the following section 7 for test results on both previous described methods.

7 Results

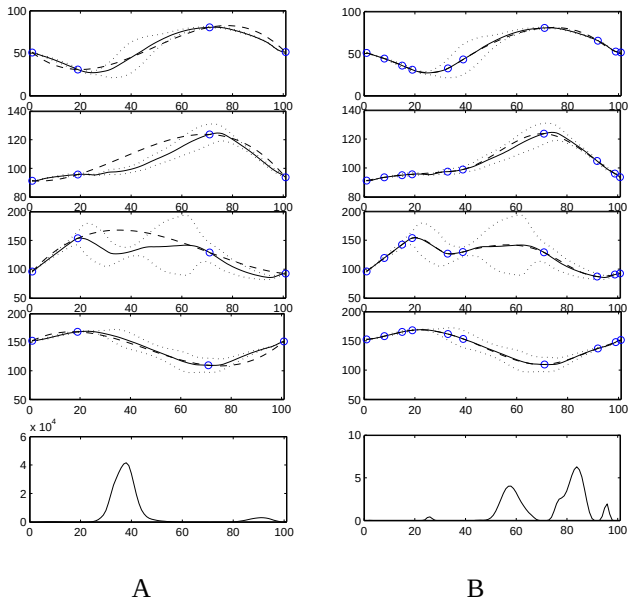


Figure 4: Reconstruction and error. Solid: The mean gesture trajectory. Dashed: Interpolated cubic spline. Dotted: Variance of training data. Circles: Selected primitives and endpoints. A: With 2 primitives. C: With 8 primitives.

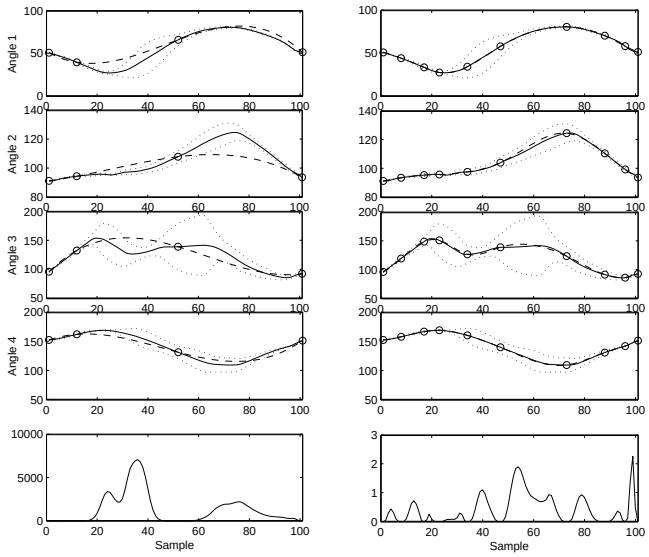
The tests described in this section were made on a training data set based on the eight one arm gestures described in section 2. Three tests persons conducted each gesture no less than ten times resulting in a total of 240 gestures[2].

The evaluation of our approach consists of two tests for each action:

- Investigate how many primitives are required in order to reconstruct the original gestures.
- Evaluate the optimization step, and determine whether or not this should be used in our continuous work.

It is our belief that the only reasonable way to evaluate whether the reconstruction of a gesture is life like enough to look natural, is to have a robot or virtual human avatar performing the reconstructed gestures before a large number of

[2]Additional 160 training gestures were made but had to be removed from the set do to extremely low signal to noise ratio.



A B

Figure 5: Reconstruction and error (Optimizted version). Solid: The mean gesture trajectory. Dashed: Interpolated cubic spline. Dotted: Variance of training data. Circles: Selected primitives and endpoints. A: With 2 primitives. C: With 8 primitives.

test persons, and having these evaluate the result. This was however not within range of our possibilities at this point in our research. Instead, all reconstructions were evaluated by the research group from a large number of graphs such as those shown in figures 4 and 5, and a number of rotating 3D curves depicturing the trajectories in three of the four Euler angles. The graphs show the four angle spaces and error measure of the gesture *Move Left*, with two endpoints and 2, and 8 primitives. Figure 4 show the result of the reconstruction without the optimizing step, where as 5 depicture the reconstruction of the exact same angle spaces, but with the optimization.

The total error sum between original and reconstructed trajectory of each gesture, was collected with the number of primitives ranging from 1-10. Figure 6 shows four graphs of the decreasing error sums: One there the primitives are selected only as the point with the largest distance to the original trajectory. Second graph shows the same, but where the density measure have been used in the selection process. The last two graphs show each of these methods after the optimization method has been conducted.

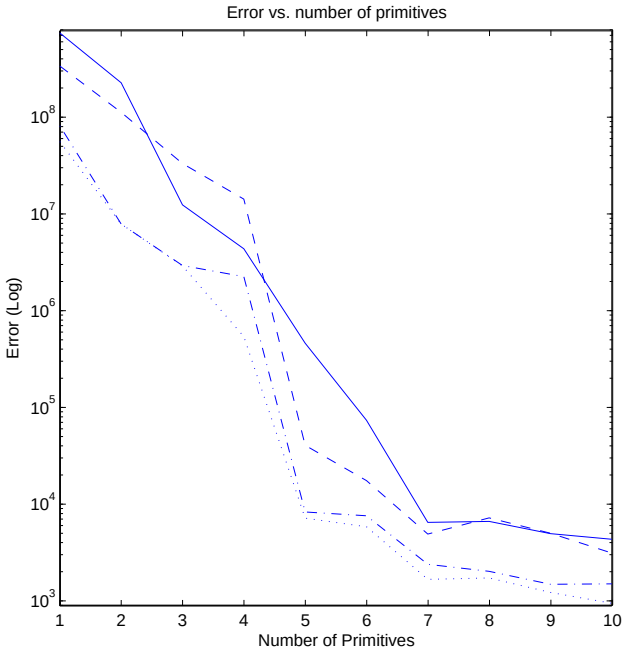


Figure 6: Logaritmic scale of error vs. number of primitives. Solid: Reconstruction error after primitive selection without the density measure. Dashed: Reconstruction error after primitive selection with the density measure. Dash-dot: Reconstruction error after primitive selection without the density measure, but with optimization. dotted: Reconstruction error after primitive selection with the density measure and optimization.

8 Conclusion

In this paper we have presented a framework for automatically finding primitives for human body gestures. A set of gestures is defined and each gesture is recorded a number of times using a commercial motion capture system. The gestures are represented using Euler angles and normalized. The normalization allows for calculation of the mean trajectory for each gesture along with the covariance of each point of the mean trajectories. For each gesture a number of primitives are found automatically. This is done by comparing the mean trajectories and cubic spline interpolated reconstructed trajectories by use of a error measurement based on density. Our framework were implemented in two slightly different versions, were the optimizted but slower version proved to be superior in repect to reconstruction. Figure 6 clearly shows that the density measure is not only usable for recognition but will also improve reconstruction by approximately a factor two for four or more primitives, as long as there position is optimized for the given number of primitives. It is a clear indication that the density measure should be taken into consideration in the future. Even thou the figure show that the density measure might result in larger errors in the reconstruction without the optimization, it will clearly have a large advantage when using the same primitives for recognition. Its is still hard to say exactly how many primitives are needed to get a natural reconstruction of a given gesture. But our tests indicate that somewhere between five and ten should be sufficient.

8.1 Near Future Work

It is my hope that I will be able to collect a larger dataset and combine the reconstruction scores of the primitives with some kind of recognition scores as well. Further more, I intend to extend the testing to include comparisons between personal primitives and none-personal primitives. Hopefully, all in time for the presentation at the conference in August 2005.

References

[1] F. Bettinger and T.F. Cootes. A Model of Facial Behaviour. In *IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17 - 19 2004.

[2] A.F. Bobick. Movemnet, Activity, and Action: The Role of Knowledge in the Perception of Motion. In *Workshop on Knowledge-based Vision in Man and Machine*, London, England, Feb 1997.

[3] A.F. Bobick and J. Davis. A Statebased Approach to the Representation and Recognition of GEstures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12), 1997.

[4] C. Bregler. Learning and Recognizing Human Dynamics in Video Sequences. In *Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997.

[5] L. Campbell and A.F. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *International Conference on Computer Vision*, Cambridge, Massachusetts, 1995.

[6] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley & Sons, Inc., 2 edition, 2001.

[7] J. Gonzalez. *Human Sequence Evaluation: The Key-Frame Approach*. PhD thesis, Universitat Autonoma de Barcelona, Barcelona, Spain, 2004.

[8] N.R. Howe, M.E. Leventon, and W.T. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. In *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.

[9] http://polhemus.com/. Polhemus, three-dimensional scanning, position/orientation tracking systems, eye tracking and head tracking systems., January 2005.

[10] http://www.3dcrimescene.com/. Typical cold case reconstruction., January 2005.

[11] O.C. Jenkins and M.J. Mataric. Deriving Action and Behavior Primitives from Human Motion Data. In *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Sep 2002.

[12] A. Just and S. Marcel. HMM and IOHMM for the Recognition of Mono- and Bi-Manual 3D Hand Gestures. In *ICPR workshop on Visual Observation of Deictic Gestures (POINTING'04)*, August 2004.

[13] A. Kale, N. Cuntoor, and R. Chellappa. A Framework for Activity-Specific Human Recognition. In *International Conference on Acoustics, Speech and Signal Processing*, Orlando, Florida, May 2002.

[14] T.B. Moeslund and E. Granum. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3), 2001.

[15] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *International Journal of Computer Vision*, 50(2), 2002.

[16] C.R. Wren and A.P. Pentland. Understanding Purposeful Human Motion. In *International Workshop on Modeling People at ICCV'99*, Corfu, Greece, September 1999.

Deformable Models for Eye Tracking

Martin Vester-Christensen, Denis Leimberg, Bjarne Kjær Ersbøll and Lars Kai Hansen
Department of Informatics and Mathematical Modelling
Technical University of Denmark
mvc@imm.dtu.dk, denis@kultvizion.dk, be@imm.dtu.dk, lkh@imm.dtu.dk

Abstract

A deformable template method for eye tracking on full face images is presented. The strengths of the method are that it is fast and retains accuracy independently of the resolution. We compare the method with a state of the art active contour approach, showing that the heuristic method is more accurate.

1 Introduction

Eye Tracking is the process of finding and tracking the eye of a human in a sequence of images. Specifically finding and tracking the iris or pupil can be used to infer the direction of interest of the human subject, this is denoted *gaze*.

Gaze is very important for human communication and also plays an increasing role for human computer interaction. Gaze can play a role, e.g., in understanding the emotional state for humans [1, 2], synthesizing emotions [5], and for estimation of attentional state [16]. Specific applications include devices for the disabled, e.g., using gaze as a replacement for a computer mouse and driver awareness monitoring to improve traffic safety [8].

It has been noted that the high cost of good gaze detection devices is a major road block for broader application of gaze technology, hence, there is a strong motivation for creating systems that are simple, inexpensive, and robust [7].

Eye tracking is an active area of research. COGAIN is a network of excellence on Communication by Gaze Interaction, supported by the European Commission's IST 6th framework program. COGAIN integrates cutting-edge expertise on interface technologies for the benefit of users with disabilities. The network aims to gather Europe's leading expertise in eye tracking integration with computers in a research project on assistive technologies for citizens with motor impairments[3]. The authors of this paper are members of this network, and it summarizes research presented in[11].

Figure 1: Examples of the dataset. The region surrounding the eyes can be found in various ways. We use a head tracking algorithm[8] based on Active Appearance Models. A subimage is extracted and subsequently processed by the eye tracking algorithms.

The paper is organized as follows. First a brief review of some of the methods used for eye tracking is given in section 2. Section 3 describes the proposed deformable template method. Section 4 describes the EM-contour method from [7] with additional constraints on the model. The two models are compared in section 5. Finally some concluding remarks are drawn in section 6.

2 Recent Work

Detection of the human eye is a difficult task due to a weak contrast between the eye and the surrounding skin. As a consequence, many existing approaches uses close-up cameras to obtain high-resolution images[7][19]. However, this imposes restrictions on head movements. The problem can be overcome by use of a two camera

setup[18][20]. One camera covering the head and controlling a second camera, which focuses on one eye of the person. Matsumoto and Zelinsky[12] utilizes template and stereo matching.

In many existing approaches the shape of iris is modeled as a circle [9][10][12][20]. Since the shape and texture of the object is known, a template model can be used with advantage[8][15]. J. Gracht et al.[17] utilizes an iris template generated by a series of wavelet filtering.

Wang et al.[18] detects the iris using thresholding, morphology and vertical edge operators. An ellipse is fitted to the resulting binary image.

A probabilistic formulation of eye trackers has the attraction that uncertainty is handled in a systematic fashion. Xie et al.[20] utilizes a Kalman filter with purpose to track the eyes. The eye region is detected by thresholding and the center of an eye is used for motion compensation. The center of this iris is chosen as tracking parameter, while the gray level of the circle modeled eye is chosen as measurement[21]. Hansen and Pece propose an active contour model combining local edges along the contour of the iris[7]. The contour model is utilized by a particle filter.

A generative model explaining the variance of the appearance of the eye is developed by Moriyama et al.[13]. The system defines the structures and motions of the eye. The structure represents information regarding size and color of iris, width and boldness of eyelid etc. The motion is represented by the position of upper and lower eyelids and 2D position of the iris. Witzner et al. utilizes an Active Appearance Model[6].

Based on the center of iris estimate, the gaze direction can be computed utilizing various methods. Stiefelhagen et al.[15] utilizes a neural network with the eye image as input. Witzner et al.[6] uses a Gaussian process interpolation method for inferring the mapping from image coordinates to screen coordinates. Ishikawa et al. [8] exploits a geometric head model, which translates from 2D image coordinates to a direction in space relative to the initial frame.

The present paper is inspired by the line of thinking mentioned above. We focus on some of the image processing issues. In particular we propose a robust algorithm for swift eye tracking in low-resolution video images. We compare this algorithm with a proven method[7] and relate the pixel-wise error to the precision of the gaze determination.

3 Deformable Template Matching

Modeling the iris as a circle is well-motivated when the camera pose coincides with the optical axis of the eye. When the gaze is off the optical axis, the circular iris is rotated in 3D space, and appears as an ellipse in the image plane. Thus, the shape of the contour changes as a function of the gaze direction and the camera pose. The objective is then to fit an ellipse to the pupil contour, which is characterized by a darker color compared to the iris. The ellipse is parameterized,

$$\mathbf{x} = (c_x, c_y, \lambda_1, \lambda_2, \theta)\,, \tag{1}$$

where (c_x, c_y) is the ellipse centroid, λ_1 and λ_2 are the lengths of the major and minor axis respectively. θ is the orientation of the ellipse.

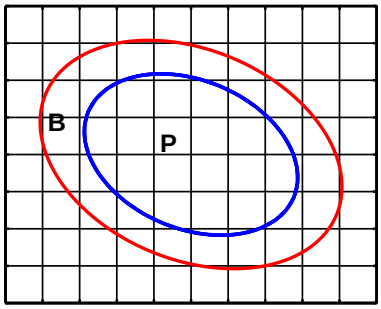


Figure 2: The deformable template model. Region P is the inner circle, and region B is the ring around it.

The model proposed here is based on the relationship between pixel values in two regions, see figure 2. The pupil region P is the part of the image I spanned by the ellipse parameterized by $\mathbf{x}$. The background region B is defined as the pixels inside an ellipse, surrounding but not included in P, as seen in figure 2. When region P contains the entire object, B must be outside the object, and thus the difference in average pixel intensity is maximal. To ensure equal weighting of the two regions, they have the same area. The area of the inner ellipse P is $A_P = \pi\lambda_1\lambda_2$. The shape parameters of B should satisfy the constraint on the area $A_{B/P} - A_P = A_P$. As a consequence, the param-

eters is defined as $\mathbf{x}_B = \left(c_x, c_y, \sqrt{2}\lambda_1, \sqrt{2}\lambda_2, \theta\right)$, while $\mathbf{x}_P$ is defined as (1).

The pupil contour can now be estimated by minimizing the cost function,

$$\mathcal{E} = \mathrm{Av}(P) - \mathrm{Av}(B), \tag{2}$$

where $\mathrm{Av}(B)$ and $\mathrm{Av}(P)$ are the average pixel intensities of the background - in this case the iris - and pupil region respectively.

The model is deformed by Newton optimization given an appropriate starting point. Due to rapid eye movements[14], the algorithm may break down if one uses the previous state as initial guess of the current state, since the starting point may be too far from the true state. As a consequence, we use a simple 'double threshold' estimate of the pupil region as starting point.

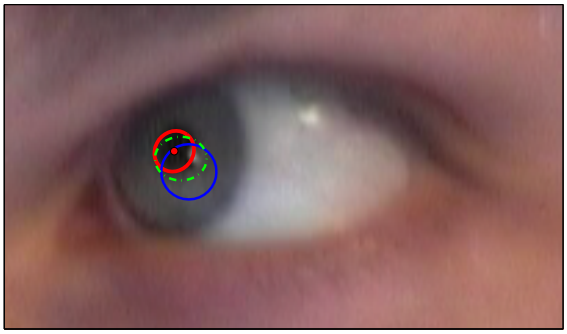


Figure 3: The blue ellipse indicates the starting point of the pupil contour. The template is iteratively deformed by an optimizer; one of the iterations is depicted in green. The red ellipse indicates the resulting estimate of the contour.

An example of the optimization of the deformable model is seen in figure 3.

3.1 Constraining the Deformation

Although a deformable template model is capable of catching changes in the pupil shape, there are also some major drawbacks. Corneal reflections, caused by illumination, may confuse the algorithm and cause it to deform unnaturally. In the worst case, the shape may grow or shrink until the algorithm collapses.

We propose to constrain the deformation of the model in the optimization step by adding a regularization term. Assume the parameters defining an ellipse is normally distributed with mean μ and covariance $\mathbf{\Sigma}$. The prior distribution of these parameters are then defined,

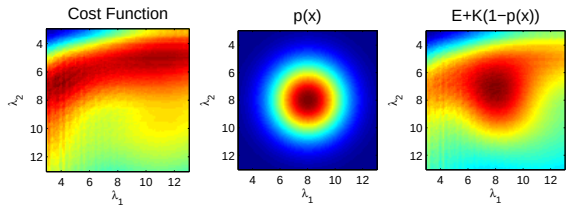


Figure 4: Given an appropriate starting point $\mathbf{x}$. The pose and orientation are kept fixed, while the shape parameters are varied. Note that the surface plots are not - as expected - smooth. This is due to rounding in the interpolation when evaluating the image evidence of the deformable template. *(Left)* The image confidence given the state - warmer colors means more likely. *(Middle)* The prior probability is a normal distribution with a given mean value μ and covariance Σ. *(Right)* Combining the image evidence and prior according to (4) yields the constrained estimate.

$$p(\mathbf{x}) = \mathcal{N}(\mu, \mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right), \tag{3}$$

where the normalization factor has been omitted. The mean and covariance are estimated in a training sequence. At last the optimization of the deformable template matching method is constrained by adding a regularization term,

$$\mathcal{E} = \mathrm{Av}(P) - \mathrm{Av}(B) + \mathcal{K}\left(1 - p(\mathbf{x})\right), \tag{4}$$

where $\mathcal{K}$ is the gain of the regularization term.

The relevance of constraining the deformation is visualized in figure 4. A suitable starting point $\mathbf{x}$ is chosen. The pose and orientation are kept fixed, while the shape parameters are varied. In this case the true shape parameters λ_1 and λ_2 are approximately eight. The image confidence as a function of the shape parameters is depicted to the left, while the prior distribution is seen in the middle of figure 4. Combining the image confidence with a prior according to (4) yields the constrained estimate, which is depicted to the right in figure 4.

By use of the shape constraints, we incorporate prior knowledge to the solution. The robustness is increased considerably and the parameters are constrained to avoid the algorithm to break down due to infinite increase or decrease of parameters.

The deformable template matching method is seen applied with and without constraints in figure 5. The constrained estimate is seen to be less sensitive to noise due to reflections.

Figure 5: The deformable template matching method applied without constraints is seen in green, while the red ellipse depicts the constrained version . The constrained estimate is seen to be less sensitive to noise due to reflections.

4 EM Contour Tracking

The iris is circular and characterized by a large contrast to the sclera. Therefore, it seems obvious to use a contour based tracker. Witzner et al.[7] describe an algorithm for tracking using active contours and particle filtering. A generative model is formulated which combines a dynamic model of state propagation and an observation model relating the contours to the image data. The current state is then found recursively by taking the sample mean of the estimated posterior probability.

The proposed method in this paper is based on [7], but extended with constraints and robust statistics.

4.1 The Dynamic Model

The dynamic model describes how the iris moves from frame to frame. Again, the iris is modeled as an ellipse and the state vector $\mathbf{x}$ consist of the five parameters defining an ellipse as defined in equation 1.

To define the problem of tracking, consider the evolution of the state sequence

$$\mathbf{x}_{t+1} = \mathbf{f}_{t+1}\{\mathbf{x}_t, t \in \mathbb{N}\}, \tag{5}$$

of a target, given by

$$\mathbf{x}_{t+1} = \mathbf{f}_{t+1}(\mathbf{x}_t, \mathbf{v}_t), \tag{6}$$

where $\mathbf{f}_{t+1}$ is a possibly non-linear function of the state $\mathbf{x}_t$ and $\{\mathbf{v}_t, t \in \mathbb{N}\}$ is an independent identically distributed process noise sequence. The objective of tracking is to recursively estimate $\mathbf{x}_{t+1}$ from the measurements,

$$\mathcal{M}_{t+1} = \mathbf{h}_{t+1}(\mathbf{x}_{t+1}, \mathbf{n}_{t+1}), \tag{7}$$

where $\mathbf{h}_{t+1}$ is a possibly non-linear function and $\{\mathbf{n}_{t+1}, t \in \mathbb{N}\}$ is an i.i.d measurement noise sequence.

The pupil movements can be very rapid and is therefore modeled as Brownian motions(AR(1)). Thus the evolution of the state sequence (6) is modeled,

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{\Sigma}_t), \tag{8}$$

where $\mathbf{\Sigma}_t$ is the time dependent covariance matrix of the noise. The time dependency compensates for scale changes, which affects the amount of movement. Larger movements is expected when the ellipse appears large, since the position of the eye is nearer to the camera. Contrary, when the eye is farther from the camera, smaller movements are expected. Hence, the first two diagonal elements of $\mathbf{\Sigma}_t$ corresponding to c_x and c_y are assumed to be linear dependent on previous sample mean.

4.2 The Observation Model

The observation model consists of two parts; a geometric component defining a probability density function over image locations of contours and a texture component defining a pdf over pixel gray level differences given a contour location. The geometric component models the deformations of the iris by assuming Gaussian distribution of all sample points along the contour. The gray level information is gathered by sampling a discrete set of points along the normals of all contour sampling points. Both components are joined and marginalized to produce a test of the hypothesis that there is a true contour present. The contour maximizing the combined hypotheses is chosen, see [7] for details.

4.3 Active Contour Tracking

The probabilistic formulation has the attraction that uncertainty is handled in a systematic fashion - Increased uncertainty results the particles to be drawn from a wider distribution, while increased confidence results the particles to be drawn from a narrower distribution.

The prediction stage involves using the system model (6) to obtain the prior pdf of the state at time $t+1$,

$$p(\mathbf{x}_{t+1}|\mathcal{M}_t) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{M}_t)\mathrm{d}\mathbf{x}_t \quad (9)$$

The observation $\mathcal{M}_t$ is independent of the previous state $\mathbf{x}_{t-1}$ and previous observation $\mathcal{M}_{t-1}$ given the current state $\mathbf{x}_t$. At time step $t+1$ a measurement $\mathcal{M}_{t+1}$ becomes available. This is used to update the prior via Bayes' rule,

$$p(\mathbf{x}_{t+1}|\mathcal{M}_{t+1}) \propto p(\mathcal{M}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\mathcal{M}_t). \quad (10)$$

With this in mind, the tracking problem is stated as a Bayesian inference problem by use of (9) and (10).

Particle filtering is used with the purpose to estimate the filtering distribution $p(\mathbf{x}_t|\mathcal{M}_t)$ recursively. This is done through a random weighted sample set $\mathcal{S}_t^N = \{(\mathbf{x}_t^n, \pi_t^n)\}$, where n is the n^{th} sample of a state at time t weighted by π_t^n. The samples are drawn from the prediction prior distribution $p(\mathbf{x}_{t+1}|\mathcal{M}_t)$. The samples are weighted proportionally to the observation likelihood $p(\mathcal{M}_t|\mathbf{x}_t)$ given by the contour hypotheses. This sample set propagates into a new sample set $\mathcal{S}_{t+1}^N$, which represents the posterior probability distribution function $p(\mathbf{x}_{t+1}|\mathcal{M}_{t+1})$ at time $t+1$.

4.4 Constraining the Hypotheses

Corneal reflections, caused by illumination, may confuse the algorithm to weigh some of the hypotheses unreasonably high compared to others. This issue is illustrated left in figure 6, where the relative normalized weighting is colored in a temperature scale - Blue indicates low, while red high scores. By using robust statistics, these hypotheses are treated as outliers and therefore rejected.

The contour algorithm may fit to the sclera rather than the iris. This is due to the general formulation of absolute gray level differences $\Delta\mathcal{M}$[4], which seeks to detect contours in a general sense. An example is depicted in figure 7, where the image evidence of the contour surrounding the sclera is greater than the one around the iris. It turns out that for a large number of particles, the maximum likelihood estimate prefers the contour around the white sclera when the gaze is turned towards the sides.

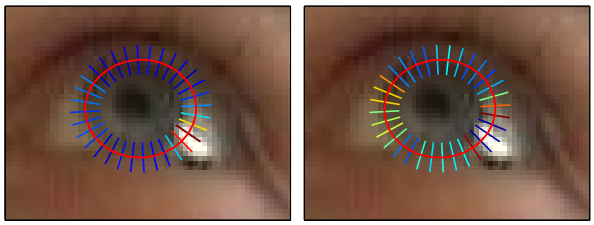


Figure 6: The relative normalized weighting of the hypotheses regarding one particle are colored in a temperature scale - Blue indicates low, while red high scores. (*Left*) Corneal reflections cause very distinct edges. Thus some hypotheses are weighted unreasonable high, which may confuse the algorithm. (*Right*) By use of robust statistics outliers are rejected. This results in a better and more robust estimate of the hypotheses regarding the contour.

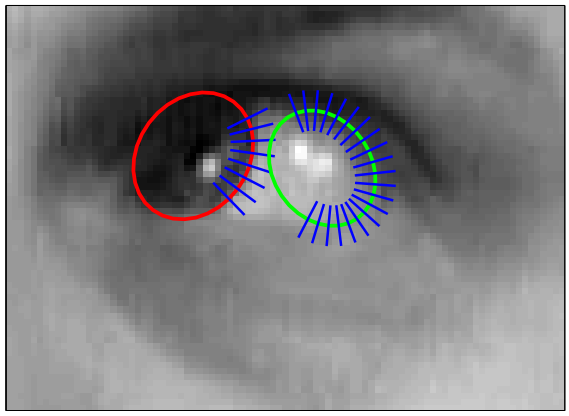


Figure 7: This figure illustrates the importance of the gray level constraint. Due to the general formulation of absolute gray level differences, the right contour has a greater likelihood, and the algorithm may thus fit to the sclera. Note the low contrast between iris and skin.

As a consequence, we propose to constrain the hypotheses. Intuitively, the average intensity value of the inner ellipse could be compared to some defined outer region as seen in expression (2). This is a poor constraint due to corneal reflection causing white blobs in the pupil area. The robustness of the active contour algorithm is increased by weighing the belief of hypotheses and utilizing robust statistics to reject outliers.

We propose to weigh the hypotheses through a sigmoid function, applied on the measurement line $\mathcal{M}$, defined as,

$$\mathcal{W} = \left(1 + \exp\left(\frac{\mu_i - \mu_o}{\sigma_w}\right)\right)^{-1} \quad (11)$$

where σ_w adjust the slope of weighting function, μ_i and μ_o are the mean values of the inner and outer sides of the contour respectively. The function is exemplified in figure 8. This has the

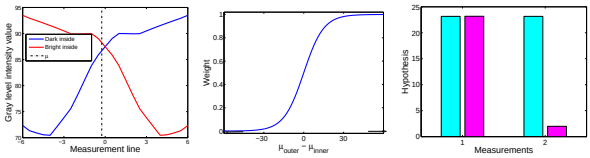


Figure 8: (*Left*) The two lines depicts the gray level intensity of two measurement lines - The blue one where the inner part of the ellipse is dark, and the red in the reverse case. (*Middle*) The shifted hyperbolic tangents is utilized as weighting function. Note, the limit values are in range $[-255; 255]$. (*Right*) The cyan bars indicates the hypothesis value before weighting, while the pink is after. Measurement *1* - The blue line - is nearly unchanged, while *2* - the red line - is suppressed.

effect of decreasing the evidence when the inner part of the ellipse is brighter than the surroundings. In addition, this relaxes the importance of the hypotheses along the contour around the eyelids, which improves the fit.

4.5 Maximum a Posteriori Formulation

The dynamic model may, in certain outlier cases, grow or shrink the contour to a degree, from where the algorithm gets lost. As a consequence, we propose to constrain on the shape of the ellipse in analogy to section 3.1. The parameters defining an ellipse is assumed normal distributed with mean μ and covariance $\mathbf{\Sigma}$. The prior distribution of these parameters are then defined,

$$p(\mathbf{x}) = \mathcal{N}(\mu, \mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right), \tag{12}$$

where the normalization factor has been omitted. The mean and covariance are estimated in a training sequence.

Combining the priors - presented in this section - with the likelihood, results in the *Maximum a Posteriori* formulation (MAP), where the goal is to maximize,

$$p(\mathbf{x}|\mathcal{M}) \propto p(\mathcal{M}|\mathbf{x})p(\mathbf{x}). \tag{13}$$

By incorporation of prior knowledge about the shape, with the prediction prior and observation likelihood (10), the robustness increases considerably and the parameters are constrained to avoid the algorithm to break down due to infinite increase or decrease of parameters.

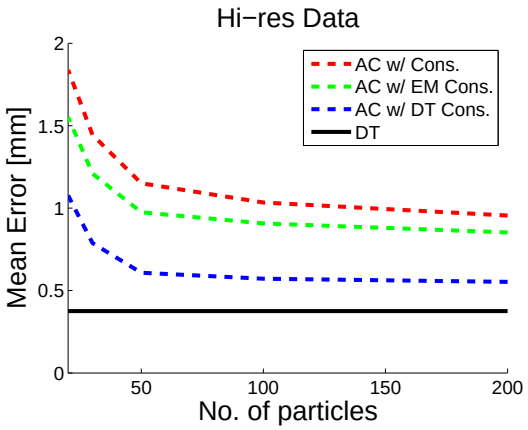


Figure 9: The error of the algorithms as a function of the number of particles for the high resolution data.

5 Results

A number of experiments have been performed with the proposed methods. We wish to investigate the importance of image resolution. Therefore the algorithms are evaluated on two datasets. One containing close up images, and one containing a down-sampled version hereof.

The algorithms estimate the center of the pupil. For each frame the error is recorded as the difference between a hand annotated ground truth and the output of the algorithms. This may lead to a biased result due to annotation error. However, this bias applies to all algorithms and a fair comparison can still be made.

Figure 9 and 10 depicts the error as a function of the number of particles used, for low resolution and high resolution images respectively. The errors for three different active contour (AC) algorithms are shows; basic, with EM refinement, with deformable template (DT) refinement. The error of the deformable template (DT) algorithm, initialized by double threshold, is inserted into the plot.

It can be seen that the proposed constraints on the active contour generally improves the accuracy of the fit. The refinement by the deformable template performs better than the EM method. The cost is an increased number of computations, which is resolution dependent. Nonetheless, the deformable template method, initialized by double thresholding, is seen to outperform all active contour algorithms.

The table in figure 5 lists the mean error in accuracy in centimeters and degrees. Also listed is the computation time in frames per section of

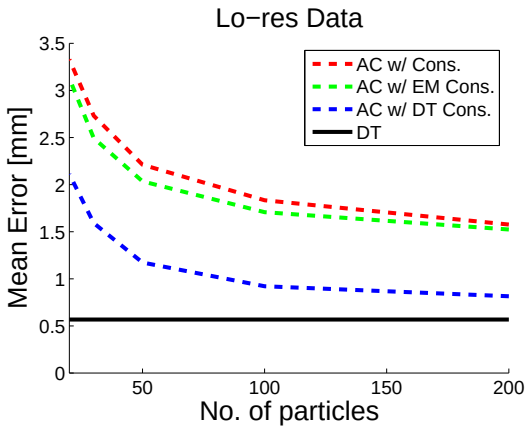


Figure 10: The error of the algorithms as a function of the number of particles for the low resolution data.

Hi-res	$E(x,y)$[mm]	$E(\theta)$	[frame/s]
AC	0.9	4.1	0.54
AC w/EM	0.8	3.7	0.49
AC w/DT	0.5	2.3	0.25
DT	0.3	1.4	2.2
Lo-res	$E(x,y)$[mm]	$E(\theta)$	[frame/s]
AC	1.5	7.3	0.57
AC w/EM	1.5	6.9	0.55
AC w/DT	0.8	3.7	0.49
DT	0.5	2.3	8.4

Table 1: Speed and precision comparison of the algorithms. The active contour uses 200 particles.

a Matlab implementation run on a 2.4Ghz PC. In general, the accuracy improves with high resolution as seen in table 5. However, the methods utilizing deformable template matching are less sensitive. The computation time for the basic active contour and EM refinement methods are independent of resolution. A significant increase in speed is noticed for the deformable template methods.

6 Conclusion

In this paper we have presented heuristics for improvement of the active contour method proposed by [7]. We have shown increased performance by using the prior knowledge that the iris is darker than its surroundings. This prevents the algorithm from fitting to the sclera as seen in figure 7.

Also presented is a novel approach to eye tracking based on a deformable template initialized by a simple heuristic. This enables the algorithm to overcome rapid eye movements. The active contour method handles these by broadening the state distribution and thus recovering the fit in a few frames. Furthermore, the accuracy is increased by fitting to the pupil rather than iris. This is particularly the case when a part of the iris is occluded as seen in figure 11.

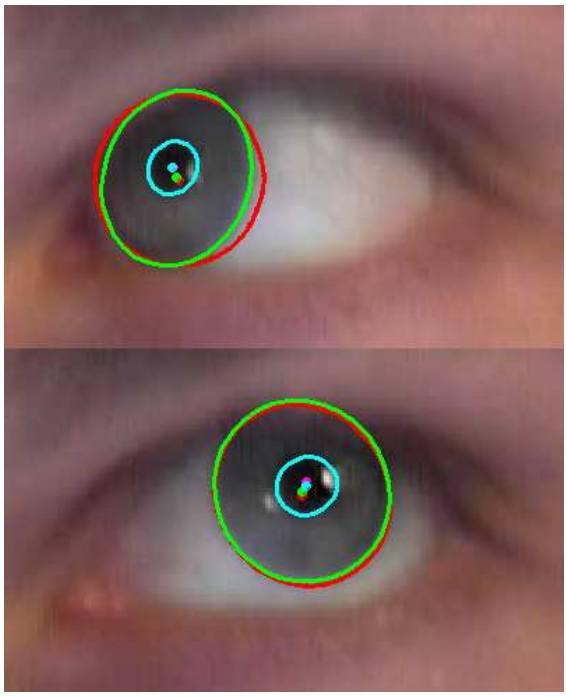


Figure 11: The resulting fit on two frames from a sequence - the red contour indicates the basic active contour, green indicates the EM refinement and the cyan indicates the deformable template initialized by the heuristic method. The top figure illustrates the benefit fitting to the pupil rather than the iris. Using robust statistic the influences from corneal reflections on the deformable template fit are ignored as depicted in the bottom image.

It is shown that the deformable template model is accurate independent of resolution and it is very fast for low resolution images. This makes it useful for head pose independent eye tracking.

Acknowledgements

We wish to thank Hans Bruun Nielsen, Department of Informatics and Mathematical Modelling - Technical University of Denmark, for providing the optimization implementation. Additionally, we wish to thank Dan Witzner Hansen, IT-University of Copenhagen, for inspiring and insightful discussions. This research is supported by the IST Network of Excellence - Communication by Gaze Interaction (COGAIN).

References

[1] Jr. Adams, R.B. and R.E. Kleck. Perceived gaze direction and the processing of facial displays of emotion. *Psychological Science*, 2003.

[2] R.B. Jr. Adams, H.L. Gordon, A.A. Baird, N. Ambady, and R.E. Kleck. Effects of gaze on amygdala sensitivity to anger and fear faces. *Science*, 300:1536–1537, 2003.

[3] Communication by Gaze Interaction. www.cogain.org.

[4] Jens Michael Carstensen. *Image analysis, vision and computer graphics*. Technical University of Denmark, Kgs. Lyngby, 2 edition, 2002.

[5] Jonathan Gratch and Stacy Marsella. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. *Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada*, June 2001.

[6] D. W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen, and M. B. Stegmann. Eye typing using markov and active appearance models. In *IEEE Workshop on Applications of Computer Vision - WACV*, pages 132–136, dec 2002.

[7] Dan W. Hansen and Arthur E. C. Pece. Iris tracking with feature free contours. In *Proc. workshop on Analysis and Modelling of Faces and Gestures: AMFG 2003*, October 2003.

[8] Takahiro Ishikawa, Simon Baker, Iain Matthews, and Takeo Kanade. Passive driver gaze tracking with active appearance models. In *Proceedings of the 11th World Congress on Intelligent Transportation Systems*, October 2004.

[9] T. Kawaguchi, D. Hidaka, and M. Rizon. Detection of eyes from human faces by hough transform and separability filter. In *ICIP00*, pages Vol I: 49–52, 2000.

[10] Kyung-Nam Kim and R.S. Ramakrishna. Vision-based eye-gaze tracking for human computer interface, 1999.

[11] D. Leimberg and M. Vester-Christensen. Eye tracking. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2005. Supervised by Prof. Lars Kai Hansen.

[12] Y. Matsumoto and A. Zelinsky. An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In *AFGR00*, pages 499–504, 2000.

[13] Tsuyoshi Moriyama, Jing Xiao, Jeffrey Cohn, and Takeo Kanade. Meticulously detailed eye model and its application to analysis of facial image. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pages 629 – 634, 2004.

[14] J. Pelz, R. Canosa, J. Babcock, D. Kucharczyk, A. Silver, and D. Konno. Portable eyetracking: A study of natural eye movements, 2000.

