ESTIMATION OF VISUAL MOTION IN IMAGE SEQUENCES

Rasmus Larsen

The Institute of Mathematical Modelling
Ph.D. Thesis No. 1
Lyngby 1995

IMM

ISSN 0909–3192

Printed by IMM/Technical University of Denmark
Bound by Hans Meyer

Preface

This thesis has been prepared at the IMSOR group of the Institute of Mathematical Modelling (IMM), the Technical University of Denmark, in partial fulfillment of the requirements for the degree of Ph.D. in engineering.

The thesis discuss different aspects of the problem of estimating the visual motion given a sequence of images. The treatment of the subject is not exhaustive, nor detailed, but is intended to provide an insight in motion detection by selected theory and examples.

A basic knowledge in the areas of image processing and statistics is required by the reader.

Lyngby, April 1994

Rasmus Larsen

Acknowledgements

During the pursuit of this work I have drawn on a lot of resources of a technical, social, and, economical nature. I would like to express my gratitude to those who have provided these resources.

First of all I would like to express my gratitude to Professor Knut Conradsen, and Dr. Bjarne Kjær Ersbøll for their guidance and encouragement during the course of this work, and for providing the excellent research facilities of our group at IMSOR, now IMM.

Furthermore, I wish to thank my colleagues of the image analysis group at IMM for providing a pleasant and inspiring research as well as social environment. The systematic effort of the group to develop and document software within the HIPS-package has proven invaluable for this work. Especially, I would like to thank Allan Aasbjerg Nielsen for his interest, help, and moral support during all phases of my study. I also thank Anders Rosholm, who implemented some to me very useful algorithms on a massively parallel computer under a CAP project. His interest in the problems concerning the algorithm exceeded anything that might have been expected by reasonable standards.

During my stay at the Computer Vision Laboratory at the University of Linköping I received very kind treatment. The stay proved very profitable, and has certainly been important for the contents of this thesis. For that I thank Professor Gösta Granlund and his co-workers. I also thank the Nordic Research Network in Computer Vision for financially supporting me during my stay.

Image data was provided by the Danish Meteorological Institute. I wish to thank Senior Scientist Hans H. Valeur for his interest in my work.

I would also like to use this opportunity to thank the friends I have gotten to know at Studenterforeningen in Copenhagen. The course of this study has coincided with a period of intense involvement in the life of Studenterforeningen. Any technical skills and experiences obtained through my study are surely rivaled by the personal experiences and general knowledge of life I have gained through participating in the activities of Studenterforeningen with them.

Finally I would like to express my indebtedness to my family, and to Annette Staal, whose moral support and encouragement has been invaluable, especially during the last months. I can only hope that some day I will get a chance to pay her back.

Rasmus Larsen

Summary

The problem of estimation of visual motion from sequences of images has been considered within a framework consisting of three stages of processing

First the extraction of motion invariants, secondly a local measurement of visual motion, and third integration of local measurements in conjunction with a priori knowledge.

We have surveyed a series of attempts to extract motion invariants. Specifically we have illustrated the use of local Fourier phase. The Fourier phase is shown to define the local shape of the signal, thus accurately localizing an event.

Different strategies for local measurement of motion under an assumption of translatory motion has been considered. Furthermore we have described methods for quantifying the directional certainty with which we have measured the local displacement. A method based on local estimation of the spatio temporal orientation is generalized to give a continuous description of certainty of the estimated motion. Examples on application of the techniques are given.

With respect to integration of local measurements, we have surveyed different techniques within a Bayesian framework. Generalization of prior distributions using 2-D Markov random fields are given. In particular we have investigated the use of smoothness of the second order derivatives, and the use of edge model and prior distributions for the field that favor discontinuities to characterize the motion field.

A succesful implementation of a temporal interpolation in a sequence of weather satellite images based on the estimated motion field is shown.

Rasmus Larsen

Resumé

Estimation af den synlige bevægelse i sekvenser af billeder er blevet behandlet som en proces bestående af tre trin.

Først udvælgelsen af en bevægelses invariant, dernæst en lokal observation af den synlige bevægelse og for det tredje integration af disse lokale observationer under hensyntagen til a priori viden.

Der er givet en oversigt over forsøg på at definere en bevægelses invariant. Specielt illustreres anvendelsen af den lokale Fourier fase. Fourier fasen vises at give en formbeskrivelse af signalet, hvor ved en hændelse præcist bliver stedfæstet.

Der gives en oversigt over forskellige strategier til at estimere den lokale bevægelse, under en antagelse om at bevægelsen lokalt kan beskrives som en translation. Desuden beskrives der metoder til at kvantificere den retningsbestemte sikkerhed hvor med den observerede forskydning er bestemt. En metode baseret på lokal estimation af den spatio temporale orientering er blevet generaliseret til at give en kontinuert beskrivelse af den estimerede bevægelse. Der gives eksempler på anvendelse af disse teknikker.

For så vidt angår integration af de lokale observationer, så gives der en oversigt over forskellige teknikker indenfor rammerne af Bayes paradigme. Forskellige generaliseringer af a priori fordelinger under anvendelse af Markov felter gives.

Specielt undersøges anvendelsen af glathed af anden ordens afledede samt anvendelsen af modeller for diskontinuiteter og a priori fordelinger der favoriserer diskontinuiteter i bevægelsesfeltet.

En implementering af en tidslig interpolation i en sekvens af vejrsatellitbilleder baseret på det estimerede bevægelsesfelt vises.

Rasmus Larsen

Contents

Chapter 1

Introduction

1.1 Visual Motion

This thesis is concerned with the estimation of visual motion in sequences of two dimensional images. Rather than stating hypotheses regarding the importance of the perception of visual motion to humans, we will focus on applications of motion estimation in image sequences.

The term visual is used in the title to emphasize that the motion estimation is restricted to the two dimensional projections of the in general three dimensional trajectories of moving objects. The collection of these two dimensional velocities is commonly referred to as the 2-D motion field, or the optical flow.

The estimation of the motion field is interesting for a number of applications. Of course the motion field holds a great deal of information about the three dimensional surface structure of objects, such as depth and orientation, the relative motion between objects and camera, and the relative motion between objects. The segmentation of moving objects might be of interest in a surveillance system.

Another application for the computation of the motion field is image sequence compression. It is evident that if we can utilize the repetition of spatial patterns in time sequences of images, e.g. video signals, in a data compression, we should be able to obtain better results, than if the compression was restricted to considering one frame at a time. In order to identify the repetitions, however, we need the motion field.

In a situation almost contrary to the one above the motion field can be used in order to perform a temporal interpolation in a time sequence of images. Given a crudely sampled sequence the perceived motion might be considerably disturbed by large displacement between images. This situation arises as we will later show in images from a meteorological satellite. Due to a narrow bandwidth in the satellite to earth communication we only get one image every half hour. Other applications of this type include the production of animated cartoons. In a temporal interpolation scheme, where the animator only would be requested to produce every fifth or tenth drawing, we should be able to considerably reduce production time.

A last example of the usefulness of the motion field is image enhancement. It is evident that repeated measurements can reduce estimation variance. Repeated measurements in the form of a time sequence of images of an object might be used for enhancement or resolution improvement if motion between frames was known.

1.2 Detection of Visual Motion

In order to pursue the estimation of motion estimation, we need to define a motion invariant, a feature the relates the projection of a surface point on the image plane at one time to the projection of the same point in the following image(s).

Secondly, we must define a method of tracking this motion invariant, or property from image to image. As we will later discover this process is concerned with considerable ambiguity. In general image sequences we can not define a motion invariant that has a unique match from image to image at all positions.

This problem is helped by an assumption that real world objects generally are coherent and move, or evolve, in constrained ways, thus causing motion fields to exhibit certain regularities. So the third step we will concern ourselves with is an integration of ambiguous matches given by the second step using assumptions of regularity.

1.3 Outline of the Thesis

Chapter 2 gives an overview of different approaches to defining a motion invariant. These include intensity, derivatives of intensity, local patterns, feature points, and curves, and finally a shape measure based on the Fourier phase.

In Chapter 3 we deal with the matching problem. How can we link points in subsequent images, and how can we describe the ambiguity in the match. A variety of methods within the classes of region based matching techniques, spatio-temporal gradient methods, methods based on frequency analysis, and feature tracking are described. Examples will be given in Chapter 4, where we will introduce a method using the Fourier phases as invariants, combined with a tracking process based on spatio temporal gradients. Furthermore we will introduce a feature tracking method based on the Euclidean distance transform.

Methods to integrate the ambiguous matches will shortly be surveyed in Chapter 5. Following this we will introduce Markov random fields as a means of integrating the matches with statistical models of the regularity of the motion fields. In Chapter 6 we will illustrate this by an example of temporal interpolation in Meteosat weather images based on the estimation of the motion field. In addition to this we will show some applications to motion based object segmentation.

1.4 Notation and Symbols

We will briefly outline some of the mathematical notation used throughout this thesis. In general, we will denote scalars using lower case letters, and vectors using bold lower case letters, e.g. $\boldsymbol{x} = (x, y)^T$, where $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$. Matrices will be denoted using upper case bold letters, e.g. the matrix $\boldsymbol{A}$ which may consist of the elements a_{ij} where i, and j denote the row and column number of the element, respectively.

In general, we will use $\boldsymbol{x} = (x, y)^T$ to denote the spatial coordinates, and t to denote the temporal coordinate. The intensity function, (or image) being $I(\boldsymbol{x}, t)$. The corresponding coordinates in frequency space will be $\boldsymbol{k} = (k_1, k_2, k_3)^T$. For the Fourier transformed functions we will use calligraphic letters, $\mathcal{I}(\boldsymbol{k})$, and we will use j for the complex unit.

In general, though small deviations occur, we will use $\boldsymbol{u} = (u, v)$ to denote the image velocity. The concept of normal flow, which will be defined later, is denoted $\boldsymbol{\mu} = (\mu, \nu)^T$. Some times we will index the flow using the pixel number, i.e. $\boldsymbol{u}_i = (u_i, v_i)^T$, at other times we will prefer a notation based on indexing the points, i.e. $\boldsymbol{u}(x_i, y_i) = (u(x_i, y_i), v(x_i, y_i))^T$.

In the context of statistics will use the usual upper case bold letter for stochastic variables, e.g. $\boldsymbol{X}$. The expectation will be denoted E. An estimator will usually be identified by a hat, e.g. $\hat{\boldsymbol{u}}$. Probabilities will be indicated by uppercase P's, e.g. $P(\boldsymbol{X} = \boldsymbol{x})$, where as probability density functions are denoted using lower case p's, e.g. $p(\boldsymbol{x})$.

The Euclidean norm will be denoted $\|\cdot\|$, and the p-norm $\|\cdot\|_p$.

Differentiation will sometimes be denoted by subscripts, e.g. I_x is the partial derivative of I with respect to x, in other places we will use $\frac{\partial I}{\partial x}$.

Chapter 2

Motion Invariants

In order to perceive the displacements occurring in a sequence of images we need to define motion invariants, i.e. observable properties that we expect to be conserved over time and space. Of course, this is a problem common also to stereopsis, registration etc. This chapter gives an overview of such motion invariants.

2.1 Intensity and Intensity Derived Properties

Early methods such as the one described by Horn & Schunk (1981) worked directly on the measured intensity, assuming the brightness of a moving point being conserved over time. Fennema & Thompson (1979) used defocused images, i.e. Gaussian filtered, in order to reduce the noise in the gradient estimates, which were based on the Sobel operator.

Time varying illumination, however, poses a problem for these methods based on the conservation of intensity. In natural images this illumination problem is quite common, due to time varying light sources, e.g. the sun, and relative motion of camera, scene and light source.

To overcome this illumination problem, and at the same time reduce noise, a number of implementations have used bandpass filtered intensity as their motion invariant.

Singh (1991) uses the difference-of-Gaussians as suggested by Burt (1981) computed on each image separately

$$K(x,y) = \frac{1}{2\pi}\frac{\exp(-(x^2+y^2)/(2\sigma_1^2))}{\sigma_1^2} - \frac{1}{2\pi}\frac{\exp(-(x^2+y^2)/(2\sigma_2^2))}{\sigma_2^2} \quad (2.1)$$

Others have used the related Laplacian-of-Gaussian operator, also known as the Marr-Hildreth operator, $\nabla^2 G$, to generate a bandpass filtered signal of each image

$$\nabla^2 G(x,y) = \frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)\exp(-\frac{x^2+y^2}{2\sigma^2}) \quad (2.2)$$

Buxton & Buxton have implemented a spatio-temporal extension of the Laplacian-of-Gaussian operator, (Murray & Buxton, 1984).

Duncan & Chou (1992) employed the second derivative in time of the temporal Gaussian smoothing function

$$S_t = \frac{\partial^2 G(t)}{\partial t^2} = -\frac{2s^3}{\sqrt{\pi}}(1 - 2s^2t^2)\exp(-s^2t^2) \quad (2.3)$$

2.2 Feature Points

Corner points, e.g. the projection of the vertices of a polyhedron, or points of high curvature on surface markings, may be regarded as projections of scene features whose appearance is invariant to rigid motion. In some sense this also applies to the boundaries of objects. The methods of this group aim at extracting feature points in subsequent images to be matched at a later stage.

These point and curve extraction methods are closely related to the assessment of locations where motion can be estimated accurately. We will review intensity corner and curve extraction methods in this context in Section 3.2.

2.3 Fourier Phase

The Fourier phase, or rather the windowed Fourier phase, holds information of the position of the structure within the window. This follows from the the well known property of the Fourier transform that a shift in the spatial domain corresponds to a shift of phase in the Fourier domain. Consider the infinite signal, $I(x)$, whose Fourier transform is $\mathcal{I}(k)$. Then the Fourier transform of the shifted version given by $I_s(x) = I(x - s)$ is

$$\mathcal{I}_s(k) = \int_{-\infty}^{\infty} I_s(x) \exp(-j2\pi kx)\, dx = \exp(-j2\pi ks)\, \mathcal{I}(k) \tag{2.4}$$

This is further illustrated in Figure 2.1. As the phase is independent of the contrast, which is measured by the magnitude of the Fourier transform, the illumination problem is eliminated by using the phase as the conserved property.

For a multi-dimensional signal, e.g. an image (2-D) or an image sequence (3-D), Knutsson (1982) has shown that we can define the phase corresponding to a quadrature[1] filter as the quotient between the odd and the even part of the filter. We will briefly review the filter design and phase definition of Knutsson.

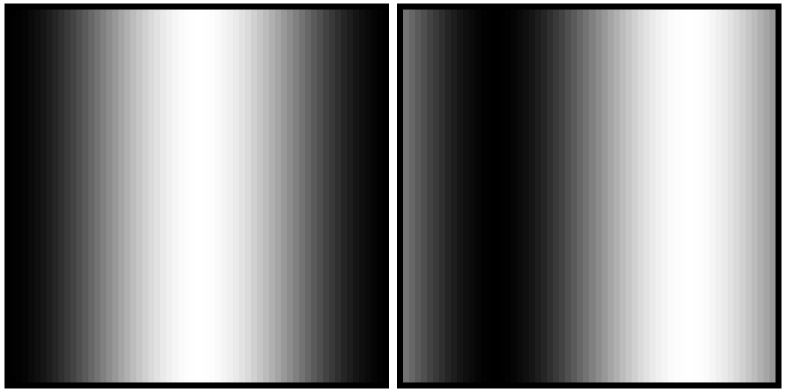
A real valued image, $I(\boldsymbol{x})$, has an Hermitian Fourier transform

$$\mathcal{I}(\boldsymbol{k}) = \mathcal{I}^*(-\boldsymbol{k}) \tag{2.5}$$

where $*$ denotes the complex conjugate. From this property it follows that measurements need only be done in one half plane of the Fourier domain. Knutsson suggested a quadrature filter given by its frequency function

$$Q_p(\boldsymbol{k}) = \begin{cases} F(\|\boldsymbol{k}\|)(\frac{\boldsymbol{k}\cdot\boldsymbol{n}_p}{\|\boldsymbol{k}\|})^{2A} & \text{if} \quad \boldsymbol{k}\cdot\boldsymbol{n}_p \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

[1]Quadrature means integration, e.g. the quadrature of a circle. In this context it is used because either component of the quadrature filter pair is the integral of the other. This becomes evident if we think of the pair as a sine and a cosine function



(a) $\phi = 0$ (b) $\phi = -\frac{\pi}{2}$

Figure 2.1: The two intensity patterns are generated using a sine-function with period equal to the side length of the image. The images have identical Fourier magnitudes. But the Fourier phases corresponding to the frequency of the sine-function are 0 and $-\frac{\pi}{2}$, respectively. Thus indicating the position of the structure.

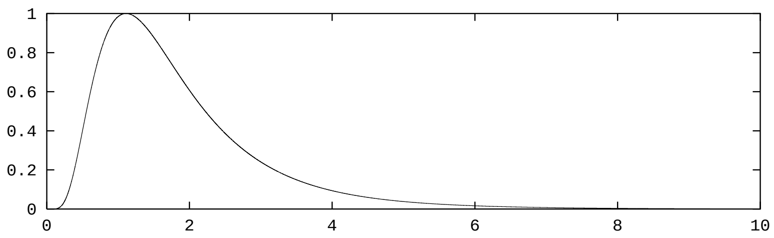


Figure 2.2: The Lognormal function. The center frequency is 1.1 and the bandwidth is 2 octaves.

where F is a bandpass function, that determines at which scale the filter has its sensitivity, $\boldsymbol{n}_p$ is the direction of the filter, and A is a parameter that specifies the angular bandwidth of the filter.

The quadrature filter can be separated into an odd and an even part

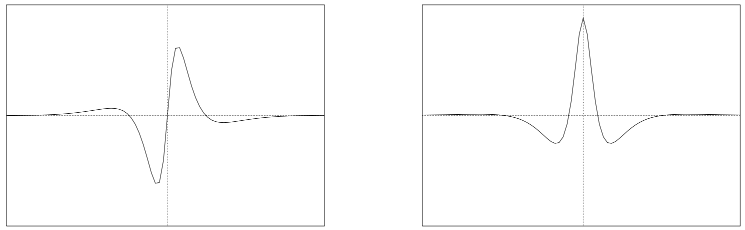
$$Q_p(\boldsymbol{k}) = H_{pe}(\boldsymbol{k}) + H_{po}(\boldsymbol{k}) \tag{2.7}$$

where

$$\begin{aligned} H_{pe}(\boldsymbol{k}) &= H_{pe}(-\boldsymbol{k}) = \frac{1}{2}F(\|\boldsymbol{k}\|)(\frac{\boldsymbol{k}\cdot\boldsymbol{n}_p}{\|\boldsymbol{k}\|})^{2A} \\ H_{po}(\boldsymbol{k}) &= -H_{po}(-\boldsymbol{k}) = \frac{1}{2}F(\|\boldsymbol{k}\|)(\frac{\boldsymbol{k}\cdot\boldsymbol{n}_p}{\|\boldsymbol{k}\|})^{2A}\mathrm{sign}(\boldsymbol{k}\cdot\boldsymbol{n}_p) \end{aligned} \tag{2.8}$$

Knutsson suggested the following lognormal function as the bandpass function in Equation (2.6)

$$F(\|\boldsymbol{k}\|) = \exp(-4\log^2(\frac{\|\boldsymbol{k}\|}{\|\boldsymbol{k}_0\|})/B^2\log(2)) \tag{2.9}$$



(a) (b)

Figure 2.3: The spatial lognormal filter. (a) The real even part of the filter. (b) The imaginary, odd part of the filter.

where $\boldsymbol{k}_0$ is the center frequency of the filter, and B is the bandwidth measured in octaves. In Figure 2.2 a plot of the lognormal function is shown. Note also that the angular function varies as $\cos^{2A}(\theta)$, where θ is the angle between the frequency and the main direction of the filter.

In Figure 2.3 a spatial realization of the one dimensional lognormal quadrature filter pair is shown. From this figure it is evident that in normal image processing terms the even filter is an edge detector and the odd filter is a line detector.

Applying the odd and even filter as convolutions between the image and their spatial transforms, h_{pe} and h_{po}, yields the output

$$\boldsymbol{q}_p = q_{pe} + jq_{po} \tag{2.10}$$

where, denoting convolution with a $*$

$$\begin{aligned} q_{pe}(\boldsymbol{x}) &= h_{pe}(\boldsymbol{x}) * I(\boldsymbol{x}) \\ q_{po}(\boldsymbol{x}) &= -jh_{po}(\boldsymbol{x}) * I(\boldsymbol{x}) \end{aligned} \tag{2.11}$$

The magnitude, $\rho(\boldsymbol{x})$, and phase, $\phi(\boldsymbol{x})$, of the filter are then defined as

$$
\begin{aligned}
\rho(\boldsymbol{x}) &= |\boldsymbol{q}_p| = \sqrt{q_{pe}^2(\boldsymbol{x}) + q_{po}^2(\boldsymbol{x})} \\
\phi(\boldsymbol{x}) &= \arg(\boldsymbol{q}_p)
\end{aligned}
\tag{2.12}
$$

This polar representation of phase is illustrated in Figure 2.4. The phase value of each shape is represented by a point on the unit circle. The arc length between points is a measure of difference between the two shapes. Using phase modulo 2π we have a continuous description of shape.

In (Weng, 1993; Langley, Atherton, Wilson, & Larcombe, 1991) implementations of the windowed Fourier phase as a conservation property for image matching is described. Fleet & Jepson (1990), Fleet, Jepson, & Jenkin (1991) have described the use of the Fourier phase in motion analysis.

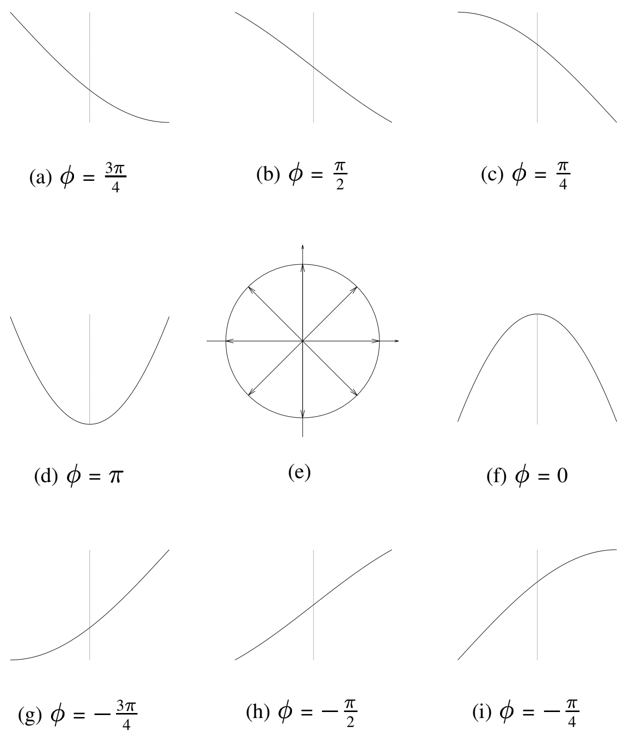


Figure 2.4: The correspondence of the windowed Fourier phase to different shapes. (a) Concave falling. (b) Falling step edge. (c) Convex falling. (d) Negative peak. (f) Positive peak. (g) Concave rising. (h) Rising step edge. (i) Convex Rising. These examples serve to illustrate the rich geometrical meaning of phase. The position of the phases in the plots is shown on the unit circle in (e).

Chapter 3

Local Measurement of Visual Motion

In general image sequences the velocity field shows complex variation, spatially as well as temporally, across the image. This is due to independently moving objects, rotation, dilation etc.

In order to obtain a good resolution of the velocity field it would appear necessary to restrict our velocity estimation to local computations.

Localization of the measurements ensure several things. First of all, with narrow apertures[1] smoothly varying velocity fields can be estimated based on translational image velocity, as opposed to more complex descriptions needed for larger neighborhoods. Secondly, localization ensures good resolution of the velocity field, that is to say we have isolated events, so that the measurement is relevant within the entire aperture. Finally, localization helps to isolate effects due to the occurrence of multiple events in close proximity, meaning that we can minimize

[1] We use the word aperture to denote the neighborhood of a point from which we draw information of the local event

the effect of occlusion on the estimation of the velocity field, and we can keep disjoint objects independent.

3.1 The Aperture Problem

As much as we need localization, restricting measurements to narrow apertures, i.e. spatio-temporal neighborhoods, often results in the measurements being based on one-dimensional intensity structures (edges and/or lines). In this case we can only determine the component of the velocity orthogonal to the intensity contour reliably. This is commonly known as *the aperture problem*. The term was introduced by Marr & Ullman (1981). In Figure 3.1 the basis of the aperture problem is outlined. The figure shows a rectangle translating at velocity $\boldsymbol{u}$ at two points in time. However, within the two marked apertures we can only determine the component of the velocity that is perpendicular to the contour. The component of the velocity orthogonal to the intensity contour, denoted $\boldsymbol{\mu}$, is referred to as the normal velocity or component velocity. Letting $\boldsymbol{n}$ denote the unity vector perpendicular to the contour, the relation to the velocity is given by $\boldsymbol{\mu} = (\boldsymbol{u}^T\boldsymbol{n})\boldsymbol{n}$.

In general the aperture problem exists in neighborhoods of the image sequence that have a one dimensional structure, and neighborhoods that have no structure at all, that is in homogeneous areas. On the other hand, in image sequence neighborhoods that exhibit two dimensional spatial structures, such as intensity corners or various textured regions, we can reliably extract the true velocity. These points though in general sparsely distributed over the image hold important information on the estimation of velocity. In Figure 3.2 we can see a neighborhood containing a one dimensional structure in a real image sequence that suffers from the aperture problem, as well as a neighborhood that does not.

It is evident at this point that local velocity estimation algorithms that are able to distinguish between, or even quantify, the nature of the neighborhood, and thus the estimated velocity - component velocity or not - should be more successful than algorithms that can not.

