Motion of disturbances: detection and tracking of multi-body non rigid motion*

Gilad Halevy and Daphna Weinshall†
Institute of Computer Science, The Hebrew University of Jerusalem
91904 Jerusalem, Israel; Email: daphna@cs.huji.ac.il

Abstract

We present a new approach to the tracking of very non rigid patterns of motion, such as water flowing down a stream. The algorithm is based on a "disturbance map," which is obtained by linearly subtracting the temporal average of the previous frames from the new frame. Every local motion creates one disturbance having the form of a wave, with a "head" at the present position of the object and a historical "tail" that indicates the previous positions of that object. These disturbances serve as loci of attraction for "tracking particles" that are scattered throughout the image. The shape of the disturbance wave does not depend on the object's shape; by tracking along these waves good separation is achieved between different objects, and very stable tracking is obtained even when there are many complex trajectories.

The method enables us to track occluded objects, as well as to simultaneously contend with a large number of independently moving objects. Our method does not employ any restrictions at all regarding the magnitude of the changes that can occur in the shape of the objects or restrictions on the magnitude of their motion. It does not require smooth trajectories, and it does not rely on stability of the velocity. The algorithm is very fast and can be performed in real time. We provide excellent tracking results on various complex sequences, using both stabilized and moving cameras, showing: a busy ant column, waterfalls, rapids and flowing streams, shoppers in a mall, and cars in a traffic intersection.

1 Introduction

The tracking of motion in computer vision can be divided into two subtopics: the motion of rigid bodies and the motion of nonrigid bodies. In the latter more complicated case it is assumed for the most part that the changes in the shape of the object are relatively slow and that it is therefore possible to compare local features; examples are edges, contours, extreme points, and principal axes, that is - anything that can provide information on the shape of the object and which remains relatively stable as the object changes. However, the shape constancy assumption is not always valid, and there are cases in which large changes occur in the shape of objects from frame to frame. An example of this is the case of water fbwing down a rushing stream. In addition, it is not always possible to extract local features since this process requires successful object segmentation, which cannot always be accomplished reliably (again, consider fbwing water or camoufhged objects that cannot be discerned while motionless).

Is local shape information the only information that can assist tracking? Introspection tells us that it is far easier to identify objects in motion than stationary objects. In fact, the identification of motion may precede the identification of shape, as in the case of camoufhged objects. Moreover, we tend to identify motion even in cases in which there are no moving objects. Take, for example, an advertising sign around which lights of different colors flush on and off. We observe circular motion of the lights around the sign, although in reality there is nothing actually moving. All this suggests that perhaps the sequence of changes itself (or disturbances) is perceived by us as motion.

*Vision research at the Hebrew University is partially supported by DARPA through ARL Contract DAAL01-97-0101.
†To whom correspondence should be addressed.

This observation leads us to the following idea: the spatial information in a picture may be used as global information that helps stabilizing large regions in an image, whereas motion is identified by means of temporal changes (disturbances) that are detected within the stabilized region. These disturbances are tracked without regard to the spatial shape of the object being tracked.

Having defined the problem in this way, our solution is the following: we compare the present frame with a "background" image obtained by the averaging of previous frames, thus obtaining an effect similar to the effect obtained by a photographer who tries to capture motion by exposing film for a long time: in the over-exposed picture every moving object creates a smeared image in the direction in which it moves. When the previous position of the object is known, there is no difficulty in attaining the new position by moving along the trail of the smeared image. But even if the object changes its direction or shape as it advances, the trail still leads to it. In fact, a sort of smooth energy field, in which every moving object causes disturbances along its trajectory, can be seen in the smeared picture. On the basis of this observation we can introduce "tracking particles" into the field, which will cling to the disturbances thus created and will thereby be attached to the object being tracked. Since the influence of this attraction is felt only along the trajectory along which the object moves, we obtain highly stable tracking even when different objects pass in close proximity to each other.

The algorithm basically does not make any assumptions regarding the smoothness of the motion of the objects, the ability to distinguish between objects and the background, or a restriction on the magnitude of the changes that can occur in an object upon passage from frame to frame. Collisions between objects, however, require special treatment. In addition, there is a certain constraint on the velocity of the objects, and the assumption is that the motion of an object from frame to frame does not greatly exceed the dimensions of the object. This assumption nearly always holds, and it is far weaker than the restrictions imposed by most of the existing algorithms in regard to the velocity of objects between consecutive frames. Thus highly stable tracking is obtained in the following difficult situations, when:

1. the simultaneous tracking of a large number of objects is required;
2. it is difficult to distinguish between the objects and the background (camouflage);
3. the objects undergo complex, non-rigid and varying motion;
4. the shape of the objects varies fast relative to the frame rate.

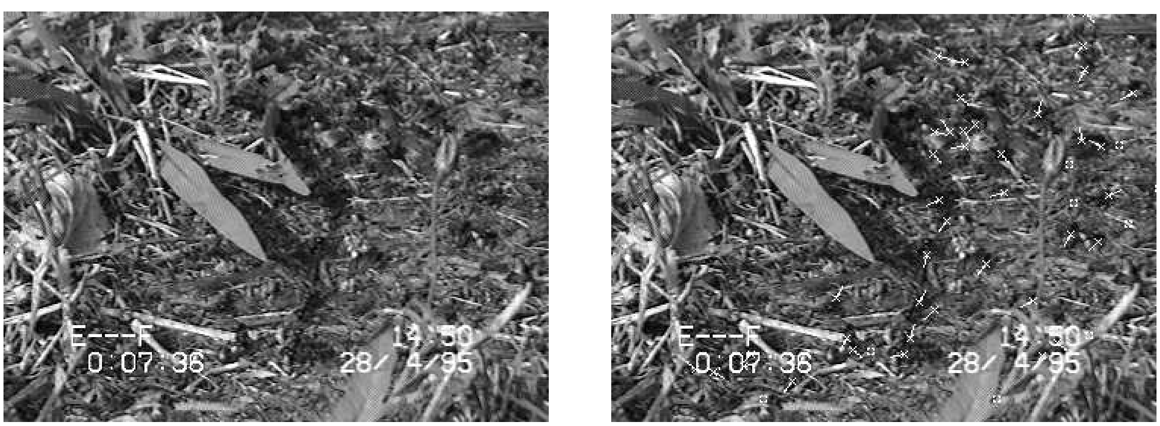


Figure 1: The ants sequence. Left: The original image, where it is very difficult to detect the ants. Right: After detection, the ants are marked and the lines indicate the direction of motion.

