Motion Analysis

PHILIPPE THÉVENAZ

Institute of Microtechnology
University of Neuchâtel
Switzerland

1 Introduction

The topic of this lecture is the automatic analysis of motion by machines. Our goal is to present some of the techniques used to solve the problem of motion detection and measurement.

1.1 Relevance of motion to image analysis

Consider the real world, and concentrate on what you can extract from it, using the visual sense only: your perception is not just like a still picture of your environment; it is rather the full time-evolution of the images you are receiving through your eyes. You would not be able to play a tennis game if you only had access to just a single still picture frame of the court, much like the court's floor plan is. On the other hand you need, in order to play, some informations on the opponent's displacement, on the ball's speed and direction, and on your own movements.

Note that you do not absolutely need an immediate perception of depth: you still are able to play even with one eye closed, without benefitting by any stereo effect. The reason is that you reconstruct the structure and proximity of your surroundings using clues found in the motion itself.

It follows that motion is a powerful characteristic to consider when analyzing visual input; its relevance to pattern recognition and image processing is of big importance.

1.2 Survey on motion application

The techniques developed for motion estimation have been applied to a wide range of applications; Nagel (in [Huan 81a], pp. 19–228) presents a survey spreading on more than 200 pages. The interested reader is encouraged to have a fruitful look at it; there, he will find full discussion and references pertaining to:

- coding. When transmitting data in a videophone application, it is interesting to predict the next frame. To this end, the knowledge of the actual frame's motion is a must;
- restoration. A blurred image can sometimes be deblurred if the causes are known. Motion of the camera or of the subject is often one of the reasons;
- interpolation. Suppose you want to get an interpolated image between two frames of a motion-picture. If there is any moving object in the image sequences, then averaging the two frames will not work, unless you distort them on the basis of motion information;
- target tracking. In this particular case, two motions have to be considered: not only the target's, but also the hunter's motion;
- traffic monitoring. The speed limitation regulation is one aspect; another one is the automatic monitoring of traffic jams;
- autonomous navigation. An autonomous mobile robot needs to select a path which avoids obstacles. Those may be moving; but even if they are still, the robot has to be fed with its own motion information;
- satellite sensors,
    1. wind/clouds. An earth orbiting satellite is so far the best tool to monitor an area as large as, say, an ocean. To detect and measure the wind velocity, simply look at the clouds it carries;
    2. snow coverage. Different time-scales may be useful: recording the melting rate of glaciers in spring may be interesting for hydrogeologic studies. Recording the growth or recession rate of polar ice-fields through the years may indicate trends in global climate;
    3. shoreline erosion. Is the erosion slow or fast? Is it wise to construct some planned building near that particular coast?
    4. forest growth. The monitoring of the deforestation yields some informations for reasonable exploitation;
- biomedical,
    1. behavior of microorganisms. The influence of some temperature or concentration gradient on the behavior of microorganisms can only be automatically measured if one has access to their motion, which can be done by the observation of Petri dishes;

2. transportation of chemicals within cells. It happens very often that the determination of the rate of transportation of chemicals within cells is crucial for the timing of some biological mechanisms. A microscope is a good tool for such studies;
3. spermatozoon activity. The spermatozoon's ability to show some characteristic movement pattern is one of the standardized quality test used in artificial insemination;

- human body medical image sequences,

1. heart beat. The physician would be very happy to give some quantitative bases to his qualitative statements;
2. blood circulation. A non-invasive method for measuring the blood velocity can be applied in conjunction with X-ray techniques, permitting to have a deep insight into the body while looking at some blood made opaque to X-rays by a marker.

1.3 Lecture outline

After a short introduction, we will define the concept of the optical flow field, and show some ways to compute it. First we will consider two domains of transformation (Fourier and Hadamard), and show how to pass from these domains to motion estimation. Then we will present a method based on intensity gradients, which allows the computation of a dense optical flow field. Unfortunately, this optical flow field is not fully specified and needs some assumptions for its full recovery; some possible assumptions are discussed. The next method is based on token correspondences between two images, and produces a sparse optical flow field. The determination of this flow field is complete, but the problems encountered in establishing the correspondences are similar to the problems found in stereo vision; some heuristics for their solution are discussed from the motion analysis point of view. A short analogy between computer motion estimation and human vision will conclude the lecture.

2 Optical flow field definition

The motion analysis proceeds usually in two steps. The first step may be considered as a generalized image processing: its purpose is to transform the raw input image sequences into a two-dimensional intermediate representation of motion, still strongly related to the retinal surface. The second step is usually considered as image interpretation, or motion understanding: its purpose is to credit each object of the acquired scene on a motion/depth information.

The purpose of this section is to define more precisely the intermediate representation to be extracted from the first step. This representation is called the optical flow field.

2.1 Definition

Consider a spot of light created by the rotating beacon of a lighthouse. Put a big screen very far from the lighthouse, at a distance so great that the spot projected onto the screen is advancing at a speed higher than the speed of light. For example, a flashing rate of one rotation pro second would result in a screen having to lie at more than fifty thousand kilometers.

Now, ask somebody to look on the screen: from the premise of bounded physical speed, he will be able to tell that what he sees is not a real object, but just some moving pattern of intensities. He will conclude that he is looking to an *optical flow field*, as opposed to a *motion flow field*.

Optical flow field: Instantaneous description of the magnitude and direction of the retinal velocities of intensity patterns.

Motion flow field: Instantaneous description of the magnitude and direction of the true projected velocities of real world objects.

These definitions are purposely left imprecise enough in order to allow them to support every kind of imaging system: flat retina vs. spherical retina, single camera vs. multi cameras, short and long image sequences, etc.

2.2 Discussion

It is important to outline that the difference between the optical and the motion flow field may not always be as obvious as in the rather artificial example from above. There are other cases where the acquired intensity patterns move in a different way than the physical surfaces supporting it; consider for example a driving-mirror, or the ground of a lake seen through clear water. Sometimes, the human brain is even disturbed by illusions, like the well-known barber's pole effect.

In each of the above cases, it would be vain to attempt the recovering of the true motion of objects on the basis of intensities alone. Some world model is required, in order e.g. to first *recognize* a magnifying glass, and then to decide *how* to treat the visual data which happens to have gone through it.

2.3 Optical and motion flow field equivalence

From now on, we will ignore the difference between motion and optical flow field. We will assume that they are identical in all aspects. For example, we will consider the optical flow field as adequate when we will attempt to reconstruct a three-dimensional scene based on motion alone, as like as the one-eyed tennis player does.

3 Optical flow field in transformation domains

The motion computation techniques we present in this section are interesting more from a theoretical point of view than from a practical point of view; they belong to a class of techniques rarely used in image processing. They first represent the image in a domain of transformation; the frequency domain, related to Fourier analysis, is one of those. Then, they extract the useful information directly from this transformation domain.

These techniques use restrictive assumptions, discussed later, and hence have low performances in a real environment. Nevertheless, their interest lies in the use of very classical signal processing tools, and they can be introduced without any special knowledge of image processing.

3.1 Image definition

Define, in the domain of real numbers, a dynamic image (time-varying intensity patterns) by

$$I_c = I_c(x, y, t) \tag{1}$$

where x and y stand for the coordinates in a two-dimensional image plane, and t denotes the time. For the case where the acquisition apparatus consists of several imaging devices, the c index is used to indicate from which camera the image is issued.

3.2 Hypothesis

We will give here some conditions required for computing an optical flow field in both the Fourier and the Hadamard domain. These conditions pertain as well to the manner in which the image sequences are acquired, as to the scene itself.

- We will use only two frames, shot at t_0 and t_1.
- We are using a single camera, so the subscript c will be subsequently dropped and replaced by the time subscripts; the corresponding frames become I_0 and I_1 respectively.
- The camera does not translate, rotate or zoom. Its focus and its diaphragm are both fixed.
- There is a single object moving before a stationary background. This object stays at any time within the image frame and is never subject to occlusions.
- The object's movement is translational only: no rotation, no change of size, no change of form is allowed.

Further, we assume that the background is black everywhere (that is, put to zero-level). This can be done by illuminating the moving object only, or by using a technique of segmentation in order to define the two areas corresponding to the background on the one hand, and to the object on the other hand. Then, setting the background to zero-level becomes easy.

These restrictions impose that

$$I_1 = I(x, y, t_1) = I(x - \Delta x, y - \Delta y, t_0) \tag{2}$$

where Δx and Δy stand for the displacement of the object along the x-axis and the y-axis respectively, achieved in the time interval $\Delta t = t_1 - t_0$. Therefore, the computed velocity corresponds to

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \Delta x / \Delta t \\ \Delta y / \Delta t \end{pmatrix} \tag{3}$$

3.3 Fourier domain

The basic idea [Huan 81a] is to compute a two-dimensional Fourier transformation for each of the I_0 and I_1 images, and then to compare them on the basis of their phase plane. As the second image is simply the shifted version of the first one, the phase difference should be linear with respect to the spatial frequency.

Define the two-dimensional Fourier transformation J of an Image I by

$$J(\nu, \omega, t) = |J| \, \exp(i \, \angle(J)) = \boldsymbol{\mathcal{F}}_{x,y} \, \{I(x, y, t)\} \tag{4}$$

where ν and ω are spatial pulsations on the x- and y-axis respectively, and where $|J|$ and $\angle(J)$ denote the magnitude and phase of the Fourier transformation respectively, with $i = \sqrt{-1}$. By the shift theorem of the Fourier analysis, we can write

$$J(\nu, \omega, t_1) = J(\nu, \omega, t_0) \, \exp(-2\pi i \, (\nu \Delta x + \omega \Delta y)) \tag{5}$$

hence,

$$\Delta\angle(J) = \angle(J(\nu, \omega, t_1)) - \angle(J(\nu, \omega, t_0)) = -2\pi \, (\nu \Delta x + \omega \Delta y) \tag{6}$$

It is obvious that the algorithm for computing the displacement of our single moving object over a black background is first to get I_0 and I_1, then J_0 and J_1. The computation of $\Delta\angle(J)$ for two arbitrarily chosen sets of pulsations ν and ω yields two linear equations in the two unknown variables Δx and Δy, which can easily be solved.

Of course, it is also possible to select every conceivable pairing of ν and ω in order to overdetermine the unknown shifts. This procedure allows the use of e.g. a least squares method for the solution of the system of equation. The additional computational cost is rewarded by increased robustness.

3.4 Separability of Fourier analysis

In the section 3.3, we have seen that we could determine the motion parameters from a pair of Fourier transformations; but the algorithmic complexity was heavy, due to the fact that they had to be computed in a two-dimensional plane. In the present section, we propose a scheme using instead a unidimensional Fourier transform. The idea is to project and accumulate the intensity pattern along each axis. Using the same hypothesis as in section 3.2, the projected image should obey the same rules concerning its displacements as the original image.

Consider the projection $I_{\Sigma y}$ along the y-axis of the image I onto the x-axis

$$I_{\Sigma y}(x,t) = \int_{y_{min}}^{y_{max}} I(x,y,t)\,dy \tag{7}$$

Define its one-dimensional Fourier transformation $J_{\Sigma y}$ by

$$J_{\Sigma y}(\nu,t) = \boldsymbol{\mathcal{F}}_x \left\{ \int_{y_{min}}^{y_{max}} I(x,y,t)\,dy \right\} \tag{8}$$

The relation 2 translates into

$$\begin{aligned} J_{\Sigma y}(\nu,t_1) &= \boldsymbol{\mathcal{F}}_x \left\{ \int_{y_{min}}^{y_{max}} I(x-\Delta x, y-\Delta y, t_0)\,dy \right\} \\ &= \int_{y_{min}-\Delta y}^{y_{max}-\Delta y} \boldsymbol{\mathcal{F}}_x \left\{ I(x-\Delta x, y, t_0) \right\}\,dy \\ &= \int_{y_{min}-\Delta y}^{y_{max}-\Delta y} \exp(-2\pi i\,\nu\,\Delta x)\,\boldsymbol{\mathcal{F}}_x \left\{ I(x,y,t_0) \right\}\,dy \\ &= \exp(-2\pi i\,\nu\,\Delta x)\,J_{\Sigma y}(\nu,t_0) \end{aligned} \tag{9}$$

where the fact that the moving object stays within the image frame has been used to replace $y_{max} - \Delta y$ and $y_{min} - \Delta y$ by respectively y_{max} and y_{min} in the integral's bounds. It follows from equation 9 that

$$\Delta\angle(J_{\Sigma y}) = \angle(J_{\Sigma y}(\nu,t_1)) - \angle(J_{\Sigma y}(\nu,t_0)) = -2\pi\,\nu\,\Delta x \tag{10}$$

For the second dimension, we get similarly

$$\Delta\angle(J_{\Sigma x}) = \angle(J_{\Sigma x}(\omega,t_1)) - \angle(J_{\Sigma x}(\omega,t_0)) = -2\pi\,\omega\,\Delta y \tag{11}$$

The shifts Δx, respectively Δy, can now be computed using a single pulsation ν, respectively ω. If one may want to achieve a higher robustness by including more pulsations in the computation, then the least squares method of section 3.3 can be substituted by a simpler averaging technique. Moreover, the burden of a two-dimensional Fourier transformation computation has been replaced by its one-dimensional counterpart.