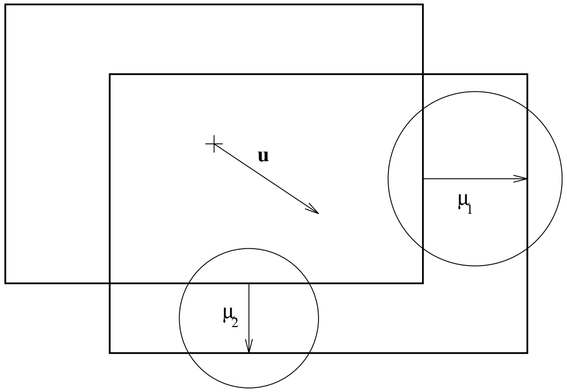


Figure 3.1: The rectangle is translating at velocity $\boldsymbol{u}$. Within apertures along the contours, however, it is only possible to determine the component of the flow perpendicular to the contour

3.2 Identification of Corners and Curves

In general, motion estimation techniques fall within one of two classes. The first class is generally referred to as token matching or correspondence analysis. Methods from this class have appeared in essentially two flavors. Either they aim at extracting points, e.g. corners, or curves, e.g. edges, at which some motion estimation technique should be applied, or they aim at tracking extracted points and curves in subsequent images. The methods of the second class apply their estimation technique at every point in the image.

As mentioned previously corners have the property that their motion can be directly computed based only on measurements made at the corner, or to pu it more accurately in a small neighborhood of the corner. Furthermore corners can be regarded as projections of scene features whose general appearance is invariant to rigid motion, e.g. an image corner can be the projection of the vertex of a polyhedron, or of a point of high curvature on the boundary of a surface marking.

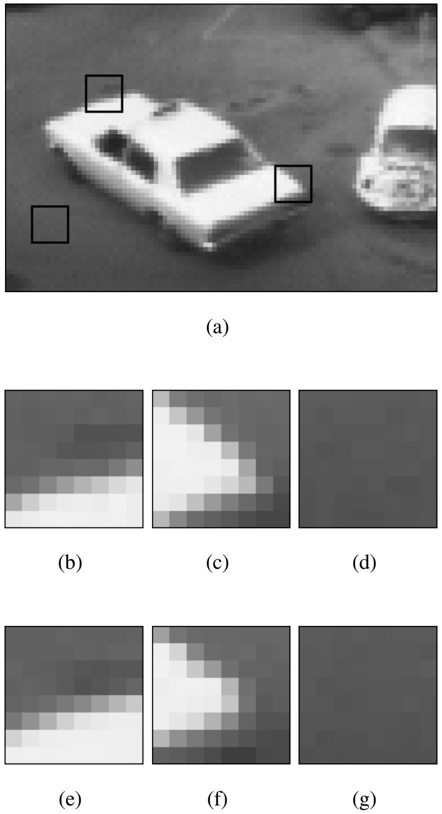


Figure 3.2: (a) Part of a frame from the Hamburg Taxi Sequence produced by the University of Hamburg. Three local neighborhoods containing a moving line, a moving corner, and a homogeneous area respectively, have been marked. The three neighborhoods are shown at two instances in time. (b),(e) From the neighborhood containing the single moving contour it is only possible to extract the normal velocity, i.e. the velocity perpendicular to the contour. (c),(f) In the case of the neighborhood containing the intensity corner we can extract the true velocity. (d),(g) In the case of homogeneous neighborhoods no reliable velocity can be estimated.

Although curves corresponding to object boundaries, or surface marking boundaries, do not locally constrain the true image flow, they can still provide us with important information about the motion.

We will briefly review some corner and curve detection techniques presented in the literature. The presentation is not intended to be exhaustive, nor to be detailed. This feature extraction is not only of interest in motion estimation applications, but also in general scene analysis and stereo matching.

3.2.1 Extraction of Points

Some methods attack this problem by first extracting edges as a chain code and then searching for significant turnings in the edges, i.e. points of high curvature. Let the spatial coordinates be (x, y). Assuming the curve to be regular and choosing a sufficiently small neighborhood, we can without loss of generality define the curve, given by the pixels of the chain code, as a function y of x within the neighborhood. The curvature of this plane curve is given by

$$\kappa = \frac{\frac{\partial^2 y}{\partial x^2}}{(1 + (\frac{\partial y}{\partial x})^2)^{3/2}} \tag{3.1}$$

Other methods work directly on the grey level images. A measure of cornerness can be defined as function of the gradient and the rate of change of the gradient. The corners are then found by a threshold. Beaudet (1978) proposed an operator called DET given by the second order derivatives of the intensity function $I(x, y)$.

$$\mathrm{DET} = I_{xx} I_{yy} - I_{xy}^2 \tag{3.2}$$

The corners are found by a threshold of the absolute value of this operator. In the vicinity of a corner DET will respond (with opposite signs) on both sides of the edge. The DET operator is related to the Gaussian curvature, K, given by Equation (A.6) as follows

$$K = \frac{\mathrm{DET}}{(1 + I_x^2 + I_y^2)^2} \tag{3.3}$$

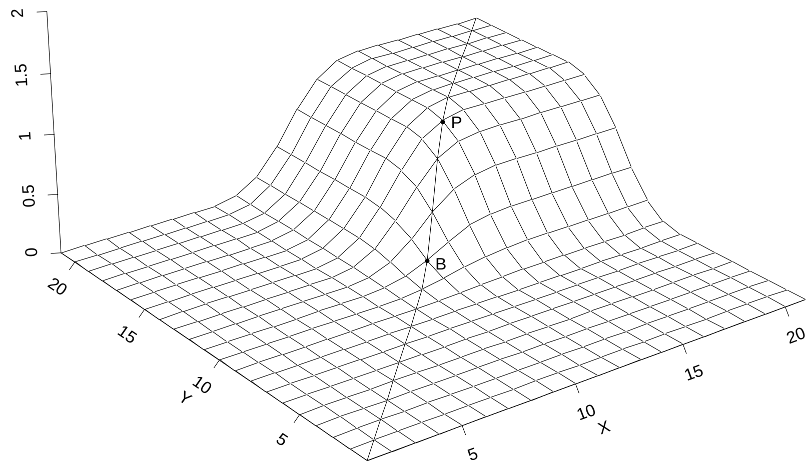


Figure 3.3: The grey level pattern is generated as the 2-D step function $I(x, y) = H(x)H(y)$, where H is the Heaviside function, blurred by a 2-D Gaussian with standard deviation 1.5. The curvature along the intersection line is non-zero around the points P and B. At the promontory point P the curvature attains its maximum and at B it attains its minimum.

Nagel (1983) describes an operator based on the Gaussian curvature, developed by Dreschler and Nagel. Assume a grey level intensity corner with high values inside the corner and low values outside the corner, as can be seen in Figure 3.3. Along the intersection curve corresponding to $x = 21$ one of the principal curvatures is identically 0. Therefore the Gaussian curvature is likewise identically zero. Along the intersection curve drawn in the plot, the Gaussian curvature will be nonzero around the promontory point P and around the bottom point B. At the promontory point one of the principal curvatures corresponds to the flattening of the intensity from the steep ascent to the constant plateau. The other principal curvature corresponds to the changing orientation of the gradient projection into the $x - y$-plane, i.e. to the corner. Since the osculating circle corresponding to both principal curvatures has its center below the surface, the Gaussian curvature will be positive. At the bottom point the principal curvature corresponding to the corner will have the same sign as at P. The other principal curvature, however, corresponds to the flattening of the descent to the constant base value. The center of its osculating circle is located at the viewers side of the surface. Since the centers are on different sides of the surface, the principal curvatures will have opposite signs, and the Gaussian curvature will be negative.

The algorithm then consists of

1. Determine Gaussian curvature
2. Find location of extremal Gaussian curvature
3. Match locations of maximum positive Gaussian curvature, such as P to points of minimum negative Gaussian curvature, such as B.
4. Select the point of the curve between P and B that has a principal curvature equal to zero. This is the corner point.

Nagel (1983) consequently refers to the Gaussian curvature as the product of the principal curvatures. However one may note that in Equation (3b) on page 89 of (Nagel, 1983), the product of the principal curvatures is defined as the DET operator and not the correct expression given in Equation (3.3).

In the approach described by Kitchen & Rosenfeld (1982) a measure of cornerness is defined as the rate of change of the gradient direction along an image contour multiplied by the local gradient magnitude. The direction of the gradient is given by

$$\theta = \arctan(I_y, I_x) \tag{3.4}$$

The partial derivatives of θ are

$$\begin{aligned} \theta_x &= \frac{I_{xy} I_x - I_{xx} I_y}{I_x^2 + I_y^2} \\ \theta_y &= \frac{I_{yy} I_x - I_{xy} I_y}{I_x^2 + I_y^2} \end{aligned} \tag{3.5}$$

Now, the gradient is directed across the edge, so the vector $(-I_y, I_x)$ is directed along the edge. Projecting the rate of change of the gradient angle along the edge, and multiplying the result by the gradient magnitude, yields the following measure of cornerness

$$C = \frac{I_{xx} I_y^2 + I_{yy} I_x^2 - 2 I_{xy} I_y I_x}{I_x^2 + I_y^2} \tag{3.6}$$

which is closely related to the curvature along the edge, i.e. perpendicular to the gradient. Using Equation (A.5) we find the curvature along the edge to be

$$\begin{aligned} \kappa &= \frac{(-I_y, I_x)}{\sqrt{I_x^2 + I_y^2}} \boldsymbol{G}^{-1} \boldsymbol{H} \frac{(-I_y, I_x)^T}{\sqrt{I_x^2 + I_y^2}} \\ &= \frac{I_{xx} I_y^2 + I_{yy} I_x^2 - 2 I_{xy} I_y I_x}{(I_x^2 + I_y^2)\sqrt{1 + I_x^2 + I_y^2}} \end{aligned} \tag{3.7}$$

Noble (1988) evaluated the Plessey corner detector, which consists of the following steps

1. Calculate the partial derivatives of $I(x, y)$, I_x and I_y
2. Compute the quantities, I_x^2, I_y^2, and $I_x I_y$ at every pixel
3. Compute the locally weighted averages, $\langle I_x^2 \rangle$, $\langle I_y^2 \rangle$, and $\langle I_x I_y \rangle$
4. Evaluate the eigenvalues, λ_1 and λ_2, of the matrix $\boldsymbol{A}$, where

$$\boldsymbol{A} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \tag{3.8}$$

   If both eigenvalues are large, declare a corner

In Plesseys implementation the cornerness C_p is calculated as the ratio

$$C_p = \frac{\text{trace}(\boldsymbol{A})}{\det(\boldsymbol{A})} = \frac{\lambda_1 + \lambda_2}{\lambda_1 \lambda_2} \tag{3.9}$$

and a corner is marked if C_p is small. Using differential geometry Noble (1988) showed that C_p is inversely proportional to the square of the curvature in the direction perpendicular to the gradient, i.e. similar to the approach of Kitchen & Rosenfeld (1982).

On a detailed comparison and an analysis of the these techniques the reader is referred to (Deriche & Giraudon, 1993). Deriche & Giraudon (1993) furthermore describe methods to improve the localization of the corner based on the techniques above.

3.2.2 Extraction of Curves

Curves can be extracted from images in a multitude of ways. A classic edge detector is the recursive implementation of the Canny-filter by Deriche, (Canny, 1986; Faugeras & Papadopoulo, 1993). In the literature of motion estimation a popular method for curve extraction is the use of the zero crossings of the Laplacian-of-Gaussian, see Equation (2.2), (Hildreth, 1984; Wahl & Simpson,

1990). As pointed out by Weng (1993) the extraction of the zero crossing contours of the Laplacian-of-Gaussian filter is equivalent to the extracting contours of the phase images corresponding to phases $\frac{\pi}{2}$ and $-\frac{\pi}{2}$, see Figure 2.4. This is not surprising as the even component of the quadrature filters, Figure 2.3(a), resemble the "Mexican hat" shape of the Laplacian-of-Gaussian.

3.3 Local Motion Estimation Techniques

Most previous approaches to the estimation of image velocity can be categorized into of one the following four groups

- region based matching
- gradient based
- frequency based
- feature tracking

All these methods are based on local estimates of image translation. We will briefly review these four basic techniques.

3.3.1 Region Based Matching Techniques

Region Based Matching Techniques assume conservation of the local distribution of intensity. The velocity at a given point is found by determining the displacement yielding the best match between an image region in one frame and regions in subsequent frames. The best match is given by some match measure or metric, which can defined in many different ways as described by Rosenfeld & Kak (1976). We could for instance use the sum of squared distances of the intensity values within a region, the sum of absolute values, the maximum absolute difference, etc. In the following we will use the cross correlation.

Consider two real, discrete two dimensional functions, $Z_1(\boldsymbol{x})$ and $Z_2(\boldsymbol{x})$. The autocovariance and the cross covariance functions are given by

$$
\begin{aligned}
C_{11}(\boldsymbol{x},\boldsymbol{\xi}) &= E\{(Z_1(\boldsymbol{x})-\eta_1(\boldsymbol{x}))(Z_1(\boldsymbol{x}+\boldsymbol{\xi})-\eta_2(\boldsymbol{x}+\boldsymbol{\xi}))\}\\
C_{12}(\boldsymbol{x},\boldsymbol{\xi}) &= E\{(Z_1(\boldsymbol{x})-\eta_1(\boldsymbol{x}))(Z_2(\boldsymbol{x}+\boldsymbol{\xi})-\eta_2(\boldsymbol{x}+\boldsymbol{\xi}))\},
\end{aligned}
$$

respectively, where E denotes expectation, $\boldsymbol{\xi}$ is a displacement vector, and $\eta_1(\boldsymbol{x})$ and $\eta_2(\boldsymbol{x})$ are the (weighted) function means. The function variances are the zero lag autocovariances. The cross correlation function is then defined as

$$
r_{12}(\boldsymbol{x},\boldsymbol{\xi}) = \frac{C_{12}(\boldsymbol{x},\boldsymbol{\xi})}{\sqrt{C_{11}(\boldsymbol{x},\mathbf{0})C_{22}(\boldsymbol{x}+\boldsymbol{\xi},\mathbf{0})}} \tag{3.10}
$$

The cross correlation function gives us a measure of similarity between the two functions at given lags. Let the two signals be the image intensities, $I(\boldsymbol{x},t_1)$ and $I(\boldsymbol{x},t_2)$, at times t_1 and t_2. Given a two dimensional windowing function, $\omega(\boldsymbol{x})$, centered at $(0,0)$, the estimated weighted cross correlation function of the intensity field at times t_1 and t_2 is

$$
\hat{r}_{12}(\boldsymbol{x},\boldsymbol{\xi}) = \frac{\sum_j \omega(\boldsymbol{x}-\boldsymbol{x}_j)(I(\boldsymbol{x}_j,t_1)-\hat{\eta}(\boldsymbol{x}_j,t_1))(I(\boldsymbol{x}_j+\boldsymbol{\xi},t_2)-\hat{\eta}(\boldsymbol{x}_j+\boldsymbol{\xi},t_2))}{\left(\sum_j \omega(\boldsymbol{x}-\boldsymbol{x}_j)(I(\boldsymbol{x}_j,t_1)-\hat{\eta}(\boldsymbol{x}_j,t_1))^2 \sum_j \omega(\boldsymbol{x}+\boldsymbol{\xi}-\boldsymbol{x}_j)(I(\boldsymbol{x}_j+\boldsymbol{\xi},t_2)-\hat{\eta}(\boldsymbol{x}_j+\boldsymbol{\xi},t_2))^2\right)^{\frac{1}{2}}}, \tag{3.11}
$$

where the sums are taken over all spatial positions, and $\hat{\eta}(\boldsymbol{x},t)$ is the local spatial intensity field mean at time t

$$
\hat{\eta}(\boldsymbol{x}_j,t) = \frac{1}{\sum_j \omega(\boldsymbol{x}-\boldsymbol{x}_j)} \sum_j \omega(\boldsymbol{x}-\boldsymbol{x}_j) I(\boldsymbol{x}_j,t) \tag{3.12}
$$

The region in the first image to be matched to the second image is given by the windowing function, $\omega(\boldsymbol{x})$. To minimize the computational load the match

measure is normally only calculated at lags smaller than an expected maximum displacement. The maximum cross correlation measure has been used by Wahl & Simpson (1990).

Some may think that the region based matching techniques are inherently based on two images only. Obviously, this is not true as the region being matched, defined by $\omega(\boldsymbol{x})$ in Equation (3.11), without any problem what so ever can be extended to be a function of temporal as well as spatial position. The sums in Equation (3.11) then being over all spatio-temporal positions.

Obviously the aperture problem will arise when the local intensity structure is effectively one-dimensional. In this case the match measure, *in casu* the cross correlation, as a function of the displacement will give rise to a ridge without a clearly defined maximum. In Figure 3.4 we show the behaviour of the cross correlation function in the three neighborhoods marked in Figure 3.2. The surfaces were generated using a 5×5 pattern tile with equal weights.

It is evident that this could cause grave errors if not dealt with properly. The maximum cross correlation in itself enables us to identify the type of neighborhood. In order to deal with this problem we could try to ensure that one-dimensional structures would not occur within our pattern tile. This would normally be impossible, either because no knowledge of the size of structures being observed is available, or because this would violate the assumption of translational velocity. Furthermore large pattern tiles are often considered impractical due the large computational cost associated with them. So what we need is a way to quantify the nature of the neighborhood, or more precisely a measure of the directional confidence in the estimated displacement.

Anandan (1989) suggested that we use confidence measures based on the local curvature of the match measure surface at the point of the best match. It is evident from Figures 3.4(a) - 3.4(c) that the curvature of the cross correlation function along a particular direction is large, respectively small, if the corresponding component of the displacement vector is well defined, respectively undefined. For a formal definition of curvature see Appendix A.

As can be seen in Appendix A the curvature of a surface at a point can be determined given the principal curvatures and the associated principal directions.

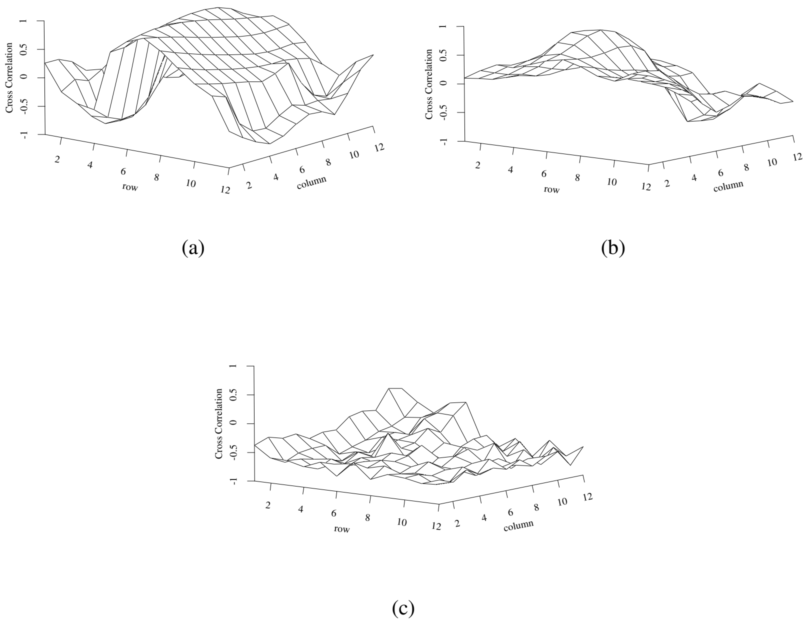


Figure 3.4: The cross correlation functions for the three neighborhoods shown in Figure 3.2. (a) In the line case, the the cross correlation function has a dominant ridge, displacements along the ridge have equally good matches, i.e. due to the aperture problem we can only determine the displacement reliably perpendicular to the ridge (contour). (b) The cross correlation function calculated at the grey level intensity corner on the other hand yields a distinct peak, thus providing us with an unequivocal displacement estimate. (c) In the homogeneous neighborhood the cross correlation function shows little variation, and low correlations, suggesting that no reliable match can be obtained.

	Line	Corner
$\boldsymbol{\mu}$	(-0.42,1.91)	(1.22,0.31)
κ_1	0.48	0.26
κ_2	0.0043	0.079
$\boldsymbol{n}_1$	(0.99,-0.11)	(0.99, 0.08)
$\boldsymbol{n}_2$	(0.11,0.99)	(-0.08, 0.99)

Table 3.1: The estimates of the displacements, the principal curvatures and the principal directions are based on a least squares fit of a quadratic surface to a 5×5 segment of the respective cross correlation functions. The vectors are in a row-column coordinate system, with origin in the upper left corner of the image.

The first and second principal directions are those two (perpendicular) directions at which the normal curvatures are maximum and minimum, respectively, and the principal curvatures are the curvatures in those directions. It is evident that at a point along an edge in the image the maximum principal direction will be oriented in a direction normal to the edge, and the minimum principal direction will be oriented along the edge. Furthermore the maximum curvature will be large and the minimum curvature will be small. At points in homogeneous areas both principal curvatures will be small, and at points on contours with high curvature, e.g. intensity corners, both principal curvatures will be high. So a measure of error between the true flow, $\boldsymbol{u}$, and our observation, $\boldsymbol{\mu}$, might be

$$E = \kappa_1((\boldsymbol{u} - \mu) \cdot \boldsymbol{n}_1)^2 + \kappa_2((\boldsymbol{u} - \mu) \cdot \boldsymbol{n}_2)^2 \qquad (3.13)$$

where κ_1 and κ_2 denote the principal curvatures, and $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ the principal directions.

Sub-pixel accuracy can be achieved by interpolation in the cross correlation function or by fitting the observed correlations to an appropriate model, e.g. a quadratic function. For the two structured neighborhoods of Figure 3.2 the estimates of the displacement, the principal curvatures and principal directions are summarized in Table 3.1. In the line-case we observe high curvature, 0.48, in the direction perpendicular to the contour, $(0.99, -0.11)$, and low curvature, 0.0043, along the curvature. In the corner-case both principal curvatures are relatively high, 0.26, and 0.079.

Singh (1991) argued that the use of the principal curvatures at the maximum of the correlation surface is not an optimal approach. If for instance more than one maximum occurs within the search window Anandan's algorithm will assign high confidence to the largest of the two maxima, if the principal curvatures at this point are high. Using an error function based on the sum of squared differences (SSD) he defined the following (mis)match measure, which he termed the response distribution

$$R(\boldsymbol{x}, \boldsymbol{\xi}) = \exp\left(-k \sum_j \omega(\boldsymbol{x} - \boldsymbol{x}_j)(I(\boldsymbol{x}_j, t_1) - I(\boldsymbol{x}_j + \boldsymbol{\xi}, t_2))^2\right) \tag{3.14}$$

Interpreting the response distribution as a frequency distribution in velocity, ξ, space, i.e the response in a point depicting the likelihood of the corresponding velocity, he estimated the true velocity and a covariance matrix based on the entire match area given by ω.

3.3.2 Gradient Based Techniques

Techniques based on establishing a relation between the velocity and the spatio-temporal gradients of the image intensity are called gradient based techniques. These techniques generally assume conservation of image intensity, that is the observed intensity of an object point is constant over time. Letting the intensity at point $(\boldsymbol{x}, t)^T$ in space-time be denoted $I(\boldsymbol{x}, t)$, and the image velocity at that same point be denoted $\boldsymbol{u}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t))^T$, or for short I and $\boldsymbol{u} = (u, v)^T$, then following Horn & Schunk (1981)

$$I(\boldsymbol{x}, t) = I(\boldsymbol{x} + \boldsymbol{\delta x}, t + \delta t) \tag{3.15}$$

Expanding the right side of Equation (3.15) as a Taylor series about $(\boldsymbol{x}, t)^T$, and discarding terms higher than first order, we find that

$$\begin{aligned} I(\boldsymbol{x} + \boldsymbol{\delta x}, t + \delta t) &= \\ & I(\boldsymbol{x}, t) + \tfrac{\partial I(\boldsymbol{x},t)}{\partial x}\delta x + \tfrac{\partial I(\boldsymbol{x},t)}{\partial y}\delta y + \tfrac{\partial I(\boldsymbol{x},t)}{\partial t}\delta t \end{aligned} \tag{3.16}$$

By combining Equations (3.15) and (3.16), dividing through with δt, and going to the limit, we find

$$I_x u + I_y v + I_t = 0, \tag{3.17}$$

where $I_x = \frac{\partial I(\boldsymbol{x},t)}{\partial x}$, $I_y = \frac{\partial I(\boldsymbol{x},t)}{\partial y}$, and $I_t = \frac{\partial I(\boldsymbol{x},t)}{\partial t}$. As Equation (3.17) constrains the velocity, $\boldsymbol{u}$, to a line in (u, v)-space, it is often referred to as the gradient constraint equation, or motion constraint line. Directly from Equation (3.17) only the normal velocity, $\boldsymbol{\mu} = (\mu, \nu)^T$, can be determined. This is given by

$$(\mu, \nu) = \frac{-I_t}{I_x^2 + I_y^2}(I_x, I_y) \tag{3.18}$$

The motion constraint line and the normal velocity are illustrated in Figure 3.5.

So further constraints are necessary. Almost all approaches couple the gradient constraint with an assumption that nearby points move in a similar manner. According to Kearney, Thompson, & Boley (1987) this combination is in general accomplished in one of three ways. *Global optimization* techniques which minimize an error function based on the gradient constraint and an assumption of global or piecewise smoothness over the entire image, (Horn & Schunk, 1981; Nagel & Enkelmann, 1986). We will return to these global smoothness approaches in Chapter 5. *Local Optimization* techniques that combine a set of constraint lines to obtain a solution for optical flow, (Fennema & Thompson, 1979; Nagel, 1983; Kearney et al., 1987; Paquin & Dubois, 1983; Bigün, 1988; Aisbett, 1989; Schunk, 1989).