[15] Rainer Stiefelhagen, Jie Yang, and Alex Waibel. Tracking eyes and monitoring eye gaze. In *Workshop on Perceptual User Interfaces*, Banff, Canada, October 1997.

[16] Rainer Stiefelhagen, Jie Yang, and Alex Waibel. Estimating focus of attention based on gaze and sound. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–9. ACM Press, 2001.

[17] Joseph van der Gracht, V. P. Pauca, Harsha Setty, Ramkumar Narayanswamy, Robert Plemmons, Sudhakar Prasad, and Todd Torgersen. Iris recognition with enhanced depth-of-field image acquistion. volume 5438, pages 120–129. SPIE, 2004.

[18] J.G. Wang and E. Sung. Study on eye gaze estimation. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernectics*, 32(3):332–350, June 2002.

[19] Jian-Gang Wang, Eric Sung, and Ronda Venkateswarlu. Eye gaze estimation from a single image of one eye. In *ICCV*, pages 136–143, 2003.

[20] X. Xie, R. Sudhakar, and H. Zhuang. A cascaded scheme for eye tracking and head movement compensation. *T-SMC*, A28:487–490, 1998.

[21] X.D. Xie, R. Sudhakar, and H.Q. Zhuang. Real-time eye feature tracking from a video image sequence using kalman filter. *SMC*, 25(12):1568–1577, December 1995.

A Comparison of Active-Contour Models Based on Blurring and on Marginalization

Arthur E.C. Pece
Heimdall Vision

Abstract

Many different active-contour methods have been proposed, but very few comparisons between alternative methods have been carried out. Further, most of these comparisons have been either exclusively theoretical or exclusively experimental. This paper presents a combined theoretical and experimental comparison between two recently proposed contour models. The two models are put into a common theoretical framework and performance comparisons are carried out on a vehicle tracking task in the PETS test sequences. Using a Condensation tracker helps to find the few frames where either model fails to provide a good fit to the image. The results show that (a) neither model has a definitive advantage over the other, and (b) Kalman filtering might actually be more effective than particle filtering for both models.

1. Introduction

Active contour methods find application to tracking when camera motion prevents the use of background subtraction methods, and/or when only specific kinds of objects need to be tracked, and the shape, but not the appearance, of these objects is known a priori.

Many active-contour methods have been proposed since the initial paper by Kass et al. [12]. However, theoretical and experimental comparisons between these methods have been very scarse. Amongst experimental comparisons in the area of tracking, a careful study [17] focused on accuracy of segmentation of a single walking human in a controlled setting, but did not address robustness and interactions between multiple targets. A recent paper on tracking motor vehicles [4] is closer to the approach followed in this paper, but is limited to an experimental comparison. A thorough theoretical comparison of methods, with a focus on segmentation is provided in [19], but it contains no experimental comparisons and is of little relevance to tracking.

This paper is based on the principle that theoretical and experimental comparisons should complement each other. Following this principle, two recently developed active-contour trackers are put into the same theoretical framework prior to experimental comparisons on the PETS test sequences. One of the methods [15] has previously been applied to vehicle tracking and the other [16] to articulated body tracking. The experimental comparisons are carried out on vehicle tracking for the following reasons:

- the methods are not sufficiently robust by themselves (i.e. without sensor fusion) for human body tracking;
- open-source software for vehicle tracking has been made available [21] that will allow the reader to replicate the experiments;

1.1. Active contour methods

Most active-contour methods can be classified as *feature-based* if the pose of the object is optimized by minimizing squared distances between contours and image features; and *contrast-based* if the pose of the object is optimized by maximizing some contrast measure (e.g. the norm of the grey-level gradient) under the contour.

Feature-based methods have found wide application in tracking (see [1] and references therein). These methods facilitate the application of Kalman filtering. However, feature extraction is a process notoriously sensitive to noise, which leads to instabilities in tracking. After the introduction of particle filtering [5], Kalman filtering is no longer the only option available.

Contrast-based methods include the original snake model [12], the model-based tracker by Kollnig and Nagel [13], and several methods making use of image statistics [9, 20]. The methods compared in this paper are contrast-based.

Amongst contrast-based methods, there is a subclass of methods in which the first processing step consist in blurring the image [12, 20, 16], or equivalently the contour [13]. The motivation for blurring is that model contours and image edges are seldom in perfect registration, due to errors both in pose estimation and in the shape model.

An alternative approach to errors in the shape model is marginalization over deformations, recently introduced in [15].

This paper compares the blurring and marginalization approaches. The comparison starts with the formulation of

a unified model which includes both the blurring approach followed by Sidenbladh and Black [16] and the marginalization approach proposed by Pece and Worrall [15]. It will be shown that the mathematical differences between the two approaches are small, but significant. This theoretical comparison is complemented by an experimental comparison based on the Condensation algorithm. This algorithm was selected because it is the most general framework for comparing different likelihood models. However, no performance comparison can conclusively prove the superiority of an entire class of methods. The analysis in this paper is meant primarily to illustrate the similarities in practice between the two models.

The theoretical comparison is presented in section 2. The experimental comparison is in section 3. The conclusions of the comparisons are in section 4.

2. Theoretical basis of blurring and marginalization

We begin by formulating a generative model along the lines proposed in [15].

The object state is described by an m-vector $\mathbf{x}(t)$ which is a function of time t. Given the state and a geometric model of the object, the object contour is projected onto the image plane. The contour is then used to estimate the likelihood of the image, given the object state.

2.1. The observation

A finite set of n sample points on the contour are used to estimate the likelihood. The image coordinates and unit normals to these sample points are computed from the geometric model together with the estimated state parameters. The normal line to a sample point will be called *observation line*. Due to the aperture problem, only the normal component of the displacement of the object boundary can be locally detected. Therefore, only the intersection between the object boundary and the observation line is of interest in the pose refinement algorithms.

A distinction must also be made between the predicted intersection (i.e. the contour intersection) and the actual intersection (i.e. the intersection of the object boundary): these differ not only because of errors in the state estimate, but also because of errors (deformations) in the geometric model. The deformations ϵ are assumed to be Gaussian, zero-mean, independent, and identically distributed on all sample points on the contours, with variance σ^2, so that the prior pdf (probability density function) of deformations $p_D(\epsilon)$ is given by

$$p_D(\epsilon) \stackrel{\text{def}}{=} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\epsilon^2}{2\sigma^2}\right) \qquad (1)$$

In the following, the symbol v_i will be used for the coordinate on the observation line indexed by i. The symbol u_i will be used for the coordinate of the contour intersection. The distance between contour and actual intersection is denoted by ϵ_i. The subscript i will be dropped when not needed.

Using a digital computer, only a finite set of grey levels can be measured on the observation line. Given regularly-spaced sampling of grey levels with spacing Δv, we define the *observation* as $\boldsymbol{I}_i = \{I_i(j\Delta v) \,|\, j \in \mathbb{Z}\}$. In the following, the subscript j will always denote location on the observation line. Fig. 1 illustrates the meaning of the symbols u, v, ϵ, Δv.

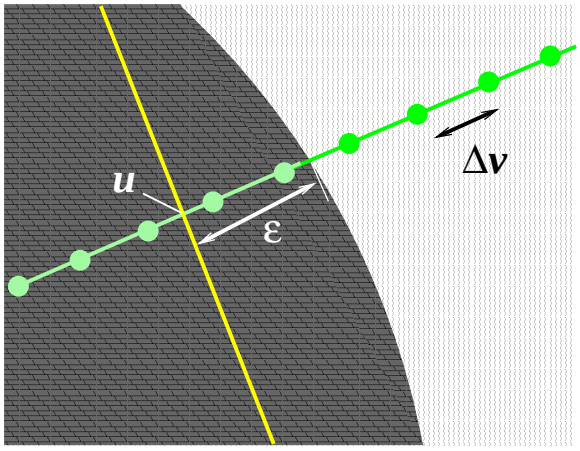


Figure 1: The yellow line represents a model contour, mismatched with the object boundary. The green line represents the normal to a sample point on the contour. u is the coordinate of the intersection with the contour and ϵ is the distance between the intersections of the normal line with the contour and with the object boundary. Grey levels are sampled on the normal with a regular spacing Δv.

2.2. Likelihoods of grey-level differences

It is assumed that the pdf of the observation depends only on grey level differences (gld's).

We define a binary indicator variable $\eta(u)$, with value 1 if the modelled boundary (i.e. the boundary modelled by the contour used to define the observation line) intersects the observation line between $u - \Delta v/2$ and $u + \Delta v/2$; and value 0 otherwise.

Given no modelled boundary between image locations $u-\Delta v/2$ and $u+\Delta v/2$, the pdf of the observation is defined

as:

$$p_L(\boldsymbol{I}, u) \stackrel{\text{def}}{=} f\left[I(u - \Delta v/2), I(u + \Delta v/2) \,|\, \eta(u) = 0\right] \tag{2}$$

The pdf p_L could be estimated by image statistics, as in [16]. However, a more robust estimate can be obtained by fitting a single-parameter pdf to the image statistics, as in [15]. Differently from [15], we use a Laplacian, rather than a generalized Laplacian. Defining $\Delta I(u) = I(u+\Delta v/2) - I(u - \Delta v/2)$, the Laplacian is of the form:

$$p_L(\boldsymbol{I}, u) = \frac{1}{2\lambda} \exp\left(-\left|\frac{\Delta I(u)}{\lambda}\right|\right) \tag{3}$$

where λ is a parameter that depends on the distance Δv.

Given a modelled boundary between image locations $u - \Delta v/2$ and $u + \Delta v/2$, the pdf of the observation is defined as:

$$p_E(\boldsymbol{I}, u) \stackrel{\text{def}}{=} f\left[I(u - \Delta v/2), I(u + \Delta v/2) \,|\, \eta(u) = 1\right] \tag{4}$$

Given that grey levels observed on opposite sides of a modelled boundary are statistically independent, this pdf is assumed to be uniform:

$$p_E(\boldsymbol{I}, u) = 1/q \tag{5}$$

where q is the number of grey levels.

2.3. Likelihood ratios

The optimal contour location is found by maximizing the likelihood ratio

$$p_R(\boldsymbol{I} \,|\, u) \stackrel{\text{def}}{=} \frac{p_E(\boldsymbol{I}, u)}{p_L(\boldsymbol{I}, u)}. \tag{6}$$

Given that the edge likelihood p_E is uniform, it can be easily seen that the likelihood ratio is a more sensible measure. The full theoretical rationale for using the ratio is given in [9, 2, 16, 15].

2.4. Estimation of contour likelihood

The previous section has defined the basic elements of the probabilistic model. However, the deformations defined by Eq. 1 have not been taken explicitly into account. To see why this is a problem, consider that the likelihood ratio, as defined above, is not a smooth function of the contour position: a displacement of Δv of the contour on the image plane means that the likelihood ratio is no longer measured across the object boundary. As a consequence, even at the optimal pose, errors in the geometrical model will make the likelihood ratio an inappropriate measure of goodness-of-fit.

There are at least two possible solutions to this problem. This subsection describes what will be called the BObs (blurred observation) model. The principle is quite simple: the observation $\boldsymbol{I}$ is convolved with a Gaussian kernel which we take to be equal to the pdf of the deformations, Eq. 1. The convolution takes the form

$$I_\sigma(v) \stackrel{\text{def}}{=} \int_{-\infty}^{+\infty} I(v - \epsilon)\, p_D(\epsilon)\, d\epsilon \tag{7}$$

In practice it is only necessary to integrate within a distance of 2σ on either side of u. After blurring, the likelihood ratio becomes:

$$p_B(\boldsymbol{I} \,|\, u) \stackrel{\text{def}}{=} \frac{p_E(\boldsymbol{I}_\sigma \,|\, u)}{p_L(\boldsymbol{I}_\sigma)} \tag{8}$$

The similarity to the edge filters proposed by Sidenbladh and Black [16] is evident.

2.5 Marginalization of likelihood ratio

Contour deformations are noise variables which are not of interest when fitting the contour to an image. In such a case, the standard Bayesian approach is to marginalize over deformations.

The likelihood ratio of the observation $\boldsymbol{I}$ given the contour location u and the deformation ϵ is $p_R(\boldsymbol{I} \,|\, u + \epsilon)$. The joint likelihood ratio of observation $\boldsymbol{I}$ and deformation ϵ, given the contour location u, is given by

$$p(\boldsymbol{I}, \epsilon \,|\, u) = p_R(\boldsymbol{I} \,|\, u + \epsilon)\, p_D(\epsilon) \tag{9}$$

The marginalized likelihood ratio (MLR) of the observation is obtained by integrating over all possible deformations:

$$p_M(\boldsymbol{I} \,|\, u) = \int_{-\infty}^{+\infty} p_R(\boldsymbol{I} \,|\, u + \epsilon)\, p_D(\epsilon)\, d\epsilon \tag{10}$$

2.6 Remarks on the models

It can be seen that the difference in practice between the BObs and MLR models is whether the observation is first filtered with a Gaussian kernel, then converted to a likelihood ratio; or *vice versa*. Conversion from grey levels to likelihood ratios is a nonlinear operation, and therefore the two operators (Gaussian filtering and conversion to likelihood ratio) do not commute.

In most practical applications, the major factor in the computational cost for either model (indeed, for most boundary-based methods) consists in accessing image pixels on the observation line. Therefore, estimating the likelihood ratio with either model will have almost the same computational cost.

3. Experimental comparison between blurring and marginalization

The Condensation filter [11] is possibly the most general tracking method, because it imposes no restrictions on the

dynamical, geometric, or observation models; therefore, it provides the least biased framework for comparing the BObs model and the MLR model. The comparisons were carried out on the PETS 2000 [7] test sequence and PETS 2001 [6] test sequence 1 (camera 1).

3.1. Implementation details

The specific dynamical model used in the tracking experiments was the steering-angle vehicle model described in [14]. The geometric model (called "wireframe" in the following, even though there is hidden-line removal) was the shape of the "average car" [8]. The "average car" was used in our experiments because, in practical applications, there is no a priori knowledge of which car enters the scene; and also to test the methods to the limit. The value of σ was set equal to the greater of 4 pixels and $(F/d)\cdot 0.1$ meters, where F is the focal length of the camera and d is the distance of the vehicle from the camera. The number of particles was fixed at 1024.

3.2. Results on the PETS 2000 test sequence

The experiments were carried out on frames 380 to 1000. In this segment, a hatchback (compact car) enters the field of view from the top left and makes two turns before parking; then a white van enters from the bottom right, passes in front of the hatchback occluding its lower edge, and finally disappears at the top left.

Fig. 2 gives examples of successful tracking. The two kinds of observed tracking failures are shown in Fig. 3: either the hatchback wireframe did not complete the turn into the parking slot; or else it was pulled back from the parking slot when the van (which had a higher contrast with the background) passed in front of it. (Note that collision is not modelled in our system.) The van itself was always successfully tracked.

Bar histograms of the numbers of failures and successful tracking are shown in Fig. 4. It can be seen that tracking failures before the hatchback reached its parking slot (failures of type 1) were only observed with the MLR model.

The tracking failures due to interference between van and hatchback (failures of type 2) were observed with both models. This kind of problem requires particle filters specifically designed for multi-target tracking, e.g. mixture particle filters [18]. However, this is beyond the scope of the present paper.

A better insight into failures of type 1 can be gained by plotting the log-likelihood ratio as a function of orientation. It can be seen in Fig. 5 that in frame 570 the BObs log-likelihood ratio has its maximum at the correct angle as determined by hand (about -150 degrees). However, the log-MLR has its maximum at about -175 degrees, a clockwise rotation of 25 degrees from the correct orientation. This explains why the tracking failure shown in Fig. 3, frame 570, is more common with the MLR model.

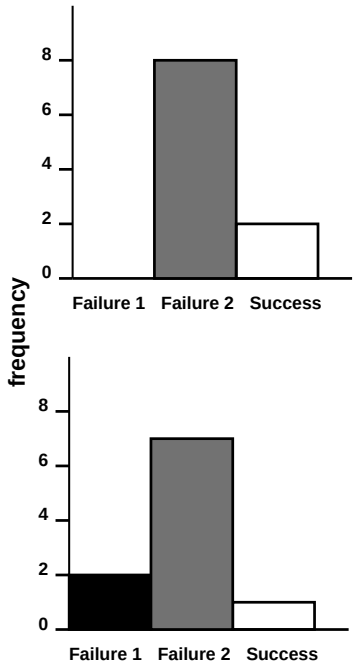


Figure 4: Summary of results on the PETS 2000 test sequence. Top: BObs model. Bottom: MLR model. Failure 1: hatchback wireframe did not reach parking slot. Failure 2: hatchback wireframe pulled out of parking slot by van.

Some insight into tracking failures of type 2 (Fig. 3, frame 825) can be obtained from the plots in Fig. 6. It can be seen that the log-likelihood ratios for both models, as functions of translation, are not very strongly peaked, so that most particles will not be found near the correct pose estimate. (Similar plots were obtained for orientation and are not shown because of space restrictions.)

3.3. Results on the PETS 2001 test sequence 1 (camera 1)

This sequence is somewhat more challenging, because of shorter focal length, less favorable viewing angle, poorer camera calibration, and large number of distractors (parked vehicles close to the trajectories of the moving vehicles).

The experiments on this sequence were carried out on frames 500 to 750, with both models and the same numbers of particles as for the PETS 2000 sequence. In this segment, the moving vehicles were again a hatchback and a white van. The hatchback entered from the lower right corner and went straight to a parking slot; then the van entered from the left, passing in front of a row of parked vehicles.

The first thing to note is that it was not possible to track

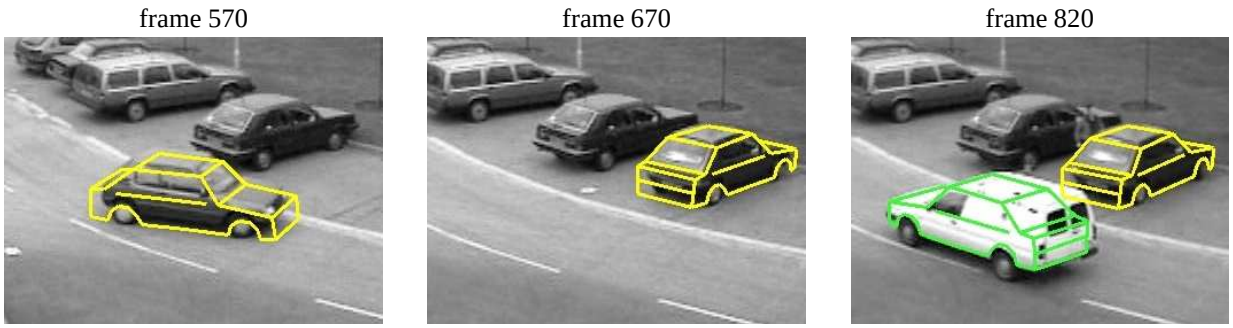


Figure 2: Successful tracking of the hatchback and van in the PETS 2000 test sequence.

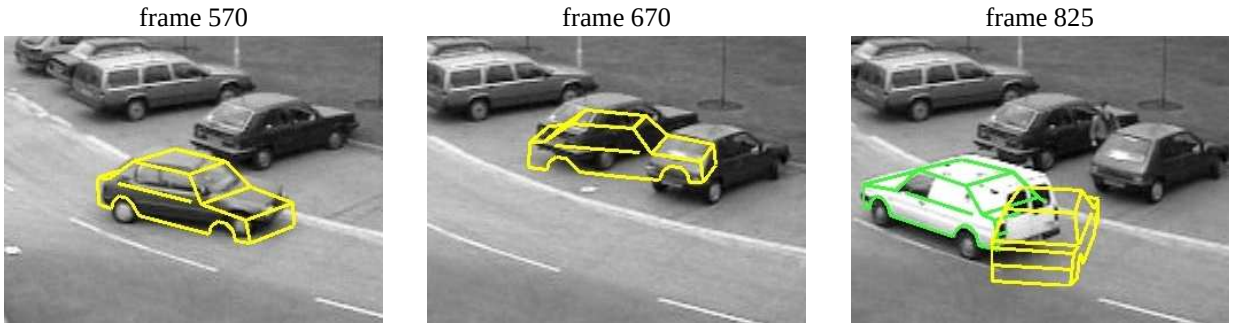


Figure 3: Typical failures in the PETS 2000 test sequence: the hatchback wireframe does not complete the turn (frames 570 and 670); or else, having completed the turn, is "pulled out" of the parking slot by the van (frame 825).

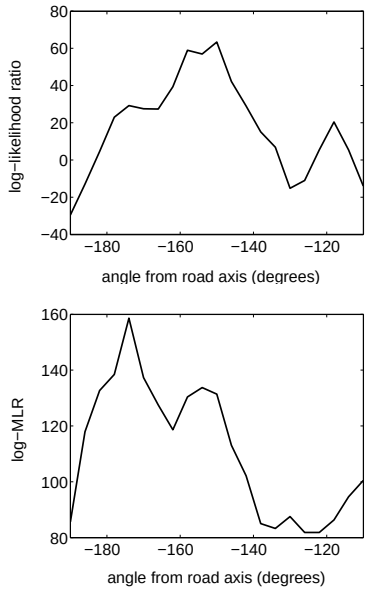


Figure 5: Log-likelihood ratios as functions of rotation of the hatchback wireframe in frame 570 of the PETS 2000 sequence, centered on the "correct" pose variables. Top: BObs model; bottom: MLR model.

the van reliably, due to the distractors (Fig. 7, frame 710b). This tracking failure cannot easily be eliminated by using a multi-target tracking algorithm, because the parked vehicles, being stationary through the sequence, might not be easy to identify as distinct targets.

The hatchback followed a straight path from its entry point to its parking slot. Nonetheless, tracking with the BObs model was not always successful: occasionally, the wireframe ended up at a wrong angle (Fig. 8, frames 600 and 710). Another distinct failure was seen when the hatchback wireframe "jumped out" of its parking slot (Fig. 8, frame 750).

Histograms of tracking results are given in Fig. 9. No results are given for the van, because it was almost always a tracking failure. The point to note is that the MLR model was more successful in tracking the hatchback in this sequence.

Once more, plots of the log-likelihood ratios give some insight into the tracking failures. Fig. 10 shows the log-likelihood ratios as functions of orientation in frame 600 of the sequence. It can be seen that the BObs log-likelihood ratio has its peak value at an orientation clockwise from the correct orientation. The BObs log-likelihood ratio was actually negative at the correct orientation. No such problems can be seen in the plot of log-MLR. This explains why the tracking failures shown in Fig. 8, frames 600 and 710, only

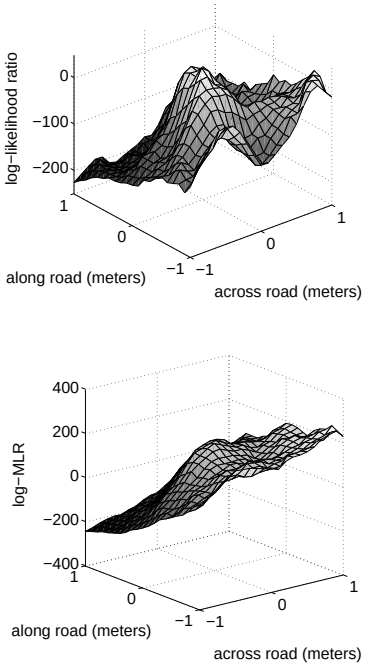


Figure 6: Log-likelihood ratios of the hatchback wireframe as functions of translation in frame 820 of the PETS 2000 sequence. Top: BObs model; bottom: MLR model.

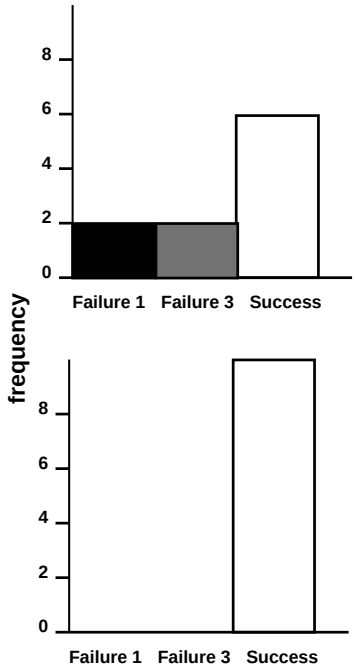


Figure 9: Summary of results on the PETS 2001 test sequence. Top: BObs model. Bottom: MLR model. Failure 1: hatchback wireframe reached parking slot at the wrong angle. Failure 3: hatchback wireframe jumped out of correct position.

happened with the BObs model.

Log-likelihood ratios for the van wireframe at frame 710 are plotted in Fig. 11. Even more clearly than in Fig. 6, it can be seen that the log-likelihood ratios for both models do not have a sharp peak at the correct position, which explains the failure seen in Fig. 7, frame 710b. (Again, similar plots were obtained for orientation and are not shown because of space restrictions.)

4. Conclusions

This paper has compared and contrasted the blurring approach and the marginalization approach to contour tracking. It has been shown that the basic mathematical difference between the models is the order in which Gaussian convolution and conversion from grey levels to likelihood are carried out.

Experiments with the Condensation tracker show that either model can fail at some specific frames. Condensation tracking helps to find the few frames for which the models fail to provide a good fit to the images. However, it is not immediately obvious why the MLR model fails in frame 570 of the PETS 2000 sequence, while the BObs model fails in frame 600 of the PETS 2001 sequence.

The MLR model, being based on marginalization, nat-

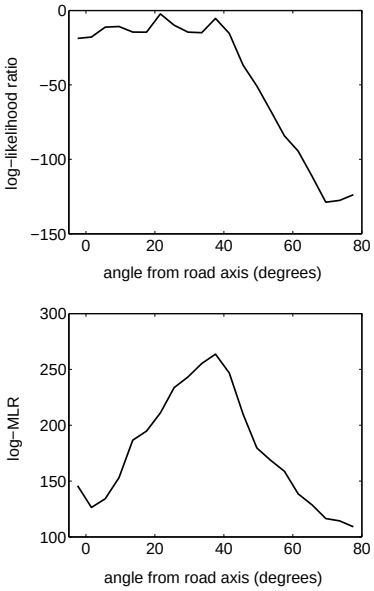


Figure 10: Log-likelihood ratios of the hatchback wireframe as functions of rotation in frame 600 of the PETS 2001 sequence. Top: BObs model; bottom: MLR model.

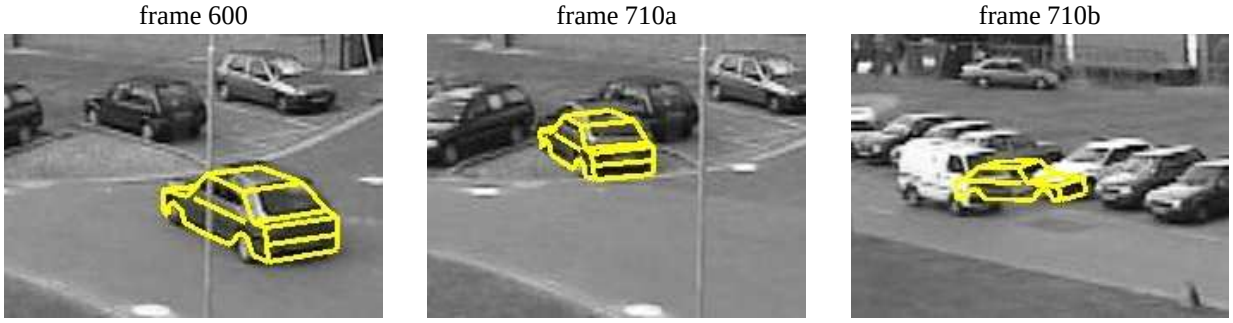


Figure 7: Typical results for the PETS 2001 test sequence 1 (camera 1): the hatchback is successfully tracked, but the van is not.

Figure 8: Tracking failures for the hatchback in the PETS 2001 test sequence: the wireframe turns clockwise at the level of the lamp-post (frame 600) and remains in that attitude (frame 710a); or else it "jumps out" of its correct pose (frame 730).

urally leads to the use of the EM algorithm, which can be easily combined with Kalman filtering. This combination has been shown to achieve better performance than the Condensation tracker at a lower computational cost [15].

Of course, a Kalman-type tracker can be implemented with any contour model: all what is needed is a pose-refinement method, i.e. a method for maximizing the posterior pdf of the object state, given the Kalman prediction and the current frame. However, in order to preserve the computational speed of Kalman filtering, a gradient-based optimization method is preferable. In the case of models based on marginalization, the EM algorithm is a natural choice for optimization. In the case of blurred-contour models, there is no obvious choice of optimization method. Kollnig and Nagel [13] used the Levenberg-Marquardt method. Testing this and other methods on the blurred-contour model formulated in this paper will be a subject of future research.

References

[1] A. Blake and M. Isard. *Active Contours*. Springer, 1998.

[2] J. Coughlan, A. Yuille, C. English, and D. Snow. Effi cient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, 2000.

[3] J. M. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proc. 7th European Conf. Comp. Vision: ECCV 2002*, LNCS 2352, pages 453–468. Springer, 2002.

[4] H. Dahlkamp, A. Ottlik, and H.-H. Nagel. Comparison of edge-driven algorithms for model-based motion estimation. In *Proc. Workshop on Spatial Coherence for Visual Motion Analysis: SCVMA'04*, 2004.

[5] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[6] J. M. Ferryman and J. L. Crowley, editors. *Proc. of the 2nd IEEE International Workshop on Performance Evaluation in Tracking and Surveillance: PETS*, 2001.

[7] J. M. Ferryman and A. D. Worrall, editors. *Proc. of the 1st IEEE International Workshop on Performance Evaluation in Tracking and Surveillance: PETS*, 2000.

[8] J. M. Ferryman, A. D. Worrall, G. D. Sullivan, and K. D. Baker. Visual surveillance using deformable models of vehicles. *Robotics and Auton. Syst.*, 19:315–335, 1997.

[9] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. PAMI*, 18(1):1–14, 1996.

[10] D. W. Hansen and A. E. C. Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, 2005.

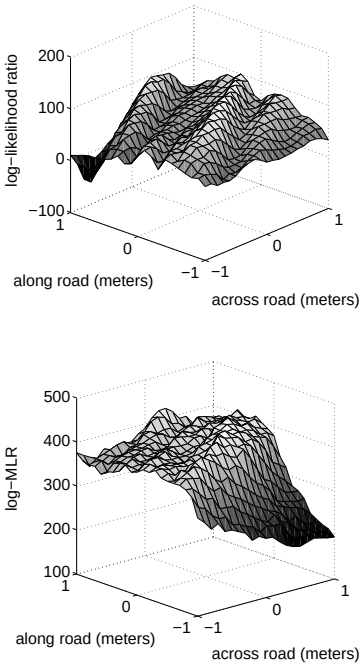


Figure 11: Log-likelihood ratios of the van wireframe as functions of translation in frame 710 of the PETS 2001 sequence. Top: BObs model; bottom: MLR model.

[11] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. In *Proc. Int. Conf. Comp. Vision: ICCV'97*, pages 259–268, 1987.

[13] H. Kollnig and H.-H. Nagel. 3D pose estimation by directly matching polyhedral models to gray value gradients. *Int. J. Computer Vision*, 23(3):283–302, 1997.

[14] H. Leuck and H.-H. Nagel. Automatic differentiation facilitates OF-integration into steering-angle-based road vehicle tracking. In *Proc. Int. Conf. Comp. Vision Pattern Rec: CVPR'99*, volume 2, pages 360–365, 1999.

[15] A. E. C. Pece and A. D. Worrall. Tracking with the EM contour algorithm. In *Proc. 7th European Conf. Comp. Vision: ECCV 2002*, LNCS 2350, pages 3–17. Springer, 2002.

[16] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. *Int. J. Computer Vision*, 54(1/2/3):183–209, 2003.

[17] P. Tissainayagam and D. Suter. Performance measures for assessing contour trackers. *Int. J. of Image and Graphics*, 2:343–359, 2001.

[18] J. Vermaak, A. Doucet, and P. Perez. Maintaining multimodality through mixture tracking. In *Proc. 9th. Int. Conf. Computer Vision: ICCV 2003*, pages 1110–1116, 2003.

[19] C. Xu, J. A. Yezzi, and J. L. Prince. On the relationship between parametric and geometric active contours. In *Proc. 34th Asilomar Conf. on Signals, Systems, and Computers*, pages 483–489, 2000.

[20] A. L. Yuille and J. M. Coughlan. Fundamental limits of Bayesian inference: order parameters and phase transitions for road tracking. *IEEE Trans. PAMI*, 22(2):160–173, 2000.

[21] `http://i21www.ira.uka.de/motris/`

Entropy of quasi-stationary measures on images with applications to 2D constrained arrays

Søren Forchhammer* and Torben V. Laursen†
Research Center COM, DTU

DSAGM2005

1 Introduction

The concept of the entropy of a Markov Source is well known. However, there are difficulties extending this concept to 2D sources such as images. Markov Random Fields (MRF), being the natural generalization of a Markov Source to 2D, have enjoyed a wide range of use in image modeling, but it is intractable to compute their partition function, and hence the entropy.

A simple class of MRFs, which also have the property of being causal, is due to Pickard [14], [13]. These are known as Pickard Random Fields (PRF). Champagnat et el. [2] have investigated these models further and given examples of their use. We can compute the entropy for a PRF, but they suffer from the fact they are first order models, and thus have limited modeling power.

We present a method for designing quasi-stationary probability measures for higher order modelling of images. Based on these models some classes of 2D constrained fields are easily analyzed. We can calculate the entropy of the measures, thus obtaining a lower bound on the entropy of the constraints considered.

Constrained codes (in 1D) have enjoyed a widespread use in communication and data storage. Recently revival of storage ideas such as holographic storage and advances in nano storage such as the milipede project [17], [4] have caused interest for 2D constrained fields as models of data storage on a surface. Our ideas are applicable to generating 2D constrained codes.

2 Basic definitions

We consider 2D fields specified by shift invariant constraints of finite extent (N, M). A constraint is defined by a list, $\mathcal{F}$, of forbidden blocks of maximum size $N \times M$ made of symbols from a finite alphabet A of size $|A|$. A configuration on an n by m rectangle having no forbidden blocks within the rectangle is called an admissible configuration.

*sf@com.dtu.dk
†tvl@com.dtu.dk

Let $E(n,m)$ be the set of admissible configurations on an n by m rectangle for a given field F and let $B(n,m) = |E(n,m)|$ be the number of distinct admissible configurations of size $n \times m$. The entropy of F is then defined as

$$H^{(2)}(F) = \lim_{n,m\to\infty} \frac{B(n,m)}{nm}. \tag{1}$$

We will describe a method for designing two-dimensional quasi-stationary measures that have entropy close to $H^{(2)}(F)$ for certain fields F.

Examples of one-dimensional constrained sequences include run-length-limited constraints. The one-dimensional (1D) (d,k)-*run-length-limited* (RLL) constraint consists of all binary words in which the run-lengths of 0s are between d and k, inclusive, except the first and the last runs which may be shorter than d. The 2D (d,k) RLL constraint consists of all configurations in which the 1-D (d,k) RLL constraint is satisfied for every row and every column.

A notable example of a first-order binary 2D RLL constraint is the 2D $(1,\infty)$ RLL. Here the extent of the constraint is $N = M = 2$ and the forbidden blocks consists of

$$\mathcal{F} = \left\{ 11, \begin{matrix} 1 \\ 1 \end{matrix} \right\}.$$

This is also referred to as the hard square constraint [7]. Calkin and Wilf [1] presented methods giving tight bounds on the entropy for the hard square constraint. Their methods apply to other (first order) constraints, but they do not apply when $N > 2$ and $M > 2$ [7].

The following sections will focus on higher order constraints ($N > 2$ and $M > 2$).

2.1 Bit-stuffing

Bit-stuffing is a simple, yet efficient way to code for 1D RLL constraints. It is applicable if it is always possible to write say a 0 at any position. The data stream is written as is, except that each time a 1 is encountered, the necessary number of zeros are stuffed immidiately after. This method can be extended to 2D constrained arrays [12], [15], [9]. Constraints of this type includes RLL(d,∞) and other checkerboard constraints [10]. An analysis of 2D bit-stuffing has been presented in [15], where it is shown that it is very efficient for the hard square constraint. In [9] bit-stuffing is used for 2D RLL (d,∞). Analysis of the hard-triangle has also been carried out [12]. In [15], [9] the bit-stuffing is performed along diagonals, writing bits from a sequence whenever possible and writing the 0s the constraint prescribes. The iid unbiased data sequences to be coded may be transformed into iid biased sequences in a precoding step in order to increase the entropy. One can utilize this further by having more than one biased sequence and choose between them depending on past data besides what is prescribed by the constraint.

For the hard-square [15] and hard-triangle [12], the entropy of the bit-stuffing scheme has been determined and optimized. For the higher order constraints 2D RLL (d,∞) lower bounds on the entropy of the bit-stuffing scheme are presented in [9]. We will provide better bounds in section 4.1.

2.2 Finite state sources

In one dimension, sequences satisfying a constraint on N consecutive symbols such as run-length-limited sequences may be described by finite state sources, where a state is characterized by $N-1$ symbols. The entropy in 1-D, defined as in Definition 1.1 but with $m = M = 1$ and $n \to \infty$, may be calculated following Shannon's approach [16]. The transfer (or adjacency) matrix $\mathbf{T}$ of the source indicates the possible transitions between two states. The largest eigenvalue Λ of the transfer matrix $\mathbf{T}$ determines the growth rate of the number of configurations [11]. Taking the logarithm gives the maximum entropy [16]:

$$H(1) = \log(\Lambda). \tag{2}$$

The one-dimensional approach is readily generalized to 2D arrays of finite (horizontal) width m and arbitrary (vertical) height n. The admissible configurations of an array of width m may for all n be described by a finite state source. For a constraint of extent (N, M), the states of the source are given by the symbols on the m by $N-1$ segment which appear as the first or last $N-1$ rows of an admissible configuration on a N by m rectangle, i.e. a configuration of $E(N, m)$. A transition from state i to state j is admissible if there is a configuration in $E(N, m)$, for which state i is identical to the top $N-1$ rows and state j to the bottom $N-1$ rows. State i and j have an overlap of $N-2$ rows. The last row of j is generated by the transition from i to j and appended to the previous rows of the output. Any admissible configuration of $E(n, m)$ with fixed m and n ($> N-1$) rows may be generated as an output by starting the source in the state specified by the first $N-1$ rows and making $n-N+1$ transitions appending one row to the output in each transition. The transfer matrix $\mathbf{T}_m$ indicates transitions which satisfy the constraint by defining the elements $t_{ij} = 1$ if the transition from state i to j is admissible and $t_{ij} = 0$ if it is not admissible. The per symbol entropy of the source on an array of width m ($n \to \infty$) is given by,

$$\frac{H(m)}{m} = \frac{\log(\Lambda_m)}{m}, \tag{3}$$

where Λ_m is the largest (positive) eigenvalue of $\mathbf{T}_m$. Equation (3) is an upper bound on the entropy $H^{(2)}$ defined by (1) [7].