3.5 Conclusion on Fourier analysis

We have defined an analytical, continuous method for extracting the motion of an object which had been subject to the restrictions of section 3.2. Unfortunately, this method is computationaly expensive (Fourier transform), and complicated by the fact that conventional techniques for computing a (discrete) Fourier transformation yield phase components values wrapped in the main domain only $[0, 2\pi[$. The solution is either to unwrap the phase component, or to impose a new condition limiting the displacement magnitude to less than half an image frame.

The assumptions needed ar thus very restrictive; on the other hand, the method is quite robust to noise, as each part of the image participates to this motion extraction algorithm. Further, after discretization (if there is any), the displacement quantization is better than the image resolution, as it is not a result of geometric computations with localized pixels, but a more global approach.

3.6 Hadamard domain

The idea [Lai 88] is that by using modern, digital computers, one is forced to forget about a continuous world and rather concentrate on its discrete version. Now, it appears that some transformation domains are more suited to analog computations, and some others to quantized processing. The Hadamard transform is one of the latter; indeed, it is defined in an world where numbers are coded using binary representation of integers only. Its peculiarity is that its associated transformation may be computed using merely additions and subtractions.

We will first give a definition of the Hadamard transform, and then see how it can be used to get motion from a set of two frames obeying to the same restrictions as in section 3.2.

3.7 Hadamard transformation

Let $I(i,j,k) \in [I_{min}, I_{max}]$ be an image subject to intensity quantization, spatial quantization $i \in [i_{min}, i_{max}]$, $j \in [j_{min}, j_{max}]$, and temporal quantization as well $k \in [k_{min}, k_{max}]$. Letting $k_0 = 0$ and $k_1 = 1$, the shift equation 2 reads now

$$I_1 = I(i,j,1) = I(i - \Delta x, j - \Delta y, 0) \tag{12}$$

The three-dimensional Hadamard transformation of this image is defined by

$$H(\xi, \zeta, \tau) = \boldsymbol{\mathcal{H}}_{i,j,k}\left\{I(i,j,k)\right\} = \sum_{i=i_{min}}^{i_{max}} \sum_{j=j_{min}}^{j_{max}} \sum_{k=k_{min}}^{k_{max}} I(i,j,k)\,(-1)^{p(i,j,k,\xi,\zeta,\tau)} \tag{13}$$

where the parity function p is defined by

$$p(i,j,k,\xi,\zeta,\tau) = \sum_{c=0}^{\log_2(i_{max}-i_{min})} \xi_c i_c + \sum_{l=0}^{\log_2(j_{max}-j_{min})} \zeta_l j_l + \sum_{t=0}^{\log_2(k_{max}-k_{min})} \tau_t i_t \tag{14}$$

In equation 14, $\log_2(*)$ denotes the base-2 logarithmic function, the selected upper-bound of summation being equal to the next or equal integer number. The notation

$a_i \in \{0,1\}$ stands here for the i-th bit of a, the least and most significant bit being respectively $a_0 = \text{LSB}$ and $a_{N-1} = \text{MSB}$ in natural binary coding.

3.8 Hadamard transformation applied to a single moving pixel

For demonstration purposes, we add now an assumption still more restrictive than the previous ones: the imaged object is reduced to a *single* non-zero moving pixel. Suppose that its grey-value is g, and that it shows up at location (i', j') in the first frame, and (i'', j'') in the second frame. The corresponding time-varying image can be described by

$$I(i,j,k) = \begin{cases} g & (i,j,k) = (i',j',0) \;\vee\; (i,j,k) = (i'',j'',1) \\ 0 & (i,j,k) \neq (i',j',0) \;\wedge\; (i,j,k) \neq (i'',j'',1) \end{cases} \tag{15}$$

In binary notation, the displacement along the x-axis is

$$\Delta i = i'' - i' = \sum_{c=0}^{N} 2^c (i''_c - i'_c) \tag{16}$$

where N is a bound big enough to hold every conceivable case

$$N = \log_2(i_{max} - i_{min}) \tag{17}$$

One gets from equations 13 and 15 two particular cases, with the definition of $p_\xi(i,\xi)$ being evident from equation 14

$$\begin{aligned} H(0,0,0) &= I(i',j',0)\,(-1)^{p(i',j',0,0,0,0)} + I(i'',j'',1)\,(-1)^{p(i'',j'',1,0,0,0)} \\ &= 2g \end{aligned} \tag{18}$$

$$\begin{aligned} H(\xi,0,1) &= I(i',j',0)\,(-1)^{p(i',j',0,\xi,0,1)} + I(i'',j'',1)\,(-1)^{p(i'',j'',1,\xi,0,1)} \\ &= g(-1)^{p_\xi(i',\xi)} + g(-1)^{p_\xi(i'',\xi)+1} \end{aligned} \tag{19}$$

Constraining ξ to the powers of two leaves only one non-zero term in equation 14, yielding

$$H(2^n,0,1) = g\left((-1)^{i'_n} - (-1)^{i''_n}\right) = 2g\,(i''_n - i'_n) \tag{20}$$

where the last equation is obtained by looking at the exhaustive list of values for i''_n and i'_n. We derive from equation 18 and equation 20 the relation linking a given bit Δi_n of the displacement Δi to the Hadamard transformations. Combining these results over the whole binary representation of the shift along the i-axis, we get

$$i''_n - i'_n = \frac{H(2^n,0,1)}{H(0,0,0)} \tag{21}$$

$$\Delta i = \sum_{n=0}^{N} 2^n \frac{H(2^n,0,1)}{H(0,0,0)} \tag{22}$$

Of course, a similar equation can be established for the j-axis. Note that $N+2$ Hadamard transformations must be computed for both axis, each requiring only additions and subtractions. That is a nice result, but it would be silly to use this sophisticated method to get the displacement of just one single point...

3.9 Hadamard transformation applied to the whole moving pattern

Suppose now that we want to apply the method developed in section 3.8 to a pattern whose size is bigger than just one single pixel. Each pixel of this moving pattern may be indexed by p, and there are P of them, ranging from 0 inclusive to P exclusive. If we term H_p the Hadamard transformation applied to the single pixel of index p, then both the property of linearity of the transformation and the conjoint use of equation 22 imply

$$\begin{aligned}
\sum_{n=0}^{N} 2^n \frac{H(2^n,0,1)}{H(0,0,0)} &= \sum_{n=0}^{N} 2^n \frac{\sum_{p=0}^{P-1} H_p(2^n,0,1)}{\sum_{p=0}^{P-1} H_p(0,0,0)} \\
&= \frac{\sum_{n=0}^{N} \sum_{p=0}^{P-1} 2^n H_p(2^n,0,1)}{\sum_{p=0}^{P-1} H_p(0,0,0)} \\
&= \frac{\sum_{p=0}^{P-1} \Delta i \, H_p(0,0,0)}{\sum_{p=0}^{P-1} H_p(0,0,0)} \\
&= \Delta i \qquad (23) \\
\sum_{n=0}^{N} 2^n \frac{H(0,2^n,1)}{H(0,0,0)} &= \Delta j \qquad (24)
\end{aligned}$$

3.10 Conclusion on Hadamard analysis

We have defined a method for extracting the motion of an object subject to the restrictions of section 3.2. This method is applicable to discrete data only, and is computationaly less expensive than the Fourier transformation, since it needs only additions and subtractions. Further, it is well adapted to the digital binary world which is actually predominant in computer architectures.

The components of motion can be retrieved separately, using a small number of values (of order $\mathcal{O}(\log n)$ in image dimension) from the transformed domain, but the assumptions needed ar still very restrictive. On the other hand, the method is quite robust to noise, as each part of the image participates to this motion extraction algorithm. Note that the resolution in displacement is better than the image resolution, as it is not a result of geometric computations, but a more global approach.

3.11 Conclusion on motion computations in transformation domains

The most striking point is the restrictiveness of the hypothesis made in the section 3.2. Remember that we are looking at the translation, *and translation only*, of a *single* moving object. Still more important is the condition of a background which is set to zero-level. In fact, as soon as this condition is fulfilled, one may apply the much simpler technique of gravity-center approach, presented below.

Consider the intensity of any image I as being a weight, and compute the gravity-center of the first and the second image

$$\mathbf{g}(t_0) = \frac{\sum_{\mathbf{x}\in I} \mathbf{x}\, I(\mathbf{x}, t_0)}{\sum_{\mathbf{x}\in I} I(\mathbf{x}, t_0)} \tag{25}$$

$$\mathbf{g}(t_1) = \frac{\sum_{\mathbf{x}\in I} \mathbf{x}\, I(\mathbf{x}, t_1)}{\sum_{\mathbf{x}\in I} I(\mathbf{x}, t_1)} \tag{26}$$

The linear displacement of the (single) pattern is then simply equal to the displacement of the center of gravity

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \mathbf{g}(t_1) - \mathbf{g}(t_0) \tag{27}$$

The velocity "field" obtained by all the above methods is very crude, as it consists of two values only: by hypothesis, the background is motionless, while a unique translational displacement value is assigned to the whole moving pattern; but in some way, this velocity field may be seen as dense, for it is defined everywhere. The content in information of this velocity field has to be considered as a whole, *including* the figure/ground segmentation. From such a point of view, much of the work is carried out by the segmentation process, the easier part being left to the motion analysis. This is one extreme of task division. Another extreme will be presented in the next section.

4 Optical flow field in gradient schemes

Let us do some introspective observations while investigating our own motion detection and measurement capabilities: it shows up very quickly that we are indeed able to detect motion in a rather unawareness manner. In particular, we do not necessarily have to *identify* any object before being able to tell that "something" is moving. I hope that still everybody has had the opportunity to experience the sudden discovery of some animal (bird? squirrel? rabbit? leaf?) while wandering in a forest, noticed by its mere movement and nothing else. Alternatively, consider the task of finding a fly on a randomly dotted wallpaper; an immobile fly or a moving fly makes a great difference. These experiences draw the attention to the fact that one is able to detect motion at a level of perception which seems not to require much information about the world's structure or meaning. A model of *local* computation of motion would perfectly cope with this.

While reproducing this behavior by machines, we are faced to a problem which is totally different from that of section 3, owing to the fact that we have no a priori knowledge of what is moving. To begin with, no figure/ground segmentation may be applied; on the contrary, we have to rely on the image intensities only in order to discover the motion parameters of maybe several patterns at the same time. It is only *after* the completion of motion analysis that we may be able to segment the scene into moving and non-moving parts.

Indeed, a scheme for the local measurement of motion using intensities does exist [Limb 75]. It is generally termed a *gradient scheme*, and is apt to yield a dense flow field. However, the full determination of motion, using this scheme, is reached only if some a priori assumptions are made. Further, some parts of the flow field may be left undefined. The scope of the present section is to show with some details the use of a gradient scheme in determining of the optical flow field; the assumptions needed for its full recovery will be discussed, too.

4.1 Aperture problem

Consider the task of determining motion on the basis of intensity patterns only: it may sometimes happen that this task does not make any sense at all. For example, the detection of a sheet of white paper sliding on top of another sheet of white paper can be considered as an intractable case. What one needs indeed, to perform the task, is the very existence of a *pattern*, that is, a non-uniform image. When fulfilled, this requirement gives automatically rise to some intensity gradient.

Consider now the task of the local determination of motion within some small window in the image; for simplicity purposes, assume that the existing gradient is replaced by a step function, oriented along e.g. a vertical direction. Then, it is clear that if this edge also slides along the vertical direction, and if one restricts the information to what is happening under the window only, then there will be no clues available to indicate any movement. The only movement which can be recovered through this small aperture is a movement which is perpendicular to the edge, or, equivalently, parallel to the gradient.

In the case just described, an infinity of different optical flow fields could fit the data obtained through our finite aperture window, for the motion in a direction parallel to the edge is lost and may take any value.

This problem is inherent to any local scheme for computing motion. It is termed the *aperture problem*, and manifests itself in a number of other cases. Basically, it states that it is impossible to fully recover motion using local informations only. This implies that determining the optical flow field on the basis of an intensity gradient scheme is formally an *ill-posed* problem. It means that this problem either

- has no solution at all,
- has no unique solution, or
- does not depend continuously on the initial data.

In the next sections, we will see first how to get the recoverable part of motion, and then how to regularize this ill-posed problem.

4.2 Optical flow derivation

Take the generalized image of equation 1, and forget about the c index, as we are making use of a single camera only. If we accept this image as being analytical, we then may expand it into a Taylor series and get

$$I(x+dx, y+dy, t+dt) = I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \epsilon(x,y,t) \tag{28}$$

where $\epsilon(x,y,t)$ denotes all the terms of order higher than 1, which will be ignored from now on.

If the image pattern at the coordinates (x,y,t) is undergoing an instantaneous translation (dx, dy) in an infinitesimal time (dt), and if we suppose that the pattern is time-invariant, then we are able to track $I(x,y,t)$ through time and space (neglecting at this time-scale any global variation of illumination is equivalent to considering motion only as responsible for the term $\partial I/\partial t$). The tracking is correct if

$$I(x+dx, y+dy, t+dt) = I(x,y,t) \tag{29}$$

which implies, by equation 28,

$$0 = \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \quad \Leftrightarrow \quad -\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} \tag{30}$$

Writing the velocities $v_x = dx/dt$ and $v_y = dy/dt$ yields

$$-\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y \tag{31}$$

or, in an equivalent vector notation

$$-\frac{\partial I}{\partial t} = (\boldsymbol{\nabla} I)^T \cdot \mathbf{v} \tag{32}$$

where $\boldsymbol{\nabla} I$ stands for the spatial gradient of the image, ^{T} is the symbol used for transposition, $\cdot$ denotes the vector-product and $\mathbf{v}$ is the vector-valued velocity.