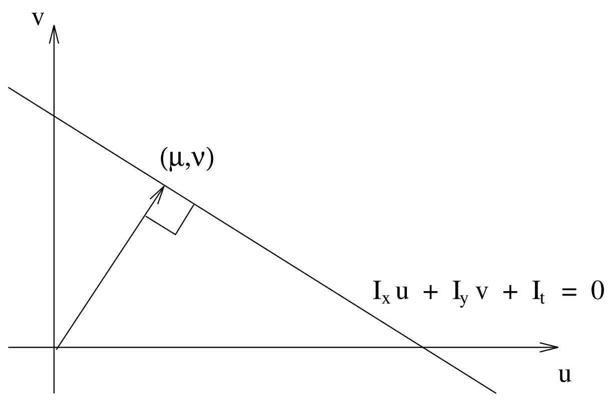
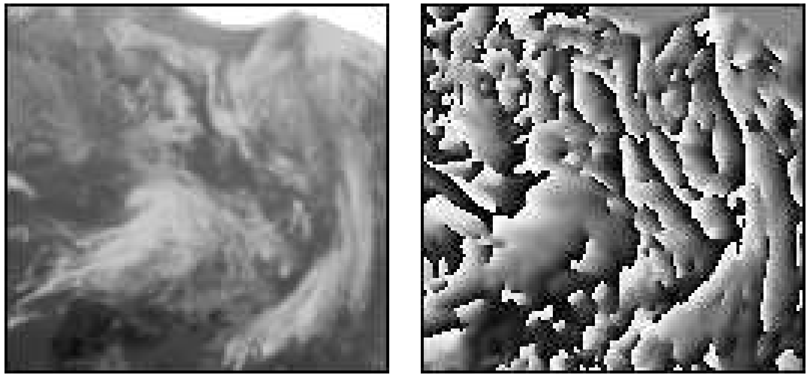


Figure 3.5: The Gradient Constraint Equation (3.17) constrains the velocity to a line in (u, v)-space. The normal velocity (μ, ν) given by Equation (3.18) is perpendicular to the motion constraint line.

Some local optimization techniques produce additional constraint lines using higher order derivatives, (Arnspang, 1988; Uras, Girosi, Verri, & Torre, 1988). If we assume that the image motion is translational, i.e (u, v) constant in a neighborhood of $(\boldsymbol{x}, t)$, Equation (3.17) gives the following constraints by differentiation

$$\begin{bmatrix} I_{xx} & I_{yx} \\ I_{xy} & I_{yy} \end{bmatrix} \boldsymbol{u} + \begin{pmatrix} I_{tx} \\ I_{ty} \end{pmatrix} = \mathbf{0}, \tag{3.19}$$

where $I_{xx} = \frac{\partial^2 I(\boldsymbol{x},t)}{\partial x^2}$, $I_{xy} = \frac{\partial^2 I(\boldsymbol{x},t)}{\partial x \partial y}$, $I_{yx} = \frac{\partial^2 I(\boldsymbol{x},t)}{\partial y \partial x}$, and $I_{yy} = \frac{\partial^2 I(\boldsymbol{x},t)}{\partial y^2}$. If the neighborhood shows significant second order spatial intensity variation, e.g at grey level corners as in Figure 3.2(c), Equation (3.19) will constrain both components of the velocity. Moreover the second order derivatives can be used to characterize the degree of two-dimensionality of the local neighborhood, thus indicating whether the velocity estimate is constrained in two dimensions, or just in one dimension, as in the case shown in Figure 3.2(b). This is analogous to the use of curvature of the match measure surfaces in the region based matching techniques to calculate a directional confidence in the flow estimate.



(a) (b)

Figure 3.6: This illustrates phase wrap around. (a) This is the ninth images in the Meteosat sequence. The original image has had its resolution reduced a factor 4. (b) This is the Fourier phase calculated from a spatio-temporal quadrature filter with the spatio-temporal orientation $(2.0, 0.0, 1+\sqrt{5})$. Note that the phase wraps around, black and white correspond to almost equal shapes.

Another method to introduce further constraints is to combine a set of motion constraint lines, Equation (3.17), in a small neighborhood, for example to a polynomial in (u, v) by a least squares fit, (Kearney et al., 1987; Paquin & Dubois, 1983; Aisbett, 1989; Bigün, 1988), or by clustering of lines to avoid blurring in the vicinity of motion boundaries, (Fennema & Thompson, 1979; Schunk, 1989).

When applying the gradient based methods to phase images, computed as described in Section 2.3 problems arise owing to the fact that it is phase modulo 2π that we assume to be conserved. In Figure 3.6 this effect is illustrated. Very bright and very dark values correspond to phases a little less than π and a little larger than $-\pi$, respectively. The shapes however are close.

We apply a set of orientation and scale specific quadrature filters each returning a complex valued output

$$R(\boldsymbol{x}, t) = \rho(\boldsymbol{x}, t) \exp(j\phi(\boldsymbol{x}, t)) \tag{3.20}$$

$\rho(\boldsymbol{x}, t)$ and $\phi(\boldsymbol{x}, t)$ denote the magnitude and phase of the filter respectively. The velocity in the direction of the iso-phase contour, i.e. the normal velocity, is in accordance with Equation (3.17) given by

$$\boldsymbol{\mu} = \frac{-\phi_t(\boldsymbol{x}, t)}{\|\nabla\phi(\boldsymbol{x}, t)\|^2} \nabla\phi(\boldsymbol{x}, t) \tag{3.21}$$

where $\nabla\phi(\boldsymbol{x}, t) = (\phi_x(\boldsymbol{x}, t), \phi_y(\boldsymbol{x}, t))$. The phase derivatives are calculated as follows

$$\phi_x(\boldsymbol{x}, t) = \frac{\mathrm{Im}(R^*(\boldsymbol{x}, t) R_x(\boldsymbol{x}, t))}{\|R(\boldsymbol{x}, t)\|^2} \tag{3.22}$$

thus avoiding problems concerning phase wrap around.

The outputs from the set of filters are combined to yield the local velocity estimates in the same way as several motion constraint lines can be combined as described in the previous sections. The magnitude of each filter output is used as an weight or certainty of the corresponding velocity.

3.3.3 Methods based on Frequency Analysis

As motion estimation in image sequence can be viewed as identification of patterns repeating themselves over time it is natural to try to describe the motion analysis in the Fourier domain.

Consider a one dimensional intensity pattern, e.g. an line, translating through time. In the spatio-temporal domain this corresponds to a neighborhood consisting of iso grey level planes. Let these planes be given by their unit normal

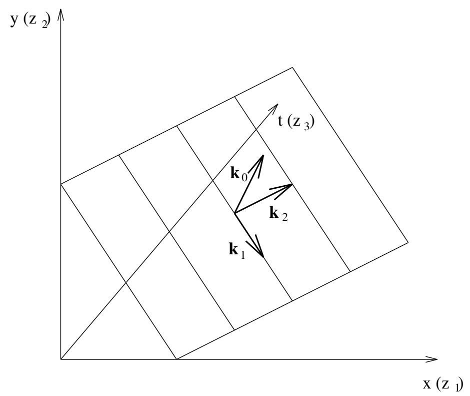


Figure 3.7: $\boldsymbol{k}_0$ is the unit normal vector of the plane generated by the translating line. The direction of normal flow is found as the projection of $\boldsymbol{k}_2$ onto the (z_1, z_2)-plane.

vector, $\boldsymbol{k}_0 = (k_{01}, k_{02}, k_{03})^T$, we will refer to this vector as the spatio-temporal orientation of the local neighborhood. For notational convenience we will use $\boldsymbol{z} = (z_1, z_2, z_3)^T$ for the spatio-temporal coordinates $(\boldsymbol{x}^T, t)^T$. The local intensity pattern is then given by

$$I(\boldsymbol{z}) = J(\boldsymbol{k}_0^T \boldsymbol{z}) \tag{3.23}$$

The correspondence between the spatio-temporal orientation and the normal velocity of the line is illustrated in Figure 3.7. The normal flow $\boldsymbol{\mu}$ is found by taking the outer product of the orientation vector, $\boldsymbol{k}_0$, and the projection onto the (z_1, z_2)-plane of the orientation vector rotated 90°, $\boldsymbol{k}_1$.

$$\boldsymbol{k}_2 = (k_{01}, k_{02}, k_{03})^T \times \left(-\frac{k_{02}}{\sqrt{k_{01}^2 + k_{02}^2}}, \frac{k_{01}}{\sqrt{k_{01}^2 + k_{02}^2}}, 0 \right)^T$$

$$= \left(-\frac{k_{03}k_{01}}{\sqrt{k_{01}^2 + k_{02}^2}}, -\frac{k_{03}k_{02}}{\sqrt{k_{01}^2 + k_{02}^2}}, \sqrt{k_{01}^2 + k_{02}^2} \right)^T$$

The length is determined by setting the temporal coordinate equal to 1, thus

$$\boldsymbol{\mu} = \frac{-k_{03}}{k_{01}^2 + k_{02}^2}(k_{01}, k_{02})^T \tag{3.24}$$

Using the rotated orthonormal coordinate system

$$\boldsymbol{\zeta} = \begin{bmatrix} k_{01} & k_{02} & k_{03} \\ -\frac{k_{02}}{\sqrt{k_{01}^2+k_{02}^2}} & \frac{k_{01}}{\sqrt{k_{01}^2+k_{02}^2}} & 0 \\ -\frac{k_{01}k_{03}}{\sqrt{k_{01}^2+k_{02}^2}} & -\frac{k_{02}k_{03}}{\sqrt{k_{01}^2+k_{02}^2}} & \sqrt{k_{01}^2+k_{02}^2} \end{bmatrix} \boldsymbol{z} = \begin{pmatrix} \boldsymbol{k}_0^T \\ \boldsymbol{k}_1^T \\ \boldsymbol{k}_2^T \end{pmatrix} \boldsymbol{z} \tag{3.25}$$

we find the Fourier transform of $I(\boldsymbol{z})$ to be

$$\begin{aligned} \mathcal{I}(\boldsymbol{k}) &= \int I(\boldsymbol{z}) \exp(-j(\boldsymbol{k}^T\boldsymbol{z}))\, d\boldsymbol{z} \\ &= \int J(\zeta_1) \exp(-j(\boldsymbol{k}^T\boldsymbol{k}_0\zeta_1 + \boldsymbol{k}^T\boldsymbol{k}_1\zeta_2 + \boldsymbol{k}^T\boldsymbol{k}_2\zeta_3))\, d\boldsymbol{\zeta} \\ &= \mathcal{J}(\boldsymbol{k}^T\boldsymbol{k}_0)\, \delta(\boldsymbol{k}^T\boldsymbol{k}_1)\, \delta(\boldsymbol{k}^T\boldsymbol{k}_2) \end{aligned} \tag{3.26}$$

where $\mathcal{J}$ is the Fourier transform of J, and δ is Dirac's delta. From Equation (3.26) it follows that all the non zero Fourier coefficients are concentrated on a line through the origin, the line is the intersection of the two planes, $k_{01}k_1 - k_{02}k_2 = 0$, and $k_{01}k_{03}k_1 + k_{02}k_{03}k_2 = (k_{01}^2 + k_{02}^2)k_3$.

Similarly, we can develop an expression for the Fourier domain in the case of a two dimensional spatial pattern, $J(z_1, z_2)$, translating with velocity $\boldsymbol{u} = (u, v)^T$, i.e the image sequence is given by

$$I(\boldsymbol{z}) = J(z_1 - uz_3, z_2 - vz_3) \tag{3.27}$$

The Fourier transform of $I(\boldsymbol{z})$ is

$$\begin{aligned} \mathcal{I}(\boldsymbol{k}) &= \int J(z_1 - uz_3, z_2 - vz_3) \exp(-j(\boldsymbol{z}^T\boldsymbol{k}))\, d\boldsymbol{z} \\ &= \mathcal{J}(k_1, k_2) \int \exp(-j((uz_3, vz_3)(k_1, k_2)^T) \exp(-jz_3k_3)\, dz_3 \\ &= \mathcal{J}(k_1, k_2)\, \delta(uk_1 + vk_2 + k_3) \end{aligned} \tag{3.28}$$

where $\mathcal{J}(k_1, k_2)$ is the Fourier transform of $J(z_1, z_2)$, and δ is Dirac's delta. From Equation (3.28) we see that the non zero Fourier coefficients are concentrated on a the plane given by $uk_1 + vk_2 + k_3 = 0$, i.e. the plane through the origin given by its normal vector $(\boldsymbol{u}^T, 1)$.

Interpretation of the Fourier Domain

From Equation (3.26) it follows that the detection of the normal flow can be transformed into the detection of a line in the Fourier domain. Viewing the Fourier coefficients as a mass distribution Bigün (1988) calculated the inertia (or variance) with respect to the axis given by the three dimensional unit vector $\tilde{\boldsymbol{k}}_0$

$$V(\tilde{\boldsymbol{k}}_0) = \int d^2(\boldsymbol{k}, \tilde{\boldsymbol{k}}_0)|\mathcal{I}(\boldsymbol{k})|^2 d\boldsymbol{k} \tag{3.29}$$

The problem then is to find the direction yielding the minimum inertia. The Euclidean distance between a point $\boldsymbol{k}$ and the axis given by $\tilde{\boldsymbol{k}}_0$ is

$$d^2(\boldsymbol{k}, \tilde{\boldsymbol{k}}_0) = ||\boldsymbol{k} - (\boldsymbol{k}^T\tilde{\boldsymbol{k}}_0)\tilde{\boldsymbol{k}}_0||^2 = \tilde{\boldsymbol{k}}_0^T(\boldsymbol{k}^T\boldsymbol{I}\boldsymbol{k} - \boldsymbol{k}\boldsymbol{k}^T)\tilde{\boldsymbol{k}}_0 \tag{3.30}$$

Thus

$$V(\tilde{\boldsymbol{k}}_0) = \tilde{\boldsymbol{k}}_0^T(\mathrm{trace}(\boldsymbol{A})\boldsymbol{I} - \boldsymbol{A})\tilde{\boldsymbol{k}}_0 \tag{3.31}$$

where the elements of the matrix are given by

$$a_{ij} = \int k_i k_j |\mathcal{I}(\boldsymbol{k})|^2 d\boldsymbol{k} = \frac{1}{4\pi^2} \int \frac{\partial I}{\partial z_i} \frac{\partial I}{\partial z_j} d\boldsymbol{z} \tag{3.32}$$

The second identity is obtained using Parseval's Theorem and the fact that differentiation in the spatial domain corresponds to multiplication by the respectively coordinate in the Fourier domain. Now it is evident that the minimum inertia given by the eigenvector corresponding to the largest eigenvalue for $\boldsymbol{A}$[1]. Bigün (1988) and Hansen (1991) developed confidence measures for the optimality of the orientation estimate based on the relation between the eigenvalues.

Obviously, in the case of a pure translation of a perfectly one dimensional signal the matrix $\boldsymbol{A}$ will have only one non zero eigenvalue, and the corresponding eigenvector will be $(\frac{\partial I}{\partial z_1}, \frac{\partial I}{\partial z_2}, \frac{\partial I}{\partial z_3})$. This in combination with Equation (3.24), give us a direct correspondence to the motion constraint Equation (3.17) derived using the gradient based techniques. Furthermore, computing the weighted sums of the products of the derivatives in a small neighborhood to obtain $\boldsymbol{A}$ corresponds exactly to the combination of constraint lines in a small neighborhood as described in Section 3.3.2. Similarity also exist with the methods for corner detection by Noble (1988), described in Section 3.2.1.

The frequency analysis based method suggested by Bigün (1988), however, suffers from a serious drawback, namely phase dependency. When sampling the inertia tensor of the Fourier domain, Equation (3.29), i.e. obtaining a description of the local neighborhood, we should use filters that collect the energy from Fourier domain irrespectively of this energy originating from a line, an edge, or any other of the shapes shown in Figure 2.4.

Knutsson (1989) developed a method with the purpose of orientation estimation based the quadrature filters of Section 2.3. These filters have exactly the

[1]Let the eigenvalues of $\boldsymbol{A}$ be labeled $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, and the corresponding eigenvectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$. Since $\boldsymbol{A}\boldsymbol{u}_j = \lambda_j \boldsymbol{u}_j$ we obtain

$$(\text{trace}(\boldsymbol{A})\boldsymbol{I} - \boldsymbol{A})\boldsymbol{u}_j = \tilde{\lambda}_j \boldsymbol{u}_j, \text{with} \qquad \tilde{\lambda}_j = (\lambda_1 + \lambda_2 + \lambda_3) - \lambda_j \tag{3.33}$$

From this equation it is evident that the matrices $\boldsymbol{A}$ and trace$(\boldsymbol{A})\boldsymbol{I} - \boldsymbol{A}$ share the same eigenvectors. The relationship between the eigenvalues stem from trace$(\boldsymbol{A}) = \lambda_1 + \lambda_2 + \lambda_3$.

property of phase independence. According to Knutsson we need at least 5 quadrature filter pairs symmetrically distributed over the orientation space in the three dimensional case[2]. In order obtain a symmetrical distribution of orientation we need 6 filters, as the only possible number of filters is given by half the number of vertices or faces of a diametrically symmetric regular polyhedron, leaving only the numbers 3, 4, 6, and 10.

The center axes of a realization of the 6 filters are given by these unit vectors

$$\begin{array}{rcllll} \boldsymbol{k}_1 &=& c\,(& a, & 0, & b\ )^T \\ \boldsymbol{k}_2 &=& c\,(& -a, & 0, & b\ )^T \\ \boldsymbol{k}_3 &=& c\,(& b, & a, & 0\ )^T \\ \boldsymbol{k}_4 &=& c\,(& b, & -a, & 0\ )^T \\ \boldsymbol{k}_5 &=& c\,(& 0, & b, & a\ )^T \\ \boldsymbol{k}_6 &=& c\,(& 0, & b, & -a\ )^T \end{array} \tag{3.34}$$

where $a = 2$, $b = 1 + \sqrt{5}$ and $c = (10 + 2\sqrt{5})^{-\frac{1}{2}}$. Using 6 quadrature filters with a radial function varying as cosine squared ($A = 1$ in Equation (2.6)) the estimator for the orientation tensor is given by

$$\begin{aligned} \hat{\boldsymbol{T}} &= \sum_p q_p(\boldsymbol{K}_p - \frac{1}{5}\boldsymbol{I}) \\ \boldsymbol{K}_p &= \boldsymbol{k}_p \boldsymbol{k}_p^T \end{aligned} \tag{3.35}$$

where q_p is the output from the quadrature filter p. $\boldsymbol{K}_p$ is the orientation tensor associated with the quadrature filter p, $\boldsymbol{k}_p$ is the unit vector defining the center axis of pth filter. $\boldsymbol{I}$ is the identity tensor.

Other noteworthy implementations of a frequency based motion estimation includes (Adelson & Bergen, 1985; Heeger, 1987).

[2]Consider the case of having 2^{N-1} quadrature filters, having symmetry axes passing through the corners of a cube in N dimensions. Consider further the contribution to the filters from frequencies on a line through the center of two opposing cube faces. Since the angle between the line and any filter axis is the same every filter will yield the same output. Consequently using this set of filters we can not determine which set of faces the line is passing through. Consequently we have to use more than 2^{N-1} filters.

3.3.4 Feature Tracking

Having detected a number of image features, i.e points and/or curves, in subsequent images the task is now to establish a correspondence and thereby find the motion of the features.

Waxman, Wu, & Bergholm, (Singh, 1991), argued that the gradient based techniques applied to the points of zero crossings of the Laplacian-of-Gaussian would fail because neither the intensity nor the intensity derivatives are truly conserved. Having extracted the contours, i.e. having generated a binary edge image, they convolved these binary images with a spatio-temporal Gaussian, thus creating what they called activation profiles. They then applied a gradient based technique to these activation profiles.

A widely used technique is that of active contour models or snakes introduced by Kass, Witkin, & Terzopoulos (1988). From a crude initialization the snake evolves in a energy minimization scheme under the action of so called internal and external forces. Representing the snake position by the following parameterization $\boldsymbol{v}(s) = (x(s), y(s)), \quad s \in [0, 1]$ we can write the snake energy functional as

$$E_{\text{snake}} = \int_0^1 E_{\text{int}}(\boldsymbol{v}(s)) + E_{\text{image}}(\boldsymbol{v}(s)) + E_{\text{con}}(\boldsymbol{v}(s))\, ds \tag{3.36}$$

where E_{int} represent the internal forces of the snake. These forces impose a piecewise smoothness, and are usually defined as a weighted sum of the first and second derivative of the position.

$$E_{\text{int}}(\boldsymbol{v}(s)) = \alpha(s) \left\| \frac{\mathrm{d}\boldsymbol{v}(s)}{\mathrm{d}s} \right\|^2 + \beta(s) \left\| \frac{\mathrm{d}^2\boldsymbol{v}(s)}{\mathrm{d}s^2} \right\|^2 \tag{3.37}$$

The image image forces can be expressed as the weighted combination of three terms

$$E_{\text{ext}} = w_{\text{line}} E_{\text{line}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{term}} E_{\text{term}} \tag{3.38}$$

The edge, line, and termination forces are forces that aim to attract the snake to certain features of the image, e.g lines, edges , and points. These forces might be constructed from any of the point and curve measures described in Section 3.2. Kass et al. suggested a functional $E_{\text{edge}} = -\|\nabla I(x, y)\|$, that would attract the snake to contours with large image gradients, as well as one based on attraction to the zero crossings of the Laplacian-of-Gaussian, $E_{\text{line}} = -\|\nabla^2 G(x, y) * I(x, y)\|^2$. In order to attract the snake to terminating line segments and corners they suggested a functional based on curvature, i.e setting E_{term} equal to the curvature in the direction perpendicular to the gradient as given by Equation (3.7). In their derivation of the curvature, however, they made an error, see equation (8) on page 327 of (Kass et al., 1988).

The last term of the snake forces is the external constraint functional. This functional specifies external forces attacking the snake. In particular a snake might be attached by a spring to a fixed point or to another snake. Such a spring between the points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ can be created by the term, $E_{\text{con}} = -k\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$.

The energy minimization is then performed in a stochastic relaxation scheme. In the application of snakes to motion estimation Kass et al. (1988) used the contour found in the previous image as initialization for the current image. No inter frame constraints where used. Inter frame constraints imposing continuity over time may improve the convergence rate and robustness of the results.

Other implementations of this algorithm include Herlin & Ayache (1992)

Deriche & Faugeras (1990) described a Kalman filtering based approach to line segment tracking. The correspondence between segments in the search areas was done based on Mahalanobis distance between chosen attributes of the segments, e.g. orientation, length.

Finally, distance maps can be used for efficient matching in binary images, as proposed by Barrow et al, (Ragnemalm, 1993). Borgefors (1988) has proposed an efficient algorithm using pyramid structures.

The general technique for matching two binary images, containing the extracted features, e.g. edges and corners, has been described by Borgefors (1988) as follows. First a distance map is generated for the first image. In the distance map each non object pixel is given a value that is a measure of the distance to the nearest object pixel, i.e edge. The object pixels get the value zero. Now, at every position in the binary second image a template is matched to the distance map of the first image. The match measure is a an average of the distances from the distance map corresponding to the edge positions in the template. The smoothness of the distance map allows a relatively fast hill climbing search for the optimal match, (Ragnemalm, 1993).

Chapter 4

Examples of Motion Measurement

In this chapter we will present three motion estimation techniques. First an extension of an algorithm based on the local spatio-temporal orientation description suggested by Knutsson (1989). We propose a continuous measure of directional certainty of the extracted flow.

Second, we will use conservation of the local Fourier phase in a number of directions at each point in the image sequence to estimate the image flow. Furthermore we will extract a directional confidence in the estimated flow.

In the third example we will describe how the Euclidean distance transform can be used to match points corresponding to different time instances of particles flowing in a fluid. In this special case a the imagery consists of a double exposed image.

4.1 A Method Based on 3-D Orientation Estimation

The following method for extracting information about the motion in a spatio temporal neighborhood is based on the characterization of this neighborhood by a orientation tensor suggested by Knutsson, and described in Section 3.3.3. As with Bigün's approach the orientation tensor can be evaluated by an eigenvalue analysis. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ denote the eigenvalues of $\hat{\boldsymbol{T}}$, and let $\boldsymbol{e}_i = (e_{i1}, e_{i2}, e_{i3})^T$ be the eigenvector corresponding to λ_i. Haglund (1992) distinguish between the three cases

1. $\lambda_1 > 0; \lambda_2 = \lambda_3 = 0;$
   This case corresponds to a neighborhood that consists of iso-grey level planes, i.e. a one dimensional spatial pattern translating through time.
2. $\lambda_1 = \lambda_2 > 0; \lambda_3 = 0;$
   This case corresponds to a neighborhood that consists of iso-grey level lines, i.e a two dimensional spatial pattern translating through time.
3. $\lambda_1 = \lambda_2 = \lambda_3 > 0;$
   This case corresponds to an isotropic neighborhood, i.e. the energy is homogeneously distributed in the neighborhood as for instance in the case of white noise.

In order to discriminate between these three cases Haglund computed the following "probabilities" for each case

$$
\begin{aligned}
p_1 &= \frac{\lambda_1 - \lambda_2}{\lambda_1} \\
p_2 &= \frac{\lambda_2 - \lambda_3}{\lambda_1} \\
p_3 &= \frac{\lambda_3}{\lambda_1}
\end{aligned}
$$

and subsequently chose the case having the highest probability. In the first case normal flow is calculated using Equation (3.24) as

$$\boldsymbol{\mu} = \frac{-e_{13}}{e_{11}^2 + e_{12}^2}(e_{11}, e_{12})^T \tag{4.1}$$

In the second case the two eigenvectors corresponding to the the two largest eigenvalues are contained in the plane of the nonzero Fourier coefficients. Consequently a normal vector to this plane is the third eigenvector. According to Equation (3.28) we can estimate the true flow by

$$\boldsymbol{u} = \frac{1}{e_{33}}(e_{31}, e_{32})^T \tag{4.2}$$

In the third case no velocity is computed.

4.1.1 Extension to a Continuous Description

We propose a representation of the local velocity along with an estimate of the directional certainty of the velocity.