For constraints where any two configurations, X and Y, on arrays of width m may admissibly be concatenated (or cascaded) by padding a merging array, V, of c columns to form the admissible configuration XVY, the entropy is lower bounded by $H(m)/(m+c)$.

A probability measure may be induced by defining transition probabilities p_{ij} such that $p_{ij} \geq 0$ if $t_{ij} > 0$, $p_{ij} = 0$ if $t_{ij} = 0$, and $\sum_j p_{ij} = 1$ for all i. Let $\mathbf{P}_m$ denote the transition probability matrix.

3 Quasi-stationary measures

Let $\mathbf{W}$ denote a stochastic variable defined on an n by m array over some alphabet A. Let $\mathbf{X}$ and $\mathbf{Z}$ denote variables representing the first and last $M-1$ columns, ie. they are defined on n by $M-1$ arrays. Let $\mathbf{Y}$ denote a variable representing the middle $m-2M+1$ columns. A quasi stationary measure may be introduced by concatenating these bands.

In the general case, the stochastic variables can take on any of the $|A|^{n\times m}$ possible values. However, we will restrict ourselves for the time being to measures agreeing with a constraint defined on an alphabet. That is, configurations having forbidden blocks are assigned probability zero.

Given a probability measure for $\mathbf{W}$ we assume that the measures on the boundaries $\mathbf{X}$ and $\mathbf{Z}$ are identical, i.e.

$$P(\mathbf{X}) = P(\mathbf{Z}). \tag{4}$$

Starting with $\mathbf{X}_0\mathbf{Y}_0\mathbf{Z}_0$, arrays $\mathbf{Y}_i\mathbf{Z}_i$ may repeatedly be added to form

$$\mathbf{X}_0, \mathbf{Y}_0, \mathbf{Z}_0, \mathbf{Y}_1\mathbf{Z}_1, \ldots, \mathbf{Y}_K\mathbf{Z}_K,$$

such that $\mathbf{Z}_{i-1}\mathbf{Y}_i\mathbf{Z}_i$ has the same measure as $\mathbf{W}$. The entropy of $(\mathbf{Y}_i\mathbf{Z}_i|\mathbf{Z}_{i-1})$ is given by

$$\frac{H_W(m) - H_X(M-1)}{m - M + 1}. \tag{5}$$

where $H_W(m)$ is the entropy of $\mathbf{W}$ (per row) and $H_X(M-1)$ is the entropy of $\mathbf{X}$ (per row).

Now assume that $\mathbf{W}$ is described by a finite state source with states of height $N-1$ and width m. Let $\mathbf{P}_m$ be the transition probabilities, with the stationary solution $\pi\mathbf{P}_m = \pi$. Let $\mathbf{X}_0\mathbf{Y}_0\mathbf{Z}_0$ be initiated by π and all $\mathbf{Y}_i\mathbf{Z}_i$ is initiated by π conditioned on the initial state of $\mathbf{Z}_{i-1}$. The entropy for $\mathbf{W}$ may be found from $\mathbf{P}_m$ and π. The entropy is given by

$$H_W = \sum_i \sum_j \pi_i p_{ij} \log(1/p_{ij}). \tag{6}$$

Given the finite state source for $\mathbf{X}$ along with its transition probabilities and the stationary distribution we can compute H_X in the same way. Hence we can compute (5).

For the sequence of arrays, $\mathbf{X}_0, \{\mathbf{Y}_j, \mathbf{Z}_j\}_0^K$, the measure, based on $\mathbf{P}_m$ with the initialization based on π, is quasi-stationary in the sense that each subset $\mathbf{Z}_{i-1}\mathbf{Y}_i\mathbf{Z}_i$ is stationary and each column within these subsets are stationary. Considering an ensemble with random phase yields a stationary measure.

Actually assuming that $\mathbf{W}$ is described by a finite state source with a transition probability matrix $\mathbf{P}_m$ it is not necessarily important that the columns are initialized using π. If transitions from all states to all states are possible, which is the case for the constraints considered here, then asymptotically ($m \to \infty$) the measure converges to the stationary solution for any width specified by m, d, and K.

3.1 Analysis of the boundaries

For the hard square (and other first order constraints) the boundaries $\mathbf{X}$ and $\mathbf{Z}$ are just one column each. Further the constraint is symmetric (left-to-right) making it easy to satisfy the prerequisite of (5), $P(\mathbf{X}) = P(\mathbf{Z})$. The maxentropic solution for the finite state source yields a transition matrix $\mathbf{P}_m$ with this property. The entropy $H_W(m)$ in (5) is determined by (3), whereas $\mathbf{X}$ may be described as a function of a Markov chain. Therefore $H_X(M-1)$ may be bounded from both sides using standard techniques, [3], [7]. To efficiently

describe the process of $\mathbf{Y}, \mathbf{Z}$ given $\mathbf{X}$, a backward pass on a trellis given by the possible states of $\mathbf{Y}, \mathbf{Z}$ given $\mathbf{X}$ is combined with a forward pass. The transition probabilities of $\mathbf{P}_m$ may be changed to optimize (5).

For higher order constraints it is more difficult to ensure that the boundaries $\mathbf{X}$ and $\mathbf{Z}$ have identical measures, i.e. $P(\mathbf{X}) = P(\mathbf{Z})$ (4).

A simple solution is to decompose the probabilities such that

$$P(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = P(\mathbf{X})P(\mathbf{Z}|\mathbf{X})P(\mathbf{Y}|\mathbf{X}, \mathbf{Z}). \tag{7}$$

This may be viewed as the boundaries are generated first and thereafter the interior, $\mathbf{Y}$, given the boundaries. A further simplification is obtained by having independent boundaries, i.e.

$$P(\mathbf{X}, \mathbf{Z}) = P(\mathbf{X})P(\mathbf{Z}). \tag{8}$$

In the following we will use bit-stuffing to determine the transition probabilities. However, we modify the basic scheme slightly in order to satisfy the requirement of boundary independence (8).

3.1.1 RLL(d, ∞)

Bit-stuffing shall be used to define the transition probability matrix, $\mathbf{P}_m$. The states are $(N-1 =)d$ by m elements and a new row of m elements is generated with each transition.

Let $(x_0, ..., x_{d-1}, y_0, ..., y_{m-2d-1}, z_0, ..., z_{d-1})$ denote these new elements. To satisfy (8), the ordering of the bit-stuffing is altered slightly as

$$(x_0, ..., x_{d-1}, z_0, ..., z_{d-1}, y_0, ..., y_{m-2d-1}).$$

The transition probability, p_{ij}, is given by the product of the bit-stuff conditional probabilities considered in the order given above, i.e.

$$p_{ij} = \prod_{l=0}^{d-1} p(x_l|c_x(l)) \prod_{l=0}^{d-1} p(z_l|c_z(l)) \prod_{l=0}^{m-2d-1} p(y_l|c_y(l)), m \geq 3d, \tag{9}$$

where $p(x_l = 1|c_x(l)) = p_1(l)$ if a $x_l = 1$ is admissible, likewise $p(z_l = 1|c_z(l)) = p_1(l)$ if admissible and $p(y_l = 1|c_y(l)) = p_{y=1}(l)$. $c_\cdot(l)$ is the context at the given position. When a 1 is not admissible, obviously the conditional probability is set to 0. Whether it is possible to write a 1 in a given position at the time of writing is only dependent on the d previous elements in the same column and the previous elements of the current row after reordering. The ordering assures that $\mathbf{X}$ and $\mathbf{Z}$ may be described independently (8) and we have $P(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = P(\mathbf{X})P(\mathbf{Z})P(\mathbf{Y}|\mathbf{X}, \mathbf{Z})$. Using the same conditional probabilities for $\mathbf{X}$ and $\mathbf{Z}$ ensures (4) is satisfied. The entropy of the modified bit-stuffing, $C_{mb}(d, \infty)$ is given by (5) and it may be written as

$$C_{mb}(d, \infty) = \frac{H_W(m) - H_X(d)}{m - d}. \tag{10}$$

where $H_W(m)$ and $H_X(d)$ are the entropies (6) of $\mathbf{W}$ and $\mathbf{X}$, respectively. The entropy gives the expected values for all the interior elements of the cascaded

bands $\mathbf{X}_0, \{\mathbf{Y}_j, \mathbf{Z}_j\}_0^K$. The overall average converges to this value for $n \to \infty, K \to \infty$.

The transition probability matrix $\mathbf{P}_m$ may be modified to optimize the entropy (10) of the code, subject to the constraints on $\mathbf{P}_m$ and the prerequisites (4),(7). That is the conditional probabilities $p(x_l = 1|c_x(l))$ may depend on all (causal) elements of $\mathbf{X}$ within one transition, $p(z_l = 1|c_z(l))$ may depend on all (causal) elements of $\mathbf{X}, \mathbf{Z}$ within one transition, and $p(y_l = 1|c_y(l))$ may may depend on all (causal) elements of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ within one transition. Further the order of the process describing the boundaries $\mathbf{X}, \mathbf{Z}$ may be increased to counteract the freedom the boundaries of $\mathbf{W}$ has in comparison with the interior, $\mathbf{Y}$, or equivalently that they are to be concatenated with arrays, $\mathbf{Y}_{j-1}$ and $\mathbf{Y}_j$ on both sides. Results for RLL$(2, \infty)$ is given in the next section.

Considering the extension in the (quarter-)plane, $\mathbf{X}_0, \{\mathbf{Y}_j, \mathbf{Z}_j\}_0^K$, it may be written either row by row or column by column, writing one element at a time. Whenever admissible an element from the (biased) sequence designated to the column is chosen. Within each instance of the $m-d$ elements of $\mathbf{Y}_j\mathbf{Z}_j$, element z_i must be written before y_{m-3d+i}. Traversing the elements row by row, this introduces a latency of $m-d-1$ elements if $z_0, ...z_{d-1}$ is written before y_i. This latency may be reduced to d if the writing of zs and ys are interleaved. Traversing the plane column by column, the biased sequence designated to the column is used. Thus the choice of biased sequence is only changed once for each column. The column with z_i must be written before the column with y_{m-3d+i}.

3.1.2 Diamond constraints

Another class of constraints is the diamond constraint [10] (Or min. distance by 1-norm between 1s [7]). For a binary alphabet the minimum distance between 1s by the 1-norm is $(N =)M$. In this case $\mathbf{X}$ and $\mathbf{Z}$, besides being written before $\mathbf{X}$, must also be $M-2$ elements ahead to ensure a (1-norm) distance of M between the newest element, x_{M-2}, of $\mathbf{X}$ and the old elements of $\mathbf{Y}$. Besides this modification the construction of $\mathbf{P}_m$ may proceed as for the RLL(d, ∞) constraint, defining the transition probabilities, p_{ij}, by a product of conditional probabilities derived from the bit-stuffing probabilities.

4 Numerical results

We have considered the following examples: Three instances of the RLL(d, ∞) constraint, for $d = 2, 3$ and 4 as well as the diamond constraint.

4.1 Entropies for RLL(d, ∞)

We have computed several bounds for each of the fields. H_U gives an upper bound on the entropy using the simple (3). H_V offers an improvement on this upper bound using a more advanced technique described in [8]. As an estimate of the entropy we use the expression $\tilde{H} = H_W(m) - H_W(m-1)$ as this can be seen as an estimate of the entropy per column. $H_{p=1/2}$ gives the entropy using an unbiased bit-stuffer. H_p is the optimized entropy over a single biased stream, whereas H_{p_X,p_Y} is optimized choosing different biased sequences for the border X and interior Y respectively. We have collected the results in the following

table where the width of the band used is also noted.

F	m	H_U	H_V	$\tilde{H}$	$H_{p=1/2}$	H_p	H_{p_X,P_Y}
RLL$(2,\infty)$	19	0.4530	0.4459	0.4455	0.3917	0.4398	0.4410
RLL$(3,\infty)$	16	0.3784	0.3686	0.3675	0.3050	0.3606	0.3628
RLL$(4,\infty)$	15	0.3299	0.3188	0.3167	0.2487	0.2982	0.3110

It can be seen that the codes using an optimized probability for the border and intererior come within 1% of the estimated value of the entropy.

4.1.1 Optimizing the entropy further

One could use a different biased sequence for each column and then optimize the entropy over all columns. We have tried this using a depest descent approach viewing the entropy as a function of the column probabilities $p_1, \ldots, p_{m-d}$ and searching in the direction of the gradient. This yielded the following results.

F	RLL$(2,\infty)$	RLL$(3,\infty)$	RLL$(4,\infty)$
H_{opt}	0.4416	0.3640	0.3125

This offers a slight improvement over the result obtained where we only used two biased sequences.

4.2 Entropy of the diamond constraint

We have computed bounds for the diamond constraint as well using a band of width $m = 15$. The results are shown below. A more elaborate scheme for specifying $\mathbf{W}$ in (5) was also devised, resulting in the value H_{opt}. The probabilities p_1 were made dependent on the other elements on the $(N-1=)$ 2 previous rows. The next row of $\mathbf{X}$ (and $\mathbf{Z}$) is specified by probabilities conditioned on the two previous rows. The new row of $\mathbf{Y}$ is specified by probabilities conditioned on 3 rows of $\mathbf{X}$ and $\mathbf{Z}$ and 2 rows of $\mathbf{Y}$. These conditional probabilities were obtained from the maxentropic solution [7] for $\mathbf{W}$ (with two rows forming the states).

F	$\tilde{H}$	$H_{p=1/2}$	H_p	H_{p_X,p_Y}	H_{opt}
$M=3$	0.3503	0.276	0.344	0.3477	0.3497

5 Discussion and further work

One might suspect that the entropies of bit-stuffing with and without altering the order are very close. For the simple case where the same probability is used in all columns one would suspect that the entropy without reordering is greater.

These hunches have been supported by simulations of the bit-stuffing scheme without alterering the order.

5.1 Other types of constraints

There are constraints for which (the modified) bit-stuffing is not straight forward and maybe not a good solution. Etzion [5] studied 2D (d,k) SRLL constraints.

These constraints are symmetric with regards to the symbols, such that the run-lengths apply to all symbols in the alphabet.

Methods for constructing a merging array, V, given any two admissible arrays, X and Y, were given. The (minimum) width of the merging array, V, for which merging is always possible was expressed in terms of d. The existence of a solution in between two given arrays is a prerequisite for applying the modified bit-stuffing.

Another example is domino tiling, where the whole plane is tiled by one by two vertical and horizontal domino pieces. For this constraint a merging array, V, of finite width does not exist for merging any pair of arrays X and Y. A counter example is the case where X has a zig-zag boundary of all horizontal pieces where the piece in every other row is displaced one position relative to its two neighbors. In this case there is only one solution extending off the boundary of X, namely that which locks up with the boundary. (By induction this extends to the entire plane.) For such constraints, the boundaries have to be restricted to avoid configurations for which there is no admissible interior.

However, the frame work of utilizing borders and interiors still seems promising in this setting. Indeed some preliminary work has already been carried out in [6].

6 Conclusion

We have presented a higher order image model using quasi-stationary measures. We demonstrated one application of the model with the modified bit-stuffing scheme for constrained 2D fields. The scheme presented is easy to analyze based on well-known 1D techniques. The entropy of the scheme may be calculated once the $m - d$ conditional probabilities are chosen. The numerical results for the 2D RLL(d, ∞) and the big diamond constraints are within 1% of the estimated entropy of these constraints.

References

[1] N. J. Calkin and H. S. Wilf. The number of independent sets in a grid graph. *SIAM Journal of Discrete Mathematics*, 11(1):54–60, 1998.

[2] F. Champagnat, J. Idier, and Y. Goussard. Stationary markov random fields on a finite rectangular lattice. *IEEE Transactions on Information Theory*, 44:2901–2916, November 1998.

[3] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley series in telecommunications. John Wiley & sons, 1991.

[4] E. Eleftheriou, T. Antonakopoulos, GK Binnig, G. Cherubini, M. Despont, A. Dholakia, U. Duerig, M.A. Lantz, H. Pozidisand H.E. Rothuizen, and P. Vettiger. Millipede-a mems-based scanning-probe data-storage system. *IEEE Transactions on Magnetics*, 39(2), 2003.

[5] T. Etzion. Cascading methods for runlength-limited arrays. *IEEE Transactions on Information Theory*, 43(1):319–324, January 1997.

[6] S. Forchhammer and T. V. Laursen. Cascading 2d constrained arrays using periodic merging arrays. In *International Symposition on Information Theory*, page 109. IEEE, 2003.

[7] Søren Forchhammer and Jørn Justesen. Entropy bounds for constrained two-dimensional random fields. *IEEE Transactions on Information Theory*, 45(1):118–127, 1999.

[8] Søren Forchhammer and Jørn Justesen. Bounds on the capacity of constrained two-dimensional codes. *IEEE Transactions on Information Theory*, 46(7):2659–2666, 2000.

[9] S. Halevy, J. Chen, R.M. Roth, P.H. Siegel, and J.K. Wolf. Improved bit-stuffing bounds on two-dimensional constraints. In *Proc. ISIT'02 Lausanne, Switzerland*, page 385. IEEE, 2002.

[10] W. Weeks IV and R.E. Blahut. The capacity and coding gain of certain checkerboard codes. *IEEE Transactions on Information Theory*, 44(3):1193–1204, 1998.

[11] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

[12] Z. Nagy and K. Zeger. Entropy bounds for the hard-triangle model. *IEEE Transactions on Information Theory*, Submitted. Preprint.

[13] D. Pickard. Unilateral markov fields. *Adv. Appl. Probability*, 12:655–671, 1980.

[14] D.K. Pickard. A curious binary lattice process. *J. Appl. Probab*, 14:717–731, 1977.

[15] Ron Roth, Paul Siegel, and Jack Wolf. Efficient coding schemes for the hard-square model. *IEEE Transactions on Information Theory*, 47:1166–76, March 2001.

[16] Claude E. Shannon and Warren Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949.

[17] P. Vettiger, G. Cross, M. Despont, U. Drechsler, U. Durig, B. Gotsmann, W. Haberle, W. M.A. Lantz, H.E. Rothuizen, R. Stutz, and G.K. Binnig. The "millipede" - nanotechnology entering data storage. *IEEE Trans. Nanotechnol*, 1(1):39–55, 2002.

Confidence sets around critical points

Bo Markussen
Department of Computer Science
University of Copenhagen, DK-2100 Copenhagen
email: boma@diku.dk

Abstract

The critical lines and the top-points in the scale space of an image carry important information about the image, and can be used *e.g.* for image reconstruction and matching. In pratical applications these descriptors are always computed from a finite sample of the image, *i.e.* a finite number of image pixels. This implies that the critical points and the top-points in non-trivial images inherently are measured with an imprecision $\epsilon > 0$. Borrowing ideas from the theory of parameter estimation we construct confidence sets on the true position of a critical point given the position of a with imprecision observed critical point. The construction relies on a probabilistic image model, which also will be described.

1 Introduction

Scale space theory provides a method of multiscale data analysis, notably image analysis [3]. The linear Gaussian scale space $\{f_s\}_{s>0}$ of a two-dimensional gray scale image $f : \mathbb{R}^2 \to \mathbb{R}$ is defined through Gaussian blurring

$$f_s(x) = \big(g_s * f\big)(x) = \int_{\mathbb{R}^2} g_s(x-y)\, f(y)\, \mathrm{d}y, \qquad g_s(x) = \frac{1}{2\pi s}\, \mathrm{e}^{-\frac{x_1^2+x_2^2}{2s}}.$$

Observe that the scale is parameterized by the variance of the blurring kernel, hence following the notation of [6]. Local extrema and saddles in the image at the different scales are called critical points, *i.e.* the set of points given by

$$\big\{(x,s) \in \mathbb{R}^2 \times \mathbb{R}_+ \;\big|\; \nabla_x f_s(x) = (0,0)\big\}.$$

The critical points are known to move along lines as scale increases, and generically [2] are created and annihilated in pairs at the so-called top-points collected

in the set

$$\big\{(x,s) \in \mathbb{R}^2 \times \mathbb{R}_+ \;\big|\; \nabla_x f_s(x) = (0,0),\ \det H_x f_s(x) = 0\big\}.$$

The critical lines and the top-points are believed to carry essential information about the image [4, 5], and hence it is of importance to compute these quantities from image data. However, in practice a given image f is only observed at a finite set of grid points. This implies that the gradients $\nabla_x f_s(x)$ and the determinant of the Hessians $H_x f_s(x)$ can only be computed via a finite discretization of the convolution integrals, and inherently are specified with an imprecision $\epsilon > 0$. This paper investigates the associated confidence sets. We emphasize that the imprecision typically decreases as the scale increases. This important fact should be remembered when interpreting our results. However, the quantification of the imprecision as a function of scale lies outside the scope of the paper.

Stability of top-points has previously been studied in [1], where the variability of the top-points as a functional of additive image noise is investigated. The approach taken in this paper is different. Instead of considering a fixed image perturbed by additive noise we view the image itself as being random. Based on the probabilistic description of the image we ask for confidence sets $C_{\epsilon,\eta} \subset \mathbb{R}^2$ and $T_{\epsilon,\eta} \subset \mathbb{R}^2 \times \mathbb{R}_+$ on the position of the critical points and the top-points, respectively. Here ϵ denotes the imprecision and $\eta \in (0,1)$ designates a probability quantifying the degree of confidence. These concepts are explained more carefully in Section 3, where the sets $C_{\epsilon,\eta}$ are also derived. In order to define confidence sets in the first place we need a probabilistic image model, which is introduced in Section 2. We remark that the confidence set $C_{\epsilon,\eta}$ around a position (x,s) can be computed a priori without any image data. Thus, in difference to the results in [1] the local image structure at (x,s) observed from the image data is not used. This is neither a strength nor a weakness of either method, but simply means that the methods work along different directions. The two methods probably can be combined, but for clarity of ideas the analysis done in this paper is kept as simple as possible.

2 A scale invariant image model

In this section we describe the scale invariant image model derived in [6]. A probabilistic description is achieved considering the image f as a random function. In order to confine this model two invariance properties are assumed, namely stationarity and scale invariance. Stationarity means that the distribution of the image difference $f(x) - f(y)$ only depends on the difference $x - y$. Especially, there exists a covariance function $\rho\colon \mathbb{R}^2 \to \mathbb{R}_+$ such that

$$\mathrm{Var}\big(f(x) - f(y)\big) = \rho(x-y).$$

Scale invariance is formulated through scaling and blurring, with the interpretation that objects viewed at larger distance become smaller and blurred. Doing the scaling around the origin $x = 0$ scale invariance can be stated as the equivalence of the probability distributions of the images $f_s(s^{1/2}x)$ and $f_t(t^{1/2}x)$ for every $s, t > 0$, *i.e.*

$$\big\{ f_s\big(s^{1/2}\, x\big)\big\}_{x\in\mathbb{R}^2} \stackrel{\mathcal{D}}{=} \big\{ f_t\big(t^{1/2}\, x\big)\big\}_{x\in\mathbb{R}^2}.$$

Thus, the width in the blurring kernel can be interpreted as the physical scale in accordance with the usual definition of scale normalization. In [6] it is proved that the natural assumptions of stationarity and scale invariance imply a particular structure on the covariance $\Phi_s = \mathrm{Var}\big(\mathcal{J}_x(f_s)\big)$ of the associated jet $\mathcal{J}_x(f_s)$ of partial derivatives at some fixed point $x = (x_1, x_2) \in \mathbb{R}^2$, *i.e.* the sets

$$\mathcal{J}_x(f_s) = \left\{ \frac{\partial^{n+m} f_s(x)}{\partial x_1^n\, \partial x_2^m} \right\}_{(n,m)\in\mathbb{I}}, \qquad\qquad \mathbb{I} = \mathbb{N}_0^2 \setminus \big\{(0,0)\big\}.$$

Observe that the zero order structure $f_s(x)$ is removed due to the stationarity assumption. Let the alternating-sign anti-diagonals $\Psi_{2n,2m} \in \mathbb{R}^{\mathbb{I}\times\mathbb{I}}$ be defined by

$$\Psi_{2n,2m} = \left\{ (-1)^{\frac{i-k}{2}+\frac{j-l}{2}}\, 1_{i+k=2n, j+l=2m} \right\}_{(i,j),(k,l)\in\mathbb{I}},$$

where $1_{\text{condition}}$ is one if the condition is satisfied and zero otherwise. Then the covariance structure $\Phi_s = \mathrm{Var}\big(\mathcal{J}_x(f_s)\big) \in \mathbb{R}^{\mathbb{I}\times\mathbb{I}}$ does not depend on the spatial position x, and is given by

$$\Phi_s = \sum_{(n,m)\in\mathbb{I}} c_{n,m}(s)\, \Psi_{2n,2m},$$

where the functions $c_{n,m} \colon \mathbb{R}_+ \to \mathbb{R}$ are confined by the recurrence relations

$$c_{n+1,m}(s) + c_{n,m+1}(s) = \frac{n+m}{s}\, c_{n,m}(s). \tag{1}$$

Thus, the covariance structure on the non-trivial partial derivatives can be derived from invariance assumptions on the image structure. In order to have a complete probabilistic description three further components are needed: (i) a description of the first order structure, (ii) a description of the higher order structure, (iii) a choice of the functions $c_{n,m}$ satisfying the recurrence (1). Concerning issue (i) and (ii) we assume a mean zero Gaussian model, *i.e.* $\mathcal{J}_x(f_s) \sim \mathcal{N}(0, \Phi_s)$. Assuming zero mean is of course a minor issue, but the assumption of a joint Gaussian distribution is a mere postulate. Concerning issue (iii) we can either postulate a specific model, *e.g.* the Lévy Brownian motion image model with contrast parameter $\gamma > 0$ given by $c_{n,m}(s) = \gamma\, s^{-n-m}\, \frac{(2n)!\,(2m)!}{2^{2n+2m}\, n!\, m!\, (n+m)}$, or the coefficients

$c_{n,m}(s)$ can be estimated from empirical image data subject to the requirement (1). Certainly, the latter alternative is most appealing. Say, the images might not be isotropic as implied by the Brownian image model.

In conclusion stationarity and scale invariance combined with the assumption of zero mean and a joint Gaussian distribution imply the image model given by

$$f_s(y) = \sum_{(n,m)\in\mathbb{I}} \frac{f_s^{(n,m)}}{n!\,m!}(y_1 - x_1)^n (y_2 - x_2)^m, \quad \left\{f_s^{(n,m)}\right\}_{(n,m)\in\mathbb{I}} \sim \mathcal{N}\big(0, \Phi_s\big), \tag{2}$$

where the covariance structure $\Phi_s = \sum_{(n,m)\in\mathbb{I}} c_{n,m}(s)\, \Psi_{2n,2m}$ preferably is estimated from empirical image data subject to the recursion requirement Eq. (1).

3 Confidence sets

Given an observation f from the image model derived in Section 2 we want to compute the critical lines and their top-points. Suppose we have computed the existence of a critical point at position (x, s_0), but that the computations only can be done with imprecision $\epsilon > 0$. Since the critical points move along lines in scale we ask for a set $C_{\epsilon,\eta}(x, s_0) \subset \mathbb{R}^2$ around x containing an x_0 such that (x_0, s_0) is a true critical point with some confidence. Given a probability $\eta \in (0, 1)$ we call

$$C_{\epsilon,\eta}(x, s_0) = \left\{ y \in \mathbb{R}^2 \;\middle|\; \mathbb{P}\Big(\nabla_x f_{s_0}(x) \in [-\epsilon, \epsilon]^2 \;\Big|\; \nabla_y f_{s_0}(y) = (0,0)\Big) \geq \eta \right\}$$

an η-confidence set for x_0 given an with ϵ-imprecision computed critical point at (x, s_0). We remark that a classical η-confidence set $D_{\epsilon,\eta}(x, s_0)$ satisfies

$$\mathbb{P}\left(\bigcup_{x_0 \in D_{\epsilon,\eta}(x,s_0)} \nabla_x f_{s_0}(x_0) = (0,0) \;\middle|\; \nabla_x f_{s_0}(x) \in [-\epsilon, \epsilon]^2 \right) \geq \eta.$$

However, we believe the probabilistic analysis needed to find the sets $D_{\epsilon,\eta}(x, s_0)$ to be very demanding. Thus, although providing a weaker statement we settle for the sets $C_{\epsilon,\eta}(x, s_0)$. Let the set $\widehat{C}_{\epsilon,\eta}(x_0, s_0) \subset \mathbb{R}^2$ be given such that

$$\mathbb{P}\Big(\nabla_y f_{s_0}(y) \notin [-\epsilon, \epsilon]^2 \;\Big|\; \nabla_x f_{s_0}(x_0) = (0,0)\Big) > 1 - \eta$$

for $y \notin \widehat{C}_{\epsilon,\eta}(x_0, s_0)$. The stationarity of the image model implies that x_0 can be translated to the origin, *i.e.* $\widehat{C}_{\epsilon,\eta}(x_0, s_0) = x_0 + \widehat{C}_{\epsilon,\eta}(s_0)$ with $\widehat{C}_{\epsilon,\eta}(s_0) = \widehat{C}_{\epsilon,\eta}(0, s_0)$. Then the confidence set $C_{\epsilon,\eta}(x, s_0)$ is given by

$$C_{\epsilon,\eta}(x, s_0) = x + C_{\epsilon,\eta}(s_0) = x - \widehat{C}_{\epsilon,\eta}(s_0).$$

To see this assume $x_0 \notin C_{\epsilon,\eta}(x, s_0)$, *i.e.* $x \notin \widehat{C}_{\epsilon,\eta}(x_0, s_0)$. Then the conditional probability that the gradient $\nabla_x f_{s_0}(x)$ lies outside the box $[-\epsilon, \epsilon]^2$, and hence differs from $(0,0)$ within imprecision ϵ, is larger than $1-\eta$. Reverting this statement gives the crucial implication

$$\mathbb{P}\Big(\nabla_x f_{s_0}(x) \in [-\epsilon,\epsilon]^2 \;\Big|\; \nabla_x f_{s_0}(x_0) = (0,0)\Big) \geq \eta \implies x_0 \in C_{\epsilon,\eta}(x, s_0).$$

In Section 3.1 we find such confidence sets $C_{\epsilon,\eta}(s) \subset \mathbb{R}^2$, and examples are provided in Section 4. Finding confidence sets $T_{\epsilon,\eta}(s) \subset \mathbb{R}^2 \times \mathbb{R}$ for top-points at scale s is more involved due to the following two facts: (i) the definition of top-points involve the determinant of the Hessian, which is a non-linear functional of the image, (ii) top-points are located at unique scales, and hence also correlation across scale is needed. We leave the study of the confidence sets $T_{\epsilon,\eta}(s)$ for future research.

3.1 Spatial position of critical points

Suppose (x, s) is a critical point for an image f following the probabilistic model described in Eq. (2). Due to stationarity we can assume without loss of generality that $x = 0$. Doing this we have the image gradients

$$\begin{aligned}\nabla_y f_s(y) &= \Big(\sum_{(n,m)\in\mathbb{N}_0^2} \frac{f_s^{(n+1,m)}}{n!\,m!}\, y_1^n\, y_2^m, \sum_{(n,m)\in\mathbb{N}_0^2} \frac{f_s^{(n,m+1)}}{n!\,m!}\, y_1^n\, y_2^m \Big) \\ &= \big(f_s^{(1,0)}, f_s^{(0,1)}\big) + \Big(\sum_{(n,m)\in\mathbb{I}} \frac{f_s^{(n+1,m)}}{n!\,m!}\, y_1^n\, y_2^m, \sum_{(n,m)\in\mathbb{I}} \frac{f_s^{(n,m+1)}}{n!\,m!}\, y_1^n\, y_2^m \Big).\end{aligned}$$

Let $\widetilde{\mathbb{I}} = \mathbb{N}_0^2 \setminus \big\{(0,0),(1,0),(0,1)\big\}$ and let $\widetilde{\Phi}_s$ be the conditional variance of $\big\{f_s^{(n,m)}\big\}_{(n,m)\in\widetilde{\mathbb{I}}}$ given $f_s^{(1,0)} = f_s^{(0,1)} = 0$. This variance is calculated in the appendix. Conditionally on $\nabla_x f_s(0) = \big(f_s^{(1,0)}, f_s^{(0,1)}\big) = (0,0)$ we have

$$\begin{aligned}\nabla_y f_s(y) &= \Big(\sum_{(n,m)\in\mathbb{I}} \frac{f_s^{(n+1,m)}}{n!\,m!}\, y_1^n\, y_2^m, \sum_{(n,m)\in\mathbb{I}} \frac{f_s^{(n,m+1)}}{n!\,m!}\, y_1^n\, y_2^m \Big), \\ \big\{f_s^{(n,m)}\big\}_{(n,m)\in\widetilde{\mathbb{I}}} &\sim \mathcal{N}\big(0, \widetilde{\Phi}_s\big).\end{aligned}$$

Thus, the conditional distribution of $\nabla_y f_s(y)$ is a two-dimensional Gaussian distribution. In order to simplify the computations we separate the coordinates of

$\nabla_y f_s(y)$. Let $\sigma_1(y)^2$ and $\sigma_2(y)^2$ be the corresponding marginal variances, *i.e.*

$$\begin{aligned}\sigma_1(y)^2 &= \sum_{(\nu,\mu),(n,m)\in\mathbb{I}} \widetilde{\phi}_s^{(\nu+1,\mu),(n+1,m)}\ \frac{y_1^{\nu+n}\, y_2^{\mu+m}}{\nu!\, n!\, \mu!\, m!},\\ \sigma_2(y)^2 &= \sum_{(\nu,\mu),(n,m)\in\mathbb{I}} \widetilde{\phi}_s^{(\nu,\mu+1),(n,m+1)}\ \frac{y_1^{\nu+n}\, y_2^{\mu+m}}{\nu!\, n!\, \mu!\, m!},\end{aligned} \tag{3}$$

where $\widetilde{\Phi}_s = \{\widetilde{\phi}_s^{(\nu,\mu),(n,m)}\}_{(\nu,\mu),(n,m)\in\mathbb{I}}$. Using Boole's inequality and

$$\frac{\partial f_s(y)}{\partial y_1} \sim \mathcal{N}\big(0, \sigma_1(y)^2\big), \qquad\qquad \frac{\partial f_s(y)}{\partial y_2} \sim \mathcal{N}\big(0, \sigma_2(y)^2\big)$$

we have that $\mathbb{P}\big(\nabla_y f_s(y) \notin [-\epsilon,\epsilon]\times[-\epsilon,\epsilon] \;\big|\; \nabla_x f_s(0) = (0,0)\big)$ is larger than[1]

$$\begin{aligned}1 - \mathbb{P}\bigg(\frac{\partial f_s(y)}{\partial y_1} \in [-\epsilon,\epsilon] \;\bigg|\; \nabla_x f_s(0) = (0,0)\bigg)\\ -\mathbb{P}\bigg(\frac{\partial f_s(y)}{\partial y_2} \in [-\epsilon,\epsilon] \;\bigg|\; \nabla_x f_s(0) = (0,0)\bigg).\end{aligned} \tag{4}$$

Introducing the distribution function $F(u) = \mathbb{P}\big(U \le u\big)$ for a standard normal variable $U \sim \mathcal{N}(0,1)$, and the inverse function F^{-1}, this lower bound equals

$$1 - \Big(2F\big(\tfrac{\epsilon}{\sigma_1(y)}\big) - 1\Big) - \Big(2F\big(\tfrac{\epsilon}{\sigma_2(y)}\big) - 1\Big) = 3 - 2F\big(\tfrac{\epsilon}{\sigma_1(y)}\big) - 2F\big(\tfrac{\epsilon}{\sigma_2(y)}\big).$$

If $\sigma_1(y) \ge \frac{\epsilon}{F^{-1}(\frac{2+\eta}{4})}$ and $\sigma_2(y) \ge \frac{\epsilon}{F^{-1}(\frac{2+\eta}{4})}$, then we have

$$\mathbb{P}\Big(\nabla_y f_s(y) \notin [-\epsilon,\epsilon]\times[-\epsilon,\epsilon] \;\Big|\; \nabla_x f_s(0) = (0,0)\Big) > 1 - \eta.$$

Remembering that $C_{\epsilon,\eta}(s) = -\widehat{C}_{\epsilon,\eta}(s)$ is the complement of the set of points y satisfying these inequalities we have the ϵ-imprecision η-confidence set

$$C_{\epsilon,\eta}(s) = \left\{ y \in \mathbb{R}^2 \;\middle|\; \sigma_1(-y)^2 < \left(\frac{\epsilon}{F^{-1}\big(\frac{2+\eta}{4}\big)}\right)^2,\ \sigma_2(-y)^2 < \left(\frac{\epsilon}{F^{-1}\big(\frac{2+\eta}{4}\big)}\right)^2 \right\}. \tag{5}$$

[1]The application of Boole's inequality implies that the derived confidence set actually is larger than the exact confidence set. We will not persue this further, but see the discussion in Section 4.