We have seen that only the component of velocity which lies perpendicular to the edge (i.e. parallel to the gradient) can be measured. Its magnitude is

$$v^{\perp}(x,y,t) = \frac{|\partial I/\partial t|}{\boldsymbol{\nabla} I} = \frac{|\partial I/\partial t|}{\sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}} \tag{33}$$

The magnitude $v^{\top}$ of the velocity component lying tangentially to the edge (i.e. perpendicular to the gradient) cannot be recovered directly from the image sequences. Some more constraints are required.

4.3 Robustness with respect to intensity distortion

The cameras used to record the image sequences are usually subject to transfer characteristics which are not linear between the illumination and the camera's signal output

[Batc 85]. In order to do computations using correct intensity values, it is necessary either to find a way to eliminate the non-linearities or to cope with them in such a manner that they cancel one another.

For our case, consider the function $f(I)$ transforming the distorted intensity I, coming from the camera, into the correct, true illumination value. If we rewrite equation 33 using this transformation, we get

$$\begin{aligned} v^{\perp}(x,y,t) &= \frac{|\partial f(I)/\partial t|}{\sqrt{\left(\frac{\partial f(I)}{\partial x}\right)^2 + \left(\frac{\partial f(I)}{\partial y}\right)^2}} = \frac{\left|\frac{\partial f}{\partial I}\frac{\partial I}{\partial t}\right|}{\sqrt{\left(\frac{\partial f}{\partial I}\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial I}\frac{\partial I}{\partial y}\right)^2}} \\ &= \frac{|\partial I/\partial t|}{\sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}} \end{aligned} \tag{34}$$

We just have shown that the burden of calibration needed to estimate $f(I)$ is useless. The raw incoming intensity I is sufficient to meet our computational requirements, because any continue and monotonous transformation f yields the same result, provided that $\partial f/\partial I$ exists at the same time that it is non-zero.

4.4 Conclusion on flow field computation using gradient

We have introduced a technique for the local computation of an optical flow field. This technique makes use of the local temporal and spatial gradient of an image; there is no need for any knowledge about the world. The only assumption about the image sequences is that they have to be analytical. The velocity obtained is independent from intensity transformations constant in time; one may then choose any hardware camera without taking care about intensity calibration. These were the good news.

The bad news are that the computations are done using derivatives, which are notorious for their numerical unreliability; this may induce some noise problems. More important, only half of the flow field has been retrieved, because the aperture problem limits the recovery to the perpendicular component of motion. The $v^{\top}$ component is still missing.

5 Motion constraint

Equation 32, derived in section 4.2, is not sufficient to specify entirely the content of the optical flow, if one considers only the data; but still it can be used in certain restrictive cases. First, we will discuss some useful representations of the constraints generated by this equation. Then, we will make some assumptions assisting in the full recovery of optical flow for some special cases.

5.1 Point locus in the velocity space

Consider rewriting equation 32 in a form making use of unit vectors in the direction perpendicular and parallel respectively to the gradient

$$\mathbf{v} = v^{\top}\mathbf{u}^{\top} + v^{\perp}\mathbf{u}^{\perp} \tag{35}$$

where $\mathbf{u}^{\top}$ and $\mathbf{u}^{\perp}$ are defined as

$$\mathbf{u}^{\perp} = \frac{\boldsymbol{\nabla} I}{|\boldsymbol{\nabla} I|} \tag{36}$$

$$|\mathbf{u}^{\top}| = 1 \qquad \wedge \qquad (\mathbf{u}^{\top})^{T} \cdot \mathbf{u}^{\perp} = 0 \tag{37}$$

We can get $\mathbf{u}^{\top}$ and $\mathbf{u}^{\perp}$ from the image sequences, and compute $v^{\perp}$ as indicated in equation 33. On the other hand, $v^{\top}$ is left unspecified; but one can clearly see, from the above equations, that if we build a velocity space where v_x and v_y are drawn, then equation 35 can be plotted as a line. Actually, this equation states that if $v^{\top} = 0$, then the velocity lies at the tip of the $v^{\perp} \cdot \mathbf{u}^{\perp}$ vector; if $v^{\top} \neq 0$, then the velocity vector may be anywhere away from this tip, as long as it lies on a line whose direction is given by $\mathbf{u}^{\perp}$ and which passes through the tip.

5.2 Translating polygon

Remember the first part of the discussion in section 4.1, and also equation 33: the spatial gradient *has* to be non-zero. Suppose now that this condition is fulfilled only at the borders of a polygon built with straight line segments. Along any given segment, the constraint lines in the velocity space will all be parallel, as the aperture problem forces the locally detected motion to be perpendicular to the segment. Moreover, under the hypothesis of a translational motion only, all the constraint lines for a given segment are merged.

If one considers now a second segment of the polygon, chosen to be non-parallel to the first one, and subject to the same translation, then its corresponding constraint line in the velocity space will intersect with the constraint line originating from the first segment, and define a single possible motion for the whole polygon. Of course, this technique can be made more robust by considering all segment pairs, and by using clustering techniques for the intersections in the velocity space in order to decide both how many polygons are moving in the image, and what is their motion.

In conclusion, we pretend that the assumption of *translational patterns only* is sufficient to fully determine the optical flow, as computed in a gradient scheme using equation 32. However, it is not the only possible way to regularize the problem; the next sections will demonstrate some other techniques allowing to achieve the same goal.

5.3 Infinite inertia assumption

The idea [Wu 88] developed in this section stems from the observation that the objects of the real world all have a finite, non-zero mass. The fact that no infinite acceleration is allowed in the classical physical world implies that, for any object, the difference in velocity between time t_0 and time t_1 is small under the condition that the time interval $\Delta t = t_1 - t_0$ is kept small enough. Hence, we will accept as a supplementary constraint that the time-derivative of an optical flow is zero everywhere. This implies, using equation 31

$$\left\{ \begin{array}{lllll} 0 & = & \frac{dv_x}{dt} & = & \frac{\partial v_x}{\partial t} + \frac{\partial v_x}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial v_x}{\partial y}\frac{\partial y}{\partial t} \\ 0 & = & \frac{dv_y}{dt} & = & \frac{\partial v_y}{\partial t} + \frac{\partial v_y}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial v_y}{\partial y}\frac{\partial y}{\partial t} \end{array} \right. \tag{38}$$

Recognizing that $v_x = \partial x/\partial t$ and $v_y = \partial y/\partial t$, one may transform the system of equations 38 into

$$\left\{ \begin{array}{lll} \frac{\partial v_x}{\partial t} & = & -\frac{\partial v_x}{\partial x} v_x - \frac{\partial v_x}{\partial y} v_y \\ \frac{\partial v_y}{\partial t} & = & -\frac{\partial v_y}{\partial x} v_x - \frac{\partial v_y}{\partial y} v_y \end{array} \right. \tag{39}$$

Differentiating equation 31 with respect to x, y and t, and assuming that the image I is not only twice differentiable, but also that the order of differentiation is not relevant, one gets

$$\left\{ \begin{array}{lll} \frac{-\partial^2 I}{\partial x\,\partial t} & = & \frac{\partial^2 I}{\partial x^2} v_x + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial x} + \frac{\partial^2 I}{\partial x\,\partial y} v_y + \frac{\partial I}{\partial y}\frac{\partial v_y}{\partial x} \\ \frac{-\partial^2 I}{\partial y\,\partial t} & = & \frac{\partial^2 I}{\partial x\,\partial y} v_x + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial y} + \frac{\partial^2 I}{\partial y^2} v_y + \frac{\partial I}{\partial y}\frac{\partial v_y}{\partial y} \\ \frac{-\partial^2 I}{\partial t^2} & = & \frac{\partial^2 I}{\partial x\,\partial t} v_x + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial t} + \frac{\partial^2 I}{\partial y\,\partial t} v_y + \frac{\partial I}{\partial y}\frac{\partial v_y}{\partial t} \end{array} \right. \tag{40}$$

If the relation $\partial I/\partial y = 0$ holds true, then the next equations have to be modified in a way which is rather obvious. Suppose from now on that the intensity gradient along the y-axis is non-zero. One may then replace some terms of the system 40 by those of system 39, getting

$$\left\{ \begin{array}{lll} \frac{\partial v_y}{\partial x} & = & \frac{-1}{\partial I/\partial y}\left(\frac{\partial^2 I}{\partial x\,\partial t} + \frac{\partial^2 I}{\partial x^2} v_x + \frac{\partial^2 I}{\partial x\,\partial y} v_y + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial x}\right) \\ \frac{\partial v_y}{\partial y} & = & \frac{-1}{\partial I/\partial y}\left(\frac{\partial^2 I}{\partial y\,\partial t} + \frac{\partial^2 I}{\partial x\,\partial y} v_x + \frac{\partial^2 I}{\partial y^2} v_y + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial y}\right) \\ \frac{-\partial^2 I}{\partial t^2} & = & \frac{\partial^2 I}{\partial x\,\partial t} v_x + \frac{\partial I}{\partial x}\left(-\frac{\partial v_x}{\partial x} v_x - \frac{\partial v_x}{\partial y} v_y\right) + \frac{\partial^2 I}{\partial y\,\partial t} v_y + \frac{\partial I}{\partial y}\left(-\frac{\partial v_y}{\partial x} v_x - \frac{\partial v_y}{\partial y} v_y\right) \end{array} \right. \tag{41}$$

Merging the system 41 into one single equation, one gets

$$\begin{array}{lll} \frac{-\partial^2 I}{\partial t^2} & = & v_x \left(\frac{\partial^2 I}{\partial x\,\partial t} - \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial x} - \frac{\partial I}{\partial y}\frac{-1}{\partial I/\partial y}\left(\frac{\partial^2 I}{\partial x\,\partial t} + \frac{\partial^2 I}{\partial x^2} v_x + \frac{\partial^2 I}{\partial x\,\partial y} v_y + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial x}\right)\right) + \\ & & v_y \left(\frac{\partial^2 I}{\partial y\,\partial t} - \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial y} - \frac{\partial I}{\partial y}\frac{-1}{\partial I/\partial y}\left(\frac{\partial^2 I}{\partial y\,\partial t} + \frac{\partial^2 I}{\partial x\,\partial y} v_x + \frac{\partial^2 I}{\partial y^2} v_y + \frac{\partial I}{\partial x}\frac{\partial v_x}{\partial y}\right)\right) \\ & = & v_x \left(2\frac{\partial^2 I}{\partial x\,\partial t} + \frac{\partial^2 I}{\partial x^2} v_x + \frac{\partial^2 I}{\partial x\,\partial y} v_y\right) + v_y \left(2\frac{\partial^2 I}{\partial y\,\partial t} + \frac{\partial^2 I}{\partial x\,\partial y} v_x + \frac{\partial^2 I}{\partial y^2} v_y\right) \\ & = & v_x^2 \frac{\partial^2 I}{\partial x^2} + 2 v_x v_y \frac{\partial^2 I}{\partial x\,\partial y} + v_y^2 \frac{\partial^2 I}{\partial y^2} + 2 v_x \frac{\partial^2 I}{\partial x\,\partial t} + 2 v_y \frac{\partial^2 I}{\partial y\,\partial t} \end{array} \tag{42}$$

Rewriting v_y from equation 31 yields

$$v_y = -\frac{\frac{\partial I}{\partial t} + \frac{\partial I}{\partial x} v_x}{\partial I/\partial y} \tag{43}$$

Substituting equation 43 into equation 42 gives rise to a second-order equation in v_x, where all the terms can be found in the image. Explicitly, we write

$$
\begin{aligned}
0 \quad = \quad & v_x^2 \left(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \left(\frac{\partial I/\partial x}{\partial I/\partial y} \right)^2 - 2 \frac{\partial^2 I}{\partial x \, \partial y} \left(\frac{\partial I/\partial x}{\partial I/\partial y} \right) \right) + \\
& 2 v_x \left(\frac{\partial^2 I}{\partial x \, \partial t} - \frac{\partial^2 I}{\partial y \, \partial t} \left(\frac{\partial I/\partial x}{\partial I/\partial y} \right) - \frac{\partial^2 I}{\partial y^2} \left(\frac{\partial I/\partial x}{\partial I/\partial y} \right) \frac{\partial I}{\partial t} - \frac{\partial^2 I}{\partial x \, \partial y} \left(\frac{\partial I/\partial t}{\partial I/\partial y} \right) \right) + \\
& \left(\frac{\partial^2 I}{\partial t^2} + \frac{\partial^2 I}{\partial y^2} \left(\frac{\partial I/\partial t}{\partial I/\partial y} \right)^2 - 2 \frac{\partial^2 I}{\partial y \, \partial t} \left(\frac{\partial I/\partial t}{\partial I/\partial y} \right) \right)
\end{aligned} \tag{44}
$$

Solving this equation and making use of equation 43 yields a pair of possible answers for v_x and v_y. In fact, it is easy to select the right choice while considering the minimization of the following discrimination function s

$$
\begin{aligned}
s(x, y, t, v_x, v_y) \quad = \quad & |I(x - v_x \, \Delta t, y - v_y \, \Delta t, t - \Delta t) - I(x, y, t)| + \\
& |I(x, y, t) - I(x + v_x \, \Delta t, y + v_y \, \Delta t, t + \Delta t)|
\end{aligned} \tag{45}
$$

where it has been implicitly assumed that the three images I_{-1}, I_0 and I_1 have been taken at a regular time interval Δt in order to estimate the second order derivatives by numerical approximation.