Evidently an eigenvector corresponding to a non zero eigenvalue constrains the motion in one direction. If we have only one non zero eigenvalue the motion is constrained only in one direction, namely in the direction given by the normal flow, as computed by Equation (4.1). In the case of two non zero eigenvalues, each of the eigenvectors corresponding to these eigenvalues will impose a linear constraint on the motion. The confidence we should have in each of these linear constraints is given by the relation between the eigenvalues. The constriction of the true flow by two linear constraints is illustrated in Figure 4.1

The point of intersection between the two constraint lines given by the normal flows $\boldsymbol{\mu}_1 = (\mu_1, nu_1)^T$, and $\boldsymbol{\mu}_2 = (\mu_2, nu_2)^T$ is

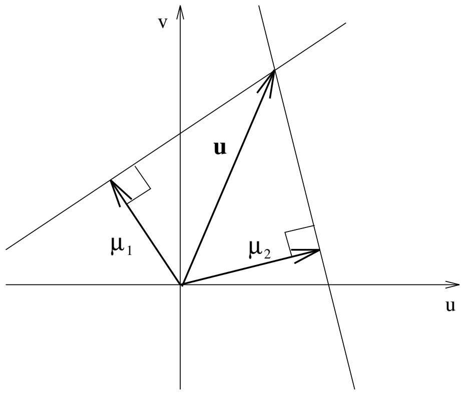


Figure 4.1: The true flow, $\boldsymbol{u}$, is constrained by the linear constraints given by the two normal flows, $\boldsymbol{\mu}_1$, and $\boldsymbol{\mu}_2$.

$$\begin{aligned} \hat{u} &= \frac{\nu_2(\mu_1^2+\nu_1^2) - \nu_1(\mu_2^2+\nu_2^2)}{\mu_1\nu_2 - \mu_2\nu_1} \\ \hat{v} &= \frac{\mu_1(\mu_2^2+\nu_2^2) - \mu_2(\mu_1^2+\nu_1^2)}{\mu_1\nu_2 - \mu_2\nu_1} \end{aligned} \tag{4.3}$$

We can furthermore determine the perpendicular distance of the true flow to either of the constraint lines, this is given by

$$d_i = \|(\boldsymbol{u} - \boldsymbol{\mu}_i)^T \frac{\mu_i}{\|\mu_i\|}\|, \qquad i = 1, 2 \tag{4.4}$$

Pure translatory motion, no noise present, and perfectly designed filters will result in the least eigenvalue being zero. Deviation from these three properties will cause the third eigenvalue to be non zero. This suggests using a confidence measure for each of the linear constraints based on the difference of the corresponding eigenvalue and the least eigenvalue. Furthermore, a normalization of this difference should be made. This is evident as a high noise free high step edge measures the motion just as well as a lower step does. We propose the following confidence measure for each of the linear constraints given by the eigenvectors corresponding to the two largest eigenvalues

$$c_i = \frac{\lambda_i - \lambda_3}{\lambda_1}, \quad i = 1, 2 \tag{4.5}$$

This confidence measure approaches zero when the difference of the corresponding eigenvalue and the least eigenvalue approaches zero, and it attains its maximum value of one, when the least eigenvalue is zero, and the corresponding eigenvalue is the largest, or is equal to the largest eigenvalue, respectively.

The distribution of this estimator for the true flow $\boldsymbol{u} = (u, v)$ will in general by a complex one, cf. Anderson (1984) on the distribution of principal components vectors. Specifically it will not be normal.

A statistical approach to defining the confidence measures above would be through simulations. We will give an example on such a simulation. Let a grey level corner be the blurred product of to Heaviside functions, as the one in Figure 3.3. Let a spatio temporal neighborhood be generated by translating this pattern with constant velocity $(0.5, 0.5)$. And let the three dimensional pattern be further corrupted by independent identically distributed Gaussian noise. We have generated a 1000 realizations of this and computed the true flow as given by Equation (4.3). The estimated true flows are shown in Figure 4.2.

Simulations like this can be used to indicate the reason in using confidence measures as the one in Equation (4.5).

Since the corner was aligned with the coordinate axes we would have expected that the estimates were symmetrically distributed around the diagonal. The reason for this not being so might be due to our disccarding the third eigenvector. The noise free pattern should result in only two non zero eigenvalues. Due to noise, however, we will in general observe that all of the eigenvalues are non zero. In some cases the noise induced eigenvalue will be greater than one or two of the eigenvalues corresponding to the signal. So by always discarding the eigenvector corresponding to the least eigenvalue and always using both the eigenvectors corresponding to the largest eigenvalues, we might have induced this bias.

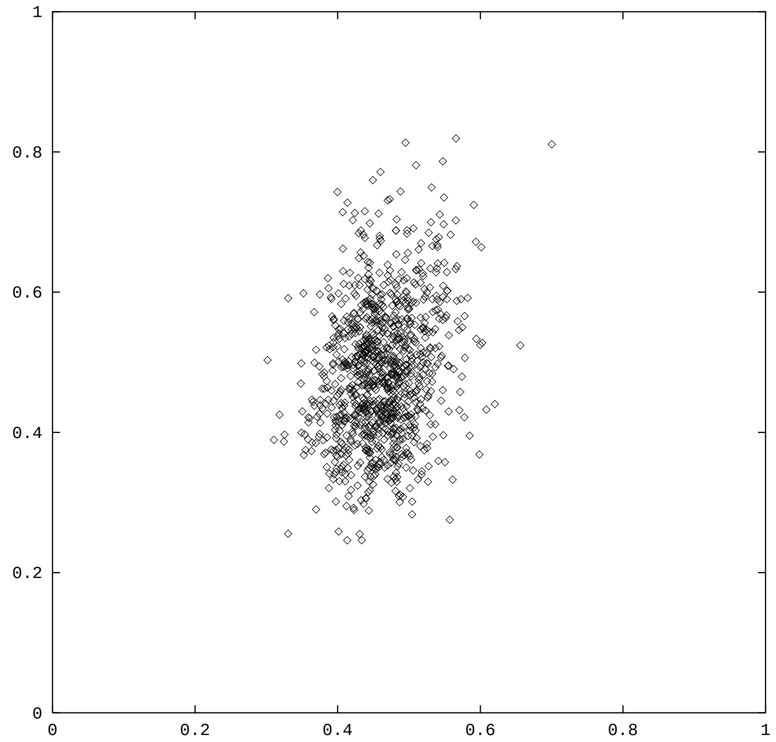


Figure 4.2: In this figure we see a scatterplot of the estimated true flow in 1000 realization of a translating grey level corner.

4.2 A Method Based on Conservation of Fourier Phase

We will now show an implementation of a local motion estimation based on conservation of the Fourier phase in a number of spatio temporal directions as measured by a set of quadrature filters. We will furthermore introduce a measure of confidence in the estimated normal flow corresponding to a certain direction based on the quadrature filter output magnitude.

4.2.1 Fourier Phase as a Motion Invariant

We will first illustrate the Fourier phase as an motion invariant. The Fourier phases are calculated using the quadrature filters described in Section 3.3.3. We have used a set containing six filter pairs given by the directions in Equation (3.34), the filter pairs have a center frequency of 1.1, and a bandwidth of two octaves. We have applied them as $11 \times 11 \times 11$ rotation symmetric spatio temporal convolution kernels.

These quadrature filters have been applied to the Hamburg taxi scene, of which 6 images are shown in Figure 4.3. The output from the first filter is shown in Figure 4.4. The total set can be seen in Appendix B. The filter outputs have been color coded with the filter output magnitude shown as intensity, and the phase on a circular color scale.

From these images we get a very good illustration of the motion invariance of the Fourier phase. The color of specific shapes, i.e. curves, and corners of the moving cars, is seen to be conserved over the images, i.e. over time.

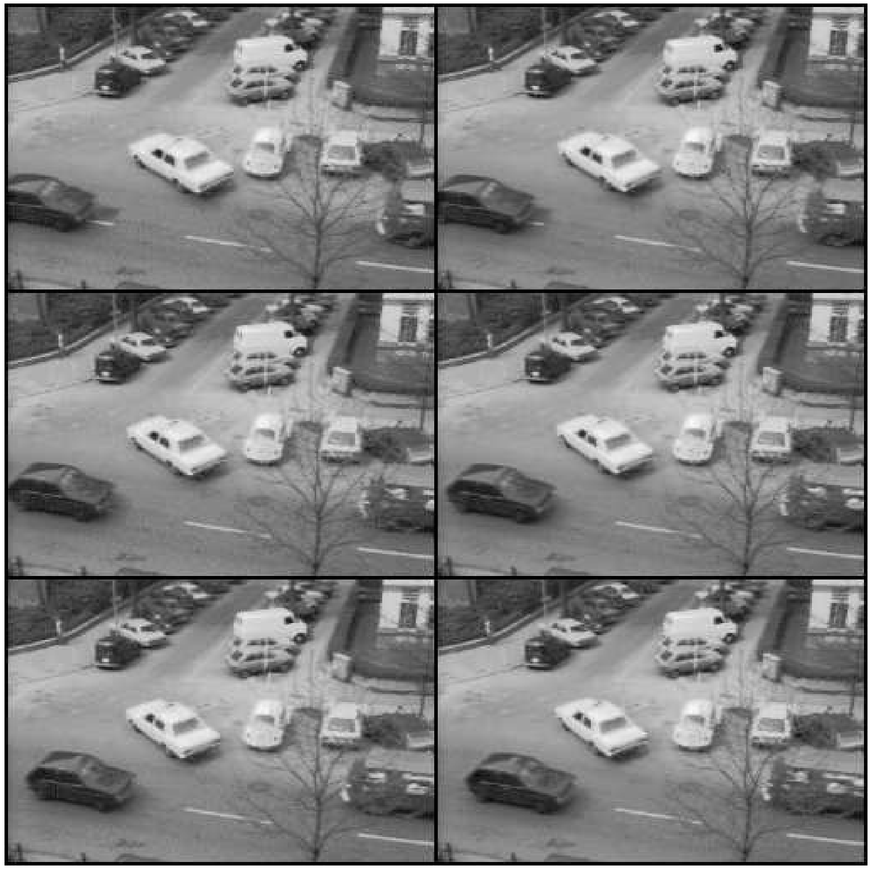


Figure 4.3: In this Figure, 6 images from the Hamburg Taxi scene is shown. In the first column it is image numbers 5, 7, and 9, and in the second column it is image numbers 11, 13, and 15. Note that we have four moving objects. A car going from left to right, a car going from right to left partly shaded by the branches of a tree, a car turning a corner and moving long the diagonal from the lower right to the upper left, and finally a pedestrian on the sidewalk in the left side of the image.

Figure 4.4: Quadrature filter output from filter number 2 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(-2, 0, 1 + \sqrt{5})$, i.e. to movements in the x-t plane going slowly from left to right, as the movement of the van that is partly shielded by tree branches in the right side of the image.

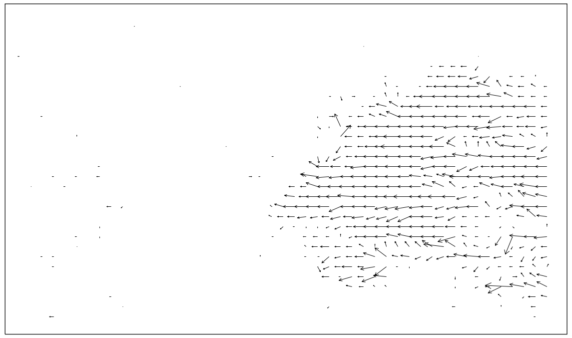


Figure 4.5: This is the normal flow of the van coming in to the image from the right as calculated from the phase from the second quadrature filter at the time corresponding to the seventh image in the sequence. The flow field has been subsampled by a factor two.

4.2.2 Motion Estimation in Phase Images

In order to estimate the motion in the scene we will apply a gradient based technique to each of the phase images. We will use rotation symmetric spatio temporal gradient filter realized as $7 \times 7 \times 7$ convolution kernels. To overcome the problem of phase wrap around we use Equation (3.22) when computing the gradients. Equation (3.21) then gives us the normal flow. In Figure 4.5 the normal flow of the van coming into the image from the right as seen using the phase from the second quadrature filter is shown at the time corresponding to the seventh image in the sequence.

At every point we have a normal flow estimated for each of the 6 quadrature filters. Each of these normal flows corresponds to a constraint line in velocity space. We will estimate the true flow by the point in velocity space that has the least sum of weighted squared distances to these constraint lines. We can formulate the distances from a point in velocity space to the constraint line either in terms of the phase gradients, or in terms of the corresponding normal flow. Letting the gradient of the phase of the pth filter output be denoted

$(\frac{\partial \phi_p}{\partial x}, \frac{\partial \phi_p}{\partial y}, \frac{\partial \phi_p}{\partial t})^T$, and the corresponding normal flow $\boldsymbol{\mu}_p = (\mu_p, \nu_p)$. Then the distance is

$$d = \frac{|\frac{\partial \phi_p}{\partial x} u + \frac{\partial \phi_p}{\partial y} v + \frac{\partial \phi_p}{\partial t}|}{\sqrt{(\frac{\partial \phi_p}{\partial x})^2 + (\frac{\partial \phi_p}{\partial x})^2}}, \tag{4.6}$$

and

$$d = \frac{|(\boldsymbol{u} - \boldsymbol{\mu}_p) \cdot \boldsymbol{\mu}_p|}{||\boldsymbol{\mu}_p||}, \tag{4.7}$$

respectively. Furthermore, we will use a weight function based on the filter output magnitude in order reduce the influence of unreliable velocity measurements. Assuming a non uniform distribution of magnitudes over the filters at a points in structured neighborhood, and a uniform distribution in noisy areas, we will use a function that assigns the magnitude normalized by the total energy at the point, i.e. the sum of all filter outputs. Denoting the filter output magnitudes, ρ_p, we find

$$w_p = \frac{\rho_p}{\sum_i \rho_i} \tag{4.8}$$

In the case of the taxi scene this results in the flow shown in Figure 4.6. As can be seen we have successfully extracted the movements of the three moving cars. It should emphasized that no prior knowledge of the moving objects has been used. It is evident that using prior knowledge, e.g of the shape and/or the type of movement to expect, more information might be extracted. We will illustrate the use of a model of the type of movement to expect in a Bayesian scheme in Chapter 5.

4.3 Measurement of the Flow in Fluids

In this section we will present an algorithm for the estimation of the flow in fluids using the Particle Image Velocimetry (PIV) technique. The algorithm is based on

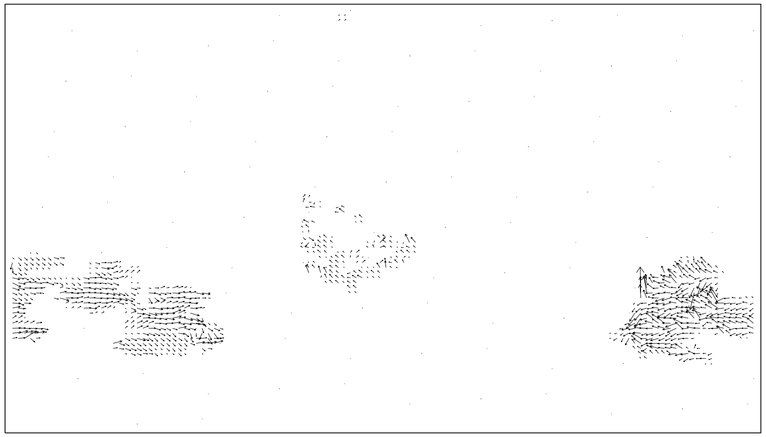


Figure 4.6: This is the result of a gradient based method applied to a set of directional Fourier phases with the purpose of estimating the motion. All computations are local. We have successfully extracted the velocities of the three moving cars.

the distance transform. It differs, however, from previously reported algorithms for matching using distance transforms, (Borgefors, 1988; Ragnemalm, 1993), due to the nature of the imagery.

4.3.1 Particle Image Velocimetry

The Particle Image Velocimetry technique is investigated at the Institute of Physics, and the Department of Fluid Dynamics at the Technical University of Denmark. The technique is developed for measurement of motion in fluids. The basic technique consists of inserting seeding particles in a fluid, then recording a double exposed image of an illuminated plane. Each particle should be exposed twice in the resulting image. Given the time between exposures, and at a match between particles it is possible to compute the velocities. For a description of an experimental setup the reader is referred to Westergaard, Kock, Larsen, & Buchhave (1993). This report further describes an approach to the measurement of the velocities based on the local Fourier transform, which is a particularly slow algorithm. A need for a faster algorithm has been expressed.

4.3.2 Initial Studies

Given two object points of distance $2d$ in two dimensional space, what does the distribution of the distance from any other position, the distance map, to the nearest point look like? In Figure 4.8 we see three iso-distance curves.

For the curves corresponding to distances less than or equal to d, the length or the curves are trivially seen to be $4\pi x$, where x is the actual distance to an object point. For the curves corresponding to distances exceeding d, we have to subtract the length of the circle segments that are not part of the iso-curve. We find the following function, a plot of which is seen in Figure 4.9

$$f(x) = \begin{cases} 4\pi x & \text{for} \quad x \leq d \\ 4x(\pi - \arccos(\frac{d}{x})) & \text{for} \quad x > d \end{cases} \tag{4.9}$$

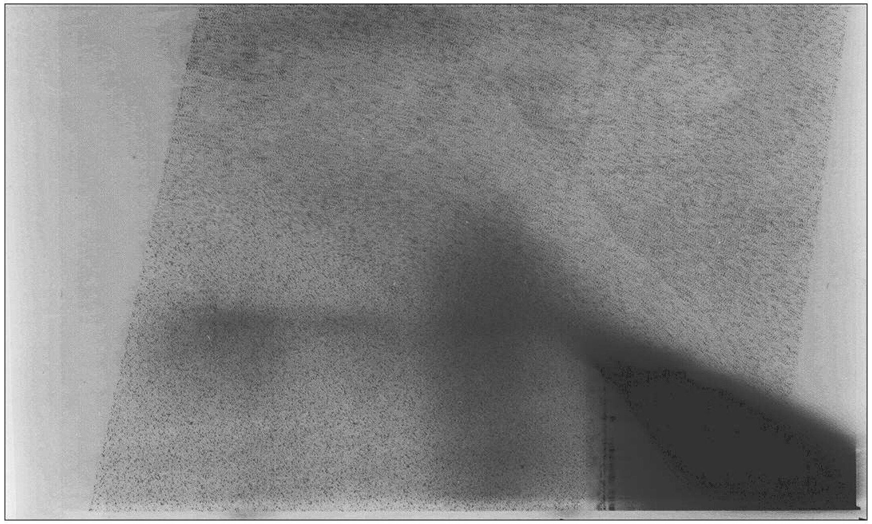


Figure 4.7: This is an image recorded using the PIV technique in a low speed wind tunnel having a turbulence generating grid in the inlet. The flow is from right to left. A disturbance is generated by a fence on the wall. The saturation is due to non separated particles.

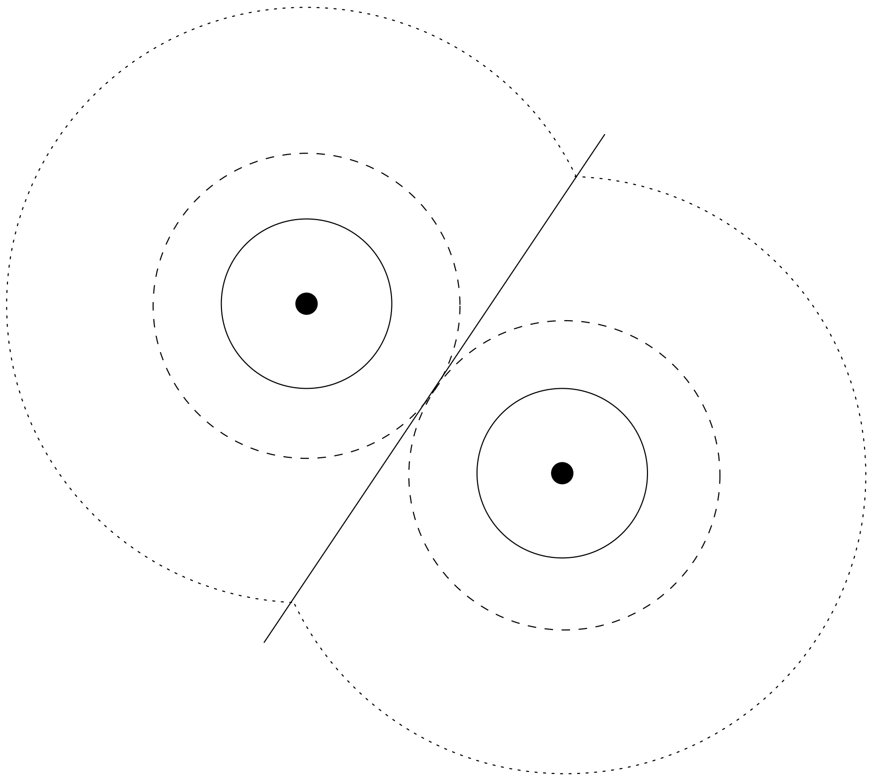


Figure 4.8: In this figure three iso-curves in the Euclidean distance transform are shown in an image containing two points.

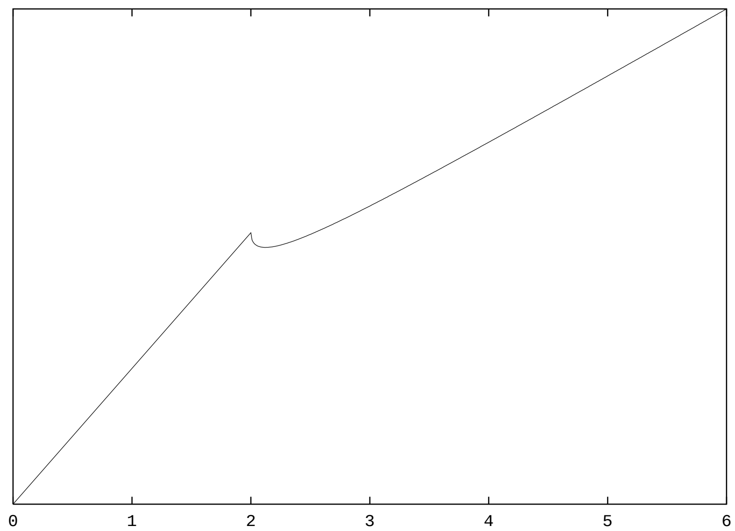


Figure 4.9: In this figure the function f of Equation (4.9) is seen.

From this figure it is evident that the half distance between points is identified by a peak in the histogram of the distance transformation.

This, however, only constrains the speed of the movement, and not the direction. First of all it is evident that we can only determine the direction modulo π, as we can not identify which of the points corresponds to the first occurrence of the particle. In order find the direction we can make use of another property of the midpoint between object points. It is located on a watershed in the distance map. This is illustrated in Figure 4.10.

4.3.3 The Algorithm

Given a continuous distance transformation, identification of the peak in the histogram, Figure 4.9, should be quite straightforward. Due to the discrete nature of digital images, however, some problems occur. The discrete grid only allows for some distances in our distance map. An isolated pixel has 4 pixels of distance 1, 4 pixels in the distance $\sqrt{2}$, 4 pixels in the distance 2, 8 pixels in the distance $\sqrt{5}$, 4 pixels in the distance 3, and so forth. So for small distances between object pixels compared to the unit distance of the grid, the peak in the histogram given by a typical half distance between object pixels might easily be confused with peaks due to the discrete nature of the imagery, such as the one corresponding to the distance $\sqrt{5}$.

We therefore propose to normalize the histogram with the nominal occurrence of the distances in the distance map corresponding to a single object pixel in infinite space.

We will illustrate this on a artificially generated PIV image. The image is shown in Figure 4.11. This image has been generated by randomly distributing pixels within a 512×512 frame, then replicating each pixel in a position 4 rows below and 4 columns to the right.

The histogram of this images normalized as described above is shown in Figure 4.12. We can compare this to the normalized histogram of the distance map

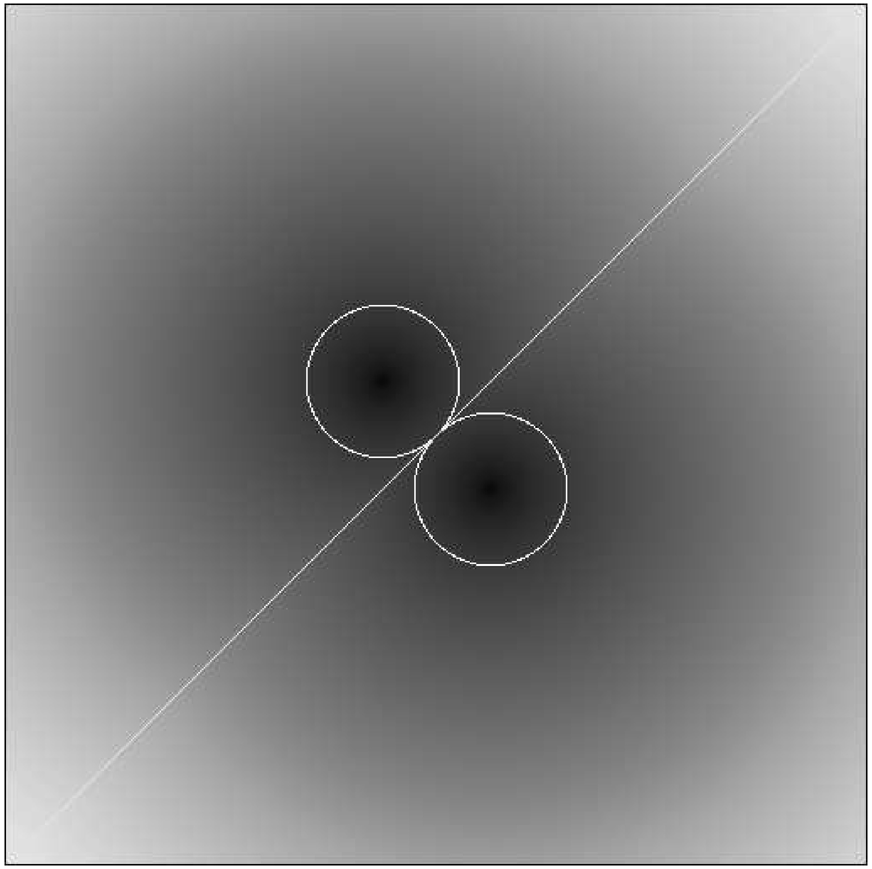


Figure 4.10: In this figure we see the distance map for the two points in Figure 4.8 shown as a grey level image. Super imposed on the distance map the iso-distance curves corresponding to the half distance between points is shown in white, and the watershed of the distance map is shown in light grey. It is evident that the two curves intersect in the midpoint between object points.

resulting from an image containing points distributed randomly with a probability twice as high as the one used to generate the one in Figure 4.11. Such an image is seen in Figure 4.13, and its normalized histogram is seen in Figure 4.14.