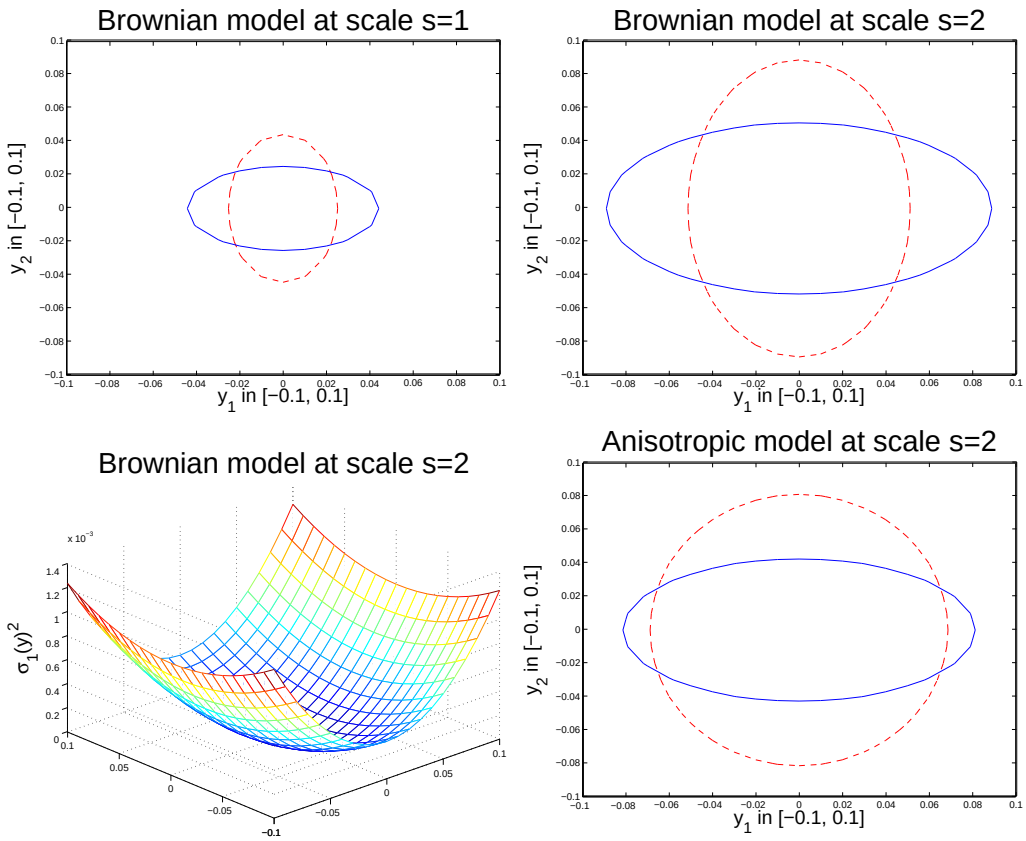


Figure 1: Confidence sets $C_{\epsilon,\eta}(0,s)$ with $\epsilon = 0.01$ and $\eta = 0.95$. Points inside the dashed red curve fulfill the condition on $\sigma_1(y)^2$, and points inside the blue curve fulfill the condition on $\sigma_2(y)^2$. The confidence set consists of the points inside both curves. The lower left panel shows the marginal variance $\sigma_1(y)^2$ as a function of y. The dashed red curve in the upper right panel is a level curve for this function, *cf.* Eq. (5).

4 Examples of confidence sets

The explicit computation of the marginal variance functions $\sigma_1(y)^2$ and $\sigma_2(y)^2$ seems to be a rather demanding task. So we are content with a few numerical experiments, where we replace the infinite sum in Eq. (3) with a finite sum over $(\nu,\mu),(n,m)$ with $\nu+n+\mu+m \le N$ for some $N \in \mathbb{N}$. Doing this facilitates the choice of coefficients $c_{n,m}(s)$ satisfying Eq. (1). Only the coefficients $c_{n,m}(s)$ with $n+m \le M \stackrel{\text{def}}{=} \lfloor \frac{N}{2} \rfloor + 2$ are needed. We freely choose $c_{n,M-n}$ for $n = 0,\dots,M$, take $c_{n,M-n}(s) = s^{-M}\, c_{n,M-n}$, and compute the remaining coefficients $c_{n,m}(s)$

using Eq. (1). We consider two examples: (i) The Brownian image model given by

$$c_{n,m}^{\text{Brownian}} = \frac{(2n)!\,(2m)!}{2^{2n+2m}\,n!\,m!\,(n+m)}.$$

(ii) An anisotropic model, where we for $N = 10$ have scaled each of the coefficients $c_{n,M-n}^{\text{Brownian}}$ by independent uniformly distributed random numbers in $[0, 2]$. We expect the confidence sets to be narrowest in directions of large variance of the image increments. The particular realization used is given by

$$\big\{c_{n,7-n}\big\}_{n=0,\ldots,7} = \big\{286.59,\ 5.36,\ 3.84,\ 1.70,\ 3.13,\ 4.82,\ 10.59,\ 5.58\big\}.$$

Examples of the associated confidence sets are given in Figure 1. These are not ellipse-like as intuitively expected, but have corners at points where the two conditions meet. This is due to the application of Boole's inequality for the derivation of the lower bound in Eq. (4). The correct, and smaller and ellipse-like, confidence sets should be derived using the two-dimensional distribution of $\nabla_y f_s(y)$. This, however, complicates the probabilistic analysis and will not be pursued further. We also observe that the confidence sets for fixed imprecision ϵ and confidence level η increase with the scale s. However, as already mentioned the imprecision is expected to decrease as scale increases. Finally, remark the notable anisotropy visible in the lower left panel, where the anisotropy $c_{7,0} \ll c_{0,7}$ is reflected by the elongation of the confidence set along the first coordinate axis.

Appendix: The conditional variance

Introducing $X = \big\{f_s^{(i,j)}\big\}_{(i,j)\in\{(1,0),(0,1)\}}$ and $Y = \big\{f_s^{(n,m)}\big\}_{(n,m)\in\widetilde{\mathbb{I}}}$ we have

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\big(0, \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}\big)$$

with conditional distribution $\mathcal{L}\big(Y|X=(0,0)\big) = \mathcal{N}\big(0, \Sigma_{YY} - \Sigma_{YX}\,\Sigma_{XX}^{-1}\,\Sigma_{XY}\big)$. We have, also using $\Psi_{2n,2m}$ for the restriction to the set $\widetilde{\mathbb{I}} \times \widetilde{\mathbb{I}}$,

$$\Sigma_{XX} = \begin{pmatrix} c_{1,0}(s) & 0 \\ 0 & c_{0,1}(s) \end{pmatrix}, \qquad \Sigma_{YY} = \sum_{(n,m)\in\mathbb{I}_2} c_{n,m}(s)\,\Psi_{2n,2m}.$$

Furthermore, $\Sigma_{YX} = \Sigma_{XY}^T$ and

$$\begin{aligned}\Sigma_{XY} &= \sum_{(n,m)\in\widetilde{\mathbb{I}}} c_{n,m}(s)\,\Big\{(-1)^{\frac{i-k}{2}+\frac{j-l}{2}}\,1_{i+k=2n,j+l=2m}\Big\}_{(i,j)\in\{(1,0),(0,1)\},\ (k,l)\in\widetilde{\mathbb{I}}} \\ &= \sum_{(n,m)\in\widetilde{\mathbb{I}}} (-1)^{n+m-1}\,c_{n,m}(s)\,\begin{pmatrix} 1_{(i,j)=(1,0),(k,l)=(2n-1,2m)} \\ 1_{(i,j)=(0,1),(k,l)=(2n,2m-1)} \end{pmatrix}_{(k,l)\in\widetilde{\mathbb{I}}}.\end{aligned}$$

Combining these covariances we have, that $\widetilde{\Phi}_s = \Sigma_{YY} - \Sigma_{YX}\,\Sigma_{XX}^{-1}\,\Sigma_{XY}$ equals

$$\sum_{(n,m)\in\widetilde{\mathbb{I}}} c_{n,m}(s)\;\Psi_{2n,2m}$$

$$- \sum_{(\nu,\mu),(n,m)\in\widetilde{\mathbb{I}}} (-1)^{\nu+n+\mu+m}\frac{c_{\nu,\mu}(s)\;c_{n,m}(s)}{c_{1,0}(s)}\left\{1_{(i,j)=(2\nu-1,2\mu),(k,l)=(2n-1,2m)}\right\}_{(i,j),(k,l)\in\widetilde{\mathbb{I}}}$$

$$- \sum_{(\nu,\mu),(n,m)\in\widetilde{\mathbb{I}}} (-1)^{\nu+n+\mu+m}\frac{c_{\nu,\mu}(s)\;c_{n,m}(s)}{c_{0,1}(s)}\left\{1_{(i,j)=(2\nu,2\mu-1),(k,l)=(2n,2m-1)}\right\}_{(i,j),(k,l)\in\widetilde{\mathbb{I}}}.$$

Thus, the $\big((i,j),(k,l)\big)$'th element $\widetilde{\phi}_s^{(i,j),(k,l)}$ in $\widetilde{\Phi}_s$ is given by

$$(-1)^{\frac{i-k}{2}+\frac{j-l}{2}}\left(c_{\frac{i+k}{2},\frac{j+l}{2}}(s) - 1_{i\text{ odd, }j\text{ even}}\frac{c_{\frac{i+1}{2},\frac{j}{2}}(s)\;c_{\frac{k+1}{2},\frac{l}{2}}(s)}{c_{1,0}(s)} - 1_{i\text{ even, }j\text{ odd}}\frac{c_{\frac{i}{2},\frac{j+1}{2}}(s)\;c_{\frac{k}{2},\frac{l+1}{2}}(s)}{c_{0,1}(s)}\right)$$

for $i+k$ and $j+l$ even, and vanishes otherwise.

References

[1] E. Balmachnova, L.M.J. Florack, B. Platel, F. Kanters & B.M.T. Haar Romeny, "Stability of top-points in scale space", proceeding 5th Scale Space conference, LNCS 3459, 67–72, 2005.

[2] J. Damon, "Local Morse theory for solutions to the heat equation and Gaussian blurring", Journal of Differential Equations, vol. 12, 368–401, 1995.

[3] L.M.J Florack, "Image Structure", The series of Computational Imaging and Vision, vol. 10, Kluwer Academic Publishers, 1997.

[4] P. Johansen, M. Nielsen & O. F. Olsen, "Branch points in one-dimensional Gaussian scale space', Journal of Mathematical Imaging and Vision, vol. 13, 193–203, 2000.

[5] F. Kanters, M. Lillholm, R. Duits, B. Janssen, B. Platel, L.M.J. Florack & B.M.T. Haar Romeny, "On image reconstruction from multiscale top points", proceedings 5th Scale Space conference, LNCS 3459, 431–442, 2005.

[6] B. Markussen, K. S. Pedersen & M. Loog, "A scale invariant covariance structure on jet space", International Workshop on Deep Structure, Singularities and Computer Vision, Maastricht, proceedings forthcoming in LNCS, 2005.

A face recognition algorithm based on multiple individual discriminative models

Jens Fagertun, David Delgado Gomez, Bjarne K. Ersbøll, Rasmus Larsen

***Abstract*— In this paper, a novel algorithm for facial recognition is proposed. The technique combines the color texture and geometrical configuration provided by face images. Landmarks and pixel intensities are used by Principal Component Analysis and Fisher Linear Discriminant Analysis to associate a one dimensional projection to each person belonging to a reference data set. Each of these projections discriminates the associated person with respect to all other people in the data set. These projections combined with a proposed classification algorithm are able to statistically deciding if a new facial image corresponds to a person in the database. Each projection is also able to visualizing the most discriminative facial features of the person associated to the projection. The performance of the proposed method is tested in two experiments. Results point out the proposed technique as an accurate and robust tool for facial identification and unknown detection.**

***Index Terms*— Face recognition, Principal Component Analysis, Fisher Linear Discriminant Analysis, Biometrics, Multi-Subspace Method.**

I. INTRODUCTION

Regrettable events which happened during the last years (New York, Madrid) have revealed flaws in the existing security systems. The vulnerability of most of the current security and personal identification system is frequently shown. Falsification of identity cards or intrusion into physical and virtual areas by cracking alphanumerical passwords appear frequently in the media. These facts have triggered a real necessity for reliable, user-friendly and widely acceptable control mechanisms for person identification and verification.

Biometrics, which bases the person authentication on the intrinsic aspects of a human being, appears as a viable alternative to more traditional approaches (such as PIN codes or passwords). Among the oldest biometrics techniques, fingerprint recognition can be found. It is known that this technique was used in China around 700 AD to officially certify contracts. Afterwards, in Europe, it was used as person identification in the middle of the 19^{th} century. A more recent biometric technique used for people identification is iris recognition [8]. It has been calculated that the chance of finding two randomly formed identical irises is one in 10^{78} (The population of the earth is below 10^{10}) [7]. This technique has started to be used as and alternative to passport in some airports in United Kingdom, Canada and Netherlands. It is also used as employee control access to restricted areas in Canadian airports and in the New York JFK airport. The inconvenient of these techniques is the necessity of interaction with the individual who wants to be identified or authenticated. This fact has caused that face recognition, a non-intrusive technique, has increasedly received the interest from the scientific community in recent years.

The first developed techniques that aimed at identifying people from facial images were based on geometrical information. Relative distances between key points, such as mouth corners or eyes, were calculated and used to characterize faces [17]. Therefore, most of the developed techniques during the first stages of facial recognition focused on the automatic detection of individual facial features. However, facial feature detection and measurements techniques developed to date are not reliable enough for the geometric feature based recognition, and such geometric properties alone are inadequate for face recognition because rich information contained in the facial texture or appearance is discarded [6], [13]. This fact produced that gradually most of the geometrical approaches were abandoned for color based techniques, which provided better results. These methods aligned the different faces to obtain a correspondence between pixels intensities. A nearest neighbor classifier used these aligned values to classify the different faces. This coarse method was notably enhanced with the appearance of the Eigenfaces technique [15]. Instead of directly comparing the pixel intensities of the different face images, the dimension of these input intensities were first reduced by a principal component analysis (PCA). This technique settled the basis of many of the current image based facial recognition schemes. Among these current techniques, Fisherfaces can be found. This technique, widely used and referred [2], [4], combines the Eigenfaces with Fisher linear discriminant analysis (FLDA) to obtain a better separation of the individuals. In Fisherfaces, the dimension of the input intensity vectors is reduced by PCA and then FLDA is applied to obtain a good separation of the different persons.

After Fisherfaces, many related techniques have been proposed. These new techniques aim at providing a projection that attain a good person discrimination and also are robust at differences in illumination or image pose. Kernel Fisherfaces [16], Laplacianfaces [10] or discriminative common vectors [3] can be found among these new approaches. Typically, these techniques have been tested assuming that the image to be classified corresponds to one of the people in the database. In these approaches, the image is usually classified to the person with the smallest Euclidean distance.

However, some inconveniences appear when the person to be analyzed may not belong to the data set. In this case, a criterium to decide if the person belongs to the data set has to be chosen. E.g. only people with an euclidian distance less than a given threshold are considered as belonging to the data set. However, this threshold has not to be necessarily the same

for all the classes (different persons) and different thresholds would need to be found. The estimation of these thresholds is not straightforward and additional data might be needed.

In this work, a new technique that addresses the different inconveniences is proposed. The proposed techniques takes advantage of two novelties in order to deal with these inconveniences. First, not only the texture intensities are taken into account but also the geometrical information. Second, the data are projected into n one-dimensional spaces instead of a $(n-1)$-dimensional space, where n is the number of people in the data set.

Each of these individual models aims at characterizing a given person uniquely. This means that every person in the data set is represented by one model. These multi one-dimensional models allow to statistically interpret the "degree of membership" of a person to the data set and to detect unknowns. Furthermore, these two facts have several advantages in interpretability, characterization, accuracy and easiness to update the model.

II. Algorithm Description

The proposed algorithm is made up of two steps. In the first step, an individual model is built for each person in the database using the color and geometrical information provided by the available images. Each model characterizes a given person and discriminates it from the other people in the database. The second step carries out the identification. A classifier, related with the standard Gaussian distribution, decides if a face image belongs to one person in the database or not. In this section, the two parts of the algorithm are described in detail. A diagram of the algorithm is displayed in Fig. 1. This diagram will be referred during the description of the algorithm to obtain an easier understanding.

A. Creating the individual models

1) Obtaining the geometry of the face: The geometrical characterization of a given face is obtained by means of the theory of statistical shape analysis [1]. In this theory, objects (faces) are represented by shapes. According to Kendall [11], a shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object. In order to describe a shape, a set of landmarks or points of correspondence that matches between and within populations are placed on each face. As an example, Fig. 2A displays a set of 22 landmarks. These landmarks indicate the position of the eyebrows, eyes, nose, mouth, jaw and size of a given face.

To obtain a shape representation according to the definition, the obtained landmarks are aligned in order to remove the location, rotational and scaling effects. To achieve this goal, the 2D-full Procrustes analysis is carried out. Briefly, let:

$$\mathbf{X} = \{\mathbf{x}_i\} = \{x_i + i \cdot y_i\}, \quad i = 1, \ldots, n$$

be a set of n landmarks expressed in complex notation. In order to apply full Procrustes analysis, the shapes are initially

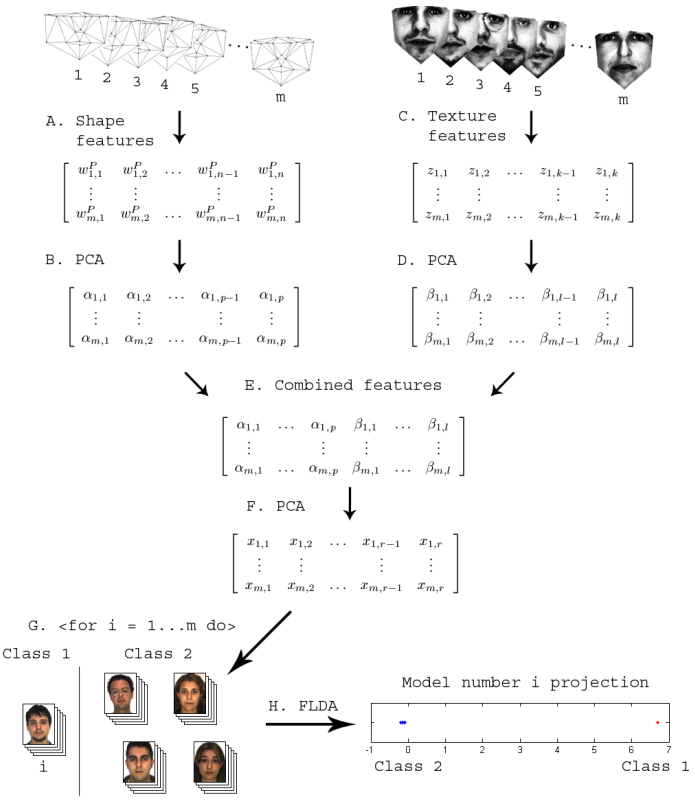


Fig. 1. Algorithm overview. A: Landmarks alignment using full Procrustes analysis. B: PCA on aligned landmarks to remove redundancy. C: Texture normalization using global histogram equalization. D: PCA on normalized texture to remove redundancy. E: Combining shape and texture features. F: PCA on combined features to remove redundancy. G & H :In turn build the individual model using FLDA.

(A) (B)

Fig. 2. (A) Set of 22 landmarks placed on a face image. (B) The Delaunay triangulation of the 22 landmarks.

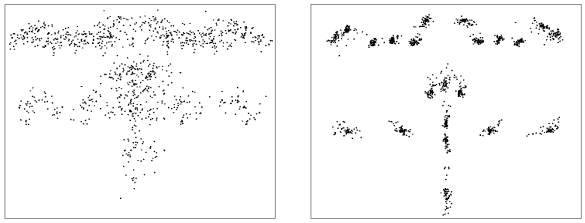
centered. To center the different shapes, the mean of the shape, $\bar{\mathbf{x}}$, is subtracted from each landmark:

$$\mathbf{w}_i = \mathbf{x}_i - \bar{\mathbf{x}}, \quad i = 1, \ldots, n$$

.

The full Procrustes mean shape [12], $\hat{\mu}$, is found as the eigenvector corresponding to the largest eigenvalue of the complex sum of squares and products matrix

$$\sum_{i=1}^{n} \mathbf{w}_i \mathbf{w}_i^* / (\mathbf{w}_i^* \mathbf{w}_i)$$



(A) (B)

Fig. 3. (A) Superimposition of the sets of 22 landmarks obtained over 49 different face images. (B) Alignment of the landmarks.

where $\mathbf{w}_i^*$ denotes the transpose of the complex conjugate of $\mathbf{w}_i$. Using this Procrustes mean shape, the full Procrustes coordinates of $\mathbf{w}_1, \ldots, \mathbf{w}_n$ (Fig. 1) A) are obtained by

$$\mathbf{w}_i^P = \mathbf{w}_i^* \hat{\mu} \mathbf{w}_i / (\mathbf{w}_i^* \mathbf{w}_i) \;\; i = 1, \ldots, n$$

Fig. 3A displays the superimposition of the set of 22 landmarks described in Fig. 2, obtained on 49 different face images. The result obtained after applying the full Procustres alignment on theses landmarks can be observed in Fig. 3B. In order to remove redundancy in the data, a Principal Component Analysis is applied to the aligned landmarks (Fig. 1 B).

2) Texture formulation: To form a complete model of the face appearance, the algorithm also captures the texture information provided by the pixels. In order to collect this texture representation, the Delaunay triangulation of every shape is obtained. The Delanuay triangulation connects the aligned landmark set of each image by a mesh of triangles, so no triangle has any of the other points of the set inside its circumcircle. The Delaunay triangulation obtained for each image is warped onto the Delaunay triangulation of the mean shape. The Delanuay triangulation of the 22 landmarks is displayed in Fig. 2B.

Formally, let I be a given image and M the mean shape previously obtained. Let $\mathbf{u}_1 = [x_1, y_1], \mathbf{u}_2 = [x_2, y_2]$ and $\mathbf{u}_3 = [x_3, y_3]$ denote the vertices of a triangle T in I, and let $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ be the associated vertices of the corresponding triangle in M. Given any internal point $\hat{\mathbf{u}} = [x, y]$ in the triangle T, the corresponding point in the associated triangle in the mean shape can be written as $\hat{\mathbf{v}} = \alpha \mathbf{v}_1 + \beta \mathbf{v}_2 + \gamma \mathbf{v}_3$ where:

$$\begin{aligned}
\alpha &= 1 - (\beta + \gamma) \\
\beta &= \frac{yx_3 - x_1 y - x_3 y_1 - y_3 x + x_1 y_3 + x y_1}{-x_2 y_3 + x_2 y_1 + x_1 y_3 + x_3 y_2 - x_3 y_1 - x_1 y_2} \\
\gamma &= \frac{x y_2 - x y_1 - x_1 y_2 - x_2 y + x_2 y_1 + x_1 y}{-x_2 y_3 + x_2 y_1 + x_1 y_3 + x_3 y_2 - x_3 y_1 - x_1 y_2}
\end{aligned}$$

This transformation extracts the texture of a given face image. A histogram equalization is applied to the collected texture to reduce the effects of differences in illumination [9]. This histogram equalization is performed independently in each of the three color channels. Afterwards, the three color channels are converted into gray scale to obtain a more compact representation (Fig. 1 C).

Similarly to the shape analysis, a PCA is conducted in the texture data to reduce dimensionality and data redundancy (Fig. 1 D). However, notice that the large dimension of the texture vectors will produce memory problems because of the huge dimension of the covariance matrix. In order to avoid this difficulty, the Eckart-Young theorem is used [5]. Formally, let $\mathbf{D}$ represents the texture data matrix composed by s n-dimensional texture vectors after the mean of the texture vectors has been subtracted from each one of them ($s << n$). Then the $n \times n$ dimensional covariance matrix can be written as:

$$\mathbf{\Sigma_D} = \frac{1}{s} \mathbf{D} \mathbf{D}^T$$

Let $\mathbf{\Sigma_S}$ be the smaller $s \times s$ dimensional matrix defined by

$$\mathbf{\Sigma_S} = \frac{1}{s} \mathbf{D}^T \mathbf{D}$$

Then the non-zero eigenvalues of the matrices $\mathbf{\Sigma_S}$ and $\mathbf{\Sigma_D}$ are equal. Moreover, the columns of:

$$\mathbf{\Phi_D} = \mathbf{D} \cdot \mathbf{\Phi_S}$$

where the columns of $\mathbf{\Phi_S}$ contain the eigenvectors of $\mathbf{\Sigma_S}$, correspond with the the eigenvectors associated to the non-zero eigenvalues of $\mathbf{\Sigma_D}$ in the sense they have the same direction. Therefore, if the columns of $\mathbf{\Phi_D}$ are normalized, then $\mathbf{\Phi_D}$ holds the normalized eigenvectors of $\mathbf{\Sigma_D}$ that has eigenvalues bigger than zero. This not only avoid problems with the memory but also it gives a substantial speed up of the calculations.

3) Combining color and geometry: The shape and texture features are concatenated in a matrix (Fig. 1 E). In order to remove correlation between shape and texture and also to make the data representation more compact, a third PCA is performed on the concatenated shape and texture matrix (Fig. 1 F).

4) Building an individual model: Once the geometry and texture of the face have been captured, the proposed algorithm builds an individual model for each person in the data set. Each model is built using Fisher linear discriminant analysis. Formally, let $\mathbf{X}$ be the data obtained after combining the shape and texture and applying the PCA. Let n_1 be the number of data elements corresponding to the person for whom the model is being created (class 1) and let n_2 be the number of elements corresponding to the other people (class 2), (Fig. 1 G). Let $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ be the class mean vectors, $\bar{\mathbf{x}}$ be the total mean vector and $\mathbf{x}_{i,j}$ be the jth sample in the ith class. Then the between matrix is defined by:

$$\mathbf{B} = n_1 (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}})(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}})^T + n_2 (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}})(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}})^T$$

and the within matrix is defined by:

$$\mathbf{W} = \sum_{i=1}^{2} \sum_{j=1}^{n_i} (\mathbf{x}_{i,j} - \bar{\mathbf{x}}_i)(\mathbf{x}_{i,j} - \bar{\mathbf{x}}_i)^T$$

The projection that best discriminates the two populations is given by the direction of the eigenvector associated to the maximum eigenvalue of $\mathbf{W}^{-1}\mathbf{B}$ (Fig. 1 H). To ensure that

the within matrix $\mathbf{W}$ is not singular, only the f first data variables are taken into account, where f is the number of non-zero eigenvalues of the within matrix $\mathbf{W}$.

B. classification

In order to obtain a method to classify a given image, the different individual models are firstly standardized so they can be compared. The standardization of model $i = 1, \ldots, m$ is based on two assumptions. First, the number of observations for person i is much smaller than the number of the observations of all other people. The second assumption is that the projection of the other people follows a Gaussian distribution. These two assumptions imply that the distribution of all the projected facial images on a particular discriminative individual model can be assumed as a Gaussian distribution with outliers. The standardization of model i is then achieved by transforming the projections into a standard Gaussian distribution, keeping the projections of the person i positive. Formally, let $\bar{x}_i$ be the mean of the projections on model i, σ_i the standard deviation, and let $x_{i,j}$ be the projection of image j in model i. These projections are standardized by:

$$\hat{x}_{i,j} = (x_{i,j} - \bar{x}_i)/\sigma_i$$

If the standardized projection for the images corresponding to person i are negative, then $\hat{x}_{i,j}$ are replaced by $-\hat{x}_{i,j}$ for all projections. This causes the projection of the images corresponding to person i to be positive and far from the mean of the gaussian.

Once that the model i is standardized, the probability of a projected image of belonging to the person i is given by the value of the standard normal cumulative function in the projected value. This fact is used to classify a given image. If it is assumed that the image belongs to a person from the data set, the image is projected by all the models and classified as belonging to the model that gives the largest probability. Moreover, it is also statistically possible to decide if a given person belongs to the data set or it is unknown. This can be achieved by comparing the largest projection obtained in all the models with a probabilistic threshold. E.g, if a 99.9% of probability is required, a given image will only be considered as belonging to the database if the projection in one of the individual models is higher than 3.1 standard deviations.

III. Experimental Results

Two experiments are conducted in order to evaluate the performance of the proposed method. The objective of the first experiment is to evaluate the recognition ability in terms of correct classification rates. This first experiment also aims at ranking the importance of shape and texture. The second experiment aims at analyzing if the proposed method can be incorporated into a biometrical facial recognition scheme. The robustness of the proposed method to the presence of unknowns is considered in this second experiment.

A. Experiment one

The first experiment aims at comparing the performance of the proposed method with respect to the Fisherfaces method in terms of correct classification rates. In order to be consistent with a previously published work [15], unknown people are not taken into account.

To achieve this first goal the AR face database [14] is used. The database is composed of two independent sessions, recorded 14 days apart. At both sessions, each person was recorded 13 times, under various facial poses (all frontal), lighting conditions and occlusions. The size of the images in the database is 768 $\times$ 576 pixels, represented in 24 bits RGB color format.

In this study, a subset of 50 persons (25 male and 25 female) from the database was randomly selected. Seven images per person without occlusions are used from each session. Therefore, the experiment data set is composed of 700 images, with 14 images per person. An example of the selected images for one person is displayed in Fig. 4.

Fig. 4. The AR data set: (Top row) The seven images without occlusions from first session, (Bottom row) The seven images without occlusions from the second session.

All the images were manually annotated with the 22 landmarks previously mentioned.

The data set was divided into two sets. The images of the first session were used to build the individual discriminative models, and images from the second session were subsequently used to test the performance.

The landmarks corresponding to the images in the training set were aligned using full Procrustes analysis. The 44 (x,y)-coordinates were obtained to represent the geometrical configuration of each face. In order to obtain the texture of each face in the training set, the different images were warped with respect to the mean shape. Each of the textures received a histogram equalization in each color band to reduce the differences in global illumination. The textures were converted to gray scale and represented by 41337 pixels. The geometrical and color representation of each face was combined, reduced and the individual models were built as described in Section II.

The test set was used to evaluate and compare the proposed method with respect to the Fisherface technique. In order to evaluate the importance of the geometrical information, the Fisherface technique was modified replacing the texture data with the shape data and also combining the shape with the texture. These two modified techniques will be referred to as Fishershape and Fishercombined from now on. The Euclidean Nearest-Neighbor algorithm was used as classifier algorithm

Method	Input features	Correct Classification Rate[1]
Proposed method	Shape	86.4% (95)
Proposed method	Texture	99.6% (3)
Proposed method	Texture and Shape	99.9% (1)
Fishershape	Shape	85% (105)
Fisherface	Texture	98.9% (8)
Fishercombined	Texture and Shape	99.7% (2)

TABLE I

AVERAGE CORRECT CLASSIFICATION RATES.

in the Fisher methods. The proposed method classified the images as the person associated to the model that yields the highest probability.

The test was repeated a second time changing the roles of the training and test sets. So session two was used as training data and session one as test data. The average correct classification rates for the different techniques are shown in Table I.

From Table I, it is observed that the proposed method has a slightly better performance than the Fisher methods. Moreover, it is also noticed that using the texture data one obtains a higher accuracy than when the shape is used. This implies that the information contained in the texture is more significant than that included in the shape. However, the information contained in the shape data is not insignificant. The highest correct classification rate in both techniques is attained when both shape and texture are considered.

Fig. 5. The 10, 15 and 25% most important pixels (shown in red) for discriminating between the 50 test persons.

An interesting property of the proposed algorithm are that it is possible to determine which are the most discriminative features of a given person. In order to illustrate this fact, four models were built using only the texture. The pixels of the faces corresponding to these models which received the 10, 15 and 25% highest weights in the model are displayed (in red) in Fig. 5. It is clear that important discriminating features include eyes, noses, glasses, moles and beards. Notice that the algorithm detects the glasses and the two moles of person 43 as discriminate features.

B. Experiment two

The objective of this second experiment is to test the possibility of incorporating the proposed technique into a biometrical facial recognition scheme. This conveys the identification of people in a data set and also the detection of unknown people. The good performance of the proposed technique in person identification was shown in the previous experiment. Therefore, this second experiment aims at evaluating the performance of the technique in detection of unknown people.

To achieve this goal, the data set used in the previous experiment is selected. In order to evaluate the performance of the technique, a 25-fold crossvalidation was conducted. The seven face images from one male and other seven face images from one female were left out in each iteration. These two people are considered as not belonging to the data set and therefore unknowns. The images of the remaining 48 people were used to train the algorithm.

The average False Acceptance Rate (FAR) and average False Rejection Rate (FRR) graph, can be observed in Fig. 6. The corresponding average Receiver Operating Characteristic curve (ROC) is displayed in Fig. 7.

Both graphs show that the known and unknown populations have a good separability. The best separation happens at the Equal Error Rate (3.1 standard deviations), giving a FAR and FRR of 2%. Moreover, notice that, if the algorithm belongs to a security scheme, the degree of accessibility can be established by increasing or diminishing the standard deviation threshold. E.g., if in the test a false rejection rate of 5.5% is allowed, then a 0% false acceptance rate is obtained. This accommodates biometrical security systems that requires a high level of control access.

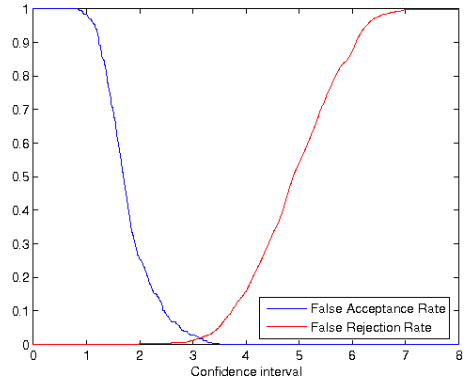


Fig. 6. Average False Acceptance Rate/False Rejections Rate graph obtained by the 25-fold crossvalidation.

[1]Number of misclassified images reported in parentheses.

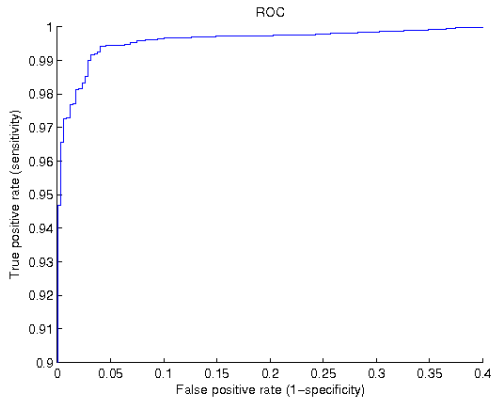


Fig. 7. Average Receiver Operating Characteristic (ROC) curve obtained by the 25-fold crossvalidation. Notice that only the top left part of the ROC curve is displayed here.

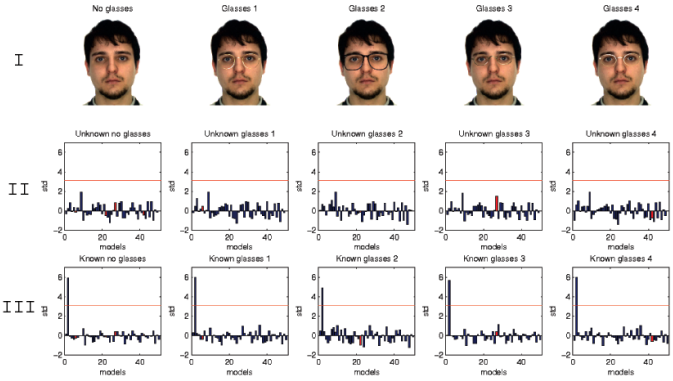


Fig. 8. Impact of changing glasses. (I) Person without glasses and syntectic fitted with 4 glasses form the data set. (II) The corresponding projections in the models as unknown. (III) The corresponding projections in the models as known. Red columns is the model corresponding to the superimposed glasses.

A second test is conducted in order to assess the robustness of the proposed method. This test also aims at showing that the method not only discriminates on removable features, such as glasses. To achieve this goal, eight people (four male and four female) are synthetically fitted with four different glasses taken from people belonging to the data set, giving 32 synthetic images.

This second test consists of two steps. First, these eight people are not used to built the individual models. The goal is to examine if these eight people who do not belong to the data set are considered as one of the person in the data set. Results show that none of the 32 images is misclassified when a threshold of 3.1 standard deviations is considered (probability of correct classification of 99.9%). This fact can be noticed in Fig. 8 II, where the projections of one of the eight unknown people on the different models are displayed. It is observed that, when the person is considered unknown, his projections onto the individual models belonging to the data set are under the selected threshold. This means that the proposed method does not classify any of the unknown people as belonging to the data set.

In the second step, the eight people (without glasses) are also used to build the individuals models. In this case the goal is to analyze if the method can still recognize people belonging to the data set who has slight changes (same people with glasses). In this second step, the 32 images are also classified correctly by the method. In Fig. 8 III, it is observed that the projections onto the individual model associated with this person clearly surpass the threshold. It is also observed that the projections into the individual models associated to the glasses's owners do not increase significantly. Similar graphs are obtained for the other seven people. These results show the suitability of the proposed technique in being incorporated into a biometrical security system.

IV. Discussion and Conclusion

In this paper, a novel method to identify people from face images has been proposed. The developed technique aims at being a precise and robust algorithm that can be incorporated into biometrical security systems. The technique has been tested on face images, but it can also be used in other biometrical data, such as speech. Experimental results have proved that the method can attain better classifications rates than an other widely used technique. Moreover, the final one-dimensional projection allows for a simple interpretation of the results. If a given face image is projected onto the different individual models, it is visually possible to determine if this person belongs to one of the models. Moreover, it is also statistically possible to observe the degree of belonging to that model.

Another of the attracting characteristics of the proposed method is its ability to deal with unknowns. The degree of belonging to the data set can be determined statistically. A decision threshold can be determined in relation to a standard Gaussian distribution. This threshold value is used to set the degree of security of the system. The higher this value is set, the smaller the probability of a person being considered as belonging to the data set.

The robustness of the algorithm has been tested using both known and unknown people. The algorithm has been shown to be robust to the inclusion of artifacts such as glasses. On one hand, unknown people using glasses belonging to people from the data set are still classified as unknown. This fact implies that unknown people would not get access to a security system when they use simple removable features belonging to people from the data set. On the other hand, known people using glasses, belonging to other people from the data set, are still recognized as themselves. This means if someone gets glasses, the associated model does not need to be recalculated. Moreover, this fact suggests that the database should be composed of facial images without glasses. This was also shown by observing that the individual model projections do not change significantly when the glasses were placed.

Another interesting property of the proposed method is its easiness to be maintained and updated. If a large data set is available, it is not needed to recalculate all the existing individual models when a new person has to be registered. Simply, a new individual model for the new person is created.

Similarly, if a person has to be removed from the database, it is only needed to remove the corresponding individual model. In conclusion, an accurate, robust and easily adaptable technique to be used for facial recognition has been developed and demonstrated.

REFERENCES

[1] *Statistical Shape Analysis*. Wiley series in probability and statistics, 1998.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[3] Hakan Cevikalp, Marian Neamtu, Mitch Wilkes, and Atalay Barkana. Discriminative common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):4–13, 2005.

[4] Songcan Chen and Daohong Li. Modified linear discriminant analysis. *Pattern Recognition*, (38):441 443, 2005.

[5] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering. University of Manchester, March 2004.

[6] I. J. Cox, J. Ghosn, and P. N. Yianilos. Feature-based face recognition using mixture-distance. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 209–216, June 1996.

[7] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[8] J. Daugman. How iris recognition works. *Proceedings of 2002 International Conf. on Image Processing*, 1, 2002.

[9] G. Finlayson, S. Hordley, G. Schaefer, and G. Y. Tian. Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38, 2005.

[10] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[11] D.G. Kendall. The diffusion of shape. *Advances in Applied Probabilities*, (9):428–430, 1977.

[12] J. T. Kent. The complex bingham distribution and shape analysis. *Proceedings in current Issues in Statistical Shape Analysis*, pages 167–175.

[13] S. Z. Li and A. K. Jain. *Handbook of face recognition*. Springer, 2005.