In conclusion, we pretend that the assumption of *massive objects* is also sufficient to fully determine the optical flow, as computed in a gradient scheme using equation 32; however, it is not the only possible way to regularize the problem. We will demonstrate in the next section another technique which allows to achieve the same result.

5.4 Smoothness assumption

This section will describe the most often used assumption for regularizing the computation of optical flow in a gradient scheme. It is based [Horn 81] on a qualitative property of imaging geometry: if you project (an orthographic projection is as good as a perspective projection) a smooth, three-dimensional motion field onto a two-dimensional retinal image (planar or spherical retinal surface), then the projected motion field tends also to be smooth if one ignores any occlusion which may appear. Now, apart from object boundaries, the true velocity field of a real scene does generally fit with a smoothness hypothesis. This justifies the regularization of our problem by imposing a smoothness constraint. The constraint should be such that

- the solution obtained is uniquely defined (if one applies this constraint, a single flow field should emerge from the computations),
- the solution obtained is physically plausible (no infinite or indefinite results should occur; moreover, correct results are needed for at least some well chosen simple cases), and
- the solution obtained is consistent with human perception (if it is true, then it is a clue that the smoothness assumption is not a mere computational artifact).

There are many possible smoothness constraints; for example, consider minimizing the integral of the first order variations $\int |\boldsymbol{\nabla}\mathbf{v}|\, d\mathbf{s}$, or minimizing the integral of the second order variations $\int \nabla^2 \mathbf{v}\, d\mathbf{s}$, or minimizing only the change of motion direction $\int |\boldsymbol{\nabla}(\angle \mathbf{v})|\, d\mathbf{s}$, or only the change of motion magnitude $\int |\, \boldsymbol{\nabla}|\mathbf{v}|\,|\, d\mathbf{s}$. Consider also the joint use of several of these constraints at the same time; you will end up with a lot (an infinity, indeed) of possible ways to look for. Here, we will restrict our choice to the minimization of the integral of the squared first order variations

$$\Theta = \int_{\mathbf{s}\in I} |\boldsymbol{\nabla}\mathbf{v}|^2\, d\mathbf{s} = \int_{\mathbf{s}\in I} \left(\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 \right) d\mathbf{s} \tag{46}$$

Of course, the quantity Θ has to be minimized. It can be shown [Rudi 73] that the requirement of solution unicity is satisfied, but the proof for this statement will not be further investigated. Here, we will contend ourselves to give two examples, demonstrating the physical plausibility of the chosen smoothness constraint applied to the regularization of the optical flow field computation.

5.5 Smoothness assumption applied to a translating surface

The hypothesis, valid for this section only, states that the imaging system consists of a unique pin-hole camera looking at an object whose motion is restricted to a translation in a plane parallel to the camera's projection plane. The object is allowed neither to rotate nor to deform. We prove here that, under these hypothesis, the motion computed using the functional Θ of equation 46 is the correct one.

First, we will give some properties of the geometry involved; then we present the proof in two steps. The first step states that *every* translating object produces the minimal value $\Theta = 0$. The second step states that if one has to build a flow field under the constraint $\Theta = 0$, then one will *always* end up with a translation. The conclusion is that minimizing Θ is the bets thing to do when confronted to a mere translation, because finding the absolute minimum is *equivalent* to finding the translation.

Geometrical considerations: with no loss of generality, describe any movement in space by the combination of an axis of rotation going through the origin, and a translation. The rotation $\boldsymbol{\Omega}$ as well as the translation $\mathbf{T}$ may be represented each by a vector

$$\boldsymbol{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T \tag{47}$$

$$\mathbf{T} = (T_x, T_y, T_z)^T \tag{48}$$

The velocity of a point in space may then be described by

$$\mathbf{V} = \boldsymbol{\Omega} \times \mathbf{X} + \mathbf{T} = \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} = \begin{pmatrix} Z\Omega_y - Y\Omega z + T_x \\ X\Omega_z - Z\Omega x + T_y \\ Y\Omega_x - X\Omega y + T_z \end{pmatrix} \tag{49}$$

where $\mathbf{X} = (X, Y, Z)^T$ is the coordinate of the considered point, and $\mathbf{V} = (V_x, V_y, V_z)^T$ is its instantaneous velocity. Using a pin-hole camera whose eye is placed at the origin, and whose projection plane is set at the focal length $f = 1$, one may then determine the true motion flow field $\mathbf{w}$ (see section 2.1). Writing the projection coordinates in lower-case and the world coordinates in upper-case, one gets

$$x(t) = \frac{X + tV_x}{Z + tV_z} \tag{50}$$

$$\dot{x}(t) = \frac{V_x(Z + tV_z) - V_z(X + tV_x)}{(Z + tV_z)^2} = \frac{ZV_x - XV_z}{(Z + tV_z)^2} \tag{51}$$

$$\begin{aligned} w_x = \dot{x}(0) &= \frac{Z^2\Omega_y - YZ\Omega_z + ZT_x - XY\Omega_x + X^2\Omega_y - XT_z}{Z^2} \\ &= -\Omega_x xy + \Omega_y(1 + x^2) - \Omega_z y + \frac{T_x - T_z x}{Z} \qquad (52) \\ w_y = \dot{y}(0) &= \Omega_y xy - \Omega_x(1 + y^2) + \Omega_z x + \frac{T_y - T_z y}{Z} \qquad (53) \end{aligned}$$

In the above equations, t is the time, and the notation $\dot{x}$ stands for the derivative of the variable x with respect to the time.

Finally, we now are able to formalize our hypothesis, which states that there should be no rotation ($\Omega_x = \Omega_y = \Omega_z = 0$), and that the remaining translation is confined to a plane parallel to the image ($T_z = 0$). We may then rewrite equations 52 and 53 for the special case treated in this section, where $Z = Z_0$ is considered as fixed, yielding the true motion flow field

$$\mathbf{w} = \begin{pmatrix} T_x/Z_0 \\ T_y/Z_0 \end{pmatrix} \tag{54}$$

One clearly can see in equation 54 that the magnitude of the motion field gradient $|\mathbf{\nabla w}|$ is zero everywhere; it follows immediately that $\int |\mathbf{\nabla w}|^2 \, d\mathbf{s} = 0$. Since this value is the lowest possible bound for Θ, then we conclude that any translational intensity pattern does minimize Θ. The first step of the proof is done.

Conversely, consider the relation 46 and develop each term of the integrand according to equations 52 and 53; we get

$$\begin{aligned} \partial v_x/\partial x &= -\Omega_x y + 2\Omega_y x - \frac{T_z}{Z} \qquad (55) \\ \partial v_x/\partial y &= -\Omega_x x - \Omega_z \qquad (56) \\ \partial v_y/\partial x &= \Omega_y x + 2\Omega_x y - \frac{T_z}{Z} \qquad (57) \\ \partial v_y/\partial y &= \Omega_y y + \Omega_z \qquad (58) \end{aligned}$$

If Θ is to reach its absolute minimum, then each term of equations 55 through 58 has to be zero. Taking this into account, the addition of equations 56 and 58 yields

$$\Omega_y y = \Omega_x x \tag{59}$$

The multiplication of the relations 55 and 57 by y preserves their zero-value, and allows one to equate them

$$-\Omega_x y^2 + 2\Omega_y xy - y\frac{T_z}{Z} = \Omega_y xy - 2\Omega_x y^2 - y\frac{T_z}{Z} \tag{60}$$

Cancelling the identical terms in the above equation, and replacing the $(\Omega_y y)$ terms using equation 59, one gets

$$-\Omega_x y^2 + 2\Omega_x x^2 = \Omega_x x^2 - 2\Omega_x y^2 \tag{61}$$

$$\Omega_x(y^2 + x^2) = 0 \tag{62}$$

This last equation can be satisfied everywhere only if $\Omega_x = 0$. This implication, together with equation 59, constrains Ω_y to be zero as well; using equation 56, it also constrains Ω_z to be zero. These considerations can be summed up in the statement that no rotation can ever be produced by constructing a motion flow field under the constraint $\Theta = 0$. Furthermore, the introduction of this constraint into equation 55 shows that any translation along the optical axis is forbidden, as it implies that $T_z = 0$.

In conclusion, we have first shown that a mere translational motion flow field generates the absolute minimal value for the functional Θ; then we have shown that constructing a flow field under the hypothesis $\Theta = 0$ yields a translational flow field again. This conveys itself in an equivalence relation, implying that trying to minimize Θ is the best thing to do when faced to a mere translational motion field such that there is no movement along the optical axis.

5.6 Smoothness assumption applied to a rigid polyhedron

The new hypothesis, valid for this section only, states that the imaging system consists of a single orthographic camera looking at some moving object. This object is a rigid polyhedron, subject to the additional constraint that any intersection of edges in the image plane should correspond to a true intersection of edges in space. We prove here that, under these hypothesis, any motion computed using the functional Θ of equation 46 is the correct one.

First, we show (in a more precise way than in section 5.2) that the constraints on motion imposed by two edges meeting at a vertex are sufficient to compute its correct two-dimensional velocity; then we show that the knowledge of the velocity at the end-points of a line segment, together with the motion constraints given along the segment, are sufficient for the correct computation of the complete velocity field along the whole segment. Hence, the polyhedron's motion will be entirely determined.

We introduce, within the image plane, the unit vectors $\mathbf{u}_i^\top$ and $\mathbf{u}_i^\perp$, parallel, respectively perpendicular to the i-th segment, where i ranges from 1 to 2, and where the corresponding segments meet at a vertex. We are able to find the velocity component $v_i^\perp$ by using equation 33, while $v_i^\top$ remains undetermined. However, the vertex belongs to both segments and has a velocity which should be compatible with the constraints provided by each one. We may write

$$\mathbf{u}_1^\top = \begin{pmatrix} \cos\alpha_1 \\ \sin\alpha_1 \end{pmatrix} \qquad \mathbf{u}_1^\perp = \begin{pmatrix} -\sin\alpha_1 \\ \cos\alpha_1 \end{pmatrix} \tag{63}$$

$$\mathbf{u}_2^\top = \begin{pmatrix} \cos\alpha_2 \\ \sin\alpha_2 \end{pmatrix} \qquad \mathbf{u}_2^\perp = \begin{pmatrix} -\sin\alpha_2 \\ \cos\alpha_2 \end{pmatrix} \tag{64}$$

which leads to a set of four simultaneous linear equations, valid at the vertex. The unknowns are only three, i.e. v_x, v_y and $v_1^\top = v_2^\top$, but the system is not overdetermined since trigonometric relations do relate each of the equations to all the others.

$$\mathbf{v} = v_1^\top \mathbf{u}_1^\top + v_1^\perp \mathbf{u}_1^\perp = v_2^\top \mathbf{u}_2^\top + v_2^\perp \mathbf{u}_2^\perp \tag{65}$$

$$\left\{ \begin{array}{lcl} v_x & = & v_1^\top \cos\alpha 1 - v_1^\perp \sin\alpha 1 \\ v_x & = & v_2^\top \cos\alpha 2 - v_2^\perp \sin\alpha 2 \\ v_y & = & v_1^\top \sin\alpha 1 + v_1^\perp \cos\alpha 1 \\ v_y & = & v_2^\top \sin\alpha 2 + v_2^\perp \cos\alpha 2 \end{array} \right. \tag{66}$$

We eliminate the unknown $v_i^\top$ variables by writing

$$\left\{ \begin{array}{lclcl} v_y \cos\alpha_1 - v_x \sin\alpha_1 & = & v_1^\perp \cos^2\alpha 1 + v_1^\perp \sin^2\alpha 1 & = & v_1^\perp \\ v_y \cos\alpha_2 - v_x \sin\alpha_2 & = & v_2^\perp \cos^2\alpha 2 + v_2^\perp \sin^2\alpha 2 & = & v_2^\perp \end{array} \right. \tag{67}$$

The solution may be obtained by the Cramer method, yielding

$$v_x = \frac{\begin{vmatrix} \cos\alpha_1 & v_1^\perp \\ \cos\alpha_2 & v_2^\perp \end{vmatrix}}{\begin{vmatrix} \cos\alpha_1 & -\sin\alpha_1 \\ \cos\alpha_2 & -\sin\alpha_2 \end{vmatrix}} = \frac{v_2^\perp \cos\alpha_1 - v_1^\perp \cos\alpha_2}{-\sin\alpha_2\cos\alpha_1 + \sin\alpha_1\cos\alpha_2} \tag{68}$$

$$v_y = \frac{\begin{vmatrix} v_1^\perp & -\sin\alpha_1 \\ v_2^\perp & -\sin\alpha_2 \end{vmatrix}}{\begin{vmatrix} \cos\alpha_1 & -\sin\alpha_1 \\ \cos\alpha_2 & -\sin\alpha_2 \end{vmatrix}} = \frac{-v_1^\perp \sin\alpha_2 + v_2^\perp \sin\alpha_1}{-\sin\alpha_2\cos\alpha_1 + \sin\alpha_1\cos\alpha_2} \tag{69}$$

We have unambiguously retrieved the velocity $\mathbf{v}$ of the vertex, as far as the denominator of equations 68 or 69 is non-zero. This is the case whenever $\alpha_1 - \alpha_2 \neq k\pi$, with k being any negative, null or positive integer. This restriction is simply an other way of stating that the two segments have to cross one another.