It is evident that the simulated PIV images shows an under representation of occurrences of pixels having the distance to the nearest object equal to $\sqrt{8}$, which is exactly half the distance between the the pixel pairs. This is further demonstrated by normalizing the the histogram corresponding to the random distribution of pairs of points with the histogram corresponding to the totally random distribution. This is shown in Figure 4.15. The under representation of points having distance $\sqrt{8}$ is clearly seen,as well as an under representation of points having the distance $\sqrt{10}$.

We have thus successfully identified the shift and thereby the speed of the particles in a simulated double exposed image of particle flow.

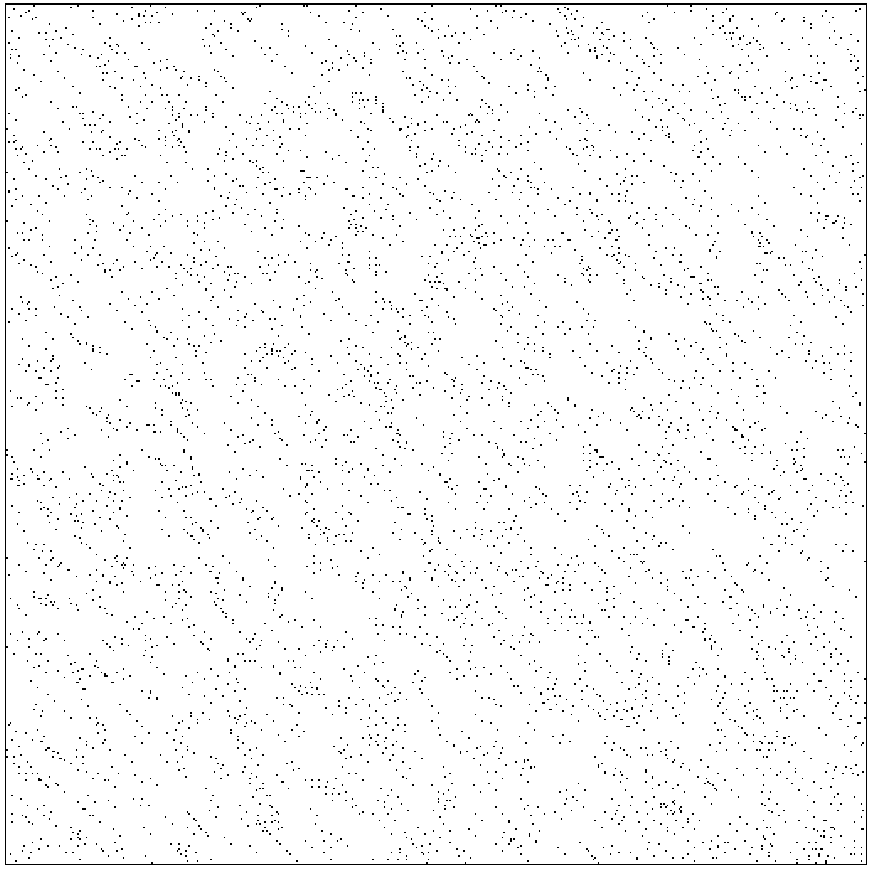


Figure 4.11: A simulated PIV image. Pixels are distributed randomly and then replicated in a position 4 rows below and 4 columns to the right. The object pixels are shown in black on a white background.

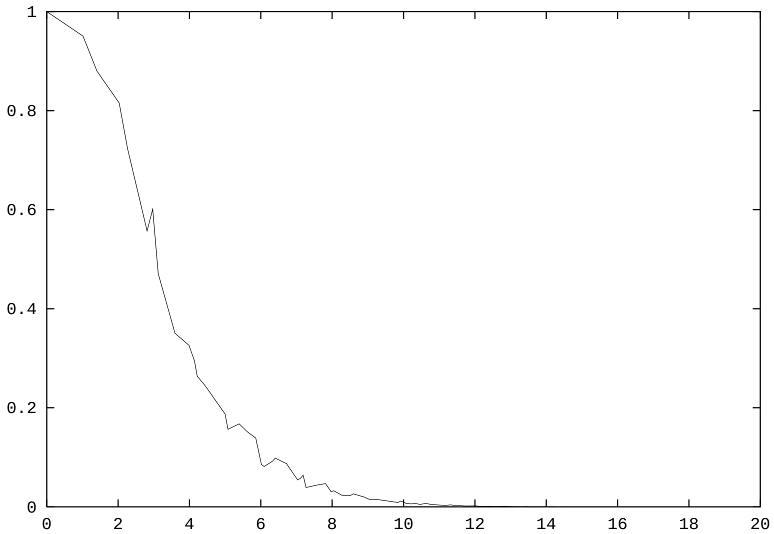


Figure 4.12: This is the histogram of the Euclidean distance transform of the image in Figure 4.11, where the number of occurrences of each distance is normalized by the number of occurrences of the distance in a distance map corresponding to a single pixel in infinite space. The histogram has further been normalized by the number of pixels in the image.

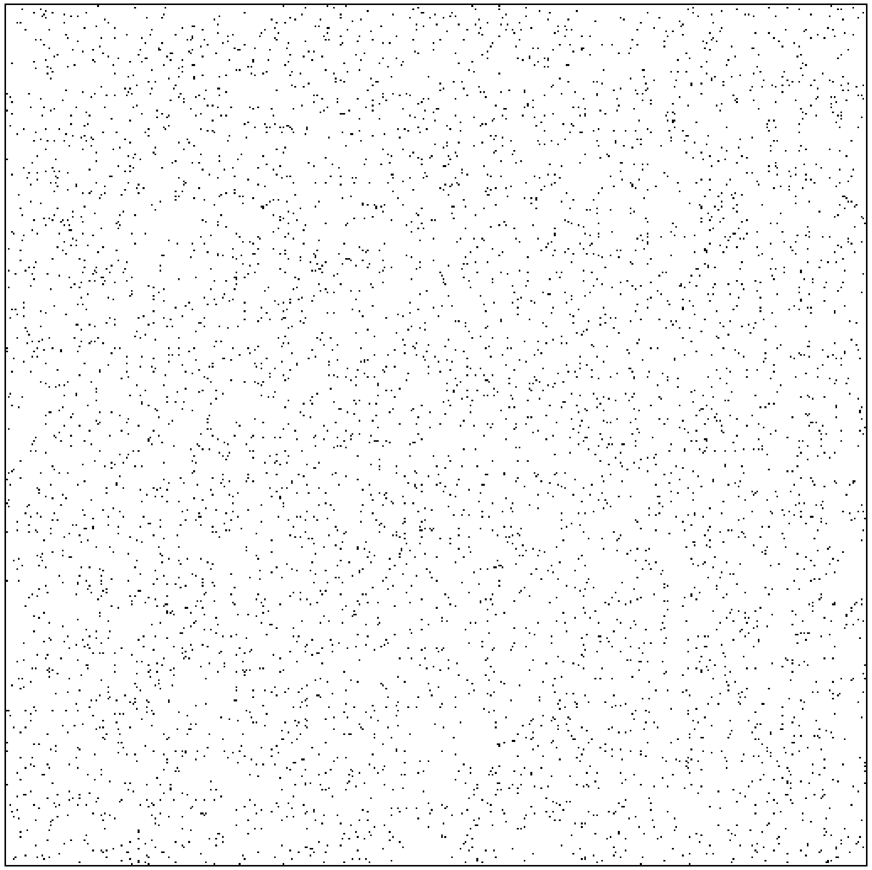


Figure 4.13: This image contains points randomly distributed the plane with a probability twice as high as the one in Figure 4.11.

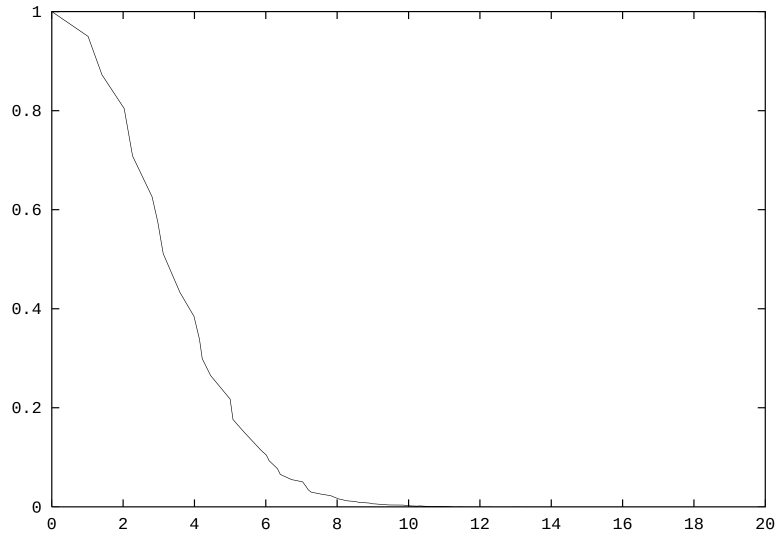


Figure 4.14: This is the histogram of the Euclidean distance transform of the image in Figure 4.13, where the number of occurrences of each distance is normalized by the number of occurrences of the distance in a distance map corresponding to a single pixel in infinite space. The histogram has further been normalized by the number of pixels in the image.

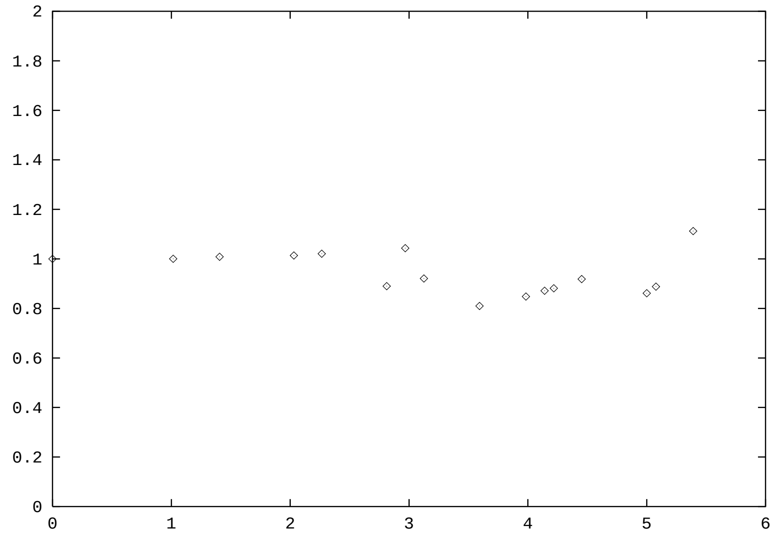


Figure 4.15: This is the normalized histogram corresponding to the random distribution of pairs of points, that are displaced by 4 rows and 4 columns, normalized with the normalized histogram corresponding to a totally random distribution of points. Under representation of points in the distance map having distances $\sqrt{8}$, as well as $\sqrt{10}$ are easily identified.

Chapter 5

Integration of Local Measurements

As described in the previous chapters, the local measurements of motion often do not constrain the true flow fully due to the aperture problem. This results in unsatisfactory flow fields in the case of image compression and temporal interpolation. In order to remedy this problem, and to lessen the effect of noisy observations, an assumption of global or piecewise smoothness is often applied.

In this chapter we will review some of the techniques used to propagate measurements of local motion at points of well defined motion to areas of less well defined motion. In particular we will describe the use of the framework given by the Bayesian Paradigm in combination with a class of models called Markov random fields.

5.1 Deterministic Techniques

We have already touched upon one of the techniques to integrate measurements. In Sections 3.3.2 and 3.3.3 methods that involved computations of the local

flow by combining neighboring measurements under the assumption of local constancy or of some parametric model, e.g linear and quadratic models, were described.

Horn & Schunk (1981) suggested a global smoothness constraint on the velocity field, which resulted in a compromise between a error derived from the motion constraint Equation (3.17)

$$\mathcal{E}_b = I_x u + I_y v + I_t \tag{5.1}$$

and a motion smoothness error given by the partial derivatives of the velocity field with respect to the spatial coordinates

$$\mathcal{E}_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \tag{5.2}$$

The total error to be minimized is then given by

$$\mathcal{E}(\boldsymbol{u}) = \int\!\!\int (\mathcal{E}_b^2 + \alpha^2 \mathcal{E}_c^2)\, dx\, dy \tag{5.3}$$

where the integration is performed over the entire image. Nagel & Enkelmann (1986) extended this method proposed by Horn & Schunk by introducing a space variant smoothing. They suggested the following "oriented" smoothness constraint

$$\mathcal{E}_c^2 = \mathrm{trace}((\nabla \boldsymbol{u})^T W (\nabla \boldsymbol{u})) \tag{5.4}$$

where $\nabla \boldsymbol{u}$ expresses the partial derivatives of the two velocity components in a matrix form

$$\nabla \boldsymbol{u} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{bmatrix} \tag{5.5}$$

and W has a structure given by

$$W = \begin{bmatrix} I_y^2 & -I_x I_y \\ -I_x I_y & I_x^2 \end{bmatrix} + \varsigma^2 \begin{bmatrix} I_{xx}^2 + I_{xy}^2 & I_{xy}(I_{xx} + I_{yy}) \\ I_{xy}(I_{xx} + I_{yy}) & I_{xy}^2 + I_{yy}^2 \end{bmatrix} \quad (5.6)$$

This change in the smoothness, i.e. using W instead of the identity matrix used by Horn & Schunk, has the following effect. At points with strong second order variation, e.g. corners, the smoothness constraint is enforced weakly, and in the vicinity of points with strong first order variation, e.g. at edges, the smoothness constraint is enforced strongly along the contour and weakly across the contour.

The approach of Nagel is one one way of overcoming the problem of over smoothing at motion boundaries. Hildreth (1984) proposed a technique according to which she only calculated local motion estimates at intensity contours, and then subsequently only applied a smoothness constraint along these contours. For each contour she minimized the following curve integral with respect the velocity along the contour, $\boldsymbol{u}$,

$$\Theta(\boldsymbol{u}) = \int \left(\left(\tfrac{\partial u}{\partial s}\right)^2 + \left(\tfrac{\partial v}{\partial s}\right)^2 \right) + \beta \left((\boldsymbol{u} - \boldsymbol{\mu}) \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|} \right) ds \quad (5.7)$$

where s is the curve length, and $\boldsymbol{\mu}$ is the measured normal flow. The normal flow measurements were done using a gradient based technique, but in principle any local measurement could have been used.

Simpson & Gobat (1993) have suggested a smoothing procedure based on vector median filtering. They applied a region based matching technique using the maximum cross correlation to estimate the local velocity. Initially points having multiple correlation peaks in the cross correlation function, e.g. points along contours where only normal flow can be extracted, where replaced by a local vector median. The purpose of this being to obtain an estimate of the true flow. Following this correction an iterative vector median filtering is applied. Simpson & Gobat successfully used this algorithm to compute flow estimates of the oceanic flow as seen in sequences of satellite data. The vector median filters are described in (Astola, Haavisto, & Neuvo, 1990).

5.2 Stochastic Approaches

Since the paper by Geman & Geman (1984) on stochastic relaxation, Gibbs distributions, and Bayesian restoration, the literature has exploded with applications of the Bayesian paradigm in image analysis and image processing. The Bayesian paradigm is a framework for incorporating stochastic models of visual phenomena into a very general set of tasks. We will give a short review of Bayesian approaches to motion estimation. First we will, however, give a short review of the Bayesian paradigm and Markov random fields.

5.2.1 Bayesian Paradigm

The Bayesian paradigm may be described in four successive stages, (Besag, 1989).

1. Construction of a prior probability distribution $P(\boldsymbol{x})$ for the phenomenon, $\boldsymbol{X}$, that we want to make inferences about. This distribution should capture our general and scene specific knowledge. In practice we can not expect to model global features of the true image, so we should concentrate on modelling the local characteristics. For instance, nearby pixel values tend to be similar, adjacent labels are usually the same, and object boundaries are generally continuous.

2. Formulation of an observation model $P(\boldsymbol{y}|\boldsymbol{x})$, that describes the distribution of the observed images $\boldsymbol{Y}$ given any particular realization $\boldsymbol{x}$ of the prior distribution. This step is governed by physical or statistical considerations concerned with the observation device.

3. Combination of the prior distribution and the observation model into the posterior distribution $P(\boldsymbol{x}|\boldsymbol{y})$ by Bayes theorem

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x})}{P(\boldsymbol{y})} \quad (5.8)$$

$P(\boldsymbol{x}|\boldsymbol{y})$ is the distribution of the phenomena $\boldsymbol{X}$ given the observed images $\boldsymbol{y}$. $P(\boldsymbol{y})$ is the prior probability of the observed image, and is generally not required.

4. Drawing of inferences about the visual phenomenon based on the posterior distribution. This resembles traditional forms of regularization, i.e. faith in the data is balanced against the regularities of the prior.

5.2.2 Markov Random Fields and Gibbs Distributions

Before entering into a discussion of motion estimation techniques we will review the class of models known as Markov random fields. We will consider discrete as well as continuous models, as both have been used extensively in motion analysis, (Murray & Buxton, 1987; Konrad & Dubois, 1992; Heitz & Bouthemy, 1993). For more thorough discussions on Markov random fields and the related Gibbs random fields the reader is referred to Geman (1990), Dubes & Jain (1989), Ripley (1988).

Graphs and Neighbourhoods

As we will be studying the the distribution of random phenomena on a spatial arrangement of points, we will start by some useful definitions from graph theory. This discussion will be strongly adapted to our needs.

Definition 1 Let $S = \{s_0, s_1, \ldots, s_{(N-1)}\}$ be a set of sites. A *neighbourhood system* $\mathcal{G} = \{G_s, s \in S\}$ is any collection of subsets of S for which

1. $s \notin G_s$
2. $r \in G_s \Leftrightarrow s \in G_r$

G_s are the *neighbors* of s. □

We will use the notation $i \sim j$ to denote that the sites i and j are neighbors.

Definition 2 A *clique* C is a subset of the set of sites S for which every pair of sites are neighbors. □

Single pixels are also considered cliques. The set of all cliques is called $\mathcal{C}$.

In the sequel we will consider the following three cases of neighbourhood systems.

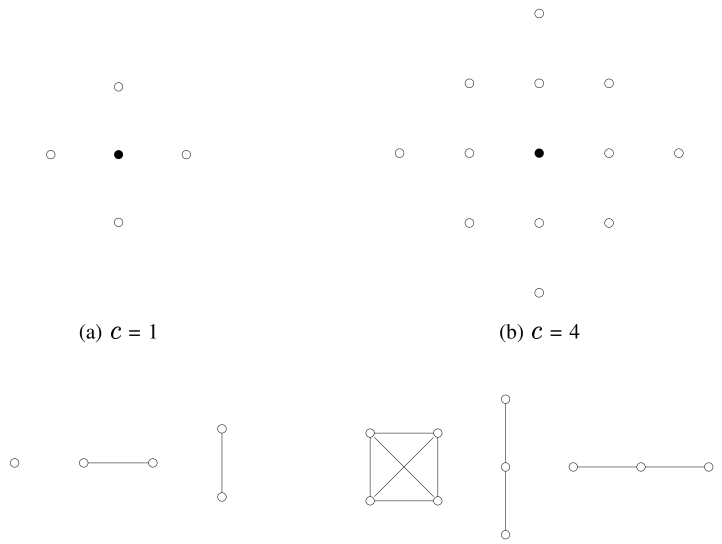
Case 1: $S = X_{m,n}$ $X_{m,n}$ is the $m \times n$ integer lattice representing the grid on which our images have been sampled, i.e. the pixel sites for the intensity process. We are interested in homogeneous neighbourhood systems given by

$$\begin{aligned} \mathcal{F}_c &= \{F_{ij} | i = 0, \ldots, m-1, j = 0, \ldots, n-1\} \\ F_{ij} &= \{(k,l) | (k-i)^2 + (l-j)^2 \le c\} \end{aligned}$$

Notice that the sites at or near the boundary of the lattice have fewer neighbors than the interior points. This is the so-called free boundary model. Other models include the toroidal model, which corresponds to first rolling the lattice as a tube and then connection of the ends of the tube to a torus. Figures 5.1(a) and 5.1(b) show the (interior) neighbourhood configuration for $c = 1$ and $c = 4$. In Figures 5.1(c) and 5.1(d) cliques corresponding to he two neighbourhood systems are shown. □

Case 2: $S = D_{m,n}$ $S = D_{m,n}$ is the dual lattice of $X_{m,n}$. The sites of $D_{m,n}$ are the midpoints of each vertical or horizontal pair of pixels, and they represent possible locations of edge elements. In Figures 5.2(a) and 5.2(b) the (interior) neighbourhood configurations for a horizontal and a vertical line element are shown in one system. In Figures 5.2(c) and 5.2(d) the neighbourhood configurations are shown for another system. □

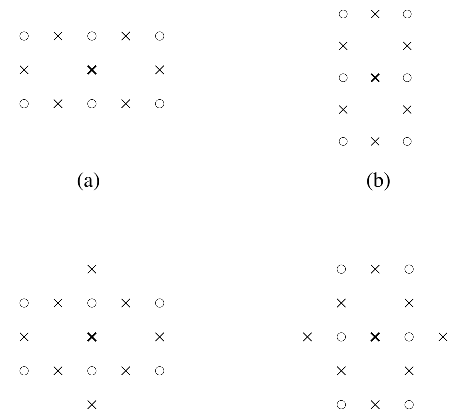
Case 3: $S = X_{m,n} \cup D_{m,n}$ This is the setup for a combined pixel and line process. The pixel neighbors of sites in $D_{m,n}$ might be the two pixels on each side, and hence each pixel has four line neighbors. □



(a) $c = 1$ (b) $c = 4$

(c) $c = 1$ (d) $c = 4$

Figure 5.1: Neighbourhood systems and cliques for the pixel process. (a),(b) The neighbourhood configurations for (interior) points. The points marked with the symbol ∘ are the neighbors of the point marked •. (c) The cliques corresponding to the neighbourhood configuration in (a). (d) Some of the additional cliques corresponding to the neighbourhood configurations in (b). This neighbourhoods contains 24 cliques in all.



(a) (b)

(c) (d)

Figure 5.2: Neighbourhood systems for the line process. Together with the line sites, denoted by ×, we see the pixel sites marked by ∘. (a),(b) these are the neighbourhoods for line sites between a vertical pair of pixels, and a horizontal pair of pixels as used by Geman & Geman (1984). (c),(d) these are the neighbourhoods for line sites between a vertical pair of pixels, and a horizontal pair of pixels as used by Konrad & Dubois (1992).

Gibbs Measures and Markov Random Fields

We shall now consider probability distributions on lattices and graphs. We will restate the definition of Markov random fields and Gibbs measures, and a theorem that shows an equivalence between these two.

Let there be given a neighbourhood system, $\mathcal{G} = \{G_s\}$, on a set of sites $S = \{s_0, \ldots, s_{n-1}\}$. We consider any family of random variables indexed by S, i.e.

$$\boldsymbol{X} = \{X_s | s \in S\} \tag{5.9}$$

The state space may be ordered, e.g

$$\Lambda = \{0, 1, \ldots, L-1\}, \tag{5.10}$$

or unordered, e.g.

$$\Lambda = \{\text{red}, \text{yellow}, \text{green}\}. \tag{5.11}$$

We let Ω denote the set of all possible configurations

$$\Omega = \{\omega = (x_{s_0}, \ldots, x_{s_{n-1}}) \,|\, x_{s_i} \in \Lambda, 0 \leq i < n\} \tag{5.12}$$

As usual, the event $\{X_{s_0} = x_{s_0}, \ldots, X_{s_{n-1}} = x_{s_{n-1}}\}$ is abbreviated $\{\boldsymbol{X} = \omega\}$. After these preparations we introduce the Markov random field in the following definition

Definition 3 $\boldsymbol{X}$ is an *Markov Random Field (MRF)* with respect to $\mathcal{G}$ if and only if

1. $P(\boldsymbol{X} = \omega) > 0, \qquad \text{for all } \omega \in \Omega$
2. $P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in G_s)$ for all $s \in S$, and $\omega \in \Omega$.

□

It should be noted that any random field satisfying the first "positivity" condition is a Markov random field, if we can specify a neighbourhood large enough to encompass the conditional dependencies of the second condition. The utility of these models in image modelling lies in that we can specify prior models with neighbourhood that are small enough to allow for reasonable computational loads, yet at the same time they are rich enough to model interesting classes of images (or fields), (Carstensen, 1992). Furthermore it should be pointed out that although the conditioning in the second condition of the definition is on the state in neighboring sites only, it is perfectly possible to obtain long range correlations.

We next consider a so-called *potential*. Let C be a clique, and let all cliques given by the neighbourhood system $\mathcal{G} = \{G_s\}$ be denoted $\mathcal{C}$. For all $C \in \mathcal{C}$ we assume we have a function

$$V_C : \Omega \to R \tag{5.13}$$

with the property that $V_C(\omega)$ only depends on the states in the sites in the clique C. The family of functions $\{V_C | C \in \mathcal{C}\}$ is called a potential. Given the potential we can constitute the *energy function*

$$U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega) \tag{5.14}$$

This leads to the following definition of the Gibbs measure.

Definition 4 The *Gibbs measure* induced by the energy function U is

$$P(\boldsymbol{X} = \omega) = \frac{1}{Z} \exp(-U(\omega)/T)$$

where T is a control parameter usually referred to as temperature, and the normalizing constant

$$Z = \sum_{\omega \in \Omega} \exp(-U(\omega)/T)$$

is called the *partition function*. □

The Gibbs measure has been extensively used in statistical physics to express the probability of a system with many degrees of freedom being in a certain state with a certain energy, mainly because it maximizes the entropy

$$-\sum_{\omega \in \Omega} p(\omega) \log(p(\omega))$$

among all distributions with the same expected energy. As entropy is a measure of uncertainty, the Gibbs measure is the probability distribution that for a given energy maximizes uncertainty.

The temperature T controls the "peaking" of the probability density function. As T tends to infinity the distribution tends to a uniform distribution among all states. As T tends to 0 the distribution tends to a uniform distribution among the minimum energy states.