[14] A.M. Martinez and R. Benavente. The ar face database. Technical Report 24, Computer Vision Center Purdue University, June 1998.

[15] M. Turk and A.P. Pentland. Face recognition using eigenfaces. *IEEE Conf. Computer Vision and Pattern Recognition*, 1991.

[16] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 205–211, 2002.

[17] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of computer Vision*, 1.

Image Uncertainty and Pose Estimation in 3D Euclidian Space

Brian Wettegren, Lars Bjerre Christensen, Bodo Rosenhahn,
Oliver Granert and Norbert Krüger

Abstract

We describe a problem of a sucessful 3D–2D pose estimation algorithm when it is applied in scenarios with large depth variation. In this case image uncertainty is inhomomogenuously reflected in the Euclidian space where the constraint equations are formulated. We introduce a scaling of the constraint equations that equalizes this inhomogenity. We can show that we can reduce the error significantly in outdoor scenarios with large depth discontinuities.

1 Introduction

The estimation of the motion of rigid bodies (rigid body motion, RBM) is an important sub–problem in computer vision for tasks such as object recognition [3], mutiple view reconstuction [7] and disambiguation of visual representations [11]. It is also important in the context of robot navigation since the ego–motion of a person or vehicle in a static scene can be described by an RBM. The mathematical formalization of this kind of motion has been studied for a long while (see, e.g., [2, 9]). An RBM can be described as a six–dimensional manifold consisting of a translation (parametrised by the three coefficients $\mathbf{t} = (t_1, t_2, t_3)$) and a rotation (parametrised by $\mathbf{r} = (r_1, r_2, r_3)$). It describes the transformation of a 3D entity[1] $\mathbf{e}$ in the first frame to a 3D entity $\mathbf{e}'$ in the second frame

$$RBM^{(\mathbf{t},\mathbf{r})}(\mathbf{e}) = \mathbf{e}^{'}. \tag{1}$$

A camera projects a scene to a 2D chip. Therefore it is often convenient to work with entities that are extracted from a 2D image. However, there occur many applications in which prior object knowledge does exist. For example in industrial robot applications CAD descriptions of objects may be available (see, e.g., [4]). 3D information can also be extracted from image sequences beforehand through stereo as done in this paper. This requires then an RBM estimation algorithm that can work on entities of different dimensions: The 3D object knowledge needs to be aligned with 2D entities in an image of this

[1] In the following 3D entities are printed in boldface while 2D entities are printed normal.

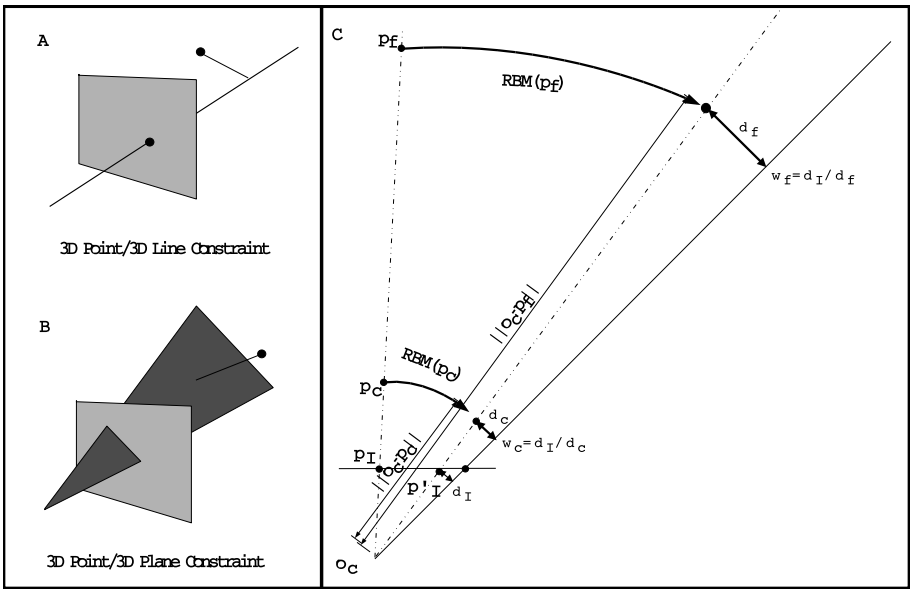


Figure 1: a) Knowing the camera gemometry a 3D line can be generated from an image point and the optical center of the camera. The 3D point/3D line constraint realizes the shortest Euclidian distance between the 3D Point and the 3D line. b) From an image line a 3D plane can be generated. The 3D point/3D plane constraint realizes the shortest Euclidian distance between the 3D Point and the 3D plane. c) Inhomogenity of Uncertainty in the Euclidian Space. The difference of a projection of a 3D point and the extracted feature corresponding to this point in the image leads to a deviation d_I. However, in case this projection comes from a far point $\mathbf{p_f}$ it results in a much larger distance d_f between the generated line and $\mathbf{p_f}$ than in case of a close point $\mathbf{p_c}$ (with associated distance d_c). An appropriate scaling factor that equalizes these differences for these two points is $w_c = d_I/d_c$ and $w_f = d_I/d_f$ respectively.

object. The problem of computing the RBM from correspondences between 3D object and 2D image entities is commonly referred to as 3D–2D pose estimation problem [6, 13]. In mathematical terms we have the following kind of constraint equations:

$$P(RBM(\mathbf{e})) = e', \tag{2}$$

where P represents the perspective projection.

There exist approaches (in the following called 'projective' approaches) that formalize constraints directly on equation (2) (see, e.g., [1]). An alternative is, instead of formalising the pose estimation problem in the image plane, to associate a 3D entity to each 2D entity: For example, a 2D image point together with the optical center of the camera spans a 3D line (see figure 1a) and an image

line together with the optical center generates a 3D plane (see figure 1b). In case of a 2D point p we denote the 3D line that is generated in this way by $\mathbf{L}(p)$. Now the RBM estimmation problem can be formulated for 3D entities

$$RBM^{(\mathbf{t},\mathbf{r})}(\mathbf{p}) \in \mathbf{L}(p).$$

where $\mathbf{p}$ is the 3D Point. Such an Euclidian formulation has been applied by, e.g., [14, 15, 5, 13]. They have coded the RBM estimation problem in a twist representation. The RBM can then be computed iteratively on a linearized approximation of the RBM.

This approach is elegant, since it deals with the full perspective projection. It works in the space where the RBM takes place (i.e., the Euclidian space) and also allows for nicely interpretable constraint equations which basically represent the Euclidian distance between the 3D entities (see figure 1,a,b). It can also deal with any kind of camera model (orthographic, perspective, paraperspective, ...): For switching between these camera models only the reconstruction of the entities change but not the actual constraint equations.

We have been sucessfully working with this algorithm which is turned out to be numerically stable and fast [10]. It is also straightforward to implement and the meaning of constraints and entities is well defined (which will become important for our improvement of the algorithm). However, one problem of such a formulation is that when dealing with natural scenes uncertainties are associated to the image features used as correspondences. These uncertainties can be for example caused by unprecise positioning or the calibration of cameras. These image uncertainties lead to an inhomogeneity in the constraint equations: The estimation of feature attributes of entities with large depth cause a higher uncertainty in the constraint equations than that of entities at a close distance. This is caused by the fact that the constraint equations are formulated on entities in the 3D–Euclidian space which however originate from 2D entities which uncertainties reproject back to the Euclidian space in a non-homogenuous way. Thus, correspondences of entities with large distance would have higher influence in the constraint equations (see figure 1c).

In this paper, we demonstrate the effect of this inhomogenity on the example of RBM estimation from stereo sequences: We can show that for scenes with large depth variation, although we get a good reduction of the error measured in the 3D constraints this can lead to quite significant errors in the 2D projections. We then introduce a scaling of the constraint equations that elliminates the inhomogenity and we can show that we achieve better results for scenes with large depth variation but not scenes with small depth variation.

The paper is structured as following: In section 2, we briefly describe the 3D–2D pose estimation algorithm. In section 4 we describe our modification of the algorithm. In section 3, we introduce the scenario in which our algorithm is applied and in section 5 we show the effect of our scaling.

2 Constraint Equations

Following [14, 15, 5, 13] an RBM can be represented as

$$RBM = e^{\tilde{\xi}\alpha} = \sum_{n=0}^{\infty} \frac{1}{n!}(\tilde{\xi}\alpha)^n \tag{3}$$

with $\tilde{\xi}$ being the 4×4 matrix

$$\tilde{\xi} = \begin{pmatrix} \tilde{w} & -\tilde{w}\mathbf{q} + \lambda\mathbf{w} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -w_3 & w_2 & w_3q_2 - w_2q_3 + \lambda w_1 \\ w_3 & 0 & -w_1 & w_1q_3 - w_3q_1 + \lambda w_2 \\ -w_2 & w_1 & 0 & w_2q_1 - w_1q_2 + \lambda w_2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

with $\tilde{w}$ being the direction of the line around which the rotation is performed, $\mathbf{q}$ being a point on this line λ being the translation along the line. A straight forward linearisation is given by $e^{\xi\alpha} \approx (I_{4\times4} + \alpha\tilde{\xi})$. We can represent a 3D point $\mathbf{p} = (p_1, p_2, p_3)$ by the null space of a set of equations

$$\mathbf{F^p}(\mathbf{x}) = \begin{pmatrix} 1 & 0 & 0 & -p_1 \\ 0 & 1 & 0 & -p_2 \\ 0 & 0 & 1 & -p_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{4}$$

Note that the value $||\mathbf{F^p}(\mathbf{x})||$ represents the Euclidian distance between $\mathbf{x}$ and $\mathbf{p}$. This will is important to derive interpretable constraint equations.

A 3D line $\mathbf{L}$ can be expressed as two 3D vectors $\mathbf{r}, \mathbf{m}$. The vector $\mathbf{r}$ describes the direction and $\mathbf{m}$ describes the moment which is the cross product of a point $\mathbf{p}$ on the line and the direction $\mathbf{m} = \mathbf{p} \times \mathbf{r}$. $\mathbf{r}$ and $\mathbf{m}$ are called Plücker coordinates. The null space of the equation $\mathbf{x} \times \mathbf{r} - \mathbf{m} = \mathbf{0}$ is the set of all points on the line. In matrix form this reads

$$\mathbf{F^L}(\mathbf{x}) = \begin{pmatrix} 0 & r_x & -r_y & -m_x \\ -r_z & 0 & r_x & -m_y \\ r_y & -r_x & 0 & -m_z \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = 0 \tag{5}$$

Note that the value $||\mathbf{F}^L(\mathbf{x})||$ can be interpreted as the Euclidian distance between the point (x_1, x_2, x_3) and the closest point on the line to (x_1, x_2, x_3) [8, 13].

We now want to formulate constraints between 2D image entities and 3D object entities. Given a 3D point $\mathbf{p}$ and a 2D point p we first generate the 3D line $\mathbf{L}(\mathbf{r}, \mathbf{m})$ that is generated by the optical center and the image point (see figure 1b).[2] Now the constraint reads:

$$\mathbf{F^{L}}^{(p)} \left((I_{4\times4} + \alpha\tilde{\xi})\mathbf{p} \right) = 0. \tag{6}$$

[2]Note that the line $\mathbf{L}$ depends on the camera parameters.

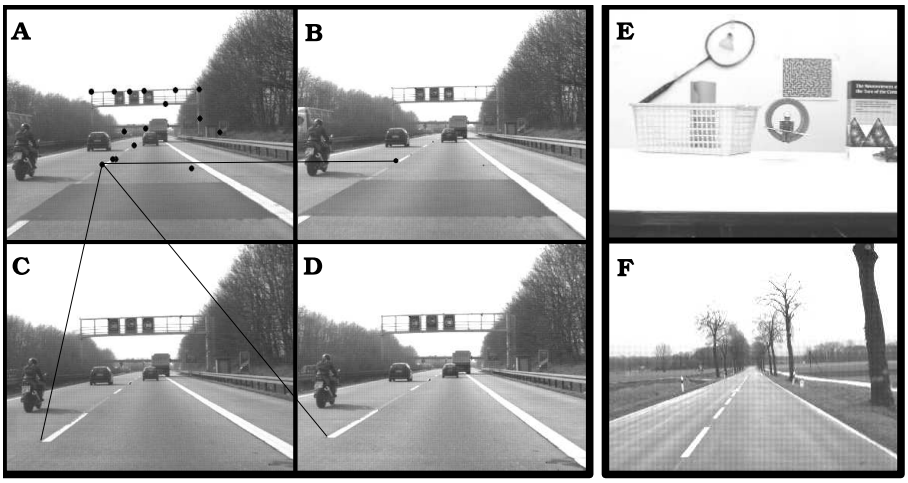


Figure 2: A,B,C,D: The scenario for RBM estimation. A,B: Left and right image of the first frame. Some of the used correspondences are displayed in A. C,D: Left and right image of the second frame. E,F: Two other scenes used for testing the pose estimation algorithm. E: Lab Scene without large differences in depth. F: Another outdoor scene.

Note that although we have 3 equations for one correspondence the matrix is of rank 2 resulting in 2 constraints. For different correspondences we get more equations. This results in a system of linear equations which solution becomes optimized iteratively (for details see [5, 12]).

3 Ego-motion estimation from Stereo Sequences

We apply the pose estimation algorithm in the context of egomotion estimation from stereo sequences (see figure 2A–D). Here we do not have any model knowledge about the scene. Therefore the 3D entities need to be computed from stereo correspondences. We provide manually derived correspondences in two consecutive stereo frames for a number of 3D points. For each 3D points we therefore get four projections, two in the first and also two in the second frame (see figure 2A,B,C.D). From the correspondences in the first frame we compute a 3D point and the correspondences in the second frame result in two 3D lines for which two constraint equations (6) can be derived.

We measured the image distances between manually determined points and points projected after the computed RBM has been performed. We noticed that for the points close to the camera there occur in average large differences. We expect that this inhomogenity results from the inhomogenity in the constraint equations.

4 Scaling of Constraint Equations according to Image Uncertainty

In the context of ego-motion estimation from stereo sequences we are faced with uncertainties in the 3D model as well as in the feature extraction. Both uncertainties are caused by the unprecision in the positioning of the corresponding 2D points. First, it results in an unprecision of stereo reconstruction.[3] Second, it leads to an unprecision in the reconstruction of the 3D line from the 2D point. Since we deal with relativeley small motions compared to depth variation in the scene we can assume that both uncertainties lead to similar distributions and can be handled by the same mechanism.

We replace equation (6) by

$$\frac{1}{w_p}\mathbf{F}^{\mathbf{L}(p)}\left((I_{3\times 3}+\tilde{\xi}\alpha)\mathbf{p}\right)=0. \tag{7}$$

where w_p is computed by

$$w_p=\frac{1}{||\mathbf{o_c}-\mathbf{RBM}(\mathbf{p})||} \tag{8}$$

where $\mathbf{o_c}$ is the optical center of the camera. Note that in our stereo context the weights for the same 3D point $\mathbf{p}$ are different for correspondences of the left and right camera since their optical centers differ.

The reason for choosing this formula is a straightforward application of the theorem of intersection of parallel lines with two intersecting lines (see also figure 1c):

$$\frac{d_p}{||\mathbf{o_c}-\mathbf{RBM}(\mathbf{p})||}=\frac{d_I}{||\mathbf{o_c}-\mathbf{P}(\mathbf{RBM}(\mathbf{p}))||}.$$

Since the weight w_p is supposed to equalize the effect of d_p we need to divide by

$$d_p=d_I\cdot\frac{||\mathbf{o_c}-\mathbf{RBM}(\mathbf{p})||}{||\mathbf{o_c}-\mathbf{P}(\mathbf{RBM}(\mathbf{p}))||}$$

We can assume the image uncertainty d_I as constant and approximate $||\mathbf{o_c}-\mathbf{P}(\mathbf{RBM}(\mathbf{p}))||$ by the focal length (i.e., by a constant as well). Both constants do not influence the relative weighting of constraint equations and can therefore be neglected such when we devide by d_p we end up with equation (8).

5 Results

We applied the scaling in 3 different scenarios: motorway (figure 2A-D), lab (figure 2E) and country road (figure 2F). In the lab scenario, depth differences were rather small compared to the ego-motion while in the other the depth

[3] In addition there is also uncertainty in the calibration. However, we negelect these effects here.

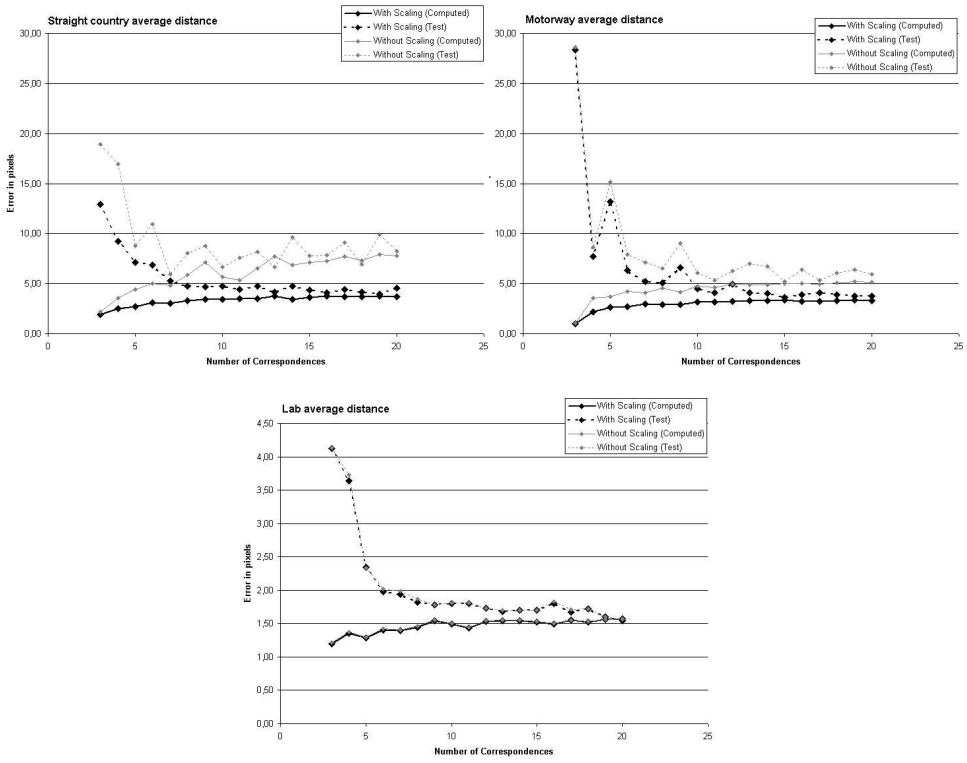


Figure 3: The average pixel distance of estimated image points depending on the number of correspondences used for computation is shown for the three scenarios: country road (top,left), motorway (top,right), and lab (bottom).

differences were rather large. From our consideration above we expect small effects for the low depth variation (lab scene) and improvement for the other two cases. For all sequences we generated 25 point correspondences manually. We computed the RBM on a subset of those (computing set). We calculated from the computing set and the set of remaining points (test set) the average pixel distance in the image plane separately.[4] The results are shown in figure 3.

Different observations are of interest. First, the average pixel error is significantly lower with our scaling compared to the non-scaling case for the motorway and the country road sequence (the error can be reduced to approximately the half). For the lab sequence there is no siginificant difference if scaling is applied or not due to the small depth variation. We can further observe that we need approximately 8 to 10 correspondences to get a good generalization. For less correspondences, we get much better results on the computing set compared to the test set.

[4]For a fixed number of correspondences we did 20 runs on different subsets.

6 Summary

We described a problem of a sucessful 3D–2D pose estimation algorithm [14, 15] when it is applied in scenarios with large depth variation. Then the image uncertainties are inhomomogenuously reflected in the Euclidian space where the constraint equations are formulated. We introduced a scaling of the constraint equations that equalizes this inhomogenity. We could show that we can reduce the error significantly in outdoor scenarios with large depth discontinuities. As expected from the motivation of the scaling method, no measurable improvement is achieved for scenes with small depth variation.

References

[1] H. Araujo, R.J. Carceroni, and C.M. Brown. A fully projective formulation to improve the accuracy of lowe's pose–estimation algorithm. *Computer Vision and Image Understanding*, 70(2):227–238, 1998.

[2] R.S. Ball. *The theory of screws*. Cambridge University Press, 1900.

[3] J. Beis and D. Lowe. Learning indexing functions for 3–d model based object recognition. *CVPR'94*, pages 275–280, 1994.

[4] C. Fagerer, D. Dickmanns, and E.D. Dickmanss. Visual grasping with long delay time of a free floating object in orbit. *Autonomous Robots*, 1(1):53–68, 1991.

[5] O. Granert. Posenschättzung kinematischer ketten. *Diploma Thesis, Universität Kiel*, 2002.

[6] W.E.L. Grimson, editor. *Object Recognition by Computer*. The MIT Press, Cambridge, MA, 1990.

[7] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[8] Selig J.M. Some remarks on the statistics of pose estimation. *Technical Report SBU-CISM-00-25, South Bank University, London*, 2000.

[9] K. Klein. *Vorlesungen über nicht–Euklidische Geometrie*. AMS Chelsea, 1927.

[10] N. Krüger, M. Ackermann, and G. Sommer. Accumulation of object representations utilizing interaction of robot action and perception. *Knowledge Based Systems*, 15:111–118, 2002.

[11] N. Krüger and F. Wörgötter. Multi-modal primitives as initiators of recurrent disambiguation processes. *Early Cognitive Vision Workshop, Isle of Skye*, 2004.

[12] N. Krüger and F. Wörgötter. Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. *Advances in Imaging and Electron Physics*, 131, 2004.

[13] B. Rosenhahn. *Pose Estimation Revisited (PhD Thesis).* Institut für Informatik und praktische Mathematik, Chrsitian–Albrechts–Universität Kiel, 2003.

[14] B. Rosenhahn, C. Perwass, and G. Sommer. Cvonline: Foundations about 2d-3d pose estimation. *In CVonline: On-Line Compendium of Computer Vision [Online]. R. Fisher (Ed). http://homepages.inf.ed.ac.uk/rbf/CVonline/"*., 2004.

[15] B. Rosenhahn and G. Sommer. Adaptive pose estimation for different corresponding entities. In L. van Gool, editor, *Pattern Recognition, 24th DAGM Symposium*, pages 265–273. Springer Verlag, 2002.

Local features for classification of structural X-ray images

Aleksandr Dubinskiy,
Informatics and Mathematical Modeling,
Technical University of Denmark,
ad@imm.dtu.dk

Abstract.

This work deals with the classification of X-ray images from Frederikssund Hospital. Previous work by Engholm & Nørgaard (2003) has attempted classification in the same domain, but the features they used were global. The goal of this work and its derivatives is to improve on this result by using features specifically suitable for the domain of X-ray images. We propose a simple region model for describing specific types of image categories, and specific feature statistics based on that model. We first experimentally evaluate this model based on manual segmentations, and then propose an algorithm for automating the segmentation process.

Introduction

This work deals with the early stages of classification of X-ray images from Frederikssund Hospital, provided by Knud-Erik Fredfeldt M.D. The problem in the medical domain is that quite often doctors and technicians make mistakes when labeling an X-ray image. This occurs for several reasons, which include, for example, fatigue, or a misclick on a computer. It is thus desirable to automate the process of X-ray labeling and reduce the workload on the doctors and technical staff. At this point, we are entrusted with the task of automatic classification among the following 10 categories of X-rays:

1. Elbow (front)
2. Elbow (side)
3. Foot joint (front)
4. Foot joint (side)
5. Hand joint (front)
6. Hand joint (side)
7. Column (front)
8. Column (side)
9. Thorax (front)
10. Thorax (side).

Figure 1 shows an example from each category. Previous work by [2] has explored the same problem in the same domain, achieving 20% error rate. While the computational techniques they employed were quite solid (decision trees / boosting), the features the computations were based on were global. Such features failed to capture the essence of the image content, i.e. bones and other solid structures, thus depriving the authors of the ability to take advantage of domain-specific knowledge. The goal of this work and its derivatives is to improve on this result by using features specifically suitable for the domain of X-ray images.

As you can see from figure 1, the images exhibit a high level of variation, but here we focus on a particular subset of X-ray images, namely that of categories with strong structural elements, such as hand / foot joints, as well as elbows. These are categories 1-6 in the original list and we will refer to them as *structural categories* from now on. We propose a simple 2-region model for the categories in question, and conduct a feasibility study by semi-automatically collecting features on the X-ray images and attempting to quantitatively assess if those features are suitable for discrimination between the structural categories. We further propose a way to automate the process of estimating the parameters of the most suitable regions, so classification can be performed without human intervention.

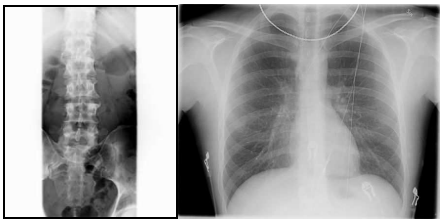
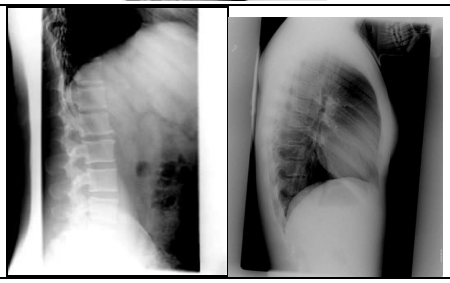
	Front	Side
Elbow		
Hand joint		
Foot joint		
Column / Thorax		

Figure 1. Examples of X-ray categories

Section 1. Classification framework

1.1 Simple regions for structural categories.

We note that the images from structural categories in the data set can be conceptually and practically subdivided into two regions, in such a way that each region has a high level of coherency in intensity and orientation. For example, the images in the category Elbow (side) can be perceived as two major parts: the bones of the upper arm, and the bones of the lower arm. We propose to similarly divide the images of the other categories into two regions (see Fig. 2)

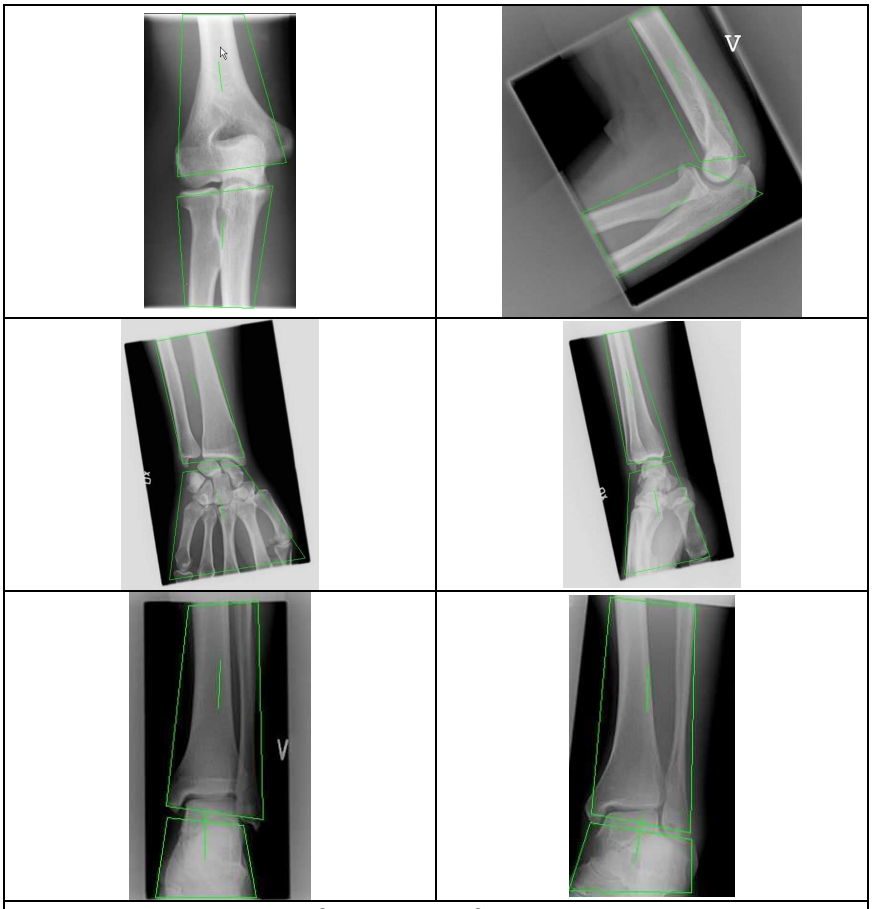


Figure 2. Examples of regions (1 for each structural category)

1.2. Region statistics

Based on these two regions, we propose a particular set of features / statistics which captures the important properties of the regions. These statistics are:

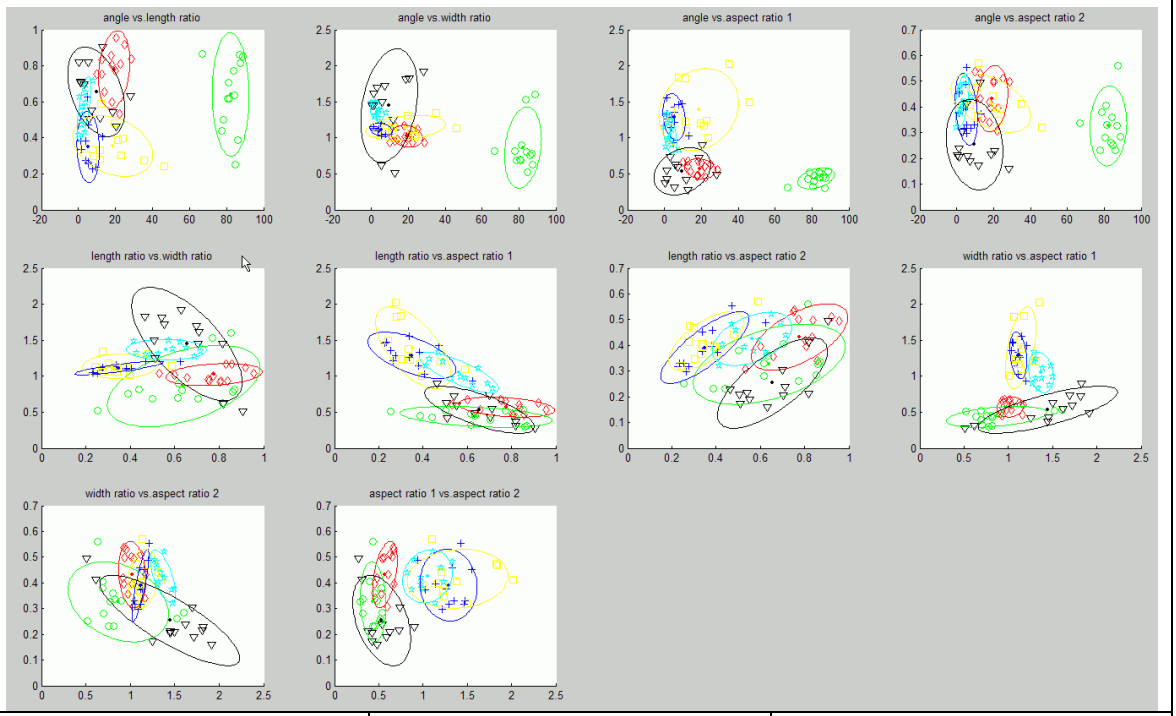
a) The angle between principal orientation of the two regions
b) The ratio of lengths of the two regions
c) The ratio of widths of the two regions
d) The aspect ratio of the longest region, and

e) the aspect ratio of the shortest region

The selected features are intuitive and invariant to orientation and scaling changes.

1.3 Data exploration and assumptions

We assume each category's feature vectors are normally distributed. Figure 3 shows all the 2-dimensional projections of the 5-dimensional feature vectors (a total of $\binom{5}{2} = 10$), as well as the projections of the Gaussian clouds for each of the image categories. It is evident, that some of the categories are clearly separable from the rest based on the features we selected. For example *elbow side* is separable based on the angle between the regions, as one would expect. Some of the other classes seem to have a consistent overlap with other classes for most of the dimensions, which makes the task of the classifier a bit harder.



◊: Elbow front ○: Elbow side	+: Foot joint front □: Foot joint side	*: Hand joint front ▼: Hand joint side

Figure 3: Separability of categories using 5 features: mutual angle, length ratio, width ratio, aspect ratio (long region), aspect ratio (short region). Pairwise scatter plots and projections of Gaussian clouds for each category on each feature pair.

1.4 Discriminant Analysis

A particular X-ray image will have a feature vector $\vec{x}$ associated with it, and is classified under class ω^* based on the maximum a posteriori principle (MAP) [3], as follows:

$$\omega^*(\vec{x}) = \arg\max_i \log p(\omega_i \mid \vec{x})$$

$$p(\omega_i \mid \vec{x}) \propto p(\omega_i)\, p(\vec{x} \mid \omega_i)$$

We base the prior for class ω_i on the number of training samples n_i in that class

$$p(\omega_i) \propto n_i$$

And the likelihood $p(\vec{x} \mid \omega_i)$ is based on the assumption of a Gaussian distribution for class ω_i with mean $\vec{\mu}_i$ and covariance Σ_i. Thus

$$p(\vec{x} \mid \omega_i) \propto \frac{1}{\sqrt{\det \Sigma_i}} \exp\left[-\frac{1}{2} (\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right]$$

and our classification decision is

$$\arg\max_i \log p(\omega_i \mid \vec{x}) = \arg\max_i \left[\log n_i - \frac{1}{2} \log \det \Sigma_i - \frac{1}{2} (\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right]$$

Section 2. Experimental results

In order to determine the suitability of our system for classification, we perform a leave-one-out test, and classify each sample using MAP, as described in the previous section. Table 1 shows that 2 categories were classified perfectly, while 2 others, only had 1 misclassification. The overall error rate was 12.33%. For comparison, Engholm & Nørgaard (2003) [2] achieved only a 20% error rate using decision trees and boosting on completely global features. Our results are not directly comparable to [2] , as we operate on fewer categories and use manual segmentation. However, the goal here is to demonstrate the potential of the framework, and the suitability of the simple structural features for discriminating between structural categories.

Category	Instances misclass-ified	Further clarification
Elbow front	1/13	1 labeled as *hand joint side*
Elbow side	0/13	Perfect classification
Foot joint front	3/12	1 labeled *foot joint side*, and 2 as *hand joint front*
Foot joint side	4/12	4 labeled as *foot joint front*
Hand joint front	1/11	1 labeled as *hand joint side*
Hand joint side	0/12	Perfect classification
Final classification error = **12.33**%		
Table 1. Misclassification of categories using the selected features		

Section 3. Automatic region segmentation

The previous section demonstrates that the two-region model and the selected features are suitable for classification of the 6 structural categories. However, as this is a real-world problem, we would like to be able to obtain such regions automatically. This can be done in a number of different ways, but we chose to attempt segmentation based on the edge map from the Canny edge detector [1]. The preprocessing consists of eliminating edges that are too short, as well as borders. Then we make initial region estimates based on the longest curves from the top and from the bottom.

3.1. Definition of desirable regions

As discussed earlier, each region has a high level of coherency in intensity and orientation, which in the domain of the edge map means, the edges in the regions must be consistent in orientation, they have to be included in the region as fully as possible, and as many edges should be included as possible (so we do not leave out any parts of the structure). Also we should include all major edges that contribute to the formation of the region. We would also like to exclude noise edges that arise from the label letters (Figure 4b).

3.2 Mathematical Formulation

Based on the above intuitions, we formulate automatic region segmentation as an energy minimization problem.

$$R^* = \arg\min_R E(R_1) + E(R_2)$$
$$E(R) = \lambda_1 E_{orient} + \lambda_2 E_{longways} + \lambda_3 E_{partial} + \lambda_4 E_{excluded}$$

Here the first term is the orientation penalty, where we penalize deviation of each edge's orientation θ_i from main region orientation $\hat{\theta}$, weighted by the edge's length L_i (longer edges should naturally have a greater impact on the region's energy than shorter ones).

$$E_{orient} = \sum_{i \in included} L_i \left(\theta_i - \hat{\theta}\right)^2$$

The second term formalizes the intuition that the outermost edges in the crosswise direction should be as close to the region's length as possible. Also the region should not include too much empty space around the edges, as we are trying to capture the main shape of the structures in the X-ray and not the space around them.

$$E_{longways} = \sum_{i \in periph} \left(L_i - \hat{L}\right)^2 + d_i^2$$

Here, $\hat{L}$ is the region's length, and d_i is the crosswise distance from the edge to the closest border of the region.
The next component of our energy function is the penalty on partial inclusions. In the desired configuration, edges will be either completely inside the regions or completely outside, the most unfavorable state being an edge half way in the region. Thus if p_i is the portion of the curve *i* included in the region, the formulation for partial inclusion energy is

$$E_{partial} = \sum_i p_i \cdot (1 - p_i)$$

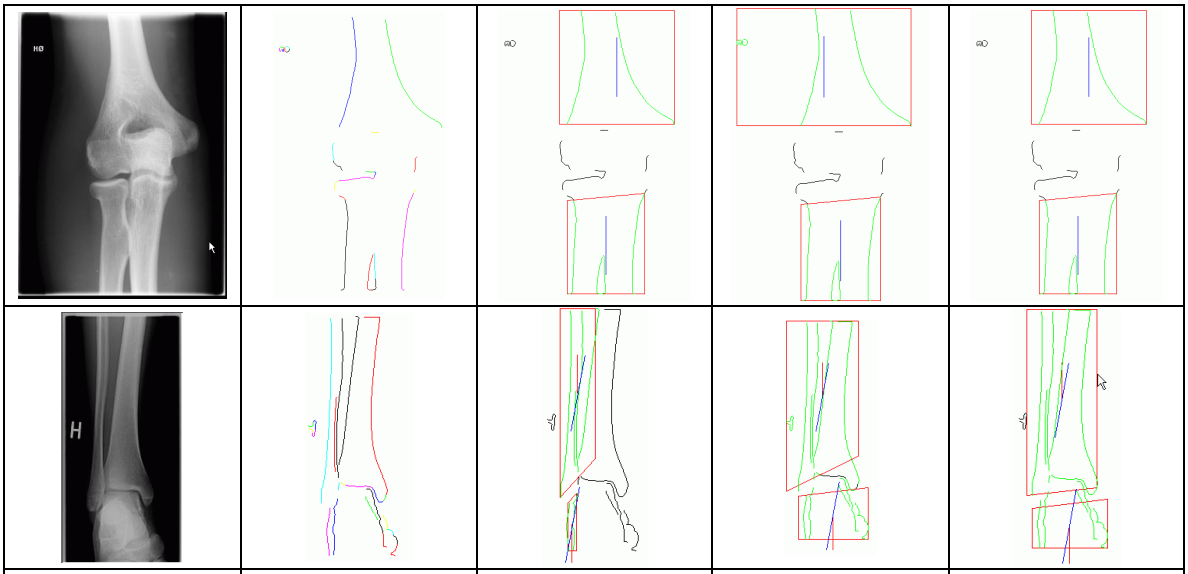
Finally, we penalize curves which are excluded in the crosswise direction from a region. The longer the projection of the excluded curve onto the principal axis of the region, the more we penalize it.

$$E_{excluded} = \sum_{i \in excluded} L_i^{\,2} \sin(\theta_i - \hat{\theta})$$

With this formulation, the penalty for excluding edges arising from labels on the X-ray will not be very large.