Now, the first step of the demonstration is terminated. We will go on by showing that if a line segment rotates and translates in space, with known orthographically projected velocity vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ at its end-points, then the two-dimensional velocity field that satisfies the constraints given by $v^\perp(s)$ along the projected line, and minimizes Θ, is the correct projected velocity field.

The orientation of the unit vectors $\mathbf{u}^\top$ and $\mathbf{u}^\perp$ is fixed along the whole segment for a given time t. Hence

$$\frac{\partial \mathbf{v}}{\partial s} = \frac{\partial v^\top}{\partial s}\mathbf{u}^\top + \frac{\partial v^\perp}{\partial s}\mathbf{u}^\perp \tag{70}$$

Remembering the second part of equation 37, one may write

$$\Theta = \int_{s_0}^{s_1} \left|\frac{\partial \mathbf{v}}{\partial s}\right|^2 ds = \int_{s_0}^{s_1} \left(\left(\frac{\partial v^\top}{\partial s}\right)^2 + \left(\frac{\partial v^\perp}{\partial s}\right)^2\right) ds \tag{71}$$

Since $\partial v^\perp/\partial s$ is known, we may ignore, in the minimization process, the last term of the integrand of equation 71. Introducing, as a constraint, our knowledge of the end-point velocities, one can describe our problem by the search of a function $v^\top(s)$ minimizing

$$\Theta = \int_{s_0}^{s_1} \left(\frac{\partial v^\top}{\partial s}\right)^2 ds \tag{72}$$

and subject to the end-points constraint

$$\int_{s_0}^{s_1} \frac{\partial v^\top}{\partial s} ds = v_1^\top - v_0^\top \tag{73}$$

It can be shown by some isoperimetric analysis that the solution to equations 72 and 73 is *always linear*; the sections 8.1 and 8.2 give a formal proof of this fact.

Since both the true component $\mathbf{w}^\top$ of the motion flow field and the computed component $\mathbf{v}^\top$ of the optical flow field are linear functions which satisfy the same end-point velocities, they must be the very same function. Therefore, the smoothest optical flow field is the correct orthographically projected two-dimensional motion flow field for a rigid polyhedron, when all intersections in the image can be attributed to intersections of edges in space.

5.7 Conclusion on the smoothness assumption

We have seen that the smoothness assumption of equation 46 is sufficient to regularize the otherwise ill-posed problem of optical flow computation; furthermore, we have shown at least two cases where the obtained optical flow field is the correct one. The hypothesis in the first case state that the projection is of a perspective type, that no rotation or deformation of the moving object is allowed, and that there is no motion along the optical axis. The hypothesis in the second case state that the projection is of an orthographic type, that the moving object consists of a rigid polygon, and that every intersection in the projection plane may be linked to some intersection of the polygon's edges in space.

After the uniqueness and physical plausibility concerns, we still have to find some qualitative case where the motion computed using this smoothness assumption is consistent with human perception, in order to completely fulfill the requirements of section 5.4. This indeed can be shown to be the case for such illusions like the *barber's pole illusion*, where one is told to look at a spiral wrapped around a rotating cylinder. Human beings usually perceive a pattern moving along the cylinder's axis, instead of the local motion produced by the rotation, tangent to the cylinder and perpendicular to the axis. Now, the velocity flow field recovered using a smoothness assumption is subject, for this particular case, to exactly the same behavior.

5.8 Smoothness computation along a curve

The sections 5.4 through 5.7 have demonstrated that the smoothness assumption was a good choice for regularizing the computation of the optical flow. We want now to show some numerical techniques for solving equation 46 in the discrete world of numerical computations. The task is to build the flow field $\mathbf{v}$ minimizing $\Theta = \int_{\mathbf{s}\in I} |\nabla\mathbf{v}|^2\, d\mathbf{s}$ under the constraint $\mathbf{v}\cdot\mathbf{u}^\perp - v^\perp = 0$. Merging these two conditions, and introducing a confidence weight γ^2 whose purpose is to take into account the possible errors of measurement, one may want to minimize

$$\Theta = \oint_{\mathbf{s}\in\Gamma} \left(|\nabla\mathbf{v}|^2 + \gamma^2(\mathbf{s}) \left(\mathbf{v}(\mathbf{s})\cdot\mathbf{u}^\perp(\mathbf{s}) - v^\perp(\mathbf{s}) \right)^2 \right) d\mathbf{s} \tag{74}$$

where we have temporarily restricted ourselves to the computation of the optical flow along some curve Γ only. After discretization, the gradient operator ∇ will be replaced by some finite difference. This leads to the minimization of

$$\theta = \sum_{i=0}^{N-1} \left((v_{x,i} - v_{x,i-1})^2 + (v_{y,i} - v_{y,i-1})^2 \right) + \gamma_i^2 (v_{x,i}u_{x,i}^\perp + v_{y,i}u_{y,i}^\perp - v_i^\perp)^2 \tag{75}$$

where N is the number of points used during the minimization process. Note that the notation above is tuned to a case studied in [Hild 83], where the optical flow field is computed along some contours which are generally closed, and do originate from the zero-crossings of a Laplacian of a Gaussian of the image. The consequence is that equation 75 is more suited to a unidimensional, circular case; hence, $v_{-1} = v_{N-1}$. We will present in section 5.12 the more general case of a full two-dimensional optical flow field.

We have now to solve the $2N$ simultaneous linear equations

$$\left\{ \begin{array}{rcl} 0 & = & \partial\theta/\partial v_{x,i} \\ 0 & = & \partial\theta/\partial v_{y,i} \end{array} \right. \qquad \forall i \in [0, N[\tag{76}$$

which can be developed into

$$\left\{ \begin{array}{rcl} \left(4 + 2\gamma_i (u_{x,i}^\perp)^2\right) v_{x,i} - 2v_{x,i+1} - 2v_{x,i-1} + 2\gamma_i u_{x,i}^\perp u_{y,i}^\perp v_{y,i} & = & 2\gamma_i v_i \perp u_{x,i}^\perp \\ \left(4 + 2\gamma_i (u_{y,i}^\perp)^2\right) v_{y,i} - 2v_{y,i+1} - 2v_{y,i-1} + 2\gamma_i u_{x,i}^\perp u_{y,i}^\perp v_{x,i} & = & 2\gamma_i v_i \perp u_{y,i}^\perp \end{array} \right.$$
$$\forall i \in [0, N[\tag{77}$$

The right hand side contains the known, measured terms γ_i and $v_i^\perp$. The left hand-side contains the unknown variables $v_{x,i}$ and $v_{y,i}$. Stated in a more concise frame of matrix formulation, we have to solve

$$A \cdot \mathbf{v} = \mathbf{b} \tag{78}$$

or, equivalently

$$\sum_{j=0}^{2N-1} a_{i,j} v_j = b_i \qquad \forall i \in [0, 2N[\tag{79}$$

where the correspondence between the terms of equation 79 and those of equation 77 should be obvious.

5.9 Solution by matrix inversion

The solution of equation 78 is well known, and quite easy to write in a matrix algebraic form

$$\mathbf{v} = A^{-1} \cdot \mathbf{b} \tag{80}$$

However, it is also very clear that the computational cost of this direct matrix inversion is, to say, high. Keep in mind that N may grow to the size of several of hundreds, which makes the burden of computing A^{-1} quiet overwhelming. Notwithstanding that this matrix inversion scheme is the ultimate, correct solution to our problem, we will try in the next sections to alleviate the computational load by finding some approximations to the solution, which may be less accurate, but more tractable.

5.10 Solution by relaxation algorithms

Consider a reordering of the terms of equation 79; one gets

$$a_{i,i} v_i = b_i - \sum_{j \in [0,2N[\setminus \{i\}} a_{i,j} v_j \qquad \forall i \in [0, 2N[\tag{81}$$

where $\setminus$ denotes the suppression operator. This equation can be used as basis for an iterative search of the solution, trying at each step k to refine some initial approximation described by

$$k = 0 \qquad v_i^k = 0 \qquad \forall i \in [0, 2N[\tag{82}$$

The Jacobi relaxation is inherently parallel. It updates simultaneously each component of the whole velocity vector to get $\mathbf{v}^{k+1}$ from step k. It proceeds by letting

$$a_{i,i} v_i^{k+1} = b_i - \sum_{j \in [0,2N[\setminus \{i\}} a_{i,j} v_j^k \qquad \forall i \in [0, 2N[\tag{83}$$

The Gauss-Seidel relaxation is inherently sequential. Each component of the velocity vector is refined one after the other, using the actual value of all the other components.

The update equation is

$$a_{i,i}v_i^{k+1} = b_i - \sum_{j\in[0,i[} a_{i,j}v_j^{k+1} - \sum_{j\in]i,2N[} a_{i,j}v_j^k \qquad i \in [0,2N[\tag{84}$$

In the notation above (equation 84 only), the i-indexes have to be drawn in an ascending order from the $[0, 2N[$ set for the method to work.

Convergence considerations: the matrix A corresponding to the system 77 can be shown to be symmetric and positive definite. This feature is welcome, since in this case both Jacobi and Gauss-Seidel relaxation algorithms will converge for $k \to \infty$.

Speed of convergence: the parallelism involved in the Jacobi relaxation is appealing, but since it usually has to be simulated on a sequential machine, the expected gain in computation speed is lost. In fact, the Gauss-Seidel relaxation shows a faster convergence rate in term of number of iterations needed to get an approximation with some given accuracy.

5.11 Solution by the conjugate gradient algorithm

The algorithm presented in this section does also exhibit convergence for any symmetric and positive definite matrix; its particularity is that the number of iterations needed to attain the final convergence is finite. The upper bound on the number of iterations can be shown to be equal to the dimension $2N$ of the matrix A. The initialization step comes first; then runs the algorithm loop, without any further comments.

Step–I	$k = 0$	$\mathbf{q}^0 = A \cdot \mathbf{v}^0 - \mathbf{b}$	$\mathbf{p}^0 = \mathbf{b} - A \cdot \mathbf{v}^0$
Step–II	$k = k + 1$		
Step–III	$\alpha^k = -\frac{(\mathbf{q}^{k-1})^T \cdot \mathbf{p}^{k-1}}{(\mathbf{p}^{k-1})^T \cdot A \cdot \mathbf{p}^{k-1}}$		
Step–IV	$\mathbf{v}^k = \mathbf{v}^{k-1} + \alpha^k \mathbf{p}^{k-1}$		
Step–V	$\mathbf{q}^k = A \cdot \mathbf{v}^0 - \mathbf{b}$		
Step–VI	$\beta^k = -\frac{(\mathbf{q}^k)^T \cdot A \cdot \mathbf{p}^{k-1}}{(\mathbf{p}^{k-1})^T \cdot A \cdot \mathbf{p}^{k-1}}$		
Step–VII	$\mathbf{p}^k = -\mathbf{q}^k + \beta^k \mathbf{p}^{k-1}$		
Step–VIII	$(\mathbf{q}^k \neq \mathbf{0}) \;\Rightarrow\;$ Step–II	$\wedge$	$(\mathbf{q}^k = \mathbf{0}) \;\Rightarrow\;$ End.

This algorithm, originating from the mathematical programming field [Luen 73], is very efficient in the sense that its convergence is attained in a linear number of steps (specifically, $2N$). Furthermore, each step is built with blocks which can be parallelized.