In general the partition function is impossible to compute due the usually enormous number of configurations in Ω. It turns out, however, that very often we need not compute exact probabilities, ratios between probabilities suffices, and then Z is cancelled.

The Markov random fields and the Gibbs measure are related through the following theorem.

Theorem 1 (Hammersley-Clifford). Let $\mathcal{G}$ be a neighbourhood system, Then $\boldsymbol{X}$ is a Markov random field with respect to $\mathcal{G}$ if and only if $P(\boldsymbol{X} = \omega)$ is a Gibbs distribution with respect to $\mathcal{G}$. □

A simple proof is given by Geman (1990). This equivalence is important because it provides us with a simple way of specifying Markov random fields, namely by

specifying potentials, which is easy, instead of local characteristics for which it is hard to establish consistence of the joint distribution. Furthermore the Gibbs measure gives us a joint distribution of $\boldsymbol{X}$.

Finally, we shall make remarks on some extensions of the description given above. For simplicity it was assumed that all sites had a common state space. Without any problems we can extend this to a site-dependent state space, which is appropriate when S consists of both pixel and line sites as described in the previous section.

The extension from a one dimensional Markov random field to multivariate random fields is also straightforward. A multivariate random field can be reduced to a univariate random field by redefining the state space. Consider a p-dimensional process, $X_s = (X_s^{(1)}, \ldots, X_s^{(p)})$. If the cardinality of the ith component is L_i, then we construct a single state space with $\prod_i L_i$ states where each state is a p-tuple $(x_s^{(1)}, \ldots, x_s^{(p)})$. This will be needed for our purposes as we in some cases aim to model the velocity field as a Markov random field.

The description has only been given for a finite state space. It is, however, possible to extend to the case of X_i having a countable number of outcomes, as well as the case of X_i having a continuous distribution. In the first case

$$\sum_{\omega \in \Omega} \exp(-U(\omega)/T)$$

must be finite, and in the second case $\exp(-U(\omega)/T)$ must be integrable.

5.2.3 Prior Distributions

In this section we will describe some of the choices of prior distributions for motion estimation in the literature. In general we can choose to model the joint distribution of velocity vectors by using a pixel prior, or we can choose a labeling prior, with the purpose of segmenting different moving objects within the image field. Template priors can be used when we want to make inferences about geometrical shapes in the image, e.g. in the case of contour tracking.

Finally we can incorporate a description of discontinuities in the velocity field by including edge models.

Pixel Priors

A very useful class, and the most widely used, of pixel priors are based on differences between neighbors. Besag (1989) presented this class of distribution in the univariate case, given by their energy function

$$U(z) = \sum_{i \sim j} \phi(z_i - z_j) \tag{5.15}$$

where ϕ is a specified function satisfying

$$\phi(w) = \phi(-w), \quad \phi(w) \text{ increasing with } |w| \tag{5.16}$$

If furthermore ϕ is convex and differentiable, computational advantages can be obtained. Joint distributions defined by Equation (5.15) are improper in the sense that they are non-integrable, and thus can not be normalized. The impropriety arises because Equation (5.15) only addresses differences, the overall level of the process being unrestricted. They do however have a perfectly proper conditional density

$$P(z_i | z_j, j \neq i) \propto \exp(-\sum_{j \in G_i} \phi(z_i - z_j)) \tag{5.17}$$

where G_i is the neighbourhood of i.

These models bear resemblance to some of the integrations described in Section 5.1 in the sense that the partial derivatives of the velocity field can be implemented as differences between pairs of pixel values.

In the following we will be concerned with the modelling of a two dimensional vector field $\boldsymbol{u} = \{\boldsymbol{u}_i, i \in S\}$, where S is the sites of our lattice. Note that

we have left the cumbersome double indexing of the sites in favor of a a more simple numbering. The coordinates of site i will be denoted $\boldsymbol{x}_i = (x_i, y_i)$.

In the context of surface reconstruction Terzopoulos (1986, 1988) described two energy functions[1]. In the two dimensional case they correspond to

$$\begin{aligned} U_1'(\boldsymbol{u}) &= \int (u_x^2 + u_y^2 + v_x^2 + v_y^2)\, dx\, dy, & (5.18)\\ U_2'(\boldsymbol{u}) &= \int (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2 + v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2)\, dx\, dy, & (5.19)\end{aligned}$$

where the integration is performed over the entire image. The first function is recognized as the measure of continuity used by Horn & Schunk (1981), Equation (5.2). If the partial derivatives are implemented like this

$$\begin{aligned} w_x &= w(x_i+1, y_i) - w(x_i, y_i) \\ w_y &= w(x_i, y_i+1) - w(x_i, y_i) \\ w_{xx} &= w(x_i+1, y_i) - 2w(x_i, y_i) + w(x_i-1, y_i) \\ w_{yy} &= w(x_i, y_i+1) - 2w(x_i, y_i) + w(x_i, y_i-1) \\ w_{xy} &= w(x_i+1, y_i+1) - w(x_i, y_i+1) - w(x_i+1, y_i) + w(x_i, y_i) \end{aligned} \tag{5.20}$$

where w can be substituted for either u, or v, the priors corresponding to Equation (5.18) and (5.19) can be implemented using the neighbourhood systems given by the neighbourhoods shown in Figures 5.1(a) and 5.1(b), respectively.

[1]These models are the two lowest order cases of the two-dimensional generalized spline functional,

$$|v|_m^2 = \int \sum_{i=0}^{m} \binom{m}{i} \left(\frac{\partial^m v}{\partial x^i \partial y^{m-i}} \right)^2 dx\, dy$$

For the first model, attained for m=1, the functional is proportional to the small deflection energy of a membrane. The associated Euler-Lagrange equation is Laplace's equation, $-\Delta u = 0$, where $\Delta u = u_{xx} + u_{yy}$. For the second model, attained for m=2, the functional is proportional to the small deflection bending energy of a thin plate. The associated Euler-Lagrange equation is the biharmonic equation, $\Delta^2 u = 0$, where $\Delta^2 u = u_{xxxx} + 2u_{xxyy} + u_{yyyy}$,(Terzopoulos, 1986).

For the two models, the discrete version of the energy function becomes upon substituting (5.20) into (5.18) and (5.19), respectively

$$U_1(\boldsymbol{u}) = \sum_i V_2^h(\boldsymbol{u}, i) + \sum_i V_2^v(\boldsymbol{u}, i). \tag{5.21}$$

$$U_2(\boldsymbol{u}) = \sum_i V_3^h(\boldsymbol{u}, i) + \sum_i V_3^v(\boldsymbol{u}, i) + \sum_i V_4(\boldsymbol{u}, i), \tag{5.22}$$

The potentials V_2^h, V_2^v, V_3^h, V_3^v, and V_4 corresponding to the multi pixel cliques shown in Figures 5.1(c) and 5.1(d) are given by

$$\begin{array}{rcl} V_2^h(\boldsymbol{u}, i) & = & \|\boldsymbol{u}(x_i{+}1, y_i) - \boldsymbol{u}(x_i, y_i)\|^2 \\ V_2^v(\boldsymbol{u}, i) & = & \|\boldsymbol{u}(x_i, y_i{+}1) - \boldsymbol{u}(x_i, y_i)\|^2 \\ V_3^h(\boldsymbol{u}, i) & = & \|\boldsymbol{u}(x_i{+}1, y_i) - 2\boldsymbol{u}(x_i, y_i) + \boldsymbol{u}(x_i - 1, y_i)\|^2 \\ V_3^v(\boldsymbol{u}, i) & = & \|\boldsymbol{u}(x_i, y_i{+}1) - 2\boldsymbol{u}(x_i, y_i) + \boldsymbol{u}(x_i, y_i - 1)\|^2 \\ V_4(\boldsymbol{u}, i) & = & \|\boldsymbol{u}(x_i{+}1, y_i{+}1) - \boldsymbol{u}(x_i, y_i{+}1) - \boldsymbol{u}(x_i{+}1, y_i) + \boldsymbol{u}(x_i, y_i)\|^2 \end{array} \tag{5.23}$$

where $\|\cdot\|$ is the Euclidean norm.

The energy function in Equation (5.21) corresponds to an intrinsic Gaussian autoregression, Besag (1989). These Gaussian priors seem well suited when the flow is known to be smooth, but they will be unsatisfactory in the presence of discontinuities, which they will smear out.

Gaussian priors for vector field modelling have been used by Konrad & Dubois (1992)

An alternative to the Gaussian pixel prior in the univariate case is the median pixel prior given by the energy

$$U(z) = \sum_{i \sim j} |(z_i - z_j)| \tag{5.24}$$

The conditional distribution of Markov random field corresponding to this energy function has its mode at the median of the neighboring z_j's rather than at the

mean, as was the case for the Gaussian pixel prior. This property makes it attractive for fields with discontinuities. The energy function, however, is not differentiable for this prior.

The univariate case can be generalized to the multivariate case by using the $\|\cdot\|_1$-norm. In fact, we might generalize further and introduce the $\|\cdot\|_p$-norm,

$$U(\boldsymbol{u}) = \sum_{i \sim j} \|(\boldsymbol{u}_i - \boldsymbol{u}_j)\|_p^p, \quad 1 \leq p \leq 2 \tag{5.25}$$

The generalization has been introduced in the univariate case by Bouman & Sauer (1993).

Other approaches to avoid the smearing of discontinuities attained by the Gaussian prior include the following function suggested by Blake & Zisserman

$$U(z) = \sum_{i \sim j} (\min\{z_i - z_j|, \tau\})^2 \tag{5.26}$$

where τ is a threshold parameter. Due to the non convexity of this function, any estimation in the posterior function relies heavily on the method used for minimization.

Label Priors

When the image sequence in question contains different moving objects, and we want to make inferences about the movement of the objects, we need to perform a segmentation as well as a velocity estimation. One way of doing this is to apply a Markov random field with a label prior. As mentioned above the label prior should incorporate the assumption that adjacent pixels tend to belong to the same object. Bouthemy & François (1993) describe an approach based on the following energy function

$$U(\boldsymbol{e}) = \sum_{i \sim j} \eta(1 - \delta(\boldsymbol{e}_i, \boldsymbol{e}_j)) \tag{5.27}$$

Where $e = \{e_i, i \in S\}$ is the label field, $\eta > 0$ is a predetermined parameter, and δ is Kronecker's delta. Bouthemy & François used the binary cliques in a second order neighbourhood, (3×3). As described by Carstensen (1992) using only binary cliques we can not model structural differences between the phases. In order to do this we need cliques containing odd numbers of pixels. So in would seem appropriate in this case, where structural differences between moving objects on a non moving background seems likely, that such cliques should be used. For instance based on the morphological Markov random fields described by Carstensen (1992).

Template Priors

Template priors are prior distributions that describe the geometrical relations in objects. We have touched upon this in Section 3.3.4, where we made inferences about the geometry of a contour based on the prior information given by the estimated position in the previous image. We might also use a template prior that describes our prior knowledge of the shapes we are observing. For instance Kervrann & Heitz (1993) have applied such an algorithm to the tracking of a hand.

Inclusion of Edge Priors

The inclusion of edge priors is a way to cope with discontinuities in the field. This is the situation described by Case 3, where we are considering a pixel/label process coupled with a line process. The prior distributions should be generated satisfying for instance that adjacent pixels have similar values, or labels, and boundaries are continuous. We will now consider a coupled pixel and line process, $(\boldsymbol{u}, \boldsymbol{l})$. We will index the pixel process by i, and use the index ij for the line process, where ij denotes the position between pixels i and j. Geman & Geman (1984) suggested a neighbourhood system, where the pixel process had the neighbourhood system given by the four nearest pixel neighbors, as seen in Figure 5.1(a), supplemented with the four edge positions given by the midpoints

of the four binary pixel cliques. The line process had the neighbourhoods system given for the horizontal and the vertical case in Figures 5.2(a) and 5.2(b). We will consider the case of a continuous state space for the pixel process and a binary state space (line on/off) for the line process.

$$\boldsymbol{l}_{ij} = \begin{cases} 1 & \text{line on} \\ 0 & \text{line off} \end{cases} \tag{5.28}$$

For the Gaussian pixel prior in the 4-nearest neighbors system this causes the following modified energy function

$$U(\boldsymbol{u}, \boldsymbol{l}) = \sum_{i \sim j} \|(\boldsymbol{u}_i - \boldsymbol{u}_j)\|^2 (1 - \boldsymbol{l}_{ij}) + \sum_{c_l} V_{c_l}(\boldsymbol{l}) \tag{5.29}$$

where c_l is the index for the line cliques. They only considered the four element cross shaped cliques. The potentials are given for the different configurations of these in Figure 5.3. These line field potentials characterize the local properties of the boundaries.

Geman & Geman (1984) also suggested line processes having state spaces including different orientations of the discontinuity. Besag (1989) preferred the line sites of the honeycomb lattice generated by using a hexagonal pixel array. To overcome problems of micro edges being generated at fuzzy boundaries, Konrad & Dubois (1992) suggested larger neighbourhood for the line process and used a potential that punished parallel lines. Geman, Geman, Graffigne, & Dong (1990) tried to overcome that problem by defining line sites and neighbourhoods on a coarser scale than the pixel process.

5.2.4 Observation Models

In the these cases of motion estimation the observations are usually computations of local velocity estimates based on the measured intensity values. The techniques for obtaining these local observations are described in Chapter 3. The

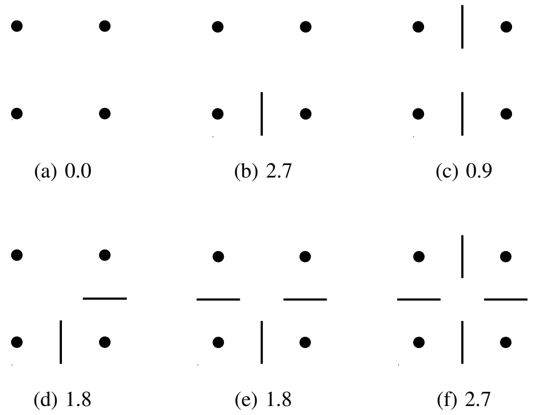


Figure 5.3: Potentials associated with various configurations (up to a rotation): (a)-(f) four-element cross shaped cliques; (•–vector site, — –line element "on").

observation models for the continuous pixel priors are based on assumptions such as normal distribution of the difference of the displaced intensity values, (Konrad & Dubois, 1992), i.e. $d = I(\boldsymbol{x}, t) - I(\boldsymbol{x}+\boldsymbol{u}, t+1) \in N(\mu, \varsigma^2)$. Others use normal distribution of the residual of the gradient constraint Equation (3.17), i.e. $d = (I_x, I_y)\boldsymbol{u} + I_t \in N(\mu, \varsigma^2)$.

In the case of the label prior, one might use an assumption of normality of the difference to a local estimate of the object velocity based on the object segmentation from the previous image, (Bouthemy & François, 1993).

5.2.5 Maximum a Posteriori Estimates

The maximum a posteriori estimate $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$ given observations $\boldsymbol{y}$ is defined by

$$\hat{\boldsymbol{u}} = \arg\max p(\boldsymbol{u}|\boldsymbol{y}) \tag{5.30}$$

If the prior model and the observation model both are described by Gibbs measures, so is the posterior model. The maximization of the posterior probability is transformed into the minimization of the sum of the energy functions for the prior distribution and the observation model. This usually involves a huge computational cost due to the high dimensionality of the parameter space.

Iterated Conditional Modes

The iterated conditional modes (ICM) algorithm, (Besag, 1986), consists of a series of sweeps over the sites of the process, in which each site visited and set to the value that maximizes the conditional probability given the observations $\boldsymbol{y}$, and the current reconstruction elsewhere

$$\hat{\boldsymbol{u}}_i = \arg\max p(\boldsymbol{u}_i|\boldsymbol{y}, \hat{\boldsymbol{u}}_{S\setminus i}) \tag{5.31}$$

Convergence is only guaranteed if the energy function of the posterior distribution is convex. In not, an ICM scheme is likely to get trapped in a local maximum of the posterior density.

Simulated Annealing

A simulated annealing scheme to obtain the maximum a posteriori estimate in a non convex posterior density was given by Geman & Geman (1984). The idea is that we for a fixed temperature T simulate in the distribution

$$p_T(\boldsymbol{u}|\boldsymbol{y}) \propto (p(\boldsymbol{u})p(\boldsymbol{y}|\boldsymbol{u}))^{\frac{1}{T}} \tag{5.32}$$

by means of the Gibbs sampler, i.e. at each update we select a new value at a site by randomly selecting a new value from the conditional distribution given the neighbors and the observations. The simulations will allow states with higher energy which enables us to escape local minima. If we decrease the temperature slowly we will finally end up in the maximum a posteriori estimate. For details on the algorithm the reader is referred to Geman & Geman (1984), Aarts & Korst (1989).

Chapter 6

Examples on Integration of Local Measurements

In this chapter we will present some examples of model based integration of local measurements of motion.

In the first example the local measurements of image flow based on the 3-D orientation estimation technique described in Section 4.1 in a sequence of images from the Meteosat satellite, are integrated using a Markov random field with a pixel prior. We will apply priors that involve continuity of the partial derivatives of the first order, as well as the second order. The use of second order derivatives is a generalization of the implementation of Konrad & Dubois (1992), who used energy functions based on first order derivatives as priors for motion fields, and the implementation of Kass et al. (1988), who used energy functions based on second order derivatives as priors for curves.

The second example serves to illustrate how it is possible to handle image motion discontinuities, either by choosing pixel priors that favor this, or by including edge priors. The first example is a realization of a model including a line process described by Konrad & Dubois (1992). We will, however, combine it with an

observation model based on the spatio temporal orientation observation. In the second example we will show a generalization of an algorithm for univariate reconstruction by Bouman & Sauer (1993). The priors of this algorithm are based on the p-norm of pixel value differences. We will generalize this to the estimation of the two dimensional motion field.

6.1 Temporal Interpolation in a Meteosat Satellite Sequence

6.1.1 Background

Meteosat is a geostationary satellite belonging to the European Space Agency. It is positioned over the Eastern Atlantic Ocean, (Colwell, 1983). The satellite covers Europe, Africa, and the Atlantic Ocean. Due to limited bandwidth it can not transmit more than one image every 30 minutes. The satellite has a visual and a thermal channel, and it scans the surface of the earth in concentric circles.

The images obtained from the satellite are used in preparation of weather forecasts, as well as illustration in TV weather reports. We have received 14 images of a sequence from the satellite covering Europe by the Danish Meteorological Institute (DMI). The images have been rectified by DMI so that they are in correspondence with commonly used maps. The spatial resolution is 498×512 pixels with a 8-bit grey scale resolution.

In Figure 6.1 and 6.2 some of the images from the sequence are shown. It is evident that due to the rotation of the earth relative to the sun, the visual channel is of little use when the earth is positioned between the sun and the satellite. Because of this we will consider only the thermal images. Note also the artifact in the upper right corner which is due to the rectification.

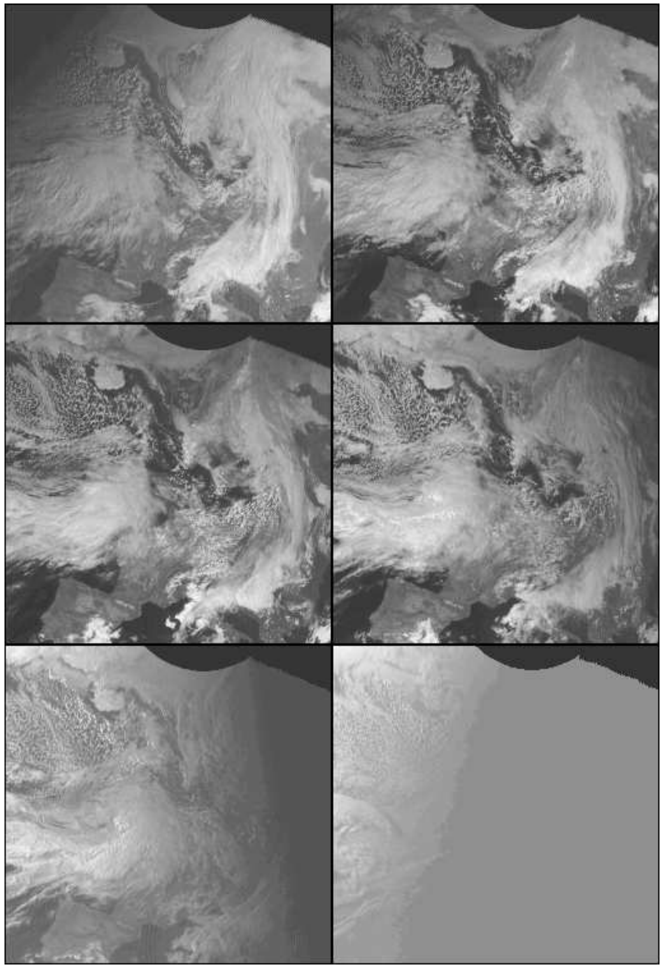


Figure 6.1: Images number 2, 4, 6, 8, 10, and 12 from the visual channel of the Meteosat sequence. Note that as Europe rotates away from the sun we loose signal.

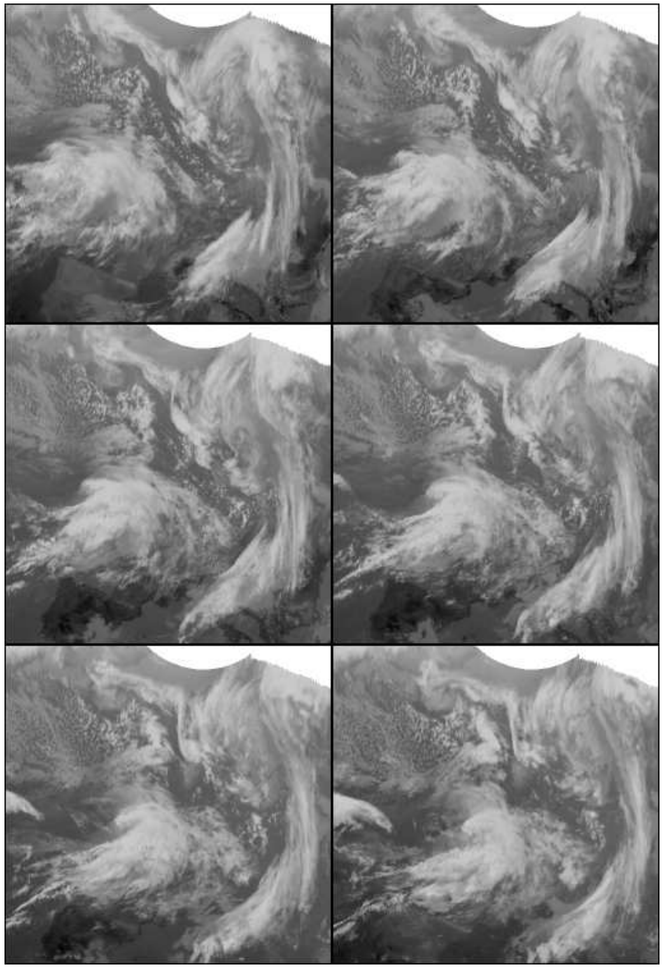


Figure 6.2: Images number 2, 4, 6, 8, 10, and 12 from the thermal channel of the Meteosat sequence.

6.1.2 The Algorithm

Prior Distribution

We choose to model the motion using a Markov random field. Although the clouds are expected to deform over time, we will assume that they evolve in such a coherent way that we can use the energy functions in Equation (5.21) and (5.22), i.e

$$\begin{aligned}&P(\boldsymbol{u}) \propto \exp(\\ &-\beta_1 \textstyle\sum_i [\|\boldsymbol{u}(x_i{+}1, y_i) - \boldsymbol{u}(x_i, y_i)\|^2 + \|\boldsymbol{u}(x_i, y_i{+}1) - \boldsymbol{u}(x_i, y_i)\|^2] \\ &-\beta_2 \textstyle\sum_i [\|\boldsymbol{u}(x_i{+}1, y_i) - 2\boldsymbol{u}(x_i, y_i) + \boldsymbol{u}(x_i{-}1, y_i)\|^2 + \\ &\qquad 2\|\boldsymbol{u}(x_i{+}1, y_i{+}1) - \boldsymbol{u}(x_i, y_i{+}1) - \boldsymbol{u}(x_i{+}1, y_i) + \boldsymbol{u}(x_i, y_i)\|^2 + \\ &\qquad \|\boldsymbol{u}(x_i, y_i{+}1) - 2\boldsymbol{u}(x_i, y_i) + \boldsymbol{u}(x_i, y_i{-}1)\|^2])\end{aligned} \tag{6.1}$$

This Gibbs distribution corresponds to a Markov random field, with a third order neighbourhood. β_1 is seen to control the penalty of high values of the gradient, and β_2 high values of the second order partial derivatives.

Observation Model

Due to the assumed smoothness of the motion field we will estimate the motion on a coarser scale than the original. We choose to subsample the original images by a factor four.

We use the algorithm for local measurement of the motion field based on the estimation of the local orientation in the spatio temporal domain, which was described in Section 4.1. The quadrature filters used have been implemented as $11 \times 11 \times 11$ convolution kernels. The filters have a center frequency of 1.1 and a bandwidth of two octaves, which by a visual inspection corresponds to the motion in the image sequence.

Given the true motion field $\boldsymbol{u}$, we now specify an observation model for these local measurements, $\boldsymbol{y}$. We will do this by a conditional Gibbs distribution

$$P(\boldsymbol{y}|\boldsymbol{u}) \propto \exp(-\alpha \sum_i \sum_{k=1}^{2} c_{ik} d_{ik}^2) \tag{6.2}$$

where d_{ik} is the difference between the projection of the true flow onto the normal given by the kth eigenvector at the ith pixel and the normal flow it self as described by Equation (4.4). c_{ik} is the certainty measure corresponding to this normal flow given by Equation (4.5).