3.3 Computation and results

Using the energy definition $E(R)$ above, we find automatically find the two-region configuration that minimizes this energy by using Gibbs sampler [4] on the parameters of the initialized regions. Figure 4 shows the progression of the automatic algorithm. We start with the original image (4a), obtain the edge map, removing borders and short curves (4b), initialize the regions based on some simple heuristics (4c), extend the regions crosswise, including all curves in the crosswise direction (4d), and finally use the Gibbs sampler to achieve the final result for the automatic segmentation (4e). Unfortunately at the moment, the heuristics we use in step (c), and the energy formulations are not entirely robust, so we are only able to achieve automatic segmentation for a select subset of X-ray images.



(a) Original Image	(b) Edge map with some noise and borders removed	(c) Initialization	(d) Expansion	(e) Automatic result (label noise excluded)

Figure 4. Automatic segmentation of regions.
Top: an instance of *elbow front*
Bottom: an instance of *foot joint front*

Section 4. Future Work

This work is still in its early stages, and the achieved results can be greatly improved upon in several areas.

1. Coming up with new and better features. Recognition results based on manual segmentation could be improved, if we include additional statistics to our feature vectors, for example region intensity or crosswise gradient profiles.
2. Making automated segmentation more robust. This can be done improving the energy function definitions. In the current implementation, we make some simplifying assumptions about the orientation of the image and the shape of the regions, but that needs improvement. For example we assume the image and the regions are oriented vertically and that the regions start close to borders, which is often not the case. As suggested by Engholm and Nørgaard (2003), we can use the Hough transform to eliminate the frame, and estimate the global orientation of the image.
3. Basing the regions on cues more stable than edges. Edges may have been a quick first solution, but they lose much useful information, such as the intensity and gradient. With intensity information we may be able to analyze connected components, and form regions in a more simple way than from edges, which would incidentally also be more robust.

4. Being able to discriminate non-structural categories by using other types of features, such as global or textural (e.g. the images from *column side* category contain recurring square patterns, and thorax contains recurring ribs).
5. Incorporating up to 50-100 categories, and increasing the complexity of the models accordingly.
6. Using generative modeling for proposing simple regions and more complex structures and employing Data Driven Markov Chain Monte-Carlo to explore the solution space [5].

Conclusion

In this work, we have proposed a simple 2-region model for classifying structural X-ray categories, and conducted a feasibility study by collecting features on the X-ray images from manual segmentations. From those features, we conducted an experimental evaluation of a MAP classifier based on the features, where a leave-one-out test yielded only a 12.33% error rate. This is promising, since if we could achieve this result without manual segmentation, it would be a significant improvement over the 20% error rate from previous work [2]. In addition we proposed a way to automate the process of estimating the parameters of the most suitable regions, so classification can be performed without human intervention, thus laying the foundation for automatic recognition.

References

1. J. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698, 1986.
2. R. Engholm, L. Nørgaard, *Classification of Medical Images in a Radiology Department*. Polytechnical Midterm Project, Technical University of Denmark, IMM, June 2003.
3. B.Ersbøll, K. Conradsen, *An introduction to statistics Volume 2*, DTU-tryk, Kgs. Lyngby, 2003
4. G. Winkler, *Image Analysis, Random Fields, and Markov Chain Monte Carlo Methods*, Springer, 2003
5. Z. Tu, S.C. Zhu, H.Y. Shum, *Image Segmentation by Data Driven Markov Chain Monte Carlo*. International Conference on Computer Vision, pp. 131-138, Vancouver, Canada 2001.

Exemplar Based Recognition of Visual Shapes

Søren I. Olsen

Department of Computer Science, University of Copenhagen, Denmark

Abstract. This paper presents an approach of visual shape recognition based on exemplars of attributed keypoints. Training is performed by storing exemplars of keypoints detected in labeled training images. Recognition is made by keypoint matching and voting according to the labels for the matched keypoints. The matching is insensitive to rotations, limited scalings and small deformations. The recognition is robust to noise, background clutter and partial occlusion. Recognition is possible from few training images and improve with the number of training images.

1 Introduction

Several recent successful approaches to shape recognition [3, 5, 6, 8, 9] are based on attributed image keypoints. By choosing the descriptors carefully, such approaches has shown robust to rotations, scalings, illumination changes, 3D camera viewpoint including minor deformations, and - probably most important - background clutter and partial occlusion. The present approach follows this line of research. In the recognition phase the focus is not on instance detection, but on semantic contents report. The semantic contents of a training image is supplied as a list of names (labels), e.g. 'house', 'chair' etc. The set of names form the vocabulary by which test images can be recognized. Given a new unlabeled images the system reports the label(s) with strongest support from the matched keypoints. Also an image of the recognized structures is produced.

2 Previous work

Probably the most influential early work using keypoints for recognition is the paper of Schmid and Mohr [9]. Here keypoints are detected at multiple scale levels using a Harris corner detector, and a vector of differential invariants is used as descriptor. Recognition is done using multidimensional indexing and voting and by applying a model of the shape configuration, i.e. the spatial relationship between the keypoints defining a model. In later work [10, 6] the use of different interest point operators has been evaluated, and the Harris detector has been combined with other approaches to result in a scale and affine invariant detector.

Another influential work is the papers by Lowe [3–5]. Here the keypoints are detected in scale-space as the local extremes in the convolution of the image with the difference of Gaussians. The descriptor is chosen by sampling the image

gradient orientation in a rectangular grid centered at the keypoint and aligned with the dominating local gradient orientation. By using a variant of the k-d-tree for indexing, input keypoint descriptors are matched to their most similar neighbor in the database of trained descriptors. Then sets of 3 matched keypoints defining the pose are grouped using the Hough transform [5]. Next, an accurate pose of the recognized object is fitted by an iterated least squares affine fit with outlier removal. Decision to reject or accept the model hypothesis is finally made based on a probabilistic model [4].

The present work may be seen as a refinement of [8]. In this work keypoints are detected in scale-space using a model of end-stopped cells [2] and using centers of circular contours. The former type of keypoint identify corners (2 legs), junctions (3 legs) and more complicated structures with 4 legs. The directions of the legs is well suited for indexing. In the present approach the keypoint types are unchanged, but the detection method has been improved. Compared with the keypoint types used in [5, 6] fewer keypoints are in general detected. These are larger scale features more likely to mark positions with high semantic contents.

In [8] a 2-dimensional histogram of local edge point gradient orientation located within an angular sector relative to the keypoint is used as descriptor. Comparison between an input and a database keypoint is made by a modified χ^2-test. Due to quantization problems this descriptor often does not perform well. To achieve a recognition invariant to rotations, scalings act. one method is to choose descriptors that are invariant to such transforms [9, 6]. This approach is reasonable only if the transformations can model the expected deformations well. In the present work this is not appropriate, because the chosen descriptor is not very local. To achieve rotational invariance the descriptor measurements may be made in a coordinate system aligned with the dominating local image gradient orientation [5]. For the keypoints chosen in the present system there will be either several or no such orientations. In [5] the problem of multiple dominating orientations is solved by storing as many descriptors as there are such orientations. This reduces the time for single descriptor comparisons but increases the size of the database significantly. We take a third approach, using a descriptor that is not invariant to the transforms mentioned. Instead rotational invariance and insensitivity to minor deformations is left to the matching process. This choices will lower the size of the database at the expense of a larger computational complexity during the matching.

The present work is focused on reporting the semantic contents of an image in terms of one or several labels introduced during training. This is different from detecting the existence and pose of a specific object in an image [5, 9]. Loosely speaking, the method is aimed at object shape categorization rather than image-to-image matching. Thus a requirement of positional consistency between a group of neighboring query and database keypoints is not necessary. In the present system there is no spatial relations between keypoints defining a shape. The lost discriminative power is regained by choosing a descriptor with a larger spatial support. Each query keypoint may be matched to multiple database keypoints. The classification is then made using a simple voting mechanism where

each match votes on the database keypoint label with a strength determined by the similarity between the two descriptors.

3 Keypoint detection

First the essential aspects of the visual shape such as object outline contours and texture contours are extracted by edge detection using Gaussian derivative convolution and spatial non-maximum suppression of the gradient magnitude. The edge points are then used to detect the two types of keypoints: Circular structures and junctions (including corners). These keypoints mark image positions where the semantic content of the local image shape is locally rich. The core of the detection method is described in [8]. Experiments however have shown that for junctions neither the estimated localization nor the number and directions of the legs defining the junction are sufficiently stable or accurate.

In the present work each detected junction is validated in a refinement step. For each detected junction the edge points in the local neighborhood are grouped according to their local angular positioning and gradient orientation. Then a straight line is fitted to the points in each group. If few points contribute to the fit, if the residual of the fit is bad, or if the distance from the junction point to the fitted line is large, then the group is discarded. If less than 2 groups remain the junction is discarded. Otherwise the fitted lines are intersected using least squares. If the residual of the fit is large or if the intersection point is far from the initial position of the junction this is discarded. Otherwise the intersection point becomes the new position of the junction. Thus, the remaining keypoints are well localized and defined by locally straight edges with a well defined orientation.

To achieve an optimal detection of keypoints previous methods have used a multi-resolution (scale-space) approach where keypoints initially are detected at a set of scale levels, then linked through scale-space, and accepted only if some strength measure is extreme. In [6] a single optimal scale for each keypoint is found. In [5] several (extremal) keypoints may be selected. Experiments have shown that both corners and junctions may be inconsistently detected over scale, i.e. the number of legs and their orientation may differ. In the present approach a scale-space approach is used - not to ensure optimal detection - but to enable recognition of shapes scaled in size and to prune unstable detections. First keypoints are detected in scale-space, using a sampling of $\sqrt[4]{2}$ corresponding to 3 samples per octave, and grouped in scale-space according to their type and positioning. For each group the dominating directions of the legs are found by histogram analysis. Then a number of the registered representations that are consistent with respect to the number and directions of the keypoint legs are selected by sampling. The sampling is made such that the scale distance between two selected items is at least 2 scale levels. Finally, all isolated non-selected keypoints (with no other selected keypoints in their neighborhood spatially as well as w.r.t. scale) is selected as well. This guarantees that only stable or unique representatives are chosen. The sampling is chosen as a compromise between a small amount of redundancy among the representatives and a good coverage of

different neighborhood sizes coded by the descriptors. The net result is a reduction in the number of selected keypoints (often by 50 %) and that most of the selected ones are supported by similar detections at other scale levels. Especially for junctions, most spurious leg detections are removed in this process.

4 Keypoint descriptors

In the present work the choice of descriptors to a large extend follows the approach of local gradient sampling proposed by Lowe [3, 5]. However, both the sampling points, the quantization, and the descriptor matching is different. The exemplar approach, relying solely on the statistics of isolated image patch matches, requires that the keypoint descriptor is chosen to code the shape in a sufficiently large image patch in order to possess sufficient discriminative power. However it also must not be too selective. A reasonable trade off between discriminative power and generalization is experimentally found to correspond to a support radius between 10 and 20 pixels. In [5] a rectangular sampling grid is used. This grid is aligned with the dominating direction of gradient orientation. If several peaks in the histogram of gradient orientation are present, several keypoints are generated. This will double/triple the size of the database for corners/junctions. The advantage of the redundancy is a simpler matching of descriptors to the database. The disadvantage is that more descriptors should be matched to a larger base. We choose to represent each keypoint only once and to pay the price of having to "rotate" the descriptor vector for a each descriptor comparison. For this choice a sampling in a rectangular grid is inappropriate. We choose to sample the gradient information in 5 rings with radii about $3i, i = 1..5$, and with a sampling distance of about 3 pixels in each ring, corresponding to 6, 12, 18, 24 and 30 samples (90 in total). The segmenting of the disc is made using the k-means algorithm. Each pixel within a distance of about 18 pixels from the keypoint is classified to the nearest segment sample point and the gradient with largest magnitude within each segment is chosen (see Figure 1). We then

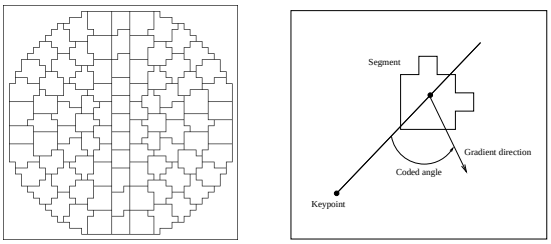
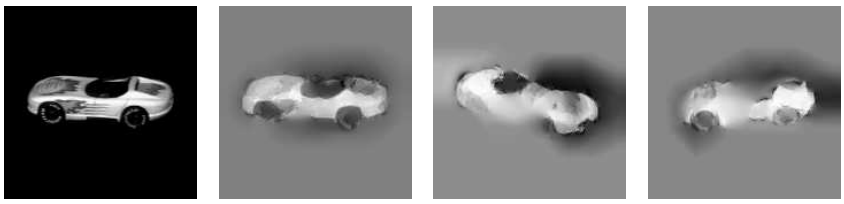


Fig. 1. Sampling regions (left), and coded angle for a segment (right).

code the gradient magnitude and the angle between the gradient orientation and segment direction. Each of the two segment component are quantized to 4 bits each. Thus the description vector has the size of 90 bytes. The coding of the

individual sample elements is invariant to rotations, but their position in the descriptor vector is not.

The motivation for inclusion of the (quantized) gradient magnitude in the descriptor is that this information allows a (more or less pleasing) reconstruction of a coded image. Also, a "mental image" of the recognized structures in a query image can be made based on the stored data, not the query data. As an example Figure 2 shows a query image, an image reconstructed from the coded data, and two reconstructions based on matched database descriptors. The database was constructed using 89 objects from the COIL-100 database [7], each object seen from 8 different viewing angles (45 degree separation in depth). The two query images differed from the nearest training image by a rotation of 10 degree (in depth).



(a) (b) (c) (d)

Fig. 2. Image (a) and reconstruction (b) based on the 63 detected keypoints in the image. Two reconstructions (c) and (d) based on 22 and 14 keypoints.

The reconstruction is made coarse to fine. The solution at a coarser level is used as initial value in the reconstruction at the next finer level. At each level the reconstruction is made in three steps. First a map of gradient magnitudes is constructed from the database descriptors at the positions defined by the matching input keypoints. Only matches to the winning label is used. Next, this map is anisotropically smoothed and the ridges detected. Ideally, these correspond to the recognized shape contours. For each point on the detected ridge the coded gradient information defines two equations in the partial derivatives of the reconstructed surface. Using standard regularization, a simple iterative updating procedure is finally applied. For simplicity a fixed number of iterations is used. Since no absolute intensity values are known the reconstruction can be made up to a additive constant only. In areas with few keypoints the reconstruction will be poor. Areas with no recognized keypoints will be reconstructed by a constant initial value of middle-gray. Thus the reconstructed image probably will be of low quality, but nevertheless show the recognized structures.

5 Shape recognition

The database of stored keypoints is organized in four groups containing circular structures, corners, junctions, and structures with more that 3 legs. Corners are indexed by the quantized angle between the two legs. Junctions are accessed using a 2-dimensional array indexed by a similar quantization of the angles between the first and the other two legs. The stored keypoint descriptions for circular structures and structures with more than 3 legs are relatively few and compared to the query keypoints through an exhaustive search. To handle large rotations and angular quantization errors several bins are checked for corners and junctions. For an n-legged keypoint $n \cdot 2^{n-1}$ bins are checked corresponding to the n possible ways one input leg can be matched to the first leg of the stored keypoint and the 2^{n-1} combinations the $n-1$ angles can be quantized to the two nearest integer values. Experiments show that the indexing step reduce the number of further comparisons by a factor of 5-20. Next each input keypoint is compared to the stored representations in the union of checked bins. In [8] a fast comparison using a measure of asymmetry was used to further limit the computational burden. Such fast tests are still relevant but are - for simplicity - omitted here.

A query keypoint is compared to a database keypoint by comparing the descriptors of gradient sample values. First the query descriptor is rotated to make the orientation of the first legs of the keypoints match. To eliminate quantization problems three rotation angles, corresponding the nearest three integral sampling numbers, is determined for each ring in the descriptor. Based on a score value the best of the three rotations is chosen independently for each sample in each ring, and a total match score is computed as a weighted sum over the 5 rings. This procedure ensures rotational invariance and that a significant amount of non-trivial deformation can be handled. The weights are chosen inversely proportional to the number of samples in each ring, i.e. inversely proportional to the sample distance. Within each ring the score is computed as a sum of gradient sample differences. Let $\mathbf{g}^q = (v^q, m^q)$ and $\mathbf{g}^{db} = (v^{db}, m^{db})$ be the quantized sample values of orientation and magnitude for a query and a database segment, and let $dm = |m^q - m^{db}|/16$ and $dv = |v^q - v^{db}|$, where the value 16 corresponds to the number of magnitude sampling intervals. Then the gradient sample difference is defined by:

$$dist(\mathbf{g}^q, \mathbf{g}^{db}) \;=\; \begin{cases} dm \cdot (dv+1) & \text{if } dv < 2 \text{ and } dm < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

Thus small gradient orientation differences are punished mildly, and larger differences equally hard. Finally, the match score is converted to a value $\in$ [0:1] with 1 corresponding to a perfect match. The match is accepted if this value is above an empirically determined threshold.

Each query keypoint may be linked to zero, one or a few database keypoints, each link attributed with a match-score. Since in the present study we want to report the semantic contents as specified by the list of training names, a simple

voting procedure is applied. Each match votes with its score as strength and has as many votes as there are names in the list associated with the matched database elements. Then the list of names are sorted according to the accumulated score value and the top-ranking name selected.

For the matches contributing to the highest ranking semantic association, a confidence C of the naming is estimated. This is based on the total support score S_0 for the most likely naming and computed by: $C = (S_0 / \sum S_i) \times (1 - exp(\frac{s_0^2}{2\sigma^2}))$. Thus C will be low if the naming is not unique or if the support score for the naming is not strong.

When training the system with a new image, a list of names is supplied. A keypoint in a new training image is installed if it cannot be matched to any previously seen keypoint. If the keypoint can be matched to a database keypoint, the list of labels for the database keypoint is extended with the new names. The description vector of the database exemplar keypoint is not updated in any way.

6 Experiments

In the experiments reported below the system was first fed with a number of images, each annotated with one label. Then the system was tested by presenting it to new unnamed images and the results of the classification was monitored. In most of the reported experiments the COIL-100-database [7] was used. This contains 100 objects each imaged from a full circle of 72 view positions. Prior to the experiments the images were preprocessed to correct for a 2-pixel column warp-around error, an incorrect constant background value and noise level, and added a supplementary border of 20 pixel width to enable detection of keypoints near the old image border. Also, the images were converted to gray-scale.

First the performance on the COIL-100-database was tested with respect to the number of objects. For subsets of the 100 objects, 8 views with an angular separation of 45 degrees were used for training and the remaining 64 views for test. First the system was first tested on all 100 objects. Then, approximately 10 objects that had the worst classification results were iteratively removed, and the system was retrained and retested on the smaller database. This procedure results in an optimal assessment curve. In all runs the threshold on the confidence C was zero, implying that all images were classified. Below in Figure 3 the results are summarized. As expected the performance decreases as the number of objects increases. For less than approximately 50 objects the misclassification rate was below 4 %.

Next, the ability of the system to perform well, when trained on few training images, were tested on subsets of the 37 object images of the COIL-100 database. The test also shows the ability to recognize objects rotated in depth. The ratio of training images to test images was varied from 36/36 to 4/68 corresponding to an angular interval between the training image from 10 to 90 degrees. As before, a confidence level of zero was used. Figure 4 below shows that a misclassification rate below 4 % is possible with as few as 6 training images per object corresponding to an angular separation of 60 degrees. Analysis showed

that the average confidence value for correct and false classifications was 0.51 and 0.35 with standard deviations 0.08 and 0.06. The two distributions are highly overlapping making a threshold-based separation difficult. This was typical for other experiments as well. Assuming normal distributions an optimal confidence threshold of 0.32 was found. Using this value will equal the number of accepted false classifications and the number of rejected true classifications. Figure 4 also shows the misclassification rate and the rate of unclassified images for $C = 0.32$.

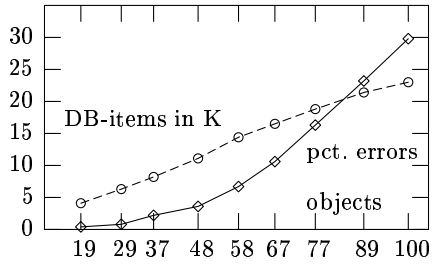


Fig. 3. Misclassification rate on subsets of the COIL-100 database, and the size of the build databases in kilo keypoints.

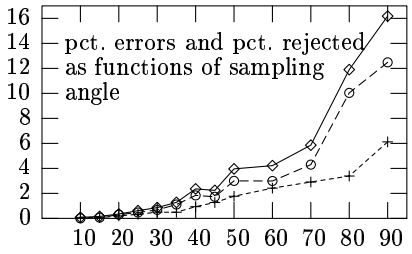


Fig. 4. Misclassification rate for different number of training images using $C = 0$ (upper) and $C = 0.32$ (middle), and the percentage of unclassified images for $C = 0.32$ (lower).

Rotation in the image plane was tested similarly. For subsets of zero-angle images of the COIL-100 database used for training, the system was tested on 71 synthetically rotated versions of each training image using a rotation step of 5 degree. Figure 5 shows that the misclassification rate was low when less that about 45 objects were to be recognized, and that the misclassification rate increased smoothly until the break-down at about 85 objects. For a small number of objects the recognition rate was independent of the rotation angle. For large databases the recognition rate was slightly better for images rotated approximately a multiplum of 90 degrees. Misclassification may happen when several object share substructures making them alike from certain view angles. In such cases, and when the queries are expected to show several objects, a list of the top-ranking classifications may be useful. For two subsets of objects of the COIL-100 database, the system was trained on 8 images with 45 degree separation, and tested on the remaining images. Then an accumulated histogram of the ranking of the correct classification was constructed. As shown in Figure 6 the correct naming was in most cases among the first few elements in the classification priority list. However, for the larger object subset still many queries seems difficult to classify correctly. One reason is that for these subsets several of the objects were very alike. Another reason is that object names trained from images with many keypoints tend to receive more votes than object names trained from images with

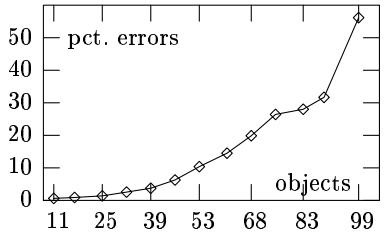


Fig. 5. Misclassification rate for images rotated in the image plane as a function of the number of training images.

ranking	37 objects	89 objects
1	97.8	76.9
1-2	99.1	83.4
1-3	99.4	86.8
1-4	99.7	88.5
1-5	99.7	89.8
1-6	99.8	90.9
1-7	99.9	91.8
1-8	99.9	92.5
1-9	99.9	93.1
1-10	99.9	93.7

Fig. 6. Accumulated histograms of the recognition rate for the top-10 ranking of the correct classification

few keypoints. Thus images of simple objects may be misclassified. This is due to the simple unnormalized voting scheme. Please note that in general neither normalization with respect to the number of keypoints in each training images nor to the number of identically labeled keypoints seems reasonable. The first choice will penalize shapes trained from images also showing background clutter. The second choice will penalize shapes with large variability. Experiments showed that in general neither type of normalization improved the performance.

Finally, the database build on 8 views of each of 37 objects from the COIL-100 collection was extended with training images of cars in natural (mostly urban) scenes. The latter images as well as the test images were selected from the car database [1]. The training images had a size of 40×100 pixels. The 144 query images were much larger and showed a significant amount of background clutter and partial occlusion. Figure 7 shows the amount of misclassification as a function of the number of car images in the training set. For less than 32 training images of cars the recognition rate was poor. This is not surprising because of the difficulty of the images and because the cars may point both left and right and may be lighter or darker than the background (corresponding to 4 different types of cars). For larger training sets the misclassification rate stays constant at a level about 3 %. This is caused by 4-5 images with heavy background clutter coincidentally giving rise to keypoints matching better to some of the distractors (keypoints from the COIL-object-views).

7 Conclusion

An exemplar based recognition scheme using attributed keypoints has been described and a few preliminary experiments has been reported. The results indicate that the system is robust to rotations, limited scalings, noise, small deformations, background clutter, and partial visibility, when the number of objects are limited (e.g. < 50). The stability w.r.t. rotations in depth has been shown to be good, and it has been shown that recognition is possible based on few training images. The results show that good performance is achievable using only local information for keypoint matching. Schmid [9] reports an experiment with 20 objects from the COIL collection, using a 20 degree separation between training as well as test images. We achieve a similar recognition rate of 99.6, but using less

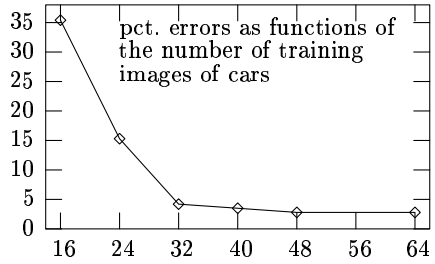


Fig. 7. Misclassification rate on images of cars in natural scenes as a function of the number of training images of cars.

than half the number of training images. Much computational effort has been put on the initial keypoint detection leaving fewer - but hopefully more stable and semantically rich - keypoints. It is left for future research to investigate whether this approach is advantageous with respect to the matching success. Automatic learning of significant keypoints as opposed to keypoints caused by background clutter and irrelevant details is of high importance for achieving a good recognition rate and to avoid the database being filled with useless data. In the present approach - having no concept of an object - this might be done by removing rarely used keypoints. The viability of this approach is left for future research.

References

1. S. Agarwal, A. Awan, D. Roth: *UIUC Image Database for Car Detection*; http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/
2. F. Heitger, L. Rosenthaler, R. Von der Heydt, E. Peterhans, O. Kubler: *Simulation of Neural Contour Mechanisms: from Simple to End-stopped Cells*, Vision Research vol. 32, no. 5, 1992, pp. 963-981
3. D. Lowe: *Object Recognition from Local Scale-Invariant Features*, Proc. 7'th ICCV, 1999, pp. 1150-1157
4. D. Lowe: *Local feature view clustering for 3D object recognition*, IEEE Conf. on Computer Vision and Pattern Recognition, Hawaii, 2001, pp. 682-688
5. D. Lowe: *Distinctive Image Features from Scale-Invariant Keypoints* Int. Jour. of Computer Vision 60(2), 2004, pp. 91-110
6. K. Mikolajczyk, C. Schmid: *Scale & Affine Invariant Interest Point Detectors* Int. Jour. of Computer Vision 60(1), 2004, pp.63-86
7. S.A. Nene, S.K. Nayar, H. Murase: *Columbia Object Image Library*, 1996; http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html
8. S.I. Olsen: *End-Stop Exemplar based Recognition*, Proceedings of the 13th Scandinavian Conference on Image Analysis, 2003, pp. 43-50.
9. C. Schmid, R. Mohr: *Local Grayvalue Invariants for Image Retrieval*, IEEE trans. PAMI, 19(5), 1997, pp.530-535
10. C. Schmid, R. Mohr, C. Bauckhage: *Evaluation of Interest Point Detectors*, Int. Jour. of Computer Vision 37(2), 2000, pp. 151-172

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Remote Sensing of Environment 91 (2004) 441–451
Remote Sensing of Environment
www.elsevier.com/locate/rse

Automatic radiometric normalization of multitemporal satellite imagery

Morton J. Canty[a,*], Allan A. Nielsen[b], Michael Schmidt[c]

[a] Systems Analysis and Technology Evaluation, Jülich Research Center, D-52425 Jülich, Germany
[b] Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
[c] Department of Geography, University of Bonn, Meckenheimer Allee 166, D-53115 Bonn, Germany

Received 24 September 2002; received in revised form 20 March 2003; accepted 13 October 2003

Abstract

The linear scale invariance of the multivariate alteration detection (MAD) transformation is used to obtain invariant pixels for automatic relative radiometric normalization of time series of multispectral data. Normalization by means of ordinary least squares regression method is compared with normalization using orthogonal regression. The procedure is applied to Landsat TM images over Nevada, Landsat ETM+ images over Morocco, and SPOT HRV images over Kenya. Results from this new automatic, combined MAD/orthogonal regression method, based on statistical analysis of test pixels not used in the actual normalization, compare favorably with results from normalization from manually obtained time-invariant features.

Keywords: MAD transformation; Orthogonal regression; Radiometric normalization

1. Introduction

Radiometric normalization of satellite imagery requires, among other things, an atmospheric correction algorithm and the associated atmospheric properties at the times of image acquisition. For most historical satellite scenes, such data are not available and even for planned acquisitions they may be difficult to obtain. A relative normalization based on the radiometric information intrinsic to the images themselves is an alternative whenever absolute surface radiances are not required, for example in change detection applications or for supervised land cover classification.

Several methods (Du et al., 2002; Furby & Campbell, 2001; Hall et al., 1991; Moran et al., 1992; Schott et al., 1988) have been proposed for the relative radiometric normalization of multispectral images taken under different conditions at different times. All proceed under the assumption that the relationship between the at-sensor radiances recorded at two different times from regions of constant reflectance is spatially homogeneous and can be approximated by linear functions. The most difficult and time-consuming aspect of all of these methods is the determination of suitable time-invariant features upon which to base the normalization.

Nielsen et al. (2002, 1998) recently proposed a change detection technique, called multivariate alteration detection (MAD), which is invariant to linear and affine scaling. Thus, if one uses MAD for change detection applications, preprocessing by linear radiometric normalization is superfluous. However, radiometric normalization of imagery is important for many other applications, such as mosaicking, tracking vegetation indices over time, supervised and unsupervised land cover classification, etc. Furthermore, if some other, non-invariant change detection procedure is preferred, it must generally be preceded by radiometric normalization.

We have applied the MAD transformation to select the no-change pixels in bitemporal images, and then used them for radiometric normalization. The procedure is simple, fast and completely automatic and compares very favorably with normalization using hand-selected, time-invariant features.

2. Selecting invariant pixels

In order to mask out the change pixels in a bitemporal scene, we first form linear combinations of the intensities for

* Corresponding author. Tel.: +49-2461-61-4885; fax: +49-2461-61-2540.
E-mail address: m.canty@fz-juelich.de (M.J. Canty).

0034-4257/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.rse.2003.10.024

all N channels in the two images, acquired at times t_1 and t_2. Representing the intensities by the random vectors $\mathbf{F}$ and $\mathbf{G}$, respectively, we have

$$U = \mathbf{a}^\top \mathbf{F} = a_1F_1 + a_2F_2 + \ldots + a_NF_N$$

$$V = \mathbf{b}^\top \mathbf{G} = b_1G_1 + b_2G_2 + \ldots + b_NG_N,$$

where $\mathbf{a}$ and $\mathbf{b}$ are constant vectors. Nielsen et al. suggest determining the transformation coefficients so that the positive correlation between U and V is minimized. This means that the difference image $U-V$ will show maximum spread in its pixel intensities. If we assume that the spread is primarily due to actual changes that have taken place in the scene over the interval $[t_2, t_1]$, then this procedure will enhance those changes as much as possible.

Specifically, we seek linear combinations such that

$$\mathrm{Var}(U-V) = \mathrm{Var}(U) + \mathrm{Var}(V) - 2\mathrm{Cov}(U,V) \rightarrow \text{maximum}, \quad (1)$$

subject to the constraints

$$\mathrm{Var}(U) = \mathrm{Var}(V) = 1 \quad (2)$$

and with $\mathrm{Cov}(U, V)>0$. Note that under these constraints

$$\mathrm{Var}(U-V) = 2(1-\rho), \quad (3)$$

where ρ is the correlation of the transformed vectors U and V,

$$\rho = \mathrm{Corr}(U,V) = \frac{\mathrm{Cov}(U,V)}{\sqrt{\mathrm{Var}(U)\mathrm{Var}(V)}}$$

The combined random vector for the bitemporal scene $(\mathbf{F}\ \mathbf{G})$ is assumed to have zero mean and variance–covariance matrix

$$\begin{pmatrix} \sum_{ff} & \sum_{fg} \\ \sum_{gf} & \sum_{gg} \end{pmatrix},$$

so that

$$\mathrm{Var}(U) = \mathbf{a}^\top \sum_{ff} \mathbf{a},\ \mathrm{Var}(V) = \mathbf{b}^\top \sum_{gg} \mathbf{b} \text{ and } \mathrm{Cov}(U,V) = \mathbf{a}^\top \sum_{fg} \mathbf{b}.$$

Extremalizing the covariance $\mathrm{Cov}(U, V)$ under the constraints (Eq. (2)) is equivalent to extremalizing the unconstrained function

$$L = \mathbf{a}^\top \sum_{fg} \mathbf{b} - \frac{\nu}{2}\left(\mathbf{a}^\top \sum_{ff} \mathbf{a} - 1\right) - \frac{\mu}{2}\left(\mathbf{b}^\top \sum_{gg} \mathbf{b} - 1\right),$$

where ν and μ are Lagrange multipliers. This leads to the coupled generalized eigenvalue problems

$$\sum_{fg}\sum_{gg}^{-1}\sum_{gf}\mathbf{a} = \rho^2\sum_{ff}\mathbf{a} \quad (4)$$

$$\sum_{gf}\sum_{ff}^{-1}\sum_{fg}\mathbf{b} = \rho^2\sum_{gg}\mathbf{b}.$$

Thus, the desired projections $U=\mathbf{a}^\top\mathbf{F}$ are given by the eigenvectors $\mathbf{a}_1 \ldots \mathbf{a}_N$ corresponding to the generalized eigenvalues

$$\rho_1^2 \geq \ldots \geq \rho_N^2$$

of $\sum_{fg}\sum_{gg}^{-1}\sum_{gf}$ with respect to $\sum_{ff}$. Similarly the desired projections $V=\mathbf{b}^\top\mathbf{G}$ are given by the eigenvectors $\mathbf{b}_1 \ldots \mathbf{b}_N$ of $\sum_{gf}\sum_{ff}^{-1}\sum_{fg}$ with respect to $\sum_{gg}$ corresponding to the *same* eigenvalues. Nielsen et al. (1998) refer to the N difference components

$$\mathrm{MAD}_i = U_i - V_i = \mathbf{a}_i^\top\mathbf{F} - \mathbf{b}_i^\top\mathbf{G}, \qquad i = 1\ldots N, \quad (5)$$

as the *multivariate alteration detection* (MAD) components of the combined bitemporal image. The covariances of the MAD components are given by

$$\mathrm{Cov}(U_i - V_i, U_j - V_j) = 2\delta_{ij}(1-\rho_j),$$

where δ_{ij} is Kronecker's delta,

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j. \end{cases}$$

The components are thus orthogonal (uncorrelated) with variances

$$\mathrm{Var}(U_i - V_i) = \sigma^2_{\mathrm{MAD}_i} = 2(1-\rho_i). \quad (6)$$

The last MAD component has maximum spread in its pixel intensities and, ideally, maximum change information. The second-to-last component has maximum spread subject to the condition that the pixel intensities are statistically uncorrelated with those in the first MAD component, and so on.

The MAD components are invariant under linear transformations of the original image intensities. We can see this as follows. Suppose the second image $\mathbf{G}$ is transformed according to some linear transformation $\mathbf{H}=\mathbf{TG}$. The relevant covariance matrices are then

$$\sum\nolimits'_{fg} = \langle \mathbf{F}\mathbf{H}^\top \rangle = \sum_{fg}\mathbf{T}^\top$$

$$\sum\nolimits'_{gf} = \langle \mathbf{H}\mathbf{F}^\top \rangle = \mathbf{T}\sum_{gf}$$

$$\sum\nolimits'_{ff} = \sum_{ff}$$

$$\sum\nolimits'_{gg} = \langle \mathbf{H}\mathbf{H}^\top \rangle = \mathbf{T}\sum_{gg}\mathbf{T}^\top.$$

The eigenvalue problems (Eq. (4)) are therefore equivalent to

$$\sum_{fg}\mathbf{T}^{\top}\left(\mathbf{T}\sum_{gg}\mathbf{T}^{\top}\right)^{-1}\mathbf{T}\sum_{gf}\mathbf{a}=\rho^{2}\sum_{ff}\mathbf{a}$$

$$\mathbf{T}\sum_{gf}\sum_{ff}^{-1}\sum_{fg}\mathbf{T}^{\top}\mathbf{c}=\rho^{2}\mathbf{T}\sum_{gg}\mathbf{T}^{\top}\mathbf{c},$$

where $\mathbf{c}$ is the desired projection for $\mathbf{H}$. These in turn are equivalent to

$$\sum_{fg}\sum_{gg}^{-1}\sum_{gf}\mathbf{a}=\rho^{2}\sum_{ff}\mathbf{a}$$

$$\sum_{gf}\sum_{ff}^{-1}\sum_{fg}(\mathbf{T}^{\top}\mathbf{c})=\rho^{2}\sum_{gg}(\mathbf{T}^{\top}\mathbf{c}),$$

which are identical to Eq. (4) with $\mathbf{b}=\mathbf{T}^{\top}\mathbf{c}$. Therefore, the MAD components in the transformed situation are

$$\begin{aligned}\mathbf{a}_i^{\top}\mathbf{F}-\mathbf{c}_i^{\top}\mathbf{H}&=\mathbf{a}_i^{\top}\mathbf{F}-\mathbf{c}_i^{\top}\mathbf{TG}=\mathbf{a}_i^{\top}\mathbf{F}-(\mathbf{T}^{\top}\mathbf{c}_i)^{\top}\mathbf{G}\\&=\mathbf{a}_i^{\top}\mathbf{F}-\mathbf{b}_i^{\top}\mathbf{G}\end{aligned}$$

as before. Given this scale invariance, we can select for radiometric normalization all pixel coordinates which satisfy

$$\sum_{i=1}^{N}\left(\frac{\mathrm{MAD}_i}{\sigma_{\mathrm{MAD}_i}}\right)^2<t,$$

where t is a decision threshold. Under the hypothesis of no-change, the above sum of squares of standardized MAD variables is approximately chi-square distributed with N degrees of freedom. We therefore choose $t=\chi^2_{N,P=0.01}$ where P is the probability of observing that value of t or lower.