5.12 Computation of a smooth velocity field on a surface

Equation 75 has been written in a way related to the minimization of a flow field defined along some curve only. If we decide to use the full two-dimensional retinal plane instead, we should develop equation 74 into

$$\Theta = \int_{\mathbf{s}\in I} \left(\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 + \gamma^2 \left(\mathbf{v}\cdot\mathbf{u}^\perp - v^\perp\right)^2 \right) ds \qquad (85)$$

where the argument $\mathbf{s}$ has been generally dropped. The derivation with respect to v_x gives the condition to be fulfilled for the minimization of Θ

$$\begin{aligned} 0 = \frac{\partial\Theta}{\partial v_x} &= \int_{\mathbf{s}\in I} \left(2\frac{\partial v_x}{\partial x}\frac{\partial(\partial v_x/\partial x)}{\partial v_x} + 2\frac{\partial v_x}{\partial y}\frac{\partial(\partial v_x/\partial y)}{\partial v_x} + \right. \\ & \left. \gamma^2 \left(2(u_x^\perp)^2 v_x + 2u_x^\perp u_y^\perp v_y - 2u_x^\perp v^\perp\right)\right) d\mathbf{s} \end{aligned} \qquad (86)$$

The same process may be applied to the y-component as well. One gets

$$\left\{ \begin{array}{lcl} u_x^\perp v^\perp - \frac{1}{\gamma^2}\left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2}\right) &=& (u_x^\perp)^2 v_x + u_x^\perp u_y^\perp v_y \\ u_y^\perp v^\perp - \frac{1}{\gamma^2}\left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2}\right) &=& (u_y^\perp)^2 v_y + u_x^\perp u_y^\perp v_x \end{array} \right. \qquad (87)$$

The Laplacians standing within the system of equations 87 have to be estimated using a discrete set of values. A convenient approximation is

$$\nabla^2 v_x = \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \approx 3\left(\bar{v}_{x,i,j} - v_{x,i,j}\right) \qquad (88)$$

where the local average $\bar{v}$ at location (i,j) in the image is defined by

$$\begin{aligned} \bar{v}_{x,i,j} &= \frac{1}{6}\left(v_{x,i-1,j} + v_{x,i,j+1} + v_{x,i+1,j} + v_{x,i,j-1}\right) + \\ & \frac{1}{12}\left(v_{x,i-1,j-1} + v_{x,i-1,j+1} + v_{x,i+1,j+1} + v_{x,i+1,j-1}\right) \end{aligned} \qquad (89)$$

The equivalent equations for $\nabla^2 v_y$ are so obviously similar to equations 88 through 89 that they are not displayed. Using these approximations, one may write now the system 87 for the discrete case

$$\left\{ \begin{array}{lcl} u_{x,i,j}^\perp v_{i,j}^\perp - \frac{3}{\gamma_{i,j}^2}\left(\bar{v}_{x,i,j} - v_{x,i,j}\right) &=& (u_{x,i,j}^\perp)^2 v_{x,i,j} + u_{x,i,j}^\perp u_{y,i,j}^\perp v_{y,i,j} \\ u_{y,i,j}^\perp v_{i,j}^\perp - \frac{3}{\gamma_{i,j}^2}\left(\bar{v}_{y,i,j} - v_{y,i,j}\right) &=& (u_{y,i,j}^\perp)^2 v_{y,i,j} + u_{x,i,j}^\perp u_{y,i,j}^\perp v_{x,i,j} \end{array} \right. \qquad (90)$$

Rearranging the terms, and solving for v_x and v_y while considering the terms $\bar{v}_x$ and $\bar{v}_y$ as fixed, one gets successively

$$\left\{ \begin{array}{lcl} v_{x,i,j}\left((u_{x,i,j}^\perp)^2 - \frac{3}{\gamma_{i,j}^2}\right) + v_{y,i,j}u_{x,i,j}^\perp u_{y,i,j}^\perp &=& u_{x,i,j}^\perp v_{i,j}^\perp - \frac{3}{\gamma_{i,j}^2}\bar{v}_{x,i,j} \\ v_{x,i,j}u_{x,i,j}^\perp u_{y,i,j}^\perp + v_{y,i,j}\left((u_{y,i,j}^\perp)^2 - \frac{3}{\gamma_{i,j}^2}\right) &=& u_{y,i,j}^\perp v_{i,j}^\perp - \frac{3}{\gamma_{i,j}^2}\bar{v}_{y,i,j} \end{array} \right. \qquad (91)$$

and

$$\left\{\begin{array}{lcl} v_{x,i,j} & = & \bar{v}_{x,i,j} + u^{\perp}_{x,i,j} \frac{\bar{v}_{x,i,j} u^{\perp}_{x,i,j} + \bar{v}_{y,i,j} u^{\perp}_{y,i,j} - v^{\perp}_{i,j}}{(3/\gamma^2_{i,j}) - 1} \\ v_{y,i,j} & = & \bar{v}_{y,i,j} + u^{\perp}_{y,i,j} \frac{\bar{v}_{x,i,j} u^{\perp}_{x,i,j} + \bar{v}_{y,i,j} u^{\perp}_{y,i,j} - v^{\perp}_{i,j}}{(3/\gamma^2_{i,j}) - 1} \end{array}\right. \tag{92}$$

This formulation leads very naturally to a Jacobi relaxation algorithm. If one denotes by k the iteration step, then we have, in a short notation

$$\left\{\begin{array}{lcl} v^k_x & = & \bar{v}^{k-1}_x + N^{k-1}/D \\ v^k_y & = & \bar{v}^{k-1}_y + N^{k-1}/D \end{array}\right. \tag{93}$$

The method just developed derives the optical flow field from a set of two time frames only; however, if the whole image sequences are available, then there is no reason to wait for the full convergence of equation 93 using just these two frames [Horn 81]. In fact, it is possible to interlace the convergence steps k and the time t in order to spread the computations over many frames, letting the partial result of some iterations from a previous frame being the initial guess for the next one. As more data is taken into account, the robustness of the algorithm should be enhanced, provided that the motion is stable enough.

6 Optical flow field in correspondence schemes

We have seen in sections 3 and section 4 two rather different approaches to the computation of motion. The first one surmised that working with the image is more sound than working with motion. This point of view emphasizes that one should first do some image analysis as preprocessing step (e.g. to segment the scene into moving and non-moving parts), before attempting to link the results over time in some kind of postprocessing step. The second approach surmised that motion is not a mere consequence, but can also be a basic, stand-alone feature for further understanding of the image.

We will now try to reconcile the two views by the introduction of a scheme for motion computation known as *correspondence scheme*, where the processing is fairly shared among image analysis and computation of motion.

6.1 Velocity issued by a correspondence scheme

Suppose that we are able to find and to label some points of interest in a first image I_0; suppose also that we are able to carry out the same job in a later image I_1. If we can highlight a correspondence between a point of interest $\mathbf{p}^0 = (x^0, y^0, t_0)^T$ in I_0 and its companion $\mathbf{p}^1 = (x^1, y^1, t_1)^T$ in I_1, then the associated velocity is indeed very simple to compute:

$$v^{0,1}_x = \frac{x^1 - x^0}{t_1 - t_0} \qquad\qquad v^{0,1}_y = \frac{y^1 - y^0}{t_1 - t_0} \tag{94}$$

where the superscripts show the origin of the points. Note that this equation is not only trivial, but also fully determined. There is no more need for an assumption like a smooth velocity field in order to get a definite solution; furthermore, there is no need for the knowledge about the world necessary to achieve the essential step of segmentation of section 3. However, there is a drawback: finding the correct set of correspondences between the points of interest is not an easy task at all. Before we attempt to solve this problem, we will first show how to extract the points of interest.

6.2 Points of interest in image sequences

Concentrate now on a single frame I of the image sequences. Our task is to detect in this frame some candidate points for later matching, subject to the requirement that they can be locally identified by some particular characteristic allowing their recognition in a subsequent frame. For this purpose, we will make use here of an interest-operator termed the *Moravec operator* [Mora 77].

The input of this operator $\mathcal{M}$ is some local patch $\mathcal{N}_0 \cup \mathcal{N}_1 \cup \mathcal{N}_2$ of the image I; the output is a binary decision stating if the location of this patch is *candidate* or is *no candidate*. The criterion used for this decision deals with the local energy within the patch; if it is too low, then there are not enough local features to be able do distinguish this patch from an other one. Conversely, if too much activity has to be considered, then any added noise will render the identification unreliable. The Moravec operator is constructed in such a way that it should find a compromise between the two activity extremes.

As a first step, compute the local energy s of the image I by

$$s^2(\mathbf{x}) = \int_{\Delta\mathbf{x} \in \mathcal{N}_0} (I(\mathbf{x} + \Delta\mathbf{x}) - I(\mathbf{x}))^2 \, d\Delta\mathbf{x} \tag{95}$$

where $\mathcal{N}_0$ is some local neighborhood. Then, the first guess for the Moravec operator is

$$M_1(\mathbf{x}) = \min_{\Delta\mathbf{x} \in \mathcal{N}_1} s^2(\mathbf{x} + \Delta\mathbf{x}) \tag{96}$$

where the size of the neighborhood $\mathcal{N}_1$ defines the precision of the operator. A good choice is usually $\mathcal{N}_1 = \mathcal{N}_0$. Only the local maxima are retained, which leading to

$$M_2(\mathbf{x}) = \begin{cases} M_1(\mathbf{x}) & M_1(\mathbf{x}) = \max_{\Delta\mathbf{x} \in \mathcal{N}_2} M_1(\mathbf{x} + \Delta\mathbf{x}) \\ 0 & M_1(\mathbf{x}) \neq \max_{\Delta\mathbf{x} \in \mathcal{N}_2} M_1(\mathbf{x} + \Delta\mathbf{x}) \end{cases} \tag{97}$$

In the above equation, the size of the neighborhood $\mathcal{N}_2$ determines in some way the fraction of the first guesses to be kept. If it is large with respect to $\mathcal{N}_1$, then only few points will be kept. Conversely, if $\mathcal{N}_2$ is empty, then all points will be considered for the next processing step. Usually, we chose $\mathcal{N}_2 = \mathcal{N}_1$ to hold true. The final value for the Moravec operator is obtained by a thresholding stage

$$\mathcal{M}(\mathbf{x}) = \begin{cases} \textit{candidate} & M_2(\mathbf{x}) > T \\ \textit{no candidate} & M_2(\mathbf{x}) \leq T \end{cases} \tag{98}$$

where T is some empirical threshold.

6.3 Correspondence problem

We dispose of some sets of points of interest. Each set belongs to a given frame. Suppose now that we want to register one frame with respect to another one, considering motion as a warping parameter. We will term the first image I_0 and give i-indexes to points of interest in this image. The second image will be I_1, and its points of interest will be indexed with j. If $\mathrm{card}(\{i\}) = N_i$ and $\mathrm{card}(\{j\}) = N_j$, and if we agree that $N_i \geq N_j$, then the number of different possible pairings (no surjections) is as high as $N_i!/(N_i - N_j)!$, which is very big since N_i is usually of a magnitude near to N_j. This observation implies that it is impossible, from a practical point of view, to select the best registration among the exhaustive set of possibilities; one needs some heuristic in order to restrict and guide the choice for a convenient correspondence.

The heuristics at hand can be stated in a way reflecting some basic knowledge of the world. Hopefully, this knowledge lies at a far lower level than was required for the segmentation needed when separating background from moving object (optical flow field in transformation domains). We present some of the heuristics hereafter.

- *Maximal speed*: We keep as potential matches to a given point of interest $\mathbf{p}_i \in I_0$ only the candidate points of interest $\mathbf{p}_j \in I_1$ such that

$$|\mathbf{p}_j - \mathbf{p}_i| \leq V_{max} \tag{99}$$

  where V_{max} is the maximum velocity allowed. Every match standing for a higher speed should be rejected.

- *Grey-level spatial coherence*: We attribute an initial probability $P^0_{i,j}$ to each match between some points of interest referenced by i in the first image, and its partner j in the second image. This initial match's probability is dependent upon the similarity of the two image patches; it is defined by

$$P^0_{i,j} = 1/\left(1 + C \int_{\Delta\mathbf{x} \in \mathcal{N}_3} (I_1(\mathbf{p}_j + \Delta\mathbf{x}) - I_0(\mathbf{p}_i))^2 \, d\Delta\mathbf{x}\right) \tag{100}$$

  where C is some normalization constant, and $\mathcal{N}_3$ a domain whose size is adjusted to fit our needs. The value of this initial probability may be compared to a threshold in order to decide if the corresponding match has to be taken into account or not.

- *Motion coherence*: We decide to exclude a potential match (k, l) if the resulting motion is not coherent with some already granted match (i, j), where $\mathbf{p}_k$ is some neighbor of $\mathbf{p}_i$. The remaining matches (m, n) consistent with the match (i, j), both in terms of neighborhood and in terms of motion, form a set $\mathcal{M}_{i,j}$ such that

$$\mathcal{M}_{i,j} = \{(m, n) \mid (|\mathbf{v}_{m,n} - \mathbf{v}_{i,j}| \leq \Delta V_{max}) \ \wedge \ (\mathbf{p}_m - \mathbf{p}_i \in \mathcal{N}_4)\} \tag{101}$$

  where $\mathbf{v}_{i,j}$ is defined by $\mathbf{v}_{i,j} = \mathbf{p}_j - \mathbf{p}_i$, ΔV_{max} is the coherence bound for velocity, and where $\mathcal{N}_4$ is the coherence domain.

- *Bijection criterion*: As occlusions may occur anywhere, one should consider the case where several points of interest share the same projection in the image plane. For that matter, it makes sense to allow a given $\mathbf{p}_i$ in the first frame to generate at the same time several correspondences with the other frame. On the other hand, the probability that such cases do occur is rather small. Therefore, the imposition of a one-to-one mapping is generally seen as more beneficial, in terms of reduction of the correspondence search space, than detrimental in terms of decrease in performances.

6.4 Solution to the correspondence problem

One solution to the correspondence problem can be found in [Barn 79], where all the heuristics described in section 6.3 have been used. In particular, the contribution from the heuristic for grey-level spatial coherence stays at the heart of the method. In fact, the order of the operations is first to restrict as far as possible the number of candidate matches by using all heuristics together. Then, we may iteratively update the matching probabilities of equation 100 by using some rules, until these probabilities converge to either a high or a low value; those matches with high probability value will be kept as solutions.