Posterior Distribution

The a posteriori distribution

$$P(\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{u})P(\boldsymbol{u}) \tag{6.3}$$

is obviously a new Markov random field, that reflects a trade-off between smoothness of the field and our trust in the local measurements. The energy of the posterior distribution is given by

$$\begin{aligned}&U = \alpha \textstyle\sum_i \sum_{k=1}^2 c_{ik} d_{ik}^2 + \\ &\beta_1 \textstyle\sum_i [\|\boldsymbol{u}(x_i{+}1, y_i) - \boldsymbol{u}(x_i, y_i)\|^2 + \|\boldsymbol{u}(x_i, y_i{+}1) - \boldsymbol{u}(x_i, y_i)\|^2] + \\ &\beta_2 \textstyle\sum_i [\|\boldsymbol{u}(x_i{+}1, y_i) - 2\boldsymbol{u}(x_i, y_i) + \boldsymbol{u}(x_i{-}1, y_i)\|^2 + \\ &\qquad 2\|\boldsymbol{u}(x_i{+}1, y_i{+}1) - \boldsymbol{u}(x_i, y_i{+}1) - \boldsymbol{u}(x_i{+}1, y_i) + \boldsymbol{u}(x_i, y_i)\|^2 + \\ &\qquad \|\boldsymbol{u}(x_i, y_i{+}1) - 2\boldsymbol{u}(x_i, y_i) + \boldsymbol{u}(x_i, y_i{-}1)\|^2]\end{aligned} \tag{6.4}$$

In this energy function we can control the properties of estimated motion field. The smoothness is controlled by β_1, and β_2, and the faith in data is controlled by α.

We can now apply a maximization scheme to the posterior distribution in order to obtain the maximum a posteriori estimate of the motion field as described in

Section 5.2.5. This has been implemented using Besag (1986)'s ICM, as well as a conjugate gradient descent routine. The ICM routine has the advantage of being implementable of a parallel computer due to the local approach as opposed to the global approach of a gradient descent routine.

6.1.3 Results of the Motion Field Estimation

Figures 6.3 and 6.4 show the estimated motion field corresponding to image number eight in the sequence. The motion field has been subsampled by a factor two. For the first field we used the parameters $\beta_1 = 0.75$, $\beta_2 = 0$, and $\alpha = 0.25$, and for the second we used the parameters $\beta_1 = 0.5$, $\beta_2 = 0$, and $\alpha = 0.5$. In both cases we iterated until the average change of the flow vectors during one sweep of the entire image was below 0.001. This corresponded to 91 and 97 ICM iterations, respectively.

In Figures 6.5 the motion field estimated using the parameters $\beta_1 = 0$, $\beta_2 = 0.5$, and $\alpha = 0.5$, i.e. a prior energy function penalizing second order derivatives only is seen. When compared to Figure 6.4 it is evident that this model model allows more abrupt changes in the field. The combined model in Figure 6.6 has properties of both models.

6.1.4 Temporal Interpolation

In order to produce temporal interpolation between two images in the sequence, we take the weighted average of the extrapolation of each image (one forwards and one backwards). The weights are inversely proportional to the time difference to the original image.

We extrapolate from the original images using the estimated flow field, the thus generated image is sampled to the original grid using a nearest neighbor interpolation.

In Figure 6.7 we see the result of a temporal interpolation between images number 7 and 8 in the original sequence.

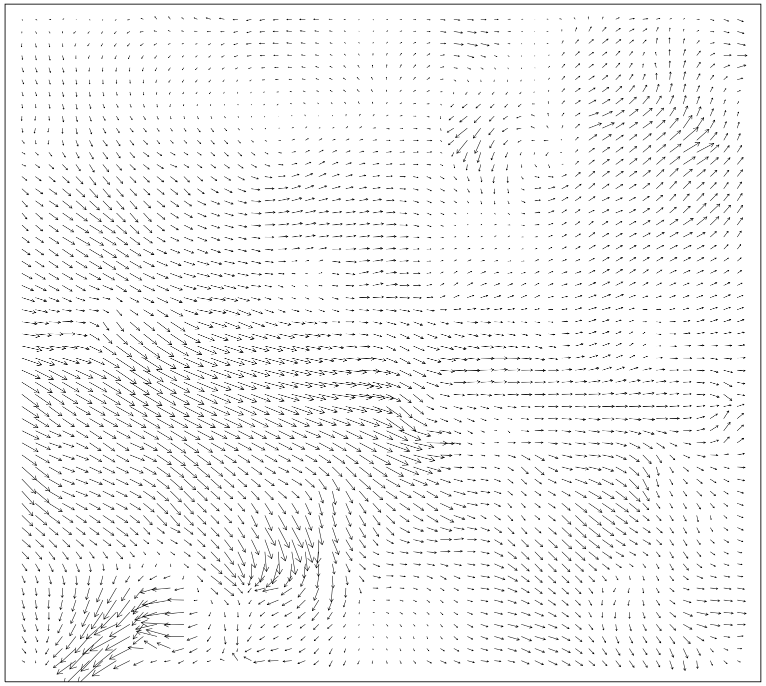


Figure 6.3: The flow field corresponding to image number nine in the Meteosat sequence. The parameters used were $\beta_1 = 0.75$, $\beta_2 = 0$, and $\alpha = 0.25$.

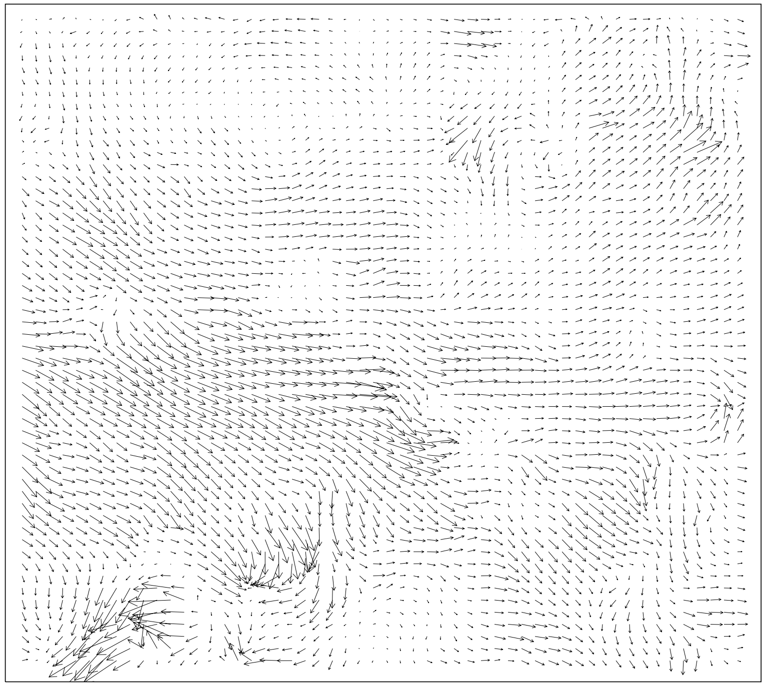


Figure 6.4: The flow field corresponding to image number eight in the Meteosat sequence. The parameters used were $\beta_1 = 0.5$, $\beta_2 = 0$, and $\alpha = 0.5$.

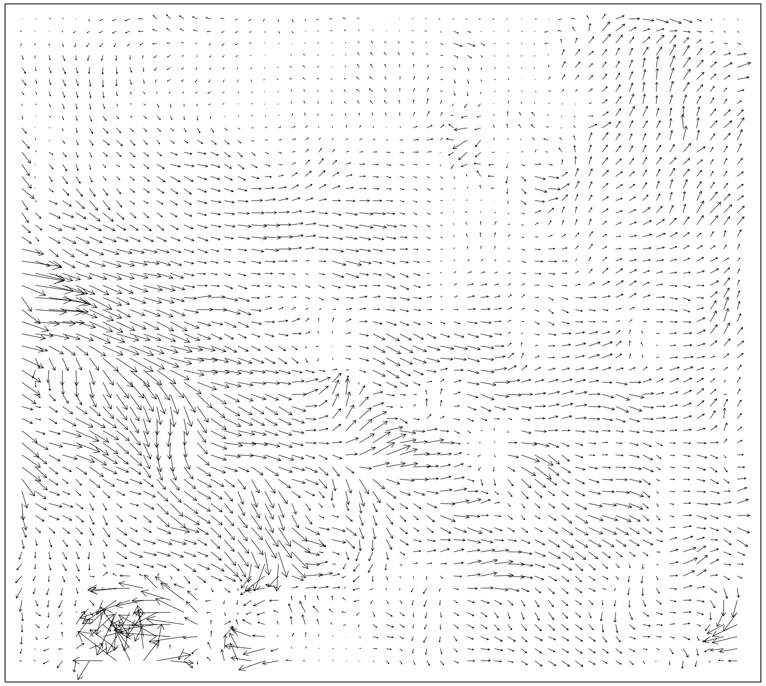


Figure 6.5: The flow field corresponding to image number eight in the Meteosat sequence. The parameters used were $\beta_1 = 0$, $\beta_2 = 0.5$, and $\alpha = 0.5$.

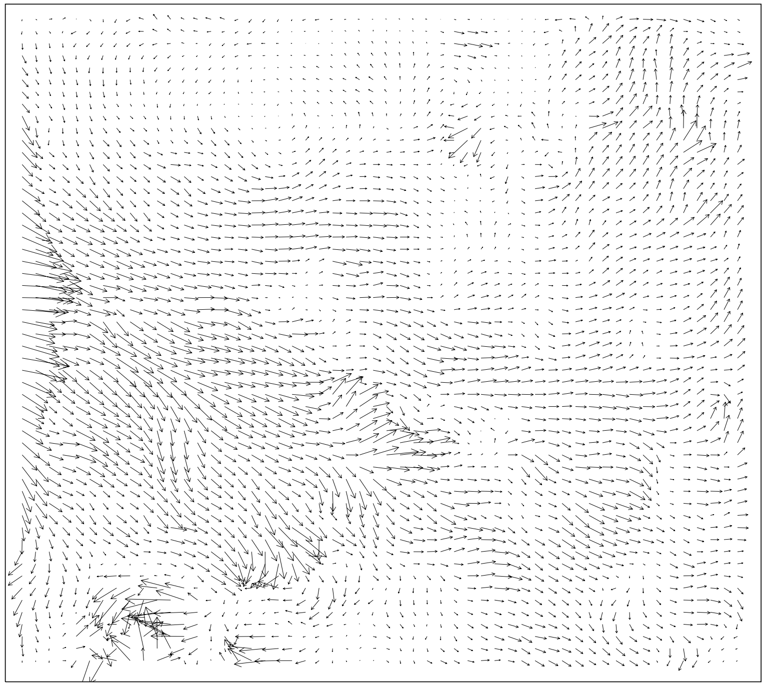


Figure 6.6: The flow field corresponding to image number eight in the Meteosat sequence. The parameters used were $\beta_1 = 0.25$, $\beta_2 = 0.25$, and $\alpha = 0.5$.

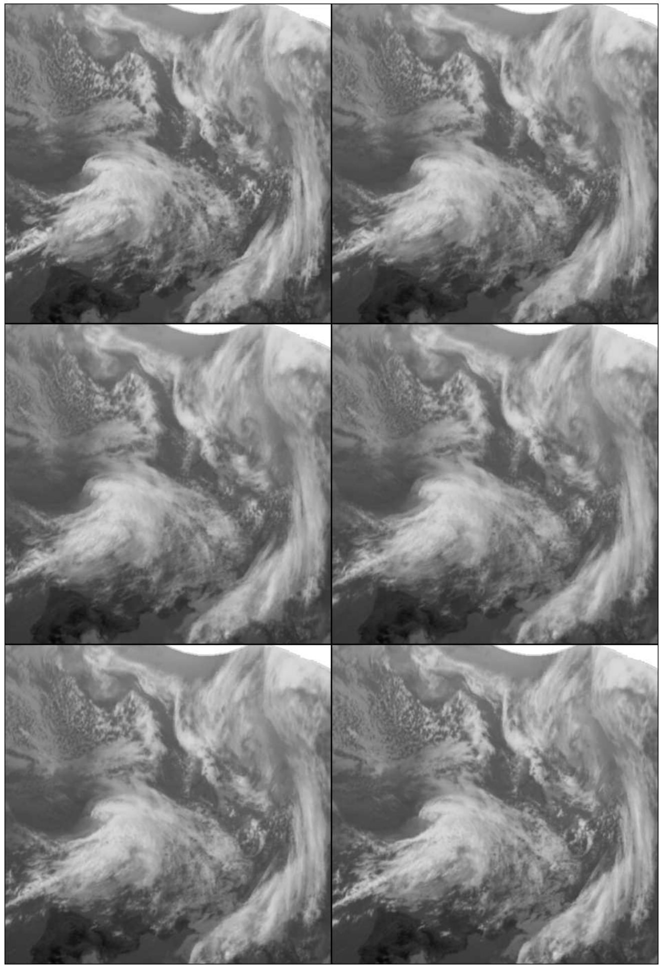


Figure 6.7: A temporal interpolation between images number 7 in the upper left corner and image number 8 in the lower right corner.

6.2 Handling Discontinuities

In this section we will present some approaches to the estimation of piecewise continuous motion fields. We will use the combination of the pixel process described in the previous section and a line process as described by Konrad & Dubois (1992). Again we will use an observation model based on the estimation of the spatio temporal orientation.

6.2.1 Inclusion of a Line Process

Prior Distribution

We will use a pixel process with a two dimensional continuous state space coupled with a binary line process, as described in Section 5.2.2. For the pixel process we will use an energy function that penalizes high values of the first order spatial derivatives, only this should not be enforced across discontinuities. In order to obtain this we will use a neighbourhood system for the pixel process given by the four nearest neighbors of a pixel supplemented with the four edge positions given by the midpoints of the four binary pixel cliques. This can be seen in Figure 6.8.

As for the line field model we will use neighbourhood system suggested by Konrad & Dubois (1992). This system consists of the system we described in Section 5.2.2 expanded to include cliques that enable us to penalize parallel lines, i.e. double edges. This second-order neighbourhood system is shown in Figure 6.9. Note that we, due to having both horizontal and vertical line elements, have two neighbourhood systems, see Figures 6.9(a) and 6.9(b). There are two types of four element line cliques. The cross-shaped cliques from Figure 6.9(c) are used to model the shape of motion boundaries, whereas the square-shaped cliques from Figure 6.9(d) are used to inhibit isolated vectors. The two-element horizontal and vertical cliques in Figures 6.9(e) and 6.9(f) are used to prevent double edges. We will let the line process take the value 1 in case of a discontinuity present, and 0 if no discontinuity is present.

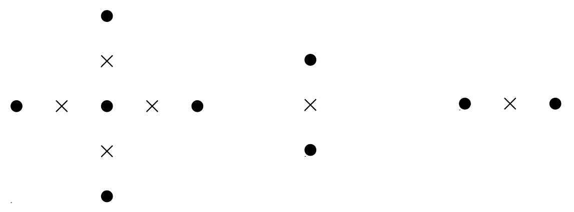


Figure 6.8: (a) First-order neighbourhood system for motion field $\boldsymbol{u}$ with discontinuities $\boldsymbol{l}$; (b) vertical cliques; (c) horizontal cliques (•–vector site, ×–line site).

The potentials associated with the various two- and four-element clique configurations are tabulated in Figure 6.10. Note from Figures 6.10(a)–6.10(f), that we apply small penalties for straight lines and high penalties for intersections. The square shaped clique configuration potential in Figure 6.10(g) prohibit isolated points. Figures 6.10(h)–6.10(k) show potentials to penalize double edges.

This results in the following prior distribution

$$p(\boldsymbol{u}, \boldsymbol{l}) \propto \exp(-\beta_1 \sum_{i \sim j} \|(\boldsymbol{u}_i - \boldsymbol{u}_j)\|^2 (1 - \boldsymbol{l}_{ij}) - \beta_2 \sum_{c_l} V_{c_l}(\boldsymbol{l})) \tag{6.5}$$

where V_{c_l} is the clique potentials shown in Figure 6.10, and the sum is over all cliques.

Observation Model

For the pixel process we will employ the same observation model as in the Meteosat case in the previous section.

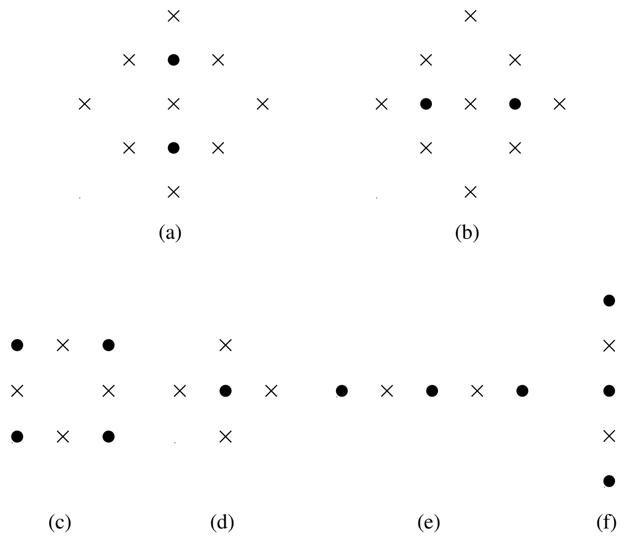


Figure 6.9: Second-order neighbourhood system for the line process. (a) Horizontal line element; (b) Vertical line element; (c),(d) Four-element cliques; (e),(f) Two-element cliques; (•–vector site, ×–line site).

We will, however, include a term that reflects that motion discontinuities in general will coincide with positions with dominant orientations, e.g edges. The observation model thus becomes.

$$P(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{l}) \propto \exp(-\alpha_1 \sum_i \sum_{k=1}^{2} c_{ik} d_{ik}^2 - \alpha_2 \sum_{i \sim j} \frac{\boldsymbol{l}_i j}{c_{i1} + c_{i2}}) \tag{6.6}$$

Posterior Distribution

The energy function of the posterior distribution is the sum of the energy functions in Equation (6.5) and (6.6). From the description above we can see that α_1

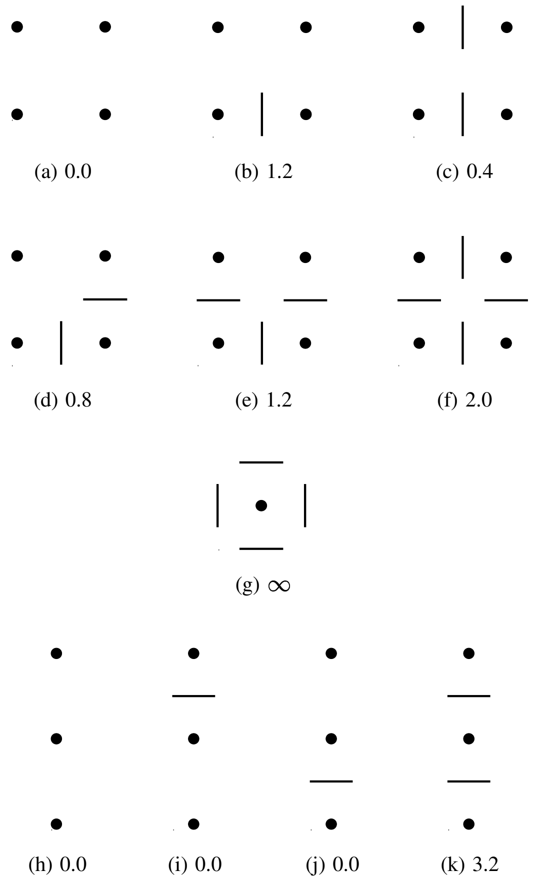


Figure 6.10: Potentials associated with various configurations (up to a rotation): (a)-(f) four-element cross shaped cliques; (g) four element square shaped clique; (h)-(k) two-element cliques (•–vector site, — –line element "on").

controls the faith in our local measurements of motion, α_2 controls the penalty for setting a discontinuity, β_1 controls the piecewise smoothness of the motion field, and β_2 is a control parameter for the discontinuity configurations.

Due to the presence of discontinuities in the field the MAP estimation results in a object function with many local extrema. To handle this we will employ a simulated annealing scheme as described in Section 5.2.2.

Results

In order to demonstrate the ability of this scheme to estimate discontinuous motion fields, we will use the sequence shown in Figure 6.11. In Figure 6.12 the problems concerning discontinuities is shown. The motion field has been estimated using the model without a line process. It is evident that we have considerable over smoothing at the motion edges.

In Figure 6.13 we see the estimated flow field using the model including the line process. It is evident that we have obtained a good distinction between the moving patch and the background. The estimated discontinuities are seen in Figure 6.14. One effect of the process is that it tends to isolate some local measurements at the motion borders. These isolated local measurements are typically subject to grave errors, because of the ambiguous neighbourhoods.

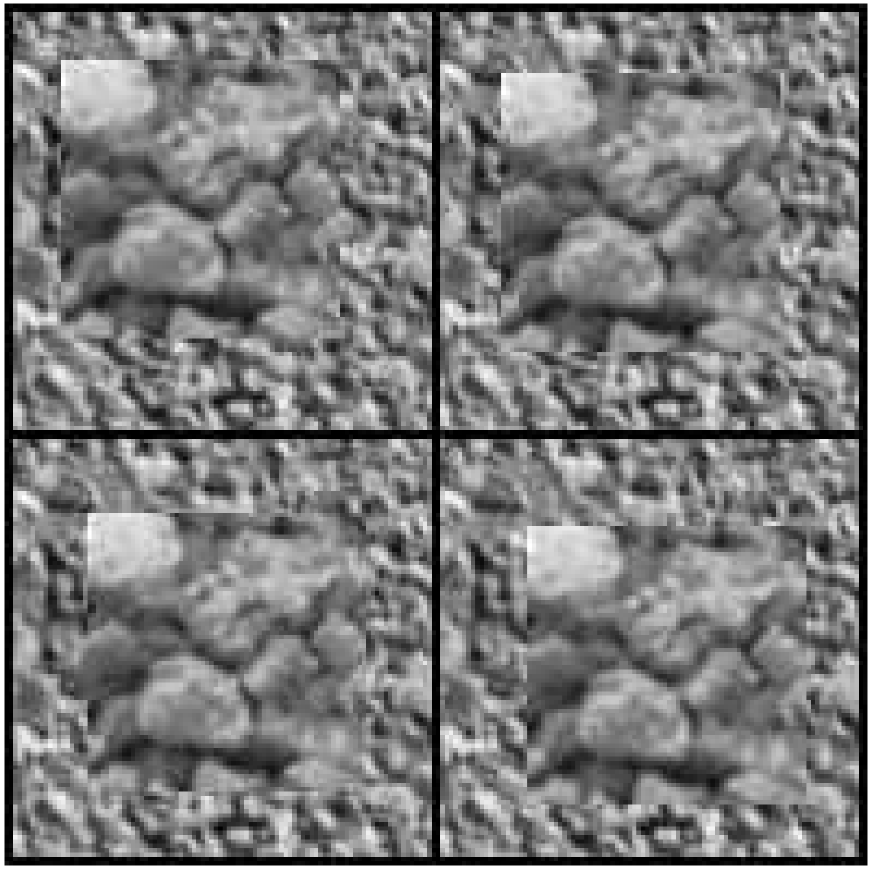


Figure 6.11: This image sequence is generated from two textures from the Brodatz (1966) catalog. We have a patch of expanded mica on a background consisting of pressed cork translating along the diagonal of the image with a constant velocity of $(1, 1)$. In this figure images numbers 1, 4, 7, and 10 from the sequence are shown.

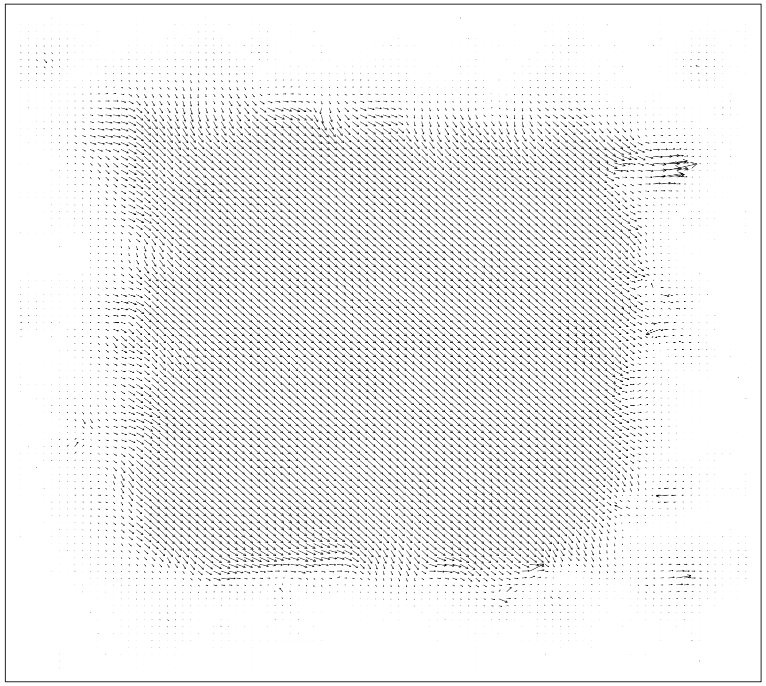


Figure 6.12: This is the motion field estimated without a line process. We see that the motion field is over smoothed along the edges of the patch.

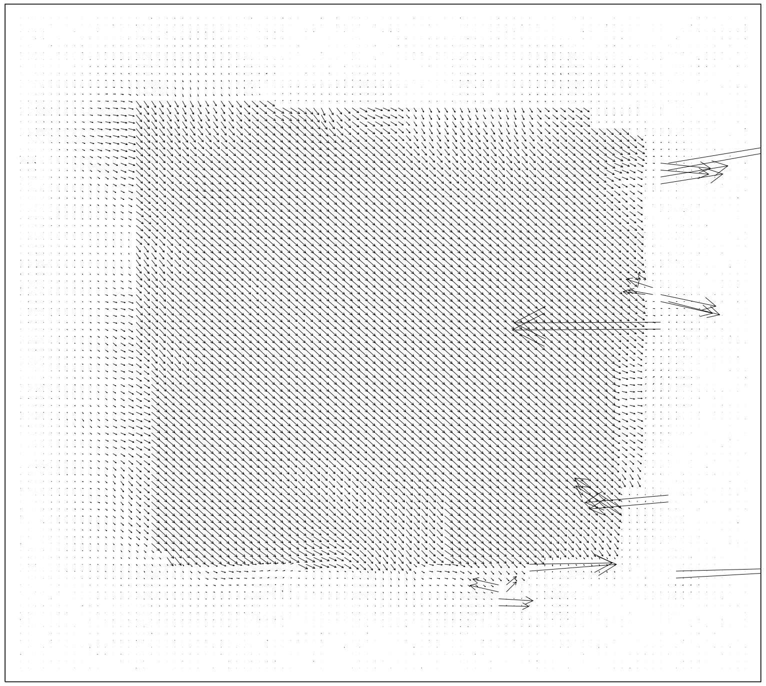


Figure 6.13: This is the motion field estimated including a line process. We see a good distinction between the moving and non moving part of the image.