The pixels thus selected should correspond to truly invariant features so long as the overall radiometric differences between the two images can be attributed to linear effects. Since this method usually identifies quite a large number of no-change pixels, we can, without serious

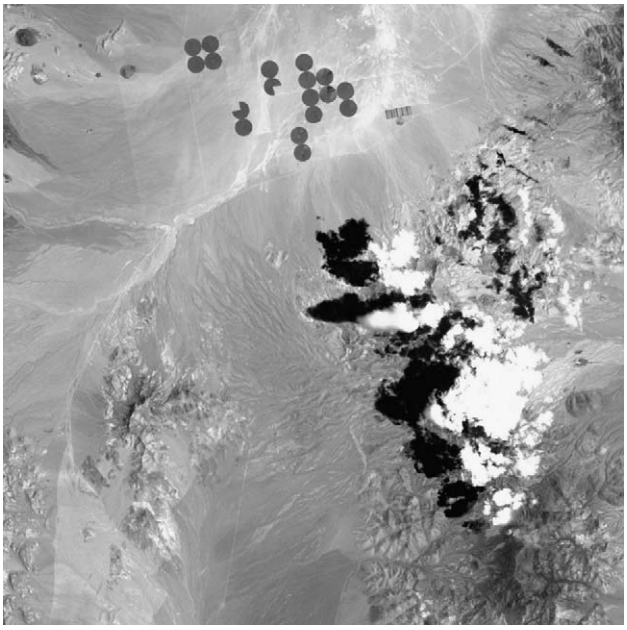


Fig. 2. Landsat-5 TM image from July, 1991 over Nevada.

Fig. 1. Landsat-7 ETM+ image from December, 1999 over Morocco.

Fig. 3. SPOT HRV image from 1987 over Kenya.

Table 1
Time-invariant features chosen for normalization to the 1999 scene

Feature	Number of pixels	Appearance
Clay	213	bright
Sand	183	bright
Fixed sand	9347	medium
Pediment1	301	medium
Quarzite	117	medium
Pediment2	365	dark
Dark stones	233	dark

Table 2
Ordinary least squares regression on manually selected training pixels for the Morocco scenes; $\hat{\alpha}$ is the fitted intercept, $\hat{\beta}$ is the fitted slope, r is the correlation and RMSE is the root mean square error

Band	$\hat{\alpha}$	$\hat{\sigma}_\alpha$	$\hat{\beta}$	$\hat{\sigma}_\beta$	r	RMSE
1	8.60	0.39	1.081	0.006	0.818	2.019
2	−3.00	0.24	1.184	0.004	0.928	1.845
3	−7.09	0.23	1.198	0.003	0.947	2.761
4	−6.37	0.18	1.258	0.003	0.961	2.020
5	4.76	0.23	1.081	0.003	0.927	2.891
7	5.31	0.24	1.077	0.003	0.910	2.870

penalty, reserve some fraction of them for subsequent testing and use the remaining pixels for performing the linear regressions.

With regard to the actual normalization on the basis of the no-change pixels, this is usually done by means of ordinary least squares (OLS) regression analysis, see, e.g. (Yang & Lo, 2000), which is a method that allows for measurement uncertainty (or error) in one variable only. For radiometric normalization, both variables involved have measurement uncertainty associated with them—in fact which variable is termed reference and which is termed unnormalized data is arbitrary. We have therefore also investigated orthogonal linear regression to perform the actual normalization, as this method treats the data symmetrically. The method is explained in detail in Appendix A.

3. Data and results

The data set used to investigate radiometric normalization consisted of Landsat TM (thematic mapper) images over Morocco and Nevada and SPOT HRV (high resolution visible) images over Kenya.

Two Landsat-7 ETM+ (extended thematic mapper) images acquired over Morocco on December 19, 1999 and October 18, 2000 (see Fig. 1) were examined for

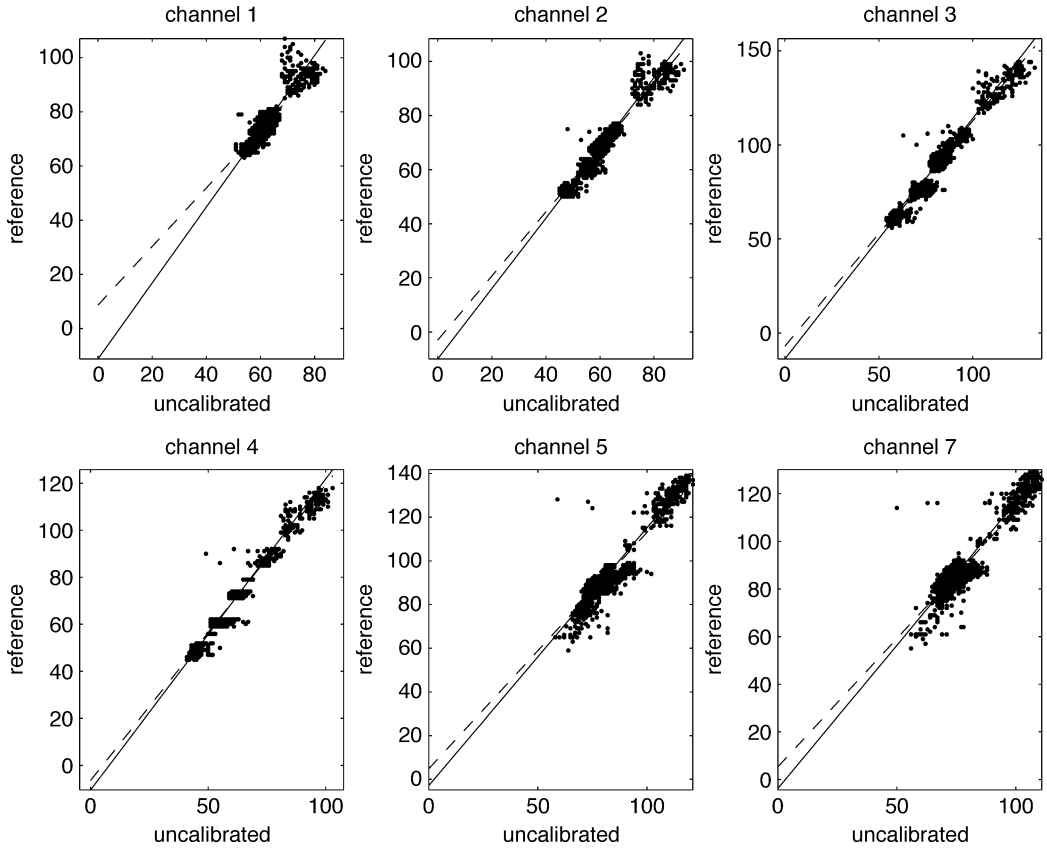


Fig. 4. Regression of the 1999 Morocco reference scene on the 2000 target (uncalibrated) scene using manually selected training pixels. Solid line: orthogonal regression; dashed line: ordinary least squares regression.

Table 3
As in Table 2, for orthogonal regression

Band	$\hat{\alpha}$	$\hat{\sigma}_\alpha$	$\hat{\beta}$	$\hat{\sigma}_\beta$	r	RMSE
1	− 11.22	0.72	1.400	0.011	0.818	1.273
2	− 9.94	0.37	1.300	0.006	0.928	1.157
3	− 13.79	0.41	1.280	0.005	0.947	1.734
4	− 10.41	0.28	1.322	0.004	0.961	1.237
5	− 2.95	0.44	1.180	0.005	0.927	1.916
7	− 3.80	0.47	1.202	0.006	0.910	1.894

comparison of the MAD procedure with normalization based on invariant features. The areas were selected on the basis of availability of ground reference data on features of constant reflectance. The dimensions of the scenes were 729 × 754 pixels. The Nevada data consisted of one Landsat-4 TM and five Landsat-5 TM scenes taken at approximately monthly intervals in the second half of 1991. A region of interest (1000 × 1000 pixels) was chosen having some agricultural activity (pivot irrigation) and significant cloud cover at the time used as normalization reference, see Fig. 2. The Kenya data consisted of two SPOT HRV images recorded in 1987 and 1989 over an agricultural region near Thika just north of Nairobi, Fig. 3. The size of scenes was 512 × 512 pixels. These data were chosen to illustrate radiometric normalization in a non-arid region.

The Morocco and Nevada scenes were registered to one another by applying an automatic contour matching algorithm due to Li et al. (1995) and using second-order polynomial, nearest-neighbor resampling. The RMS errors were less than 0.5 pixel. The Kenya data were geocoded to a common reference map with similar accuracy.

3.1. Morocco

As mentioned above, the Morocco scenes, for which ground reference data were available, were used to compare the MAD procedure with normalization based on manual selection of invariant features; see, e.g. Schott et al. (1988). The features were chosen from dark, bright and medium reflectance surfaces representative of the surface variability, see Table 1.

In their original paper on "pseudo-invariant features" (PIFs), Schott et al. (1988) do not use ordinary linear regression, but rather assume a direct (error-free) linear relation between digital numbers recorded from man-made features at two times. Since imagery is always subject to stochastic measurement error, we prefer to use regression methods which allow for this error. Fig. 4 shows the orthogonal regressions (solid lines) for normalization of the two Morocco images, based on 2/3 of the no-change pixels (referred to henceforth as "training pixels") determined from the invariant features. For comparison, the results of ordinary least squares regression are also given (dashed lines). Note that orthogonal regression leads to a consistently higher slope and correspondingly smaller intercept than ordinary regression. The fitted intercepts ($\hat{\alpha}$) and slopes ($\hat{\beta}$) for ordinary regression are shown in Table 2 for the 7200 training pixels, those for orthogonal regression in Table 3. Tables 4 and 5 show, respectively, the means and variances of the 1999 scene before and after normalization to the 2000 scene using the ordinary least squares regression line. They were determined with the 3600 holdout test pixels. Tables 6 and 7 show similar results after normalization using the orthogonal regression lines.

In contrast with the manually selected data, Fig. 5 displays the orthogonal and ordinary least squares regressions for normalization of the two Morocco images based on 11 260 no-change training pixels derived from the MAD procedure. Tables 8–13 give the corresponding information on regression statistics and on the comparisons of means and variances with 5630 test pixels.

Comparing Tables 4 and 6, we see that the paired t-tests for equal mean values of the individual bands after the manual normalization are better (the differences and t-values are closer to zero and the p-values are higher) for OLS regression for all bands except TM7. The p-value is the probability of finding a larger value of $|t|$. We also see that for all bands except TM1 for both OLS and orthogonal regression, none of the p-values are below 5%. This means

Table 4
Comparison of mean intensities of hold-out test pixels for the 2000 Morocco scene before and after normalization to the 1999 scene with ordinary least squares regression, with paired t-tests for equal means

TM band	1	2	3	4	5	7
Uncorrected(2000)	62.080	59.898	81.975	62.612	77.989	72.898
Normalized(2000)	75.720	67.969	91.143	72.400	89.117	83.820
Reference(1999)	75.650	67.969	91.115	72.455	89.114	83.771
Difference	− 0.069	0.000	− 0.027	0.055	− 0.003	− 0.049
t	− 2.207	− 0.001	− 0.589	1.668	− 0.069	− 1.062
p	0.027	0.998	0.555	0.095	0.944	0.287

Table 5
Comparison of variances of hold-out test pixels for the 2000 Morocco scene before and after normalization to the 1999 scene with ordinary least squares regression, with F-tests for equal variances

TM band	1	2	3	4	5	7
Uncorrected(2000)	6.96	14.48	44.93	29.60	40.692	31.70
Normalized(2000)	8.14	20.34	64.52	46.85	47.60	36.77
Reference(1999)	10.88	22.09	68.98	49.16	54.16	43.27
F	1.336	1.086	1.069	1.049	1.138	1.177
p	0.000	0.013	0.0443	0.147	0.000	0.000

Table 6
As in Table 4, for orthogonal regression

TM band	1	2	3	4	5	7
Uncorrected(2000)	62.08	59.90	81.98	62.61	77.99	72.90
Normalized(2000)	75.73	67.97	91.15	72.40	89.11	83.81
Reference(1999)	75.65	67.97	91.12	72.46	89.11	83.77
Difference	− 0.084	0.000	− 0.030	0.058	0.005	− 0.044
t	− 2.367	0.012	− 0.635	1.694	0.103	− 0.915
p	0.018	0.991	0.525	0.090	0.918	0.360

Table 7
As in Table 5, for orthogonal regression

TM band	1	2	3	4	5	7
Uncorrected(2000)	6.97	14.49	44.93	29.60	40.69	31.70
Normalized(2000)	13.67	24.51	73.63	51.78	56.70	45.80
Reference(1999)	10.88	22.09	68.98	49.16	54.16	43.27
F	0.796	0.901	0.937	0.949	0.955	0.945
p	0.000	0.002	0.050	0.118	0.167	0.0868

Table 8
Ordinary least squares regression on training MAD pixels for the Morocco scenes

Band	$\hat{\alpha}$	$\hat{\sigma}_\alpha$	$\hat{\beta}$	$\hat{\sigma}_\beta$	r	RMSE
1	−1.56	0.19	1.230	0.003	0.966	1.074
2	−4.68	0.13	1.191	0.002	0.978	1.372
3	−8.88	0.12	1.194	0.001	0.983	2.109
4	−8.31	0.10	1.265	0.002	0.987	1.546
5	−2.22	0.13	1.148	0.001	0.981	2.244
7	−1.33	0.14	1.146	0.002	0.976	1.983

that we can assume that the band-wise mean values are equal after normalization except for TM1. A T^2-test for equality of the mean vectors of all bands after normalization does not show significant equality. The T^2-value is lower (19.865 vs. 21.793) and the significance level is higher, i.e., better (0.0030 vs. 0.0014) for OLS regression.

Comparing Tables 5 and 7, we see that the band-wise variances are quite different after normalization for both OLS and orthogonal regression. The F-values are the ratios between the variances of the reference data and the normalized data. These values should be close to one. The significance levels show that we can assume equal variances for TM4 with OLS and for TM3, TM4, TM5 and TM7 with orthogonal regression since these are all higher than 5%.

Comparing Tables 9 and 12, we see that the paired t-tests for equal mean values of the individual bands after the MAD-based normalization are better (the differences and t-values are closer to zero and the p-values are higher) for OLS regression for all bands. We also see that for all bands for both OLS and orthogonal regression, none of the p-values are below 5%. This means that we can assume that the band-wise mean values are equal after normalization. Also the T^2-test for equality of the mean vectors of all bands after normalization shows significant equality. The T^2-value is lower (5.777 vs. 6.063) and significance level is higher, i.e., better (0.4493 vs. 0.4169) for orthogonal regression.

In Tables 10 and 13, the F-tests for equal variances show that we cannot reject the hypothesis of equal variances for any band with orthogonal regression whereas we must reject

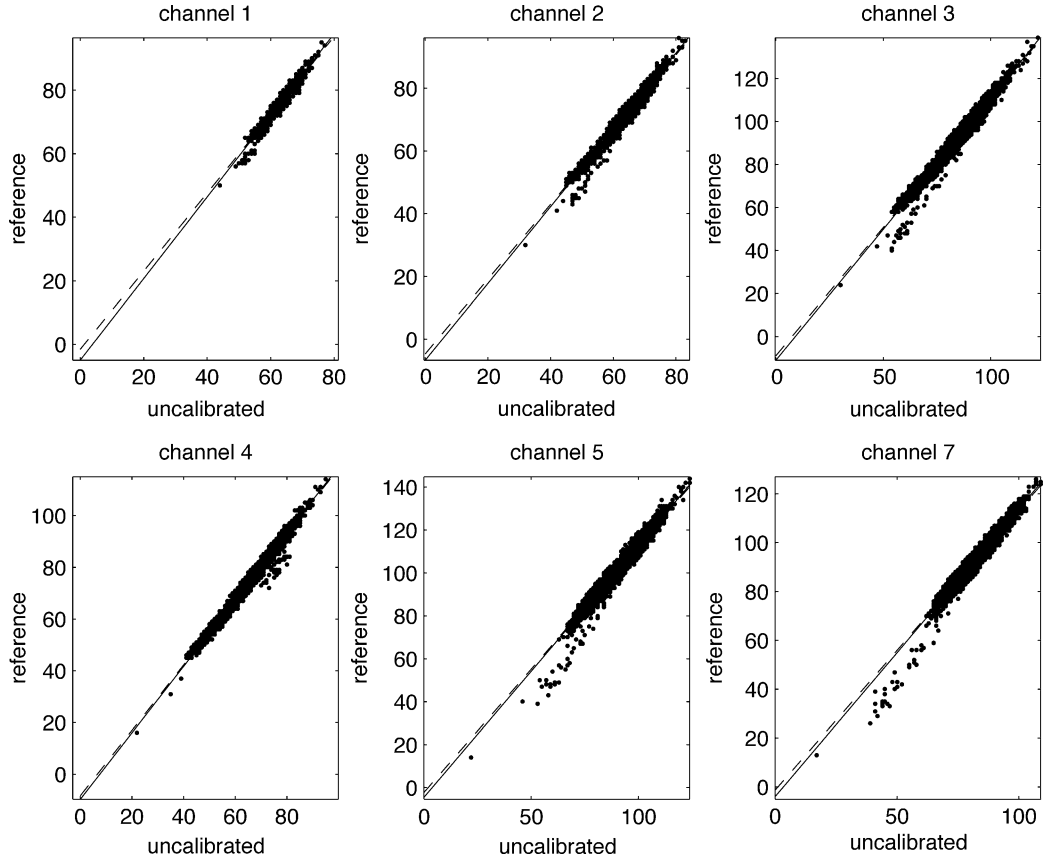


Fig. 5. Regression of the 1999 Morocco reference scene on the 2000 target (uncalibrated) scene using the MAD training pixels. Solid line: orthogonal regression; dashed line: ordinary least squares regression.

Table 9
Comparison of mean intensities of hold-out test MAD pixels for the 2000 Morocco scene before and after normalization to the 1999 scene with ordinary least squares regression, with paired *t*-tests for equal means

TM band	1	2	3	4	5	7
Uncorrected(2000)	62.734	61.544	83.894	64.573	88.128	80.094
Normalized(2000)	75.577	68.621	91.319	73.345	98.936	90.441
Reference(1999)	75.576	68.595	91.279	73.323	98.905	90.414
Difference	− 0.001	− 0.026	− 0.039	− 0.022	− 0.032	− 0.027
t	− 0.059	− 1.416	− 1.390	− 1.079	− 1.052	− 1.020
p	0.953	0.157	0.165	0.280	0.293	0.308

the hypothesis of equal variances for TM1 and TM7 for OLS regression.

Tables 2, 3, 8 and 11 show that the RMS errors are lower for MAD-based normalization and for orthogonal regression. This is true for all bands.

Finally, the plots in Figs. 4 and 5 clearly show a lot more scatter in the no-change pixels for the manual method corresponding to lower correlations as seen in Tables 2 (or 3) and 8 (or 11).

In spite of the better OLS fit for the means, all the above shows that in this case the automatic MAD-based normalization outperforms the manual normalization and that orthogonal regression is to be preferred over the OLS regression normally applied to normalization.

3.2. Nevada

Five of the Nevada images (August through December, 1991) were normalized to the July, 1991 image with the MAD procedure using orthogonal regression as described above. Fig. 6 displays the reference image (lower center) and of the December, 1991 target image before (upper left) and after normalization (upper right). The main spectral differences prior to normalization are due to Sun elevation, circular pivot plantations and clouds. Normalization to the July image as reference results in a qualitatively similar image for December. Since the clouds and irrigation pivots represent real changes, they have no influence on the calibration. The only other subjectively evident differences after normalization are the longer shadows in the December scene and some bidirectional reflectance effects in the mountainous areas.

For radiometric normalization over arid areas, both atmospheric differences and actual changes in surface reflectance are likely to be small. Fig. 7 displays the overall mean pixel intensities in the six Landsat TM images before and after normalization to the July image. The intensities have been averaged over all six non-thermal bands. The means were calculated using the 33% holdout test pixels. Also shown in the figure are the unnormalized mean intensities multiplied by the factor

$$\frac{d_i^2}{\cos\theta_i}\cdot\frac{\cos\theta_1}{d_1^2}, \qquad i = 1\ldots 6,$$

where θ_i is the Sun zenith angle and d_i is the Earth–Sun distance for each of the six acquisition dates. Since the sensor gains and offsets were constant over the acquisition period, this is equivalent to a normalization *without* atmospheric correction. Therefore, the variations may be attributed to differences in atmospheric absorption and scattering.

Table 10
Comparison of variances of hold-out test MAD pixels for the 2000 Morocco scene before and after normalization to the 1999 scene with ordinary least squares regression, with *F*-tests for equal variances

TM band	1	2	3	4	5	7
Uncorrected(2000)	10.58	28.71	86.99	54.45	95.67	59.79
Normalized(2000)	15.99	40.72	124.11	87.06	126.05	78.50
Reference(1999)	16.92	42.43	128.44	89.26	131.27	82.86
F	1.058	1.042	1.035	1.025	1.041	1.056
p	0.035	0.121	0.197	0.348	0.126	0.042

Table 11
As in Table 8, for orthogonal regression

Band	$\hat{\alpha}$	$\hat{\sigma}_\alpha$	$\hat{\beta}$	$\hat{\sigma}_\beta$	*r*	RMSE
1	− 4.96	0.20	1.284	0.003	0.966	0.670
2	− 6.66	0.15	1.223	0.002	0.978	0.875
3	− 10.98	0.18	1.219	0.002	0.983	1.346
4	− 9.65	0.13	1.285	0.002	0.987	0.954
5	− 4.53	0.20	1.174	0.002	0.981	1.465
7	− 3.95	0.20	1.179	0.002	0.976	1.293

3.3. Kenya

The Kenya data are from an agricultural region near Thika just north of Nairobi and were used to test the MAD normalization based on both OLS and orthogonal regression on data from a non-arid region. The images cover the town of Thika, large pineapple fields to the north and small coffee fields to the northwest of Thika.

Results for the test pixels (not shown) are similar to those of the data from arid regions: although we see more scatter and therefore less correlation (especially for band 3) than in the cases with arid data, both OLS and orthogonal regression give normalized data with the same mean as the reference data, OLS gives better significance. OLS regression gives significantly different variances whereas orthogonal regression gives equal variances. Also the RMSEs are smaller for orthogonal regression.

Fig. 8 shows the cumulative distribution functions for the three bands before and after MAD-based normalization with

Table 12
As in Table 9, for orthogonal regression

TM band	1	2	3	4	5	7
Uncorrected(2000)	62.734	61.544	83.894	64.573	88.128	80.094
Normalized(2000)	75.580	68.625	91.324	73.349	98.943	90.447
Reference(1999)	75.576	68.595	91.279	73.323	98.905	90.414
Difference	− 0.004	− 0.030	− 0.044	− 0.026	− 0.039	− 0.033
t	− 0.310	− 1.625	− 1.554	− 1.248	− 1.279	− 1.236
p	0.757	0.104	0.120	0.212	0.201	0.217

Table 13
As in Table 10, for orthogonal regression

TM band	1	2	3	4	5	7
Uncorrected(2000)	10.58	28.71	86.99	54.45	95.67	59.79
Normalized(2000)	17.44	42.96	129.37	89.96	131.89	83.06
Reference(1999)	16.92	42.43	128.44	89.26	131.27	82.86
F	0.970	0.987	0.993	0.992	0.995	0.997
p	0.254	0.644	0.784	0.766	0.858	0.927

orthogonal regression: a visually pleasing fit has been obtained.

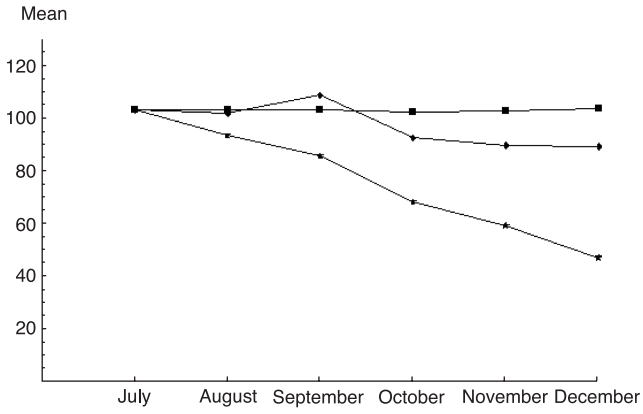


Fig. 7. Unnormalized (stars) and normalized (boxes) mean pixel intensities (in digital number units) for six Landsat TM images over Nevada from July to December, 1991. The July image was taken as reference. The diamonds are the unnormalized mean values corrected for Sun elevation and Earth–Sun distance (see text).

4. Conclusions

The procedure for radiometric normalization suggested here is automatic, very fast and requires, apart from the chi-

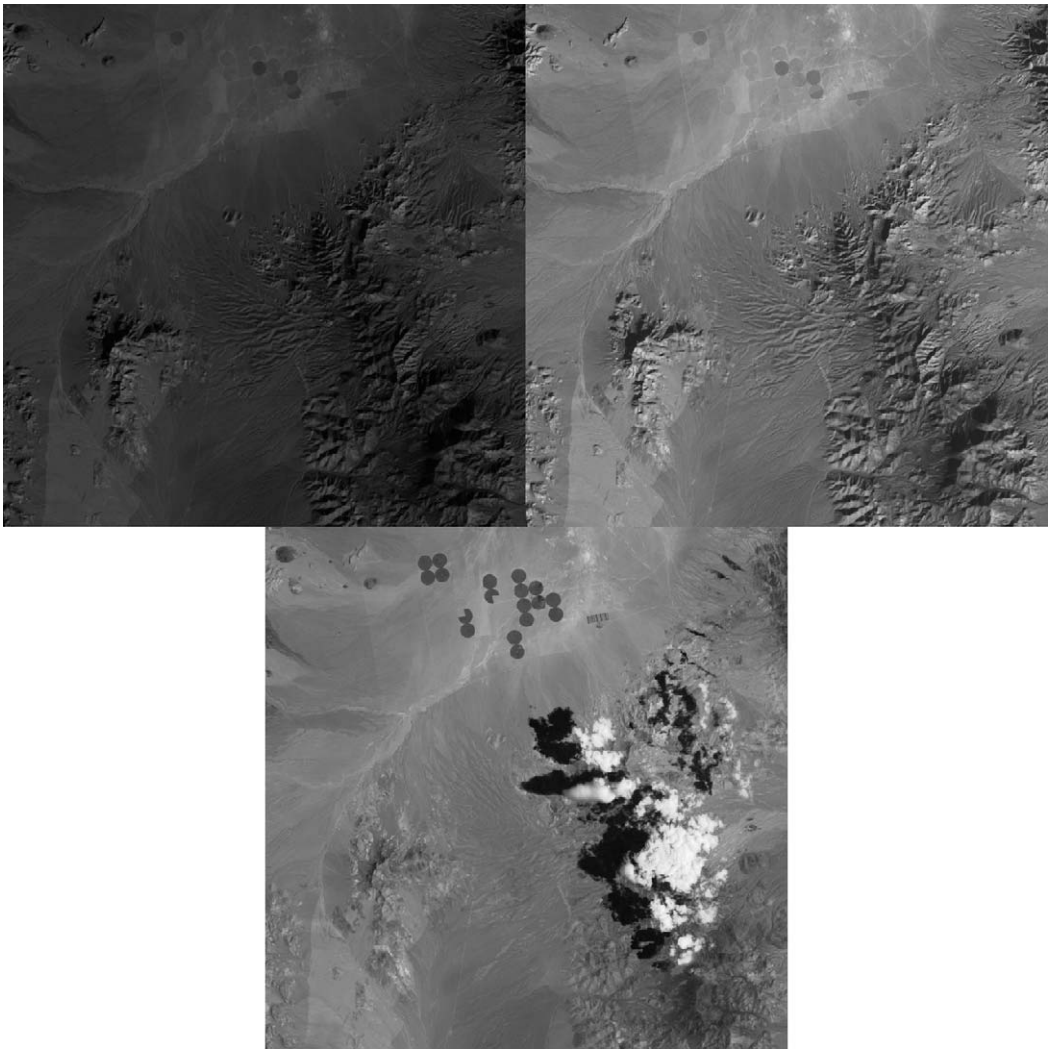


Fig. 6. Radiometric normalization of the Nevada scene. Top left: the uncorrected December, 1991 image; top right: the December scene after normalization; bottom middle: the July, 1991 reference scene.

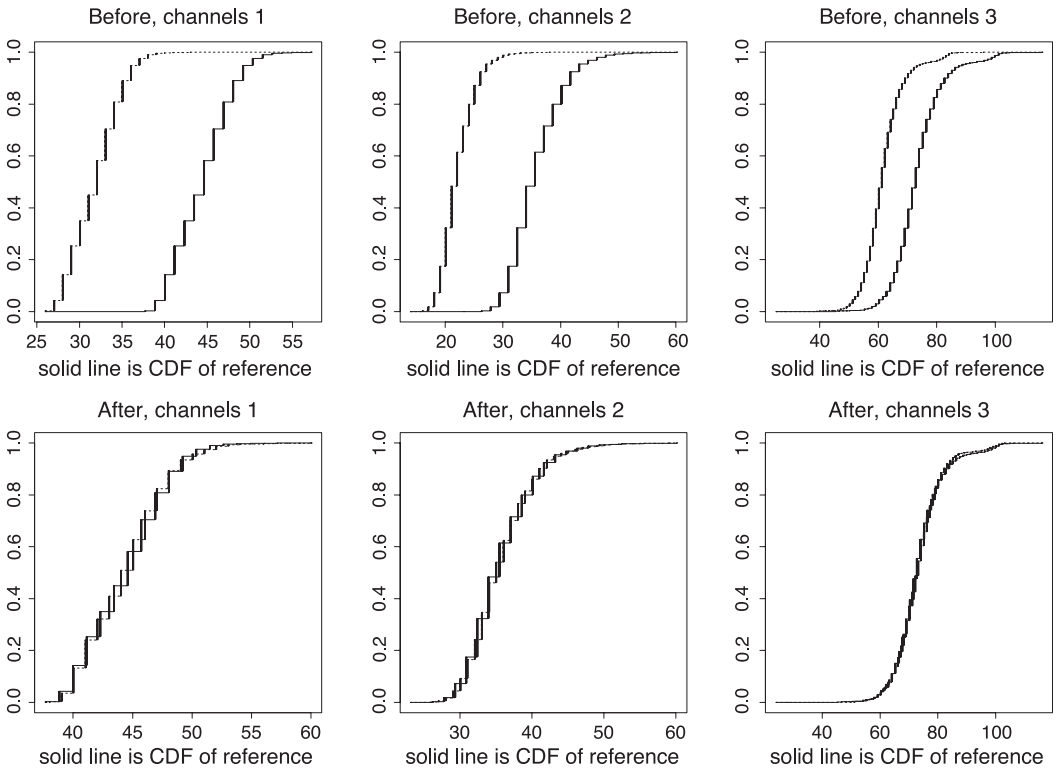


Fig. 8. Cumulative distribution functions for SPOT HRV bands before and after MAD-based normalization with orthogonal regression.

square percentile, no externally adjustable parameters such as decision thresholds or subjective criteria for defining PIF masks; everything else is entirely determined by the image data themselves. The method yields results which compare favorably to those obtained by the more laborious manual choice of time-invariant features in the images involved. On the whole, orthogonal regression using the no-change pixels is to be preferred to ordinary least squares regression. As the no-change pixels are actually selected for each image on the basis of multispectral change detection relative to the reference image, the method automatically avoids interference due to cloud cover, or indeed due to any other kind of reflectance changes that might occur.

In a recent proposal by Du et al. (2002), pseudo-invariant pixels are also selected using statistical properties rather than physical characteristics of reflecting surfaces. Their selection of suitable pixels for normalization is based on a bitemporal principal component transformation. Because of the presence of change pixels in the transformation, the principal axis must be recalculated after setting of rejection thresholds. Since the principal component transformation, unlike the MAD transformation, is not scale invariant, the method proposed here would appear to be better and more natural. Conservation of radiometric resolution after normalization, an aspect emphasized in Du et al. (2002), can of course be achieved similarly with the MAD method.

Finally, as an example of the application of relative radiometric normalization with MAD, Figs. 9 and 10 show

Fig. 9. Mosaic of two Landsat ETM+ scenes from May 2 and May 27, 2000 without radiometric normalization.

Fig. 10. As in Fig. 9, with radiometric normalization using the MAD procedure and orthogonal regression.

a part of the intersection area of a mosaic of Landsat ETM+ scenes over south Morocco on adjacent paths dating from May 2, 2000 and May 25, 2000. Fig. 9 is without, Fig. 10 with radiometric normalization. For Fig. 10, a subset of the overlap area of the images was used to calculate the regression parameters. The true changes in the surface reflectance, still apparent in the figure after normalization, are the result of rainfall prior to the acquisition of the second scene, as is the difference in the water level in the river.

Acknowledgements

AAN thanks Dr. Poul Thyregod, IMM, Technical University of Denmark, for many good discussions on normalization and calibration.

Appendix A

Some readers may not be familiar with the two types of regression analysis applied in this paper. We therefore give a very brief account of some of the more important characteristics of the two.

A.1. Ordinary least squares regression

In the model for ordinary least squares (OLS) regression

$$y_i = \alpha + \beta x_i + \gamma_i, \qquad i = 1 \ldots n \tag{7}$$

where x is considered as an independent (fixed, deterministic) predictor and y is considered as a dependent (random, stochastic) response, the x's are assumed to be uncertainty- or error-free. (This usage of the terms dependent and independent is different from the usual probabilistic meaning.) n is the number of observations and γ is a white, Gaussian noise term with mean zero and variance σ^2, white meaning that γ_i and γ_j are stochastically independent if $i \neq j$.

In this model, the estimator for β is (see any good textbook on statistics), for example (Rice, 1995)

$$\hat{\beta} = \frac{s_{xy}}{s_{xx}^2} \tag{8}$$

where

$$s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}), \tag{9}$$

$$s_{xx}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{10}$$

with $n\bar{x} = \sum_{i=1}^{n} x_i$ and $n\bar{y} = \sum_{i=1}^{n} y_i$. The estimator for α is

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}. \tag{11}$$

The variance/covariance matrix (also known as the dispersion matrix) of the vector $[\hat{\alpha}\,\hat{\beta}]^T$ is

$$\frac{\sigma^2}{n\sum x_i^2 - (\sum x_i)^2}\begin{bmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{bmatrix} \tag{12}$$

where σ^2 can be replaced by

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}\hat{\gamma}_i^2 \tag{13}$$

with $\hat{\gamma}_i = y_i - \hat{\alpha} - \hat{\beta}x_i$. The root-mean-squared error (RMSE) is $\hat{\sigma}$.

The standard errors of $\hat{\alpha}$ and $\hat{\beta}$ are the square roots of the diagonal elements of the above dispersion matrix. The test statistics for $\hat{\alpha}$ and $\hat{\beta}$ being significantly different from zero are the estimates divided by the standard errors.

A.2. Orthogonal regression

In the model for ordinary least squares regression the x's are assumed to be error-free. In the calibration case where it is arbitrary what we call the reference variable and what we

call the uncalibrated variable to be normalized, we should allow for error in both x and y. If we impose the model (we reuse the symbols $\hat{\alpha}$ and $\hat{\beta}$, later also σ)

$$y_i - \epsilon_i = \alpha + \beta(x_i - \delta_i), \qquad i = 1 \ldots n \tag{14}$$

with ϵ and δ as uncorrelated, white, Gaussian noise terms with mean zero and equal variances σ^2, we get for the estimator of β (Kendall & Stuart, 1979),

$$\hat{\beta} = \frac{(s_{yy}^2 - s_{xx}^2) + \sqrt{(s_{yy}^2 - s_{xx}^2)^2 + 4s_{xy}^2}}{2s_{xy}} \tag{15}$$

with

$$s_{yy}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{16}$$

and the remaining quantities defined in the section immediately above. The model in Eq. (14) is often referred to as a linear functional relationship in the literature.The estimator for α is

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}. \tag{17}$$

According to (Bilbo, 1989; Patefield, 1977), we get for the dispersion matrix of the vector $[\hat{\alpha}\hat{\beta}]^T$

$$\frac{\sigma^2\hat{\beta}(1+\hat{\beta}^2)}{ns_{xy}} \begin{bmatrix} \bar{x}^2(1+\hat{\tau}) + s_{xy}/\hat{\beta} & -\bar{x}(1+\hat{\tau}) \\ -\hat{x}(1+\hat{\tau}) & 1+\hat{\tau} \end{bmatrix} \tag{18}$$

with

$$\hat{\tau} = \frac{\sigma^2\hat{\beta}}{(1+\hat{\beta}^2)s_{xy}} \tag{19}$$

and where σ^2 can be replaced by

$$\hat{\sigma}^2 = \frac{n}{(n-2)(1+\hat{\beta}^2)}(s_{yy}^2 - 2\hat{\beta}s_{xy} + \hat{\beta}^2 s_{xx}^2), \tag{20}$$

It can be shown that estimators of α and β can be calculated by means of the elements in the eigenvector corresponding to the smallest eigenvalue of the dispersion matrix of the n by two data matrix with a vector of the x's in the first column and a vector of the y's in the second column (Kendall & Stuart, 1979). This can be used to perform orthogonal regression in higher dimensions, i.e., when we have, for example, more x variables than the one variable we have in our case.

Software packages to perform ordinary least squares regression (LAPACK) and orthogonal regression (ODRPACK) can be found on the Internet.

References

Bilbo, C. M. (1989). Statistisk analyse af relationer mellem alternative antistoftracere. Master's thesis, Informatics and Mathematical Modeling, Technical University of Denmark, Lyngby, 1989. In Danish.

Du, Y., Teillet, P. M., & Cihlar, J. (2002). Radiometric normalization of multitemporal high-resolution images with quality control for land cover change detection. *Remote Sensing of Environment*, *82*, 123–134.

Furby, S. L., & Campbell, N. A. (2001). Calibrating images from different dates to like-valuecounts. *Remote Sensing of Environment*, *77*, 186–196.

Hall, F. G., Strebel, D. E., Nickeson, J. E., & Goetz, S. J. (1991). Radiometric rectification: Toward a common radiometric response among multidate, mutisensor images. *Remote Sensing of Environment*, *35*, 11–27.

Kendall, M., & Stuart, A. (1979) (4th ed.). *The advanced theory of statistics*, *vol. 2*. London: Charles Griffen.

Li, H., Manjunath, B. S., & Mitra, S. K. (1995). A contour-based approach to multisensor image registration. *IEEE Transactions on Image Processing*, *4*(3), 320–334.