We define the goodness $q_{i,j}^k$ of a particular match (i, j) by

$$q_{i,j}^k = \sum_{(m,n)\in\mathcal{M}_{i,j}} P_{i,j}^{k-1} \tag{102}$$

where k is the iteration step, and where the set $\mathcal{M}_{i,j}$ has been defined in equation 101; the initial guess $P_{i,j}^0$ results from equation 100. Then, we compute the new, unnormalized probability for match (i, j), writing

$$\tilde{P}_{i,j}^k = P_{i,j}^{k-1}(\alpha + \gamma q_{i,j}^k) \tag{103}$$

where α is some update decay, and γ some gain factor related to the cardinality of the set $\mathcal{M}_{i,j}$. We normalize the probability value by simply letting

$$P_{i,j}^k = \tilde{P}_{i,j}^k \;/ \sum_{(m,n)\in\mathcal{M}_{i,j}} \tilde{P}_{m,n}^k \tag{104}$$

The convergence is attained only for $k \to \infty$, but some probability values begin to emerge rather quickly. In order to speed up the convergence rate, one may want to accept immediately their corresponding match, and remove it from the set of potential candidates.

6.5 Relation to gradient scheme

To conclude, we will compare the optical flow field, as obtained using a Gradient Scheme (GS), with the optical flow field obtained using a Correspondence Scheme (CS). This comparison demonstrates that each method has some advantages and some drawbacks, with no clear winner.

- Spatial density

GS: The optical flow field is defined almost everywhere. The only points where it is undefined are those where the gradient is non-existent, i.e. $|\nabla I| = 0$. Therefore, the velocity field in a gradient scheme is very dense.

CS: The optical flow field is defined only at certain points of interest, the number of which being under control. But the correspondence scheme is only attractive when a number of points of interest not too high has to be taken into account. Therefore, the velocity field in a correspondence scheme is low.

- Temporal density

GS: The temporal precision in a gradient scheme is high for a given velocity, since the derivatives have to be estimated over a short time interval Δt.

CS: The temporal precision in a correspondence scheme is low, since the displacement between the two frames has to be large with respect to the spatial quantization step. This constraint implies a long time interval Δt between the acquisition of the two frames.

- Motion range

GS: The upper and lower bounds of the velocities detected in a gradient scheme for motion estimation are dependent from the image itself. If the image is highly contrasted, then slow movements will be fairly computed, while high velocities will be badly estimated. On the contrary, a very smooth image will show a higher tolerance for high velocities, but will tend to have quantization problem for slow motions.

CS: The lower bound for motion detection in a correspondence scheme is dependent upon the accuracy of the position of points of interest. If a simple method is used, then this accuracy can be at most equal to the spatial quantization step. The upper bound is dependent, among other things, upon the parameter V_{max} of equation 99.

- Aperture problem

GS: The gradient scheme is notorious for the ability to compute the component of velocity in the direction of the local gradient only; the component's tangent to the edge is lost. The regularization of this problem requires the introduction of a smoothness assumption.

CS: The correspondence scheme requires the registration of one frame with respect to the other, considering motion as a warping deformation. Now, a limited number of tokens is selected in each frame, and the registration is done using these tokens. If their size is too small, then it can happen that an insufficient number of features is available to reliably distinguish one from an other. This is usually the case for the algorithm described. Then, the solution of the correspondence problem requires the introduction of several heuristics.

- Robustness

  GS: The gradient scheme is locally poorly robust, since it is based on the numerical computation of derivatives. However, it produces a dense velocity field, and many values can be fed to an averaging or smoothing filter in order to enhance the general reliability.

  CS: The correspondence scheme for computing an optical flow field is as robust as the correspondence allows; but the low density of the resulting field prevents one to use postprocessing steps for result enhancement.

- Complexity

  GS: In a gradient scheme, the complexity of computing the local component of motion along the gradient direction is low. The complexity of recovering the component of motion perpendicular to the gradient direction is much higher.

  CS: In a correspondence scheme, the complexity of finding the interest points is low, but the complexity of establishing the correspondences is high. Once the correspondences are found, the task of motion computation is of a vanishing complexity.

7 Relation to human perception

The biological visual system is basically a *continuous system*, even if the receptive field is built of a finite number of discrete cells, and even if the action potential spikes issued from these cells are discrete events. Now, this continuous system is not only able to deal with the continuous motions of the natural world, but it obviously can accept some visual input which is not continuous, too, and still treat it as if it were a smooth, uninterrupted motion. The success of motion-pictures, or of television systems, proves it.

Interestingly, the two main computational processes discussed in this lecture can find a counterpart in human vision, although these processes are discrete. The intensity-based method, or gradient scheme, can be associated to some short range process, where at least ten frames per second have to be presented, and where the visual angle velocity should not exceed fifteen minutes of arc per frame, in order to be perceived as a smooth movement. The token-based method, or correspondence scheme, can be associated to some long range process, where the frames can be as few as two and a half per second, and where the visual angle velocity can be as high as ten degrees per frame, while still allowing reliable object tracking.

As a conclusion, we should mention that the human visual system seems to solve quite efficiently the problem of smoothness computation; this is not a surprise, since we have seen that this problem could be parallelized and distributed over time. The human visual system is also capable to handle the correspondence task quite well; in this respect, the technical solution proposed in this lecture is still perfectible. Another aspect, left open until now, is the detection of motion discontinuities; here also, the race between the human beings and the silicon beings is won by the former.

8 Appendices

8.1 Euler equation

The Euler equation is a relation satisfied when minimizing an integral of a well defined kind. First, we will give the general frame for establishing this equation, and then we will apply it to the problem of section 5.6.

Consider the task of finding the minimum of

$$\Theta = \int_{x_0}^{x_1} f(x, y, y')\,dx \tag{105}$$

We build the auxiliary function $\tilde{y}$

$$\tilde{y}(x) = y(x) + \xi\eta(x) \tag{106}$$

where $\eta(x)$ is any differentiable function subject to

$$\eta(x_0) = \eta(x_1) = 0 \tag{107}$$

We may then write a new functional Φ dependent upon ξ only

$$\Phi(\xi) = \int_{x_0}^{x_1} f(x, \tilde{y}, \tilde{y}')\,dx = \int_{x_0}^{x_1} f(x, y(x) + \xi\eta(x), y'(x) + \xi\eta'(x))\,dx \tag{108}$$

Derivating equation 108 with respect to ξ yields

$$\frac{d\Phi(\xi)}{d\xi} = \int_{x_0}^{x_1} \left(\frac{\partial f}{\partial x}\frac{dx}{d\xi} + \frac{\partial f}{\partial \tilde{y}}\frac{d\tilde{y}}{d\xi} + \frac{\partial f}{\partial \tilde{y}'}\frac{d\tilde{y}'}{d\xi} \right) dx \tag{109}$$

Using equation 106 and selecting $\xi = 0$, one can minimize Φ

$$\begin{aligned}
\left.\frac{d\Phi(\xi)}{d\xi}\right|_{\xi=0} &= \int_{x_0}^{x_1} \left(\frac{\partial f}{\partial y}\eta(x) + \frac{\partial f}{\partial y'}\eta'(x) \right) dx \\
&= \int_{x_0}^{x_1} \frac{\partial f}{\partial y}\eta(x)\,dx + \left.\frac{\partial f}{\partial y'}\eta(x)\right|_{x_0}^{x_1} - \int_{x_0}^{x_1} \frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right)\eta(x)\,dx \\
&= 0
\end{aligned} \tag{110}$$

The condition we imposed on the otherwise free function $\eta(x)$ at equation 107 implies that

$$0 = \int_{x_0}^{x_1} \left(\frac{\partial f}{\partial y} - \frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) \right) \eta(x)\,dx \tag{111}$$

We may now build $\eta(x)$ in such a way that

$$\mathrm{sign}(\eta(x)) = \mathrm{sign}\left(\frac{\partial f}{\partial y} - \frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) \right) \tag{112}$$

Finally, the Euler relation stems from equations 111 and 112

$$\frac{\partial f}{\partial y} - \frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) = 0 \tag{113}$$

In our case, we have $x = s$, $y = v^\top$ and $f(x, y, y') = (\partial v^\top / \partial s)^2 = (y')^2$.

8.2 Lagrange multiplier

Consider now the more subtle minimization of an integral subject to some other *integral* condition. We keep equation 105, but we take into consideration the constraint

$$G = \int_{x_0}^{x_1} g(x, y, y')\, dx \tag{114}$$

where g is a fixed function, and G a fixed value. We may then introduce an auxiliary function h, such that

$$h(x, y, y') = f(x, y, y') + \lambda g(x, y, y') \tag{115}$$

where λ is termed the *Lagrange multiplier*. As the minimization of h is equivalent to the minimization of f, we get from the Euler relation

$$\frac{\partial h}{\partial y} - \frac{d}{dx}\left(\frac{\partial h}{\partial y'}\right) = 0 \tag{116}$$

Writing some well chosen total derivative yields

$$\begin{aligned} \frac{d}{dx}\left(h - \frac{\partial h}{\partial y'}y'\right) &= \frac{\partial h}{\partial x} + \frac{\partial h}{\partial y}y' + \frac{\partial h}{\partial y'}y'' - \left(\frac{d}{dx}\left(\frac{\partial h}{\partial y'}\right)y' + \frac{\partial h}{\partial y'}y''\right) \\ &= \frac{\partial h}{\partial x} + \left(\frac{\partial h}{\partial y} - \frac{d}{dx}\left(\frac{\partial h}{\partial y'}\right)\right)y' \end{aligned} \tag{117}$$

Remembering equation 116, and observing that h is not explicitly dependent upon x ($\partial h / \partial x = 0$), we have the following implication

$$\frac{d}{dx}\left(h - \frac{\partial h}{\partial y'}y'\right) = 0 \tag{118}$$

In our case, we are subject to the conditions $g = \partial v^\top / \partial s$ and $G = v_1^\top - v_0^\top$. The other variables have already been introduced at the end of section 8.1; this leads to

$$h - \frac{\partial h}{\partial y'}y' = \left(\frac{\partial v^\top}{ds}\right)^2 + \lambda v^\top - (2\frac{\partial v^\top}{ds} + \lambda)\frac{\partial v^\top}{ds} = -\left(\frac{\partial v^\top}{ds}\right)^2 = -C \tag{119}$$

where C is some constant obtained during the indefinite integration of equation 118. This constant has to be specified by the boundary conditions for $v^\top(s)$, yielding a linear solution

$$v^\top(s) = \frac{v_1^\top - v_0^\top}{s_1 - s_0}(s - s_0) + v_0^\top \tag{120}$$

9 References

[Abid 84] Abidi M. A., Gonzalez R. C., July 1984, "Motion Detection in Radar Images", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 2**, pp. 787–790.

[Adiv 85] Adiv G., July 1985, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–7, no. 4**, pp. 384–401.

[Adiv 89] Adiv G., May 1989, "Inherent Ambiguities in Recovering 3-D Motion and Structure from a Noisy Flow Field", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 477–489.

[Aisb 89] Aisbett J., May 1989, "Optical Flow with an Intensity-Weighted Smoothing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 512–522.

[Aloi 84] Aloimonos J., Brown C. M., July 1984, "The Relationship Between Optical Flow and Surface Orientation", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 1**, pp. 542–545.

[Aloi 90] Aloimonos J., Hervé J. Y., May 1990, "Correspondenceless Stereo and Motion: Planar Surfaces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–12, no. 5**, pp. 504–510.

[Aoki 84] Aoki M., July 1984, "Detection of Moving Objects Using Line Image Sequence", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 2**, pp. 784–786.

[Ball 82] Ballard D. H., Brown C. M., 1982, *Computer Vision*, Prentice Hall.

[Barn 79] Barnard S. T., Thompson W. B., January 1979, "Disparity Analysis of Images", *Technical Report 79-1*, Computer Science Dept., Univ. Minnesota.

[Batc 85] Batchelor B. G., Hill D. A., Hodgson D. C., 1985, *Automated Visual Inspection*, IFS Ltd.

[Blos 88] Blostein S. D., Huang T. S., April 1988, "Detection of Small Moving Objects in Image Sequences Using Multistage Hypothesis Testing", *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, New York Hilton, New York City, **Vol. 2**, pp. 1068–1071.

[Bout 89] Bouthemy P., May 1989, "A Maximum Likelihood Framework for Determining Moving Edges", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 499–511.

[Bres 87] Bresler Y., Merhav S. J., January 1987, "Recursive Image Registration with Application to Motion Estimation", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **Vol. ASSP–35, no. 1**, pp. 70–85.

[Buxt 85] Buxton B. F., Murray D. W., November 1985, "Optic Flow Segmentation as an Ill-Posed and Maximum Likelihood Problem", *Image and Vision Computing*, **Vol. 3, no. 4**, pp. 163–169.

[Chen 88] Chen J. S., Lin W. C., November 1988, "A New Surface Interpolation Technique for Reconstructing 3-D Objects from Serial Cross-Sections", *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 2**, pp. 1100–1102.

[Chri 88] Christensen H. I., Granum E., March 1988, "On Token-Matching in Real Time Motion Analysis", *Proc. BPRA Fourth International Conference on Pattern Recognition*, Cambridge, U. K., pp. 448–457.

[Dieh 86] Diehl N., Burkhardt H., October 1986, "Planar Motion Estimation with a Fast Convergence Algorithm", *Proc. IEEE Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 2**, pp. 1099–1102.

[Gius 89] Giusto D. D., Vernazza G., January 1989, "Optical Flow Calculation from Feature Space Analysis through an Automatic Segmentation Process", *Signal Processing*, **Vol. 16, no. 1**, pp. 41–51.

[Grov 84] Groves M. H., Rajala S. A., Snyder W. E., March 1984, "Calculation of Displacement Fields by Means of the Motion Detection Transform", *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, San Diego, California, **Vol. 2**, pp. 23.6.1–23.6.4.