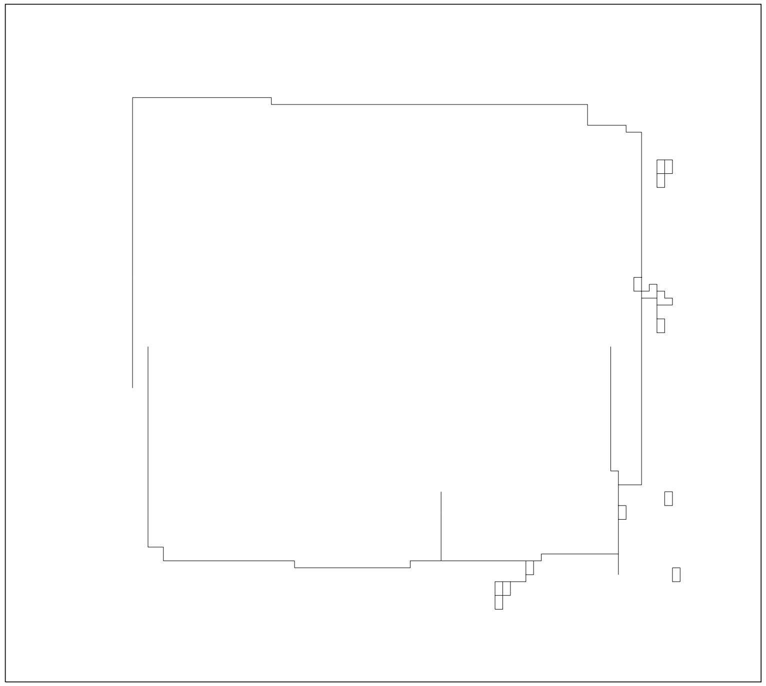


Figure 6.14: These are the estimated discontinuities in the motion field.

6.2.2 Using a Pixel Prior Based on the p-norm

Prior Distribution

Since the problem of the Gaussian pixel prior whose energy function is shown in Equation (5.21) consists of over smoothing at the edges, as shown in Figure 6.12, we might consider a pixel prior that handles discontinuities better. The median pixel prior has its mode at the median, which should make it able to preserve edges. A more general prior, however, is the one whose energy function is based on the p-norm

$$p(\boldsymbol{u}, \boldsymbol{l}) \propto \exp(-\beta \sum_{i \sim j} \|(\boldsymbol{u}_i - \boldsymbol{u}_j)\|_p^p), \qquad i \leq p \leq 2 \tag{6.7}$$

This prior furthermore has the nice property of being differentiable for $p > 1$.

Observation Model

For the pixel process we will employ the same observation model as in the two previous cases. The observation model thus becomes.

$$P(\boldsymbol{y}|\boldsymbol{u}) \propto \exp(-\alpha \sum_i \sum_{k=1}^{2} c_{ik} d_{ik}^2) \tag{6.8}$$

Posterior Distribution

The energy function of the posterior distribution is the sum of the energy functions in Equation (6.7) and (6.8). From the description above we can see that α controls the faith in our local measurements of motion, and β controls the smoothness of the motion field.

We have employed a gradient descent routine to the maximization of the posterior probability.

Results

Again we will use the sequence shown in Figure 6.11 in order to demonstrate the ability of this scheme to estimate discontinuous motion fields.

In Figures 6.15 and 6.16 the motion field using $p = 1.2$, and $p = 1.1$, respectively, along with the parameters $\alpha = 0.5$, and $\beta = 0.5$ is shown. It is clearly seen that the the motion discontinuities are better preserved than when we used using the Gaussian prior. The result of which was shown in Figure 6.12.

One effect of the ability of this model to preserve edges is that isolated measurements with high certainty and deviating values are not smoothed away.

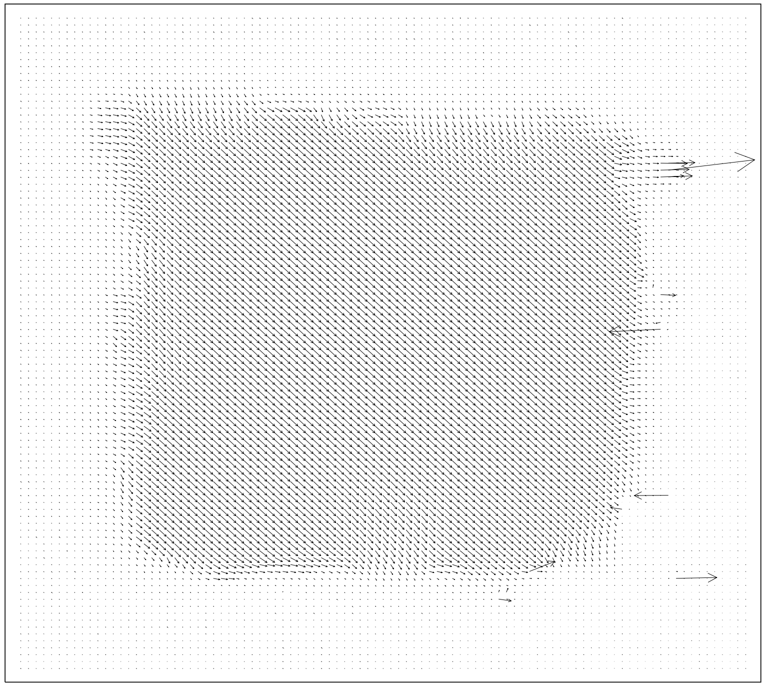


Figure 6.15: This is the motion field estimated using the pixel prior based on the p-norm. We have used $p = 1.2$, and the parameters $\alpha = 0.5$, and $\beta = 0.5$. We see that the motion field is less smoothed at the motion boundaries than in the case of the Gaussian prior used in Figure 6.12.

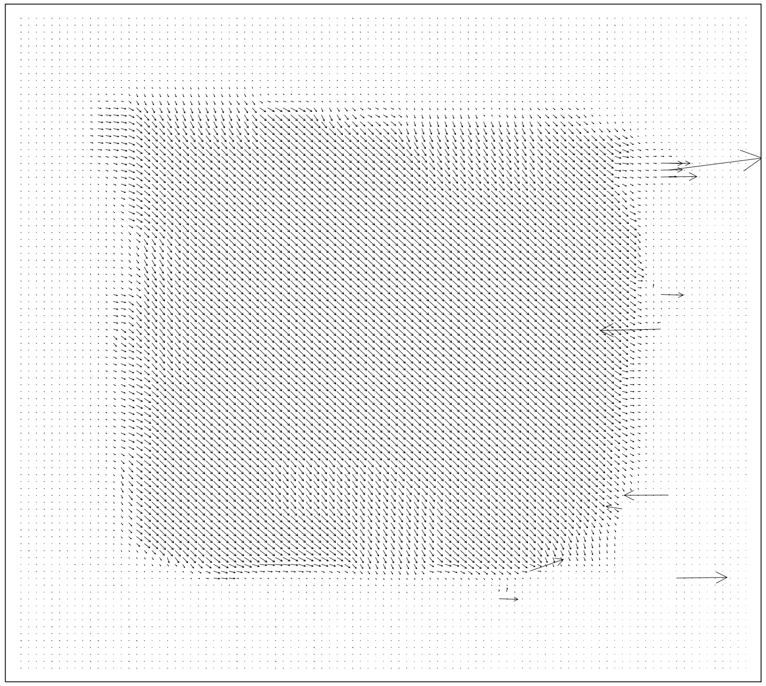


Figure 6.16: This is the motion field estimated using the pixel prior based on the p-norm. We have used $p = 1.1$, and the parameters $\alpha = 0.5$, and $\beta = 0.5$. We see that the motion field is even smoother than in the case of $p = 1.2$

Chapter 7

Conclusion

Motion is an important characteristic for visual phenomena. The literature contains a multitude of descriptions of attempts to estimate the visual motion as perceived in a sequence of images. This thesis contains a contribution to these attempts by going through selective theory and practical applications.

7.1 Summary

We have based our analysis on a framework consisting of three stages of processing

1. Extraction of motion invariants.
2. Local measurement of visual motion.
3. Integration of local measurements in conjunction with a priori knowledge.

We have surveyed a series of attempts to extract motion invariants. Approaches that extract sparsely distributed points and curves, as well as approaches that

aim at providing dense fields of invariants have been considered. Specifically we have illustrated the use of local Fourier phase, estimated in a number of directions evenly distributed in the spatio temporal domain. The phase has the attractive property of precisely localizing an event. Methods based on intensity and derivatives of intensity have also been considered.

With respect to locally measuring the visual motion we have surveyed methods based on region matching, gradient approaches, frequency approaches, and feature tracking. In particular we have described techniques for quantifying the directional certainty with which we estimated the local motion. In two cases, one based on the conservation of bandpass filtered intensity used in a frequency based estimation method, and the other one based on the conservation of Fourier phases as estimated using a set of quadrature filters, we have suggested measures of certainty.

We have applied an algorithm based on the conservation of Fourier phases using a gradient based technique to the estimation of the visual motion in a natural scene containing moving vehicles. Without the use of a prior knowledge, save an assumption of the motion being on a scale comparable with the spatial resolution, we have identified the moving vehicles by their motion.

The, in motion analysis, seldom used distance transformation has been applied in a feature tracking scheme. We have described a new algorithm, which in a special kind of double exposed images can be used to estimate the motion. Spatial coherence in the flow is assumed. The double exposed images are produced using a technique for estimating flow in fluids called Particle Image Velocimetry.

With respect to integration of local measurements of visual motion we have surveyed different approaches within a Bayesian framework. The description contains generalization of univariate Markov random fields models to two dimensional Markov random fields used to characterize motion fields. Standard integration of measurements are based of a smoothing that penalizes first order derivatives of the field. We have investigated how we might use second order derivatives, and how this affects the estimated fields.

Furthermore, we have described techniques to overcome the problem of over smoothing at motion discontinuities. A technique based on the estimation of a coupled pixel and line process in a Markov random field setting combined with a local measurements technique based of the spatio temporal orientation has been implemented. We have also investigated prior distributions of the motion field that tend to preserve motion boundaries, this includes a generalization to the multivariate case of a technique used in image restoration based on the p-norm.

Appendix A

Curvature

The term curvature is a quantitative characteristic of the degree to which some object, e.g a surface, deviates from some other object, e.g the tangent plane of the surface. Specifically we want to use the curvature to quantitatively describe the local intensity patterns of an image in terms of homogeneity, collinearity, cornerness. Let us consider the surface $\boldsymbol{S}(\boldsymbol{x})$ in three-dimensional Euclidean space associated with the grey level intensity image $I(\boldsymbol{x})$

$$\boldsymbol{S}(\boldsymbol{x}) = (\boldsymbol{x}, I(\boldsymbol{x})) \tag{A.1}$$

The concepts of curvature which we will be examining are connected with deviations which are small to the second order; hence we will assume $\boldsymbol{S}$ to be a C^2-surface.

The squared distance between two infinitesimally close points $(\boldsymbol{x})$ and $(\boldsymbol{x}+\boldsymbol{dx})$ on the surface $\boldsymbol{S}$ can be expressed by

$$\boldsymbol{dSdS} = \boldsymbol{dx}^T \begin{bmatrix} E & F \\ F & G \end{bmatrix} \boldsymbol{dx} \tag{A.2}$$

where the elements of the matrix which we will denote $\boldsymbol{G}$ are defined as

$$\begin{aligned} E &= \boldsymbol{S}_x\boldsymbol{S}_x \\ F &= \boldsymbol{S}_x\boldsymbol{S}_y \\ G &= \boldsymbol{S}_y\boldsymbol{S}_y \end{aligned}$$

This equation is known as the first fundamental form of the surface $\boldsymbol{S}(\boldsymbol{x})$. For the surface given by Equation (A.1) the coefficients E, F, and G reduce to

$$\begin{aligned} E &= 1 + I_x(\boldsymbol{x})I_x(\boldsymbol{x}) \\ F &= I_x(\boldsymbol{x})I_y(\boldsymbol{x}) \\ G &= 1 + I_y(\boldsymbol{x})I_y(\boldsymbol{x}) \end{aligned}$$

The first fundamental form gives the distance between the two points $\boldsymbol{x}$ and $\boldsymbol{x}+\boldsymbol{dx}$ measured on the tangent plane of the surface at $\boldsymbol{x}$. In order to describe how the surface curves at $\boldsymbol{x}$ we will consider the distance perpendicular to the tangent plane at $\boldsymbol{x}$ of a point on the surface at $\boldsymbol{x}+\boldsymbol{dx}$. This distance to the second order in $\boldsymbol{dx}$ is given by

$$-\boldsymbol{dSdN} = \boldsymbol{dx}^T \begin{bmatrix} L & M \\ M & N \end{bmatrix} \boldsymbol{dx} \tag{A.3}$$

where $\boldsymbol{N}$ is a positive normal unit vector to the surface at $\boldsymbol{x}$

$$\boldsymbol{N} = \frac{\boldsymbol{S}_x \times \boldsymbol{S}_y}{\|\boldsymbol{S}_x \times \boldsymbol{S}_y\|} \tag{A.4}$$

and the elements of this matrix which we will denote $\boldsymbol{H}$ are defined as

$$
\begin{aligned}
L &= \boldsymbol{N}\boldsymbol{S}_{xx} \\
M &= \boldsymbol{N}\boldsymbol{S}_{xy} \\
N &= \boldsymbol{N}\boldsymbol{S}_{yy}
\end{aligned}
$$

Equation (A.3) is known as the second fundamental form of the surface $\boldsymbol{S}(\boldsymbol{x})$. For our surface given by Equation (A.1) the coefficients L, M, and N reduce to

$$
\begin{aligned}
L &= \frac{I_{xx}(\boldsymbol{x})}{\sqrt{1 + I_x(\boldsymbol{x})^2 + I_y(\boldsymbol{x})^2}} \\
M &= \frac{I_{xy}(\boldsymbol{x})}{\sqrt{1 + I_x(\boldsymbol{x})^2 + I_y(\boldsymbol{x})^2}} \\
N &= \frac{I_{yy}(\boldsymbol{x})}{\sqrt{1 + I_x(\boldsymbol{x})^2 + I_y(\boldsymbol{x})^2}}
\end{aligned}
$$

Now, at each point of the surface, $\boldsymbol{S}$, the normal curvature κ in the direction given by $\boldsymbol{dx}$ is the ratio between the second and the first fundamental form.

$$
\kappa = \frac{\boldsymbol{dx}^T \boldsymbol{H} \boldsymbol{dx}}{\boldsymbol{dx}^T \boldsymbol{G} \boldsymbol{dx}} \tag{A.5}
$$

Of special interest is the minimum and maximum values of curvature and the directions in which these are attained. These curvatures are normally referred to as the principal curvatures and the principal directions of the surface. Evidently, they are given by the eigenvalues and eigenvectors of $\boldsymbol{H}$ with respect to $\boldsymbol{G}$. Trivial computation leads to

$$
\begin{aligned}
\kappa_{\text{max}} &= H + \sqrt{H^2 - K} \\
\kappa_{\text{min}} &= H - \sqrt{H^2 - K}
\end{aligned}
$$

where the coefficients H and K denote the Gaussian curvature and mean curvature, respectively

$$
\begin{aligned}
K &= \frac{LN - M^2}{EG - F^2} \\
H &= \frac{1}{2}\frac{EN + GL - 2FM}{EG - F^2}
\end{aligned} \tag{A.6}
$$

The Gaussian and mean curvature can easily be expressed as the product and the mean of the principal curvatures

$$
\begin{aligned}
K &= \kappa_{\text{max}}\kappa_{\text{min}} \\
H &= \frac{\kappa_{\text{max}} + \kappa_{\text{min}}}{2}
\end{aligned}
$$

which is equivalent to saying that K and H are the determinant and half the trace of the matrix $\boldsymbol{G}^{-1}\boldsymbol{H}$.

In the case of a stationary point on the surface the Gaussian and mean curvature are, respectively

$$
\begin{aligned}
K_{\text{stat}} &= I_{xx}(\boldsymbol{x}) I_{yy}(\boldsymbol{x}) - I_{xy}(\boldsymbol{x}) \\
H_{\text{stat}} &= \frac{1}{2}(I_{xx}(\boldsymbol{x}) + I_{yy}(\boldsymbol{x}))
\end{aligned}
$$

Appendix B

Phase Images

This appendix contains the quadrature filter outputs of 6 filters described in Section 3.3.3 applied to the Hamburg taxi sequence. The six images shown for each filter corresponds to the images of the original sequence shown in Figure 4.3. The images are shown with the magnitude of the filters as intensity, and the phase shown on a circular color scale.

Figure B.1: Quadrature filter output from filter number 1 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(2, 0, 1 + \sqrt{5})$, i.e. to movements in the x-t plane going slowly from left to right.

Figure B.2: Quadrature filter output from filter number 2 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(-2, 0, 1+\sqrt{5})$, i.e. to movements in the x-t plane going fast from right to left.

Figure B.3: Quadrature filter output from filter number 3 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(1+\sqrt{5}, 2, 0)$, i.e. to contours in the x-y plane that go in the direction from the upper left to the lower right.

Figure B.4: Quadrature filter output from filter number 4 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(1+\sqrt{5}, -2, 0)$, i.e. to contours in the x-y plane that go in the direction from the lower left to the upper right.

Figure B.5: Quadrature filter output from filter number 5 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(0, 1+\sqrt{5}, 2)$, i.e. to movements in the y-t plane going fast downwards in the image.

Figure B.6: Quadrature filter output from filter number 6 of Equation (3.34). This filter is tuned to the spatio-temporal direction $(0, 1 + \sqrt{5}, -2)$, i.e. to movements in the y-t plane going slowly upwards in the image.

References

Aarts, E. & Korst, J. (1989). *Simulated Annealing and Boltzmann Machines*. J. Wiley & Sons, New York. 272 pp.

Adelson, E. H. & Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America, Series A*, **2**(2), 284–299.

Aisbett, J. (1989). Optical flow with an intensity-weighted smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(5), 512–522.

Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, **2**, 283–310.

Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis* (second edition). John Wiley, New York. 675 pp.

Arnspang, J. (1988). Notes on local determination of smooth optic flow and the translational property of first order optic flow. Tech. rep. 1, Institute of Datalogy, University of Copenhagen. 26 pp.

Astola, J., Haavisto, P., & Neuvo, Y. (1990). Vector median filters. *Proceedings of the IEEE*, **78**(4), 678–689.

Beaudet, P. R. (1978). Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, pp. 579–583.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, **48**(3), 259–302.

Besag, J. (1989). Towards Bayesian image analysis. *Journal of Applied Statistics*, **16**(3), 395–407.

Bigün, J. (1988). *Local Symmetry Features in Image Processing*. Ph.D. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden. 104 pp.

Borgefors, G. (1988). Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6), 849–865.

Bouman, C. & Sauer, K. (1993). A generalized gaussian image model for edge-preserving MAP estimation. *IEEE Transactions on Image Processing*, **2**(3), 296–310.

Bouthemy, P. & François, E. (1993). Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, **10**(2), 157–182.

Brodatz, P. (1966). *Textures - A Photographic Album for Artists and Designers*. Dover, New York.

Burt, P. J. (1981). Fast filter transforms for image processing. *Computer Graphics and Image Processing*, **16**, 20–51.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6), 679–698.

Carstensen, J. M. (1992). *Description and Simulation of Visual Texture*. Ph.D. thesis, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Lyngby. 234 pp.

Colwell, R. N. (Ed.). (1983). *Manual of Remote Sensing* (2 edition)., Vol. 1. American Society of Photogrammetry, Falls Church, Virginia. 1232 pp.

Deriche, R. & Faugeras, O. (1990). Tracking line segments. *Image and Vision Computing*, **8**(4), 261–270.

Deriche, R. & Giraudon, G. (1993). A computational approach for corner and vertex detection. *International Journal of Computer Vision*, **10**(2), 101–124.

Dubes, R. C. & Jain, A. K. (1989). Random field models in image analysis. *Journal of Applied Statistics*, **16**(2), 131–164.

Duncan, J. H. & Chou, T.-C. (1992). On the detection of motion and the computation of optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(3), 346–352.

Faugeras, O. & Papadopoulo, T. (1993). A theory of the motion fields of curves. *International Journal of Computer Vision*, **10**(2), 125–156.

Fennema, C. L. & Thompson, W. B. (1979). Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, **9**, 301–315.

Fleet, D. J. & Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, **5**(1), 77–104.

Fleet, D. J., Jepson, A. D., & Jenkin, M. R. M. (1991). Phase-based disparity measurement. *Computer Vision, Graphics and Image Processing*, **53**(2), 71–210.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(6), 721–741.

Geman, D., Geman, S., Graffigne, C., & Dong, P. (1990). Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(7), 609–628.

Geman, D. (1990). Random fields and inverse problems in imaging. In *Saint-Flour lectures 1988*, Lecture Notes in Mathematics, pp. 113–193. Springer-Verlag.

Haglund, L. (1992). *Adaptive Multidimensional Filtering*. Ph.D. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden. 158 pp.

Hansen, O. (1991). *On the use of Local Symmetries in Image Analysis and Computer Vision*. Ph.D. thesis, Laboratory of Image Analysis, Institute of Electronic Systems, University of Aalborg, Denmark. 128 pp.

Heeger, D. J. (1987). Optical flow from spatio-temporal filters. In *First International Conference on Computer Vision*, pp. 181–190 London.

Heitz, F. & Bouthemy, P. (1993). Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(12), 1217–1232.

Herlin, I. L. & Ayache, N. (1992). Features extraction and analysis methods for sequences of ultrasound images. *Image and Vision Computing*, **10**(10), 673–682.

Hildreth, E. C. (1984). Computations underlying the measurement of visual motion. *Artificial Intelligence*, **23**, 309–354.

Horn, B. K. P. & Schunk, B. G. (1981). Determining optical flow. *Artificial Intelligence*, **17**, 185–203.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, **1**, 321–331.

Kearney, J. K., Thompson, W. B., & Boley, D. L. (1987). Optical flow estimation: an error analysis of gradient-based methods with local optomization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**(2), 229–244.

Kervrann, C. & Heitz, F. (1993). A hierarchical statistical framework for the segmentation of deformable objects in image sequences. Tech. rep. 2133, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis, France.

Kitchen, L. & Rosenfeld, A. (1982). Grey-level corner detection. *Pattern Recognition Letters*, **1**(2), 95–102.

Knutsson, H. (1982). *Filtering and Reconstruction in Image Processing*. Ph.D. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Knutsson, H. (1989). Representing local structure using tensors. In Pietikäinen, M. & Röning, J. (Eds.), *The 6th Scandinavian Conference on Image Analysis*, pp. 244–251 Oulu, Finland. The Pattern Recognition Society of Finland.

Konrad, J. & Dubois, E. (1992). Bayesian estimation of motion vector fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(9), 910–927.

Langley, K., Atherton, T. J., Wilson, R. G., & Larcombe, M. H. E. (1991). Vertical and horizontal disparities from phase. *Image and Vision Computing*, **9**(5), 296–302.

Marr, D. & Ullman, S. (1981). Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London*, **211**, 151–180.

Murray, D. W. & Buxton, B. F. (1984). Reconstructing the optical flow field from edge motion: an examination of two different approaches. In *The First Conference on Artificial Intelligence Applications*, pp. 382–388 Sheraton, Denver, USA.

Murray, D. W. & Buxton, B. F. (1987). Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**(2), 220–228.

Nagel, H.-H. & Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(5), 565–593.

Nagel, H.-H. (1983). Displacement vectors derived form second-order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, **21**, 85–117.

Noble, J. A. (1988). Finding corners. *Image and Vision Computing*, **6**(2), 121–128.

Paquin, R. & Dubois, E. (1983). A spatio-temporal gradient method for estimating the displacement field in time-varying imagery. *Computer Vision, Graphics and Image Processing*, **23**, 205–221.

Ragnemalm, I. (1993). *The Euclidean Distance Transform*. Ph.D. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden. 276 pp.

Ripley, B. D. (1988). *Statistical inference for spatial processes*. Cambridge University Press. 148 pp.

Rosenfeld, A. & Kak, A. C. (1976). *Digital Image Processing* (first edition). Academic Press, New York. 457 pp.

Schunk, B. G. (1989). Image flow segmentation and estimation by constraint line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(10), 1010–1027.

Simpson, J. J. & Gobat, J. I. (1993). Robust velocity estimates, stream functions, and simulated lagrangian drifters from sequential spacecraft data. Tech. rep. 1, Digital Image Analysis Laboratory, Scripps Institute of Oceanography, La Jolla, California. 31 pp.

Singh, A. (1991). *Optic Flow Computations*. IEEE Computer Society Press, Los Alamitos, California. 243 pp.

Terzopoulos, D. (1986). Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(4), 413–424.

Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(4), 417–438.

Uras, S., Girosi, F., Verri, A., & Torre, V. (1988). A computational approach to motion perception. *Biological Cybernetics*, **60**, 79–87.

Wahl, D. D. & Simpson, J. J. (1990). Physical processes affecting the objective determination of near-surface velocity from satellite data. *Journal of Geophysical Research*, **95**, 13511–16528.

Weng, J. J. (1993). Image matching using windowed Fourier phase. *International Journal of Computer Vision*, **11**(3), 211–236.

Westergaard, C. H., Kock, C. W., Larsen, P. S., & Buchhave, P. (1993). Application of PIV to turbulence studies, - and exploratory study. Tech. rep. AFM–9307, Department of Fluid Mechanics, Technical University of Denmark, Lyngby, Denmark. 49 pp.

IMM Ph.D. Theses

1. **Rasmus Larsen.** (1994). *Estimation of visual motion in image sequences.* 143 pp.