Moran, M. S., Jackson, R. D., Slater, P. N., & Teillet, P. M. (1992). Evaluation of simplified procedures for retrieval of land surface reflectance factors from satellite sensor output. *Remote Sensing of Environment*, *41*, 160–184.

Nielsen, A. A., Conradsen, K., & Andersen, O. B. (2002). A change oriented extension of EOF analysis applied to the 1996–1997 AVHRR sea surface temperature data. *Physics and Chemistry of the Earth*, *27*(32–34), 1379–1386.

Nielsen, A. A., Conradsen, K., & Simpson, J. J. (1998). Multivariate alteration detection (MAD) and MAF post-processing in multispectral, bitemporal image data: New approaches to change detection studies. *Remote Sensing of Environment*, *64*, 1–19.

Patefield, W. M. (1977). On the information matrix in the linear functional problem. *Journal of the Royal Statistical Society Series C*, *26*, 69–70.

Rice, J. A. (1995). *Mathematical statistics and data analysis* (2nd ed.). Belmont, California: Duxbury Press/Wadsworth.

Schott, J. R., Salvaggio, C., & Volchok, W. J. (1988). Radiometric scene normalization using pseudo-invariant features. *Remote Sensing of Environment*, *26*, 1–16.

Yang, X., & Lo, C. P. (2000). Relative radiometric normalization performance for change detection from multi-date satellite images. *Photogrammetric Engineering and Remote Sensing*, *66*, 967–980.

Scene Modeling, Nonlinear Dimensionality Reduction and Optimization

Brian Lading
Informatics and Mathematical Modelling, DTU
Mathematical Imaging Group, LTH

Abstract

We present results on optimization of a 3d shape model. We optimize the shape, the pose and the light. Initial estimate of the model parameters is obtained by interpolation of the neighbours of the sample image in the isomap embedding.
The presented model works on any set of registered 3d shapes, with applied texture and texture coordinates, the Jacobian is derived analytically in a form which is GPU friendly.
Results on artificial and real images are presented.

1 Introduction

Here we'll be looking at scene modeling, focusing on the well known object the human face.

Inspired by the work of giants, (ie in this field the creators of statistical shape models and active appearence model, Edwards, Cootes and Taylor eg. [1, 2], and the impressing and realistic faces generated by Blanz and Vetter [3, 4], and the intersting new approach to model fitting proposed by Matthews and Baker [5]) we have, from ranged 3d face data scans, constructed a 3d statistical shape model of the face. This model is intended to be used in the modeling of faces in images, ie for object segmentation and classification. The current model is build from data of one person, thereby concentrating on facial deformations belonging to the expression of the face.
The presented optimization approach is guaranteed to converge if the model is initialized close to the actual minimum. Therefor we need some initial classifier which can provide us with a good initial state. Here we do this by exploring a dimensionality reduction scheme, isomap, put forth by Tenenbaum et al [9], and map our unseen test images with the estimated mapping to get a weighted estimate of the scale, pose and light parameters. The model optimizes s statistical shape model parameters, seven similarity transformation parameters, aswell as the main light position and five parameters for the Phong [6] light image.

2 what we have done

From 28 3d face scans we have obtained a 3d statistical shape model of the face. The presented images are from a model with 414 vertices and 768 triangles[8] which should be considered close to a minimal model. For the initial classification we have used Isomap to embed 629 artificial model images. The training data images are constructed with varying rotation round y-axis, scale and translation. The images are embedded by use of the isomap (isometric feature mapping[7]) algorithm. From the embeddings and the images we construct a 'image space to embedded space' mapper, $\mathbf{F}$ mapping images $\mathbf{X}$ onto a low dimensional feature space $\mathbf{Y}$, $\mathbf{FX} = \mathbf{Y}$[10]. This mapping is then used to estimate the embedding of unseen test samples. Once the test images have been embedded we estimate the pose of the object from our knowledge of the parameters of the embedded training samples.
Once we have an initial estimate of the pose of our model we may initiate the optimization procedure, where the model parameters for pose, lights and

shape are optimized. The optimization procedure used is a standard Levenberg-Marquardt optimization procedure, where the model parameters are updated as $\delta\mathbf{p} = -\left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{r}$. Here $\mathbf{r}$ is the current error between the model image and the sample image, and $\mathbf{J}$ is the model Jacobian whose i'th-column is given by $\mathbf{J}_i = (\partial\mathbf{r}/\partial p_i)$.

Writing our image of the model as a product of an image describing the lights, $\mathbf{I}_l$, and an image of the mapped texture values, $\hat{\mathbf{I}}_m$, we have

$$\begin{aligned}\frac{\partial\mathbf{r}}{\partial p_i} &= \frac{\partial}{\partial p_i}\left(\mathbf{I}_l(\mathbf{p})\hat{\mathbf{I}}_m(\mathbf{p}) - \mathbf{I}_s\right) \\ &= \frac{\partial\mathbf{I}_l}{\partial p_i}\hat{\mathbf{I}}_m + \mathbf{I}_l\left(\frac{\partial\hat{\mathbf{I}}_m}{\partial\mathbf{s}}\right)^T\frac{\partial\mathbf{s}}{\partial p_i}\end{aligned}$$

for the i'th column of the Jacobian. $\mathbf{s}$ is the texture cordinates of the texture image. The parameter for the model includes k shape parameters for the statistical shape model, seven parameters for the similarity transformation and seven parameters for the ligth image.

3 results

The training database images are constructed at three diffenret scales, at 10 degrees rotation interval (from -90 to 90) for rotations round y-axis and at 11 different light positions. Image size is 50×70 pixels. Thus we know the data has an intrincis dimensionality of three dimensions. As seen in figure 1 Isomap correctly estimates this intrincic dimensionality.

Plots of two 2-dimensional embeddings obtained by Isomap are visualized in figure 2. In the first plot we observe three distinct arms, each with training samples at the three different scales. In the second plot the rotation is clearly depicted in loops.

The optimization algorithm works in two steps; first the pose is optimized in an hierarchical manner (working on Gaussian images), when converged

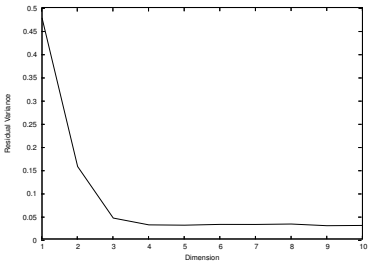


Figure 1: The residual variance of Isomap. The intrinsic dimensionality of the data is estimated to be at the 'elbow', ie Isomap estimates 3 dimensions. The number of neighbours used for the estimation of the manifold is 10.

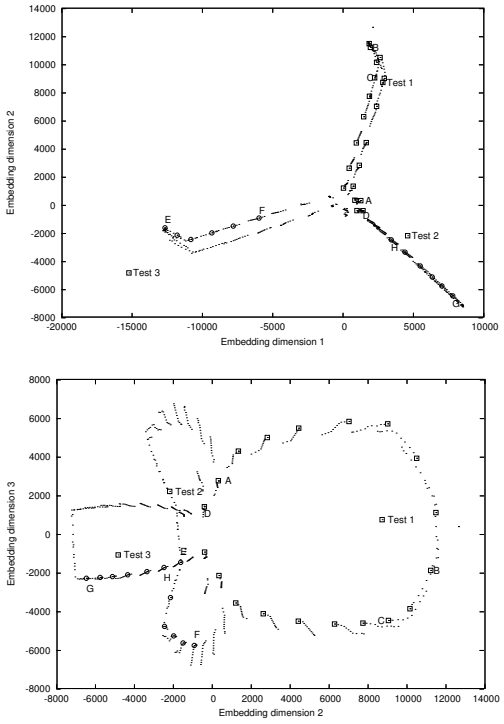


Figure 2: Isomap 2d embeddings. Markings explained in caption to figures 3 and 4

Figure 3: Three image samples. First two are images generated by the model. Last image is real world data. The images are classified in figure 2 as Test 1, 2 and 3.

Figure 4: Images of isomap training data. Top image corresponds to points on the 'upper arm' of the top plot in figure 2, top row corresponds to square marked sites A to B (going counterclockwise). Second row corresponds to marks from C to D. Top row of bottom image is marked E to F. Last row is G to H.

Figure 5: Optimized Test image 1. First image in each column is initial state, second image is final state and last image is the original sample image. Initial pose and light, for the first image sequence, is given by the weighted parameters of the four nearest neighbours in the three dimensional isomap embedding. The second image sequence is with an initial perturbed state (48 degree rotation round y-axis). Optimized state reached in under 10 iterations.

we optimize on all model parameters (also in an hierarchical manner). A typical optimization runs with 2-4 iterations for the pose and 4-10 iterations in the last 'all parameters'-step.

Three test image samples (visualized in figure 3) are classified by the Isomap mapping, and the initial estimated state is shown in the first images of the first image sequences in figures 5, 6 and 7. The first test image is classified as a frontal view, which is correct. The optimization result is good, and as the second image sequence of figure 5 shows, the optimization procedure seems to work on highly perturbed states. But as we observe in figure 6 this is not always the case. The first image sequence shows optimization results of the model with initial pose as depicted by the Isomap mapping. The system gets caught in some local minimum and the optimization fails. In the second image sequence we have altered the initial scale and pose and results are almost satisfactory.

The optimization of the last test sample is shown in figure 7. Here we have applied the model on a real world image (image of the same object as that of the statistical shape model) and with the initial pose being close to the actual pose, we obtain a good result.

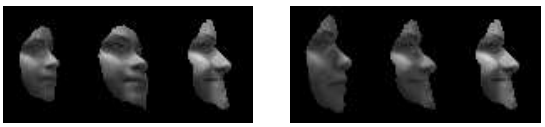


Figure 6: Optimized Test image 2. First image sequence is with initial values given by the isomap mapping. Last image sequence is with a user specified initialization.

Figure 7: Optimized Test image 3. Initial values given by the isomap mapping. At current development stage this result is satisfactory. Final state is reached in 17 iterations.

4 Conclusion

We have seen how face images may be classified for differences in pose by the use of the Isomap nonlinear dimensionality reduction scheme, and we have seen some results on the optimization of synthetic aswell as real images of the human face. Results presented indicates that the outlined scheme for initial pose classification and model optimization is applicable.

Acknowledgements

Karl Skoglund is thanked for his collaboration on data collection, as are supervisors Kalle Åström and Rasmus Larsen for advices.

References

[1] G.J. Edwards, T. F. Cootes, and C. J. Taylor. *Face recognition using active appearance models.* ECCV'98 Proc., volume 2, pages 581ŋ95. 1998.

[2] T.F.Cootes, G.J. Edwards and C.J.Taylor. *Active Appearance Models*, ECCV Vol. 2, pp. 484-498, 1998.

[3] V. Blanz and T. Vetter. *A morphable model for the synthesis of 3d faces.* SIGGRAPH'99, pages 187ŋ194, 1999.

[4] V. Blanz, S. Romdhani, and T. Vetter. *Face Identification across Different Poses and Illuminations with a 3D Morphable Model.* Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02).

[5] I. Matthews and S. Baker, *Active Appearance Models Revisited.* International Journal of Computer Vision, Volume 60, p. 135 - 164, 2004

[6] B.T. Phong, *Illumination for Computer Generated Pictures*, PhD. dissertation, Department of Computer Science, University of Utah, December 1978.

[7] Isometric, ie same metric, that is the Isomap algorithm seeks to preserve the intrinsic metric/geometry of the data.

[8] This low polygonial model is used due to rendering time. With the current octave implementation this model renders in 20 seconds. The full model, consists of 13500 vertices and 14600 polygons, renders in close to one hour. All on a standard pc.

[9] J.B. Tenenbaum, V. de Silva and J.C. Langford , *A Global Geometric Framework for Nonlinear Dimensionality Reduction.*, Science, vol 290, p 2319-2324, 22 dec 2000
Matlab software for the algorithm is available at

`http://isomap.stanford.edu/`

[10] $\mathbf{FX} = \mathbf{Y} \Rightarrow \mathbf{X}'\mathbf{F}' = (\mathbf{USV}')'\mathbf{F}' = \mathbf{Y}'$
$\Rightarrow \mathbf{F}' = (\mathbf{VS}'\mathbf{U}')^{-1}\mathbf{Y}' = \mathbf{V}(\mathbf{S}^{-1})'\mathbf{U}'\mathbf{Y}'.$

A Symmetry Set Based 2D Shape Descriptor

Arjan Kuijper & Ole Fogh Olsen
IT University of Copenhagen, Denmark
{arjan,fogh}@itu.dk

Abstract

We introduce strings, based on the Symmetry Set, to describe shapes. These strings denote links between pairs of extrema of the curvature together with a length measure. An algorithm is given to match strings of different types of shapes. Examples show the usability of the presented theory.

1. Introduction

In shape analysis, much effort has been put into the research on the skeleton, or Medial Axis [2], as a way to represent the shape in a more simplified way. As it was soon realized, the Medial Axis it itself didn't carry enough information [8] and sophisticated extensions were built, like the Shock Graph method [17]. Basically, each points on the Medial Axis is endowed with some augments related to the distance to the shape itself or related to its neighbours. Next, the potential changes of the Medial Axis were investigated, yielding a set of possible transition [9]. In that way different shapes can be related to each other for shape indexing and retrieval [15, 16].

The results on transitions boiled down from the results on the possible transitions of the Symmetry Set. This set, containing the Medial Axis as subset, has been thoroughly studied in [4]. Its transitions are described in [3]. The Symmetry Set has its advantage in being easily described in mathematical sense, but its visualization is less pleasant for the eye. So most of the research has been focused on the (augmented) Medial Axis [10].

Recently, however, a data structure was presented for the Symmetry Set [13], using information of the evolute of the shape. The data structure can be visualized by a sequence of nodes that are pair wise joined. It was claimed that its main advantage over the graph structure used for the Medial Axis is that this sequence would allow operations on it with a lower complexity.

In this paper we use the idea of representing Symmetry Sets as a sequence. In contrast to [13], we relate this sequence directly to the shape. As different shapes have different sequences $\{A_i\}_{i=1\ldots n}$ and $\{B_j\}_{i=1\ldots n}$, we propose to build a matrix M with entries $f(A_i, B_j)$. The similarity of shapes is then measured as the path $P = \{M(i_k, j_l)\}$ through M that contains each row and column at most once, and has a maximal sum of the elements $M_{i,j}$.

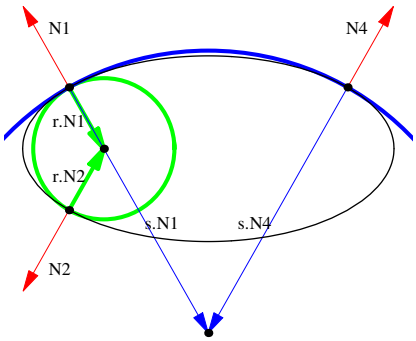


Figure 1: Definition of the Symmetry Set. See text for details.

2 Symmetry Sets

The Symmetry Set is defined as the closure of the loci of the circles tangent to a shape. See Figure 1. The shape is given by the oval. Inside a circle is tangent to it at two locations, so the *unit* normals $\mathcal{N}_1$ and $\mathcal{N}_2$ are equal for the shape and the circle. The centre of the circle is found by multiplying minus the radius r with the normals. Note that this is also a Medial Axis point Next, also outside a circle is tangent to the shape at two locations, where the unit normals $\mathcal{N}_1$ and $\mathcal{N}_4$ are equal for the shape and the circle.

From this image it follows immediately that a point on the shape relates to at least two points on the Symmetry Set, in contrast with the Medial Axis. A recipe for finding the Symmetry Set is the given by the following observations.

Let a circle be tangent to the shape as in Figure 2a. Then call the points at which it is tangent p_1 and p_2 (Figure 2b). Then the vector $p_1 - p_2$ is perpendicular to the vector $\mathcal{N}_1 + \mathcal{N}_2$ when the circle is tangent twice from the same side as shown in these images, or to the vector $\mathcal{N}_1 - \mathcal{N}_2$, when tangent from two different sides (see [9]). So to find these

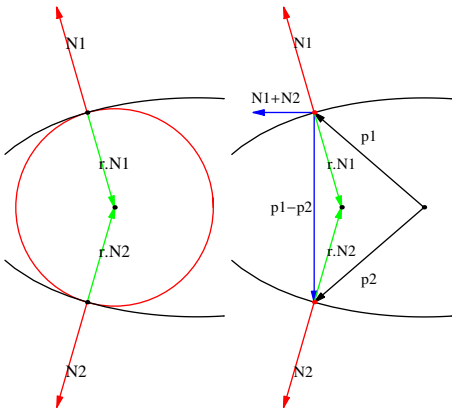


Figure 2: Deriving the Symmetry Set. See text for details.

locations it suffices to have a point p_i fixed and try all other points p_j along the shape and find zero crossings of

$$(p_i - p_j).(\mathcal{N}_i \pm \mathcal{N}_j) \tag{1}$$

Next, the centre of the circle - the location of the Symmetry Set point - is given by

$$p_i - r\mathcal{N}_i = p_j \pm r\mathcal{N}_j \tag{2}$$

2.1 Representations

A branch of the Symmetry Set is given by a connected set of centers of circles. The end points of a branch are the closures of these sets, obtained when the two points p_i and p_j coincide. For the Medial Axis, such a point is an end point of the graph. In the Symmetry Set, these points come in pairs, as the Symmetry Set consists of distinct curves.

At these points the circle has a third order of contact at the shape, or in other words, the shape has a local extremum of the curvature κ at that point. Consequently, each local extremum of the curvature can be mapped to another local extremum of the curvature.

Next, the end points are part of the evolute, which is the curve $\mathcal{S} + \mathcal{N}/\kappa$, since $r = 1/\kappa$ for these points. Following the evolute, one can label the order of appearance of the end points, yielding a sequence of end points. Connecting the end points pair wise and augmenting each connection with 'special points' that arise on the Symmetry Set, gives the string structure proposed in [13].

An example is given in Fig. 3. On the left, a fish shape is taken from a common data set [15, 16]. On the right, the string structure - without special points - is shown.

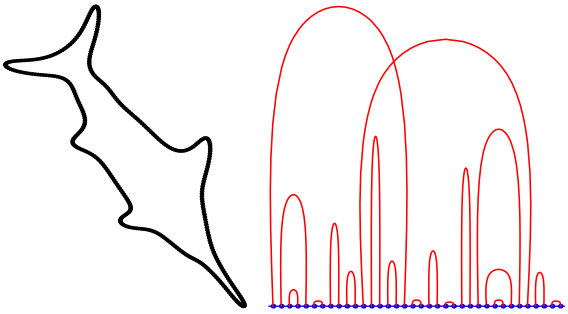


Figure 3: A fish shape and its corresponding sequential representation.

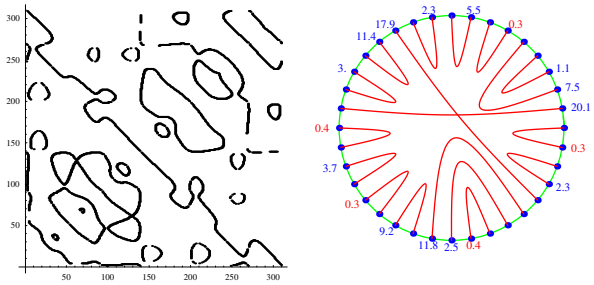


Figure 4: A fish shape and its corresponding sequential representation.

3 Closed form representation

The evolute can become complicated, especially for concave shapes. Then sometimes $\kappa = 0$ and the evolute moves to infinity. The same holds for Symmetry Set branches and the Medial Axis part outside the shape. It is therefore convenient to relate the Symmetry Set directly to the shape.

This can easily be done while computing the Symmetry Set in Eq. 2 by using the locations of the tangency of the circle, instead of its centre. This results in pairs of so-called 'pre-Symmetry Set' points, known in robotics [1]. They are shown in Figure 4 on the left.

In this diagram, branches of the Symmetry Set are visible as curves. Note that the shape is closed, so the left part of the diagram is connected to the right part, and the bottom to the top. At end point of the Symmetry Set branches, $p_i = p_j$, which is the diagonal. This diagonal can also be regarded as an identity mapping of the shape on itself, and therefore as the shape.

Consequently, points on the shape (diagonal) are connected to points on the shape (diagonal) via the curves in the pre-Symmetry Set. As the shape is closed and not self-intersecting, it can be represented as a circle. The connections of points on the shape are visible as cords. An example is given in Figure 4 on the right.

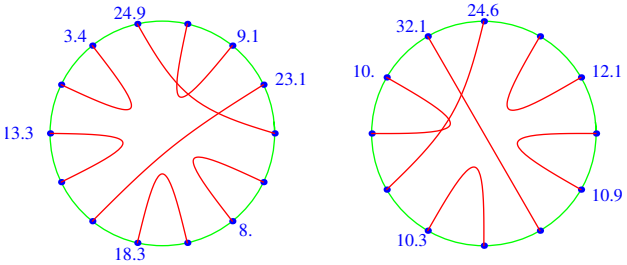


Figure 5: Two circles describing different shapes.

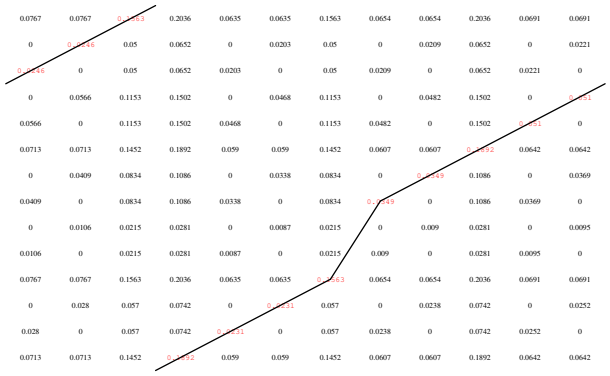


Figure 6: Cost matrix and optimal path for the shape circles in Figure 5.

Next, each cord can be assigned a weight. This weight is the number of points on a branch in the pre-Symmetry Set, divided by the sum of all branches in the pre-Symmetry Set that intersect the diagonal. So the weights sum up to 1. In Figure 4 this number is given as a percentage.

3.1 A String representation

A straightforward manner to store the information given by the circle with cords, is by creating a vector with the same dimension as the number of end points. Each coordinate of the vector then get the value of the relative length of the cord that is related to it. Consequently, the coordinates sum up to 2. When all cords have different length, the cords can easily be reproduced from this vector. However, the connectivity information is lost if two cords have the same length. Therefore, each coordinate of the vector contains besides the length also the coordinate to which it relates.

4 Matching strings

Given two shapes, comparison can done visually by comparing their circle diagrams A and B. As the information of these diagrams consists of points and cord, the points are mapped such, that the number of coinciding cords is highest. Obviously, the ordering of points may not change. As the parameterization has an arbitrary begin point, also all rotated versions of A up to 2π must be taken into account. Furthermore, the number of cords of both circles may differ, as well as the way the cords are connected, see Figure 5.

From the transitions of the Symmetry Set [3] it follows that a cord (a branch of the Symmetry Set) may (dis-) appear in a transition where two end points meet and a cord (dis-) appears. As the removal of a cord in one circle to optimize matching relates to introducing a cord in the other circle, it suffices to consider removing cords. Consequently, a cord connecting two neighbouring end points is allowed to vanish - in the mapping such a cord may be removed.

4.1 Cost Matrix

The matching of two circle diagrams A and B can be done as follows. Let $\{A_i\}_{i=1\ldots n}$ and $\{B_j\}_{i=1\ldots n}$ denote the vectors with the lengths of the branches. Then $M(i,j) = f(A_i, B_j)$ is the cost matrix, where f is some distance measure. In the remainder we shall use $f(x,y) = x.y/\|x\|\|y\|$, but other choices, like $f(x,y) = \|x-y\|$, can be applied as well.

If A=B and the starting positions are equal, tr M describes the inner product between two identical vectors and equals one. If the starting positions are different, the trace of a rotated version of M equals one.

To maximize the matching, a path $P = \{M(i_k, j_l)\}$ is to be found in M, such that each row and column i_k and j_l are present only once - each point can be matched only once. For the two examples given above, this is simple. For different shapes, it must be taken into account that two neighbouring points and their connecting cord may be removed. This relates to the matrix in removing two subsequent rows or columns.

Next, when two points are matched, automatically the two points to which they are connected, must be matched. For simplicity, one can state that when two cords are given by (i_k, i_{k+1}) and (j_l, j_{l+1}), i_k and j_l can only be matched, if i_{k+1} and j_{l+1} are matched, and that the matchings $M(i_k, j_{l+1})$ and $M(i_{k+1}, j_l)$ are forbidden.

An example of a matrix M is given in Figure 6. The origin is bottom left. The line through the matrix denotes the optimal match. As one can see, the matrix contains zeros, denoting the forbidden entries. When two subsequent values along the line are equal, the off-diagonal neighbouring points are zero, as described above. As the vectors have different length, the line makes a jump. The jump skips two rows. In general, jumps skip an even number of rows or columns, since a jump resembles the removal of a number of cords, each with two points.

Figure 7: A fish image, fish shape and a blurred fish shape.

4.2 Implementation

The derivation of the Symmetry Set given a shape is described in [4, 13]. It basically boils down in computing all zero crossings in Eqs. 1-1 for all point pairs (p_i, p_j). These points pairs form the pre-Symmetry Set as shown in Fig. 4, left. Then the distinct Symmetry Set branches that intersect the diagonal are derived, with the locations at the diagonal and their lengths. This gives a set with elements $A_i = (e_1, e_2, L)_i$, with e_1 and e_2 the e_1^{th} and e_2^{th} position on the diagonal, and l the relative length of the branch.

Next, on each cord that is allowed to vanish, the two points are marked as 'begin' or 'end' point. Note that if two cords are nested, both are allowed to vanish. If the cross each other, they cannot be removed. For more details on the type of cords, see [12]. Let $L_i \in A$ and $L_j \in B$, then the cost matrix is built up as $M(i,j) = 0$ if A_i and B_j are a combination of a begin and an end point, and $M(i,j) = L_i L_j$, elsewhere. The path with maximal value is found by using a shortest path algorithm [6] on $-M$. M can be transferred into a graph with as vertices the rectangular grid, given by the dimensions of M, and edges from $M(i,j)$ as follows.

- If $M(i+1, j+1) = M(i,j)$ and $M(i+1,j) = M(i,j+1) = 0$ two begin points of a cord are matched and the only allowed edge is $M(i+1, j+1) \rightarrow M(i,j)$ with cost $M(i+1, j+1)$.
- If $M(i+1, j+1) = 0$, this position is not allowed and the only allowed edges, denoting a possible skip, are $M(i+1, j+1) \rightarrow M(i+1, j)$ and $M(i+1, j+1) \rightarrow M(i, j+1)$, both with cost 0.
- Else three edges are possible: $M(i+1, j+1) \rightarrow M(i,j)$ with cost $M(i+1, j+1)$, and $M(i+1, j+1) \rightarrow M(i+1, j)$ and $M(i+1, j+1) \rightarrow M(i, j+1)$, both with cost 0.

Obviously, to compute the complete path from a point to itself, one should handle the boundaries of M properly. To find the shortest path solution, it suffices to take the shortest paths through the entries of one column or row and take the minimum of them.

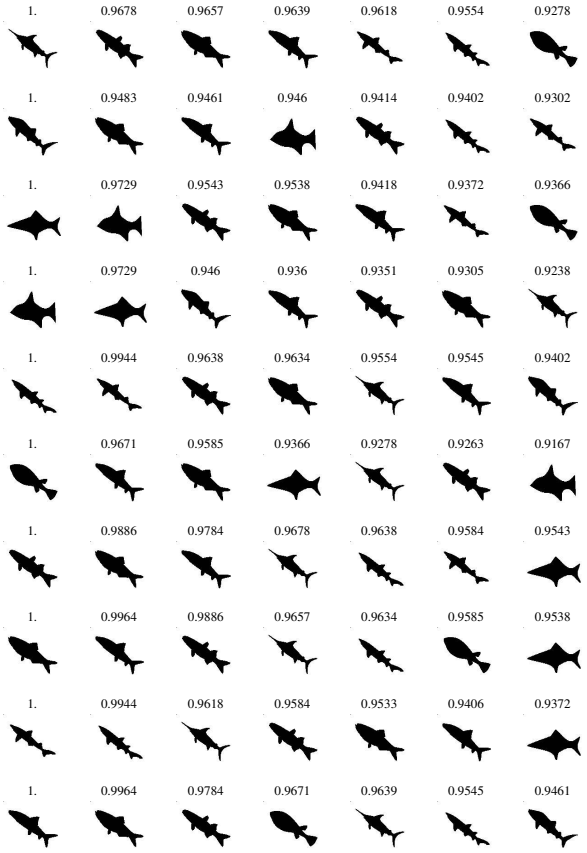


Figure 8: Matching of fishes.

5 Results

In the remaining we used shapes from an existing data base [15, 16]. These shapes are the boundary of 128×128 pixel sized black and white images, as shown in Figure 7, left. Of each image the boundary is extracted and the points are ordered, yielding a sequence of points, Figure 7, middle. The number of points ranges typically from 1200 to 1500. The derivatives of a Gaussian filter are applied to this sequence to find a reasonable estimation of the derivatives [7] of the shape parameterization. The normal vector is obtained at a scale of 4.5 pixels. We note that using a small scale resembles applying a (small) mean curvature motion to the shape [5]. The shape in Figure 7, middle, is therefore slightly blurred, see Figure 7, right.

This blurring of shapes has the property that it removes small details. This may be regarded as a disadvantage, but on the other hand no removal of spurious details, or whatever adjustments to the data need to be carried out. The corresponding string, pre-Symmetry Set and circle diagram are shown in Figures 3-4.

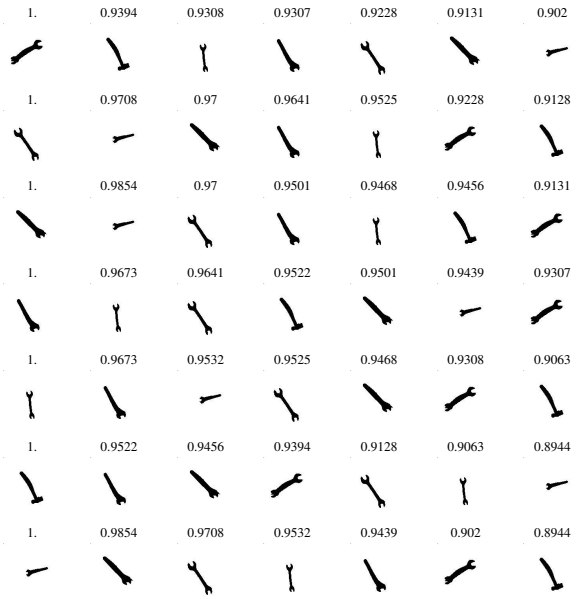


Figure 9: Matching of tools.

Next, 10 different fish shapes are compared. The results are shown in Figure 8. The images show the fish, the numbers the score of the match. The first colum shows the best match, second column the second-best match and so on. As the matching of any shape with itself matches 1, the first column also represents the shape to be matched. The fishes in row three and four are artificially drawn, and they are each others second-best match. Furthermore, the matching has a preference for matching fins. This is due to the fact that fins are introducing prominent extrema of curvature.

The second group of shapes consists of 7 tools, as shown in Figure 9. Although tool number 7 is significantly smaller than the others, it is still matched with larger tools. This is due to the normalization of the lengths of the branches of the pre-Symmetry Set.

The third test shows the comparison of all 10 fishes and 7 tools. The results are shown in Figures 10-11. Most fishes and tools have as the 5 best matches shapes from the same category. In the fishes-part, Figure 10, a wrench occasionally appears. This tool is considered as a fish with only two tail fins and no other fins. For the same reason some fishes appear in the tools-part, Figure 11.

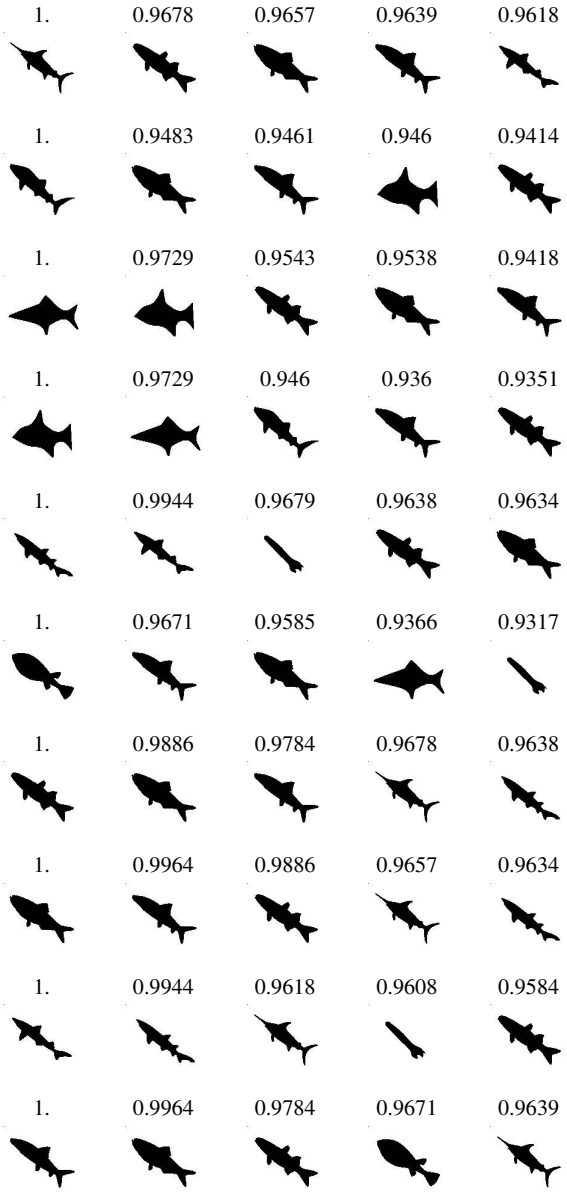


Figure 10: Matching of fishes and tools; the fish part.

6 Summary and Conclusions

We introduced a new way to represent and compare shapes based on the Symmetry Set, a generalization of the Medial Axis. This string representation uses the end point of the Symmetry Set branches and the relative length of the branch

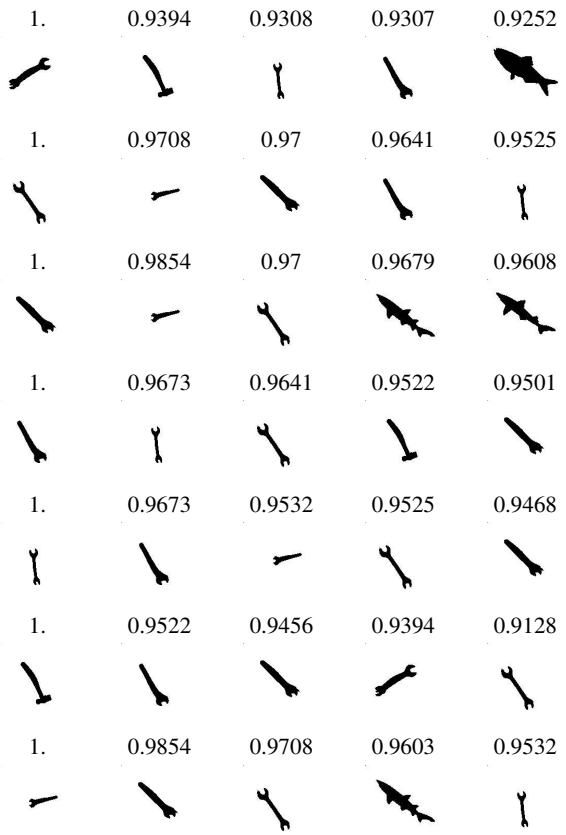


Figure 11: Matching of fishes and tools; the tools part.

in the pre-Symmetry Set diagram. The end points represent the extrema of curvature of the shape. Therefore, the representation links these extrema pair wise. This idea of pair wise linking of points on the shape relates conceptually to that of Curvature Scale Space [14], albeit that we do not use a scale space to establish a linking, but use the Symmetry Set. The representation allows the matching of shapes by comparing strings, for instance by taking the inner product of appropriate sub sets of these strings. The sub sets are defined by applying allowed changes of the Symmetry Set. The maximal matching is found by an adapted shortest pad algorithm that finds the optimal sub sets. Examples show the usability of the proposed method. Future work will focus on improvement of the shortest path based algorithm and on the influence of alternative difference measures besides the inner product.

References

[1] A. Blake, M. Taylor, and A. Cox. Grasping visual symmetry. *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 724–733, 1993.

[2] H. Blum. Biological shape and visual science (part i). *Journal of Theoretical Biology*, 38:205–287, 1973.

[3] J. W. Bruce and P. J. Giblin. Growth, motion and 1-parameter families of symmetry sets. *Proceedings of the Royal Society of Edinburgh*, 104(A):179–204, 1986.

[4] J. W. Bruce, P. J. Giblin, and C. Gibson. Symmetry sets. *Proceedings of the Royal Society of Edinburgh*, 101(A):163–186, 1985.

[5] F. Cao. *Geometric Curve Evolution and Image Processing*, volume 1805 of *Lecture Notes in Mathematics*. Springer Verlag, 2003.

[6] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, 1993.

[7] L. M. J. Florack. *Image Structure*, volume 10 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[8] P. J. Giblin and B. B. Kimia. On the intrinsic reconstruction of shape from its symmetries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):895–911, July 2003.

[9] P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, medial axes, and shocks. *International Journal of Computer Vision*, 54(1/2):143–156, 2003.

[10] B.B. Kimia. On the role of medial geometry in human vision. *Journal of Physiology - Paris*, 97(2-3):155–190, 2003.

[11] R. Kimmel, N. Sochen, and J. Weickert, editors. *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*. Springer -Verlag, Berlin Heidelberg, 2005.

[12] A. Kuijper and O.F. Olsen. The structure of shapes: Scale space aspects of the (pre-) symmetry set. In *Kimmel et al. [11]*, pages 291–302, 2005.

[13] A. Kuijper, O.F. Olsen, P.J. Giblin, Ph. Bille, and M. Nielsen. From a 2D shape to a string structure using the symmetry set. In *Proceedings of the 8th European Conference on Computer Vision - ECCV 2004, Part II (Prague, Czech Republic, May 11-14, 2004)*, volume II, pages 313–326, 2004. LNCS 3022.

[14] F. Mokhtarian and M. Z. Bober. *Curvature Scale Space Representation: Theory, Applications, & Mpeg-7 Standardization*, volume 25 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[15] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, 1999.

[16] T.B. Sebastian, P.N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. In *Proceedings of the 8th International Conference on Computer Vision (2001)*, pages 755–762, 2001.

[17] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–22, 1999.