[Hayn 83] Haynes S. M., Ramesh J., 1983, "Detection of Moving Edges", *Computer Vision, Graphics, and Image Processing*, **no. 21**, pp. 345–367.

[Hild 83] Hildreth E. C., 1983, *The Measurement of Visual Motion*, The MIT Press, Cambridge, Massachusetts.

[Horn 81] Horn B. K. P., Schunck B. G., 1981, "Determining Optical Flow", *Artificial Intelligence*, **no. 17**, pp. 185–203.

[Huan 81a] Huang T. S., Tsai R. Y., 1981a, *Image Sequence Analysis: Motion Estimation*, Huang T. S. Editor, Springer-Verlag.

[Huan 81b] Huang T. S., Tsai R. Y., April 1981b, "Three-Dimensional Motion Estimation from Image-Space Shifts", *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, Atlanta, Georgia, **Vol. 3**, pp. 1136–1139.

[Huan 88] Huang T. S., Shim Y. S., March 1988, "Linear Algorithm for Motion Estimation: How to Handle Degenerate Cases", *Proc. BPRA Fourth International Conference on Pattern Recognition*, Cambridge, U. K., pp. 439–447.

[Huan 89] Huang T. S., Lee C. H., May 1989, "Motion and Structure from Orthographic Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 536–541.

[Hutc 88] Hutchinson J., Koch C., Luo J., Mead C., March 1988, "Computing Motion Using Analog and Binary Resistive Networks", *Computer*, pp. 52–63.

[Jain 88] Jain R., November 1988, "Dynamic Vision", *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 1**, pp. 226–235.

[Kahn 85] Kahn P., July 1985, "Local Determination of a Moving Contrast Edge", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–7, no. 4**, pp. 402–409.

[Kali 88] Kalivas D. S., Sawchuck A. A., Chellappa R., April 1988, "Segmentation and 2-D Motion Estimation of Noisy Image Sequences", *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, New York Hilton, New York City, **Vol. 2**, pp. 1076–1079.

[Kana 88] Kanatani K., March 1988, "Transformation of Optical Flow by Camera Rotation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–10, no. 2**, pp. 131–143.

[Kear 87] Kearney J. K., Thompson W. B. and Boley L. D., March 1987, "Optical Flow Estimation: An Error Analysis of Gradient-Based Methods with Local Optimization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–9, no. 2**, pp. 229–244.

[Kees 88] Keesman G., September 1988, "Motion Estimation Based on a Motion Model Incorporating Translation, Rotation and Zoom", *Proc. EUSIPCO Fourth European Signal Processing Conference*, **Vol. 1**, Grenoble, France, pp. 31–34.

[Konr 88] Konrad J., Dubois E., April 1988, “Estimation of Image Motion Fields: Bayesian Formulation and Stochastic Solution”, *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, New York Hilton, New York City, **Vol. 2**, pp. 1072–1075.

[Kori 84] Kories R., Zimmermann G., July 1984, “Motion Detection in Image Sequences: An Evaluation of Feature Detectors”, *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 2**, pp. 778–780.

[Kors 90] Korsten M. J., Houkes Z., April 1990, “The Estimation of Geometry and Motion of a Surface from Image Sequences by Means of Linearization of a Parametric Model”, *Computer Vision, Graphics, and Image Processing*, **Vol. 50, no. 1**, pp. 1–28.

[Lai 88] Lai S. H., Chang S., September 1988, “Estimation of 2-D Translation via Hadamard Transform”, *Proc. EUSIPCO Fourth European Signal Processing Conference*, **Vol. 1**, Grenoble, France, pp. 35–37.

[Lawt 83] Lawton D. T., 1983, “Processing Translational Motion Sequences”, *Computer Vision, Graphics, and Image Processing*, **no. 22**, pp. 116–144.

[Limb 75] Limb J. O., Murphy J. A., 1975, “Estimating the Velocity of Moving Images in Television Signals”, *Computer Vision, Graphics, and Image Processing*, **no. 4**, pp. 311-327.

[Lin 86] Lin Z. C., Lee H., Huang T. S., October 1986, “Finding 3-D Point Correspondences in Motion Estimation”, *Proc. IAPR Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 1**, pp. 303–305.

[Liu 86] Liu Y., Huang T. S., October 1986, “Estimation of Rigid Body Motion Using Straight Line Correspondences: Further Results”, *Proc. IAPR Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 1**, pp. 306–309.

[Luen 73] Luenberger D. G., 1973, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, Massachusetts.

[Meco 89] Mecocci A., October 1989, “Moving Object Recognition and Classification in External Environments”, *Signal Processing*, **Vol. 18, no. 2**, pp. 183–194.

[Miti 86] Mitiche A., Seida S., Aggarval J. K., October 1986, “Interpretation of Structure and Motion from Line Correspondences”, *Proc. IAPR Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 2**, pp. 1110–1112.

[Miti 88] Mitiche A., November 1988, “Interpretation of Optical Flow Correspondence”, *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 2**, pp. 1097–1099.

[Mora 77] Moravec H. P., August 1977, “Towards Automatic Visual Obstacle Avoidance”, . *Proc. Fifth International Joint Conference on Artificial Intelligence*, Cambridge, Massaschussets, pp. 584.

[Murr 87] Murray D. W., Buxton B. F., March 1987, “Scene Segmentation from Visual Motion Using Global Optimization”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–9, no. 2**, pp. 220–228.

[Nage 83] Nagel H. H., 1983, “Displacement Vectors Derived from Second-Order Variations in Image Sequences”, *Computer Vision, Graphics, and Image Processing*, **no. 21**, pp. 85–117.

[Nage 84] Nagel H. H., July 1984, “Towards the Estimation of Displacement Vector Fields by "Oriented Smoothness" Constraints”, *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 1**, pp. 6–8.

[Nage 89] Nagel H. H., January 1989, "On a Constraint Equation for the Estimation of Displacement Rates in Image Sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 1**, pp. 13–30.

[Nega 89] Negahdaripour S., Lee S., Shokrollahi, December 1989, "Robust Motion Recovery: The Two-Plane Method", *Proc. Second International Conference on Intelligent Autonomous Systems*, Amsterdam, Netherlands, **Vol. 2**, pp. 619–629.

[Nels 89] Nelson R. C., Aloimonos J., October 1989, "Obstacle Avoidance Using Flow Divergence", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 10**, pp. 1102–1106.

[Niem 88] Niemann H., Arnold J., Sagerer G., November 1988, "On the Accuracy of Optical Flow Computation Using Global Optimization", *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 2**, pp. 1094–1096.

[Pan 86] Pan F., Gu W. K., Jing R. J., October 1986, "A Back Tracking Algorithm for Matching Two Line-Drawings of a 3-D Moving Object", *Proc. IAPR Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 2**, pp. 1096–1098.

[Peng 88] Peng S. L., Medioni G., November 1988, "Spatio-Temporal Analysis for Velocity Estimation of Contours in an Image Sequence with Occlusion", *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 1**, pp. 236–241.

[Prag 83] Prager J. M., Arbib M. A., 1983, "Computing the Optic Flow: The MATCH Algorithm and Prediction", *Computer Vision, Graphics, and Image Processing*, **no. 24**, pp. 271–304.

[Roac 80] Roach J. W., November 1980, "Determining the Movement of Objects from a Sequence of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–2, no. 6**, pp. 554–562.

[Rose 82] Rosenfeld A., June 1982, "Motion: Analysis of Time-Varying Imagery", *Fundamentals in Computer Vision*, Faugeras O. D. Editor, Cambridge Press, pp. 173–183.

[Rudi 73] Rudin W., 1973, *Functional Analysis*, McGraw-Hill Co., New York.

[Sabr 88] Sabri A. M., Mostafa S. A., Green R. J., November 1988, "Recognition and Velocity Computation of Large Moving Objects in Images", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **Vol. ASSP–36, no. 11**, pp. 1790–1791.

[Schä 89] Schäfer H., Murai S., September 1989, "Results Using a Machine Vision System for Automated Measurement of Object Motion", *Proc. Conference on Optical 3-D Measurement Techniques*, Vienna, Austria, pp. 281–290.

[Scha 84] Schalkoff R. J., Labuz J., July 1984, "An Integrated Spatio-Temporal Model and Recursive Algorithm for Image Motion Estimation", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 1**, pp. 530–533.

[Schu 84] Schunck B. G., July 1984, "The Motion Constraint Equation for Optical Flow", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 1**, pp. 20–22.

[Schu 89] Schunck B. G., October 1989, "Image Flow Segmentation and Estimation by Constraint Line Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 10**, pp. 1010–1027.

[Seth 87] Sethi I. K., Jain R., January 1987, "Finding Trajectories of Feature Points in a Monocular Image Sequence", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–**

9, no. 1, pp. 56–73.

[Shah 84] Shah M. A., Jain R., July 1984, "Detecting Time-Varying Corners", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 1**, pp. 2–5.

[Shar 90] Shariat H., Price K. E., May 1990, "Motion Estimation with More than Two Frames", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–12, no. 5**, pp. 417–434.

[Snyd 89] Snyder M. A., May 1989, "The Precision of 3-D Parameters in Correspondence-Based Techniques: The Case of Uniform Translational Movement in a Rigid Environment", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 523–528.

[Stui 88] Stuifbergen J. A. M., Heideman G. H. L. M., 1988, "Hierarchical Estimation of Spatio-Temporal Intensity Changes in Images", *Signal Processing IV: Theories and Applications*, Eurasip, pp. 247–250.

[Subb 89] Subbarao M., March 1989, "Interpretation of Image Flow: A Spatio-Temporal Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 3**, pp. 266–278.

[Thév 88a] Thévenaz P., Hügli H., March 1988a, "Optical Flow Estimation Using Line Image Sequences", *Proc. BPRA Fourth International Conference on Pattern Recognition*, Cambridge, U. K., pp. 468–477.

[Thév 88b] Thévenaz P., Hügli H., September 1988b, "Optimal Speed Measurement by Optical Flow Using Line Image Sequences", *Proc. EUSIPCO Fourth European Signal Processing Conference*, **Vol. 3**, Grenoble, France, pp. 1629–1632.

[Thom 81] Thompson W. B., Barnard S. T., August 1981, "Lower-Level Estimation and Interpretation of Visual Motion", *Computer*, pp. 20–28.

[Thom 84] Thompson W. B., Mutch K. M., Berzins V. A., July 1984, "Analyzing Object Motion Based on Optical Flow", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 2**, pp. 791–794.

[Tret 84] Tretiak O., Pastor L., July 1984, "Velocity Estimation from Image Sequences with Second Order Differential Operators", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 1**, pp. 16–19.

[Tsuj 86] Tsuji S., Yagi Y., Asada M., October 1986, "Finding of Objects Moving in a Pathway by a Moving Observer", *Proc. IAPR Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 2**, pp. 1103–1106.

[Tzir 89] Tziritas G., January 1989, "Recursive And/Or Iterative Estimation of the Two-Dimensional Velocity Field and Reconstruction of Three-Dimensional Motion", *Signal Processing*, **Vol. 16, no. 1**, pp. 53–72.

[Ullm 81] Ullman S., August 1981, "Analysis of Visual Motion by Biological and Computer Systems", *Computer*, pp. 57–69.

[Verr 89] Verri A., Poggio T., May 1989, "Motion Field and Optical Flow: Qualitative Properties", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 490–498.

[Waxm 86] Waxman A. M., Duncan J. H., November 1986, "Binocular Image Flows: Steps toward Stereo-Motion Fusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–8, no. 6**, pp. 715–729.

[Weng 86] Weng J., Ahuja N., Huang T. S., October 1986, "Motion Modelling and Prediction", *Proc. IAPR Eighth International Conference on Pattern Recognition*, Paris, France, **Vol. 2**, pp. 1107–1109.

[Weng 88] Weng J., Ahuja N., Huang T. S., November 1988, "Motion and Structure from Point Correspondences: A Robust Algorithm for Planar Case with Error Estimation", *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 1**, pp. 247–251.

[Weng 89] Weng J., Huang T. S., Ahuja N., May 1989, "Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 451–476.

[Werk 90] Werkhoven P., Toet A., Koenderink J. J., July 1990, "Displacement Estimates through Adaptive Affinities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–12, no. 7**, pp. 658–663.

[Wu 88] Wu L., Jian L., Faguan W., November 1988, "The Theorem Analysis on Optical Flow Estimation from Three Frames of Image Sequences", *Proc. IAPR Ninth International Conference on Pattern Recognition*, Rome, Italy, **Vol. 2**, pp. 1103–1105.

[Yama 89] Yamamoto M., May 1989, "A General Aperture Problem for Direct Estimation of 3-D Motion Parameters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI–11, no. 5**, pp. 528–536.

[Yen 84a] Yen B. L., Huang T. S., July 1984a, "Determining 3D Motion/Structure of a Rigid Body over 2 Frames Using Correspondences of Straight Lines Lying on Parallel Planes", *Proc. IAPR Seventh International Conference on Pattern Recognition*, Montreal, Canada, **Vol. 2**, pp. 781–783.

[Yen 84b] Yen B. L., Huang T. S., March 1984b, "Solving for 3D Motion Parameters of a Rigid Body by a Geometrical Approach: Uniqueness and Numerical Results", *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, San Diego, California, **Vol. 2**, pp. 23.5.1–23.5.4.