We shall illustrate the algorithm in the difficult examples of an ant column, water flowing down streams and waterfalls, shoppers in a mall, and cars at a traffic intersection. Fig. 1 presents an example of an ant column. In a single frame it is very difficult to identify the ants, because their color is very similar to that of the background. On the other hand, when we look at the motion picture, there is no difficulty in identifying the ants. Our algorithm, too, which is based on temporal changes, can easily identify the position of the ants and their individual directions of motion (right picture in Fig. 1).

The rest of this paper is organized as follows: after literature review in Section 2, Section 3 describes how the "disturbance field"" is calculated, how changes in the motion of an object or its shape influences the field, and which are the parameters that can influence the computation. Section 4 discusses the tracking of objects using "tracking particles," which are attracted to the regions where the moving objects create disturbances. At the end of the section we present results obtained using movies taken with a stabilized camera, displaying an ant column, flowing waterfalls and rapids, shoppers in a mall, and cars crossing a traffic intersection. Section 5 presents an extension of the algorithm to the case of a non-stabilized camera using the stabilization algorithm described in [21]. In Sec. 6 a comparison with other methods is described.

2 Review of literature

The identification of motion and the automatic tracking of moving objects constitute one of the most actively studied subjects in computer vision. Until recently research has focused mainly on tracking rigid bodies, which do not deform as they move. In recent years the study of the motion of nonrigid bodies attracts more attention; this is motivated in part by applications such as medical imaging: NMR and ultrasonic images, or the need in automatic tracking of moving objects such as pedestrians.

The problem of automatically tracking nonrigid motion is far more difficult than the former problem due to the large number of degrees of freedom in it. Thus many algorithms in this area are supplemented by an additional assumption, according to which the deformation of each object between consecutive frames is relatively small. On the basis of this assumption, these algorithms search for local features that characterize the shape of the object and permit a relatively easy comparison of consecutive frames.

One general approach to the study of nonrigid motion relies on the contours of the object, typically using active contours or "snakes" [1] (see also [2, 5, 4, 3]). The fundamental assumptions in these algorithms are: (1) a primary contour similar to the desired contour is supplied externally; (2) there are reliable regions from which contours can be derived; (3) the shape and position of the object do not change much from frame to frame.

These assumptions do not hold in many cases. An attempt to extend the approach to cases of large deformation can be found in [7]. The idea here was to replace the constraint of small changes by a physical or geometric model of the phenomenon that we wish to study. Under this approach, too, there are still several problems: (1) it is necessary to identify occluding contours that identify the object, which is not always possible; (2) the use of a model greatly restricts the generality of the algorithm for treating diverse problems; (3) in many cases it is very difficult to conceive of a physical or geometric model that can describe the motion (is there a model for the overall motion of an ant column?).

Our objective is to design a general tracking algorithm and thus we chose not to limit ourselves to a particular physical model. We shall see later that it is possible to achieve an abstract general model that can characterize any kind of motion.

A different approach, which was described in [8], offers a solution to the opposite problem, in which the motion of the object can indeed be great, but the deformation of the object between consecutive frames must be small. Application of the algorithm to real images requires special enhancements, including:

1. Erasing the static background regions (an assumption that the camera is static).
2. Finding "lost" objects: when adequate correspondence is not achieved between the model and the present frame, the correspondence of previous models is tested (this requires their storage and, of course, also increases the computation time).
3. When there are several objects of similar shape in the picture, errors in identification can occur; therefore comparisons are made between different alternatives to determine which of them is more suitable from the standpoint of motion smoothness (in violation of the fundamental assumption that there are no restrictions on

the motion). Overall, the system is fundamentally capable of simultaneously handling a very small number of models.

Even this principled approach would not do in cases such as water flowing down a river, where there are large changes between consecutive frames, or the motion in an ant column which includes many identical objects moving in different directions.

Exploring other applications, many attempts have been made to reliably track pedestrians. These efforts span a range of techniques, from the use of contours, through the tracking of representative points in joints, and on to the use of complex models of a person walking. For example, [6] used a cubic B-spline representation for the contours of a moving image (these contours are obtained by subtracting pictures of the background that were photographed in advance). Other studies attempt to find correspondences and to reconstruct trajectories between groups of points, e.g., [9, 10, 12, 13].

Since algorithms of this kind do not obtain all the information on the shape, color, boundary lines, and the like, and since the only information is the position of the objects, which is provided in the form of an irregular group of points, these algorithms have no choice but to rely on preliminary assumptions regarding the smoothness of the trajectories, the velocity, constant or slow motion, etc. In these cases it helps to use a prediction mechanism, such as the Kalman-Filter technique, that can reduce the number of possibilities on the basis of the information that has been accumulated. Nevertheless, the situation in nature is not so ideal, and there are many cases in which the objects do not maintain continuous motion. We shall see an example of this below for a column of ants that change their trajectories and velocities almost randomly in their attempts to overcome various obstacles.

Despite the drawbacks of the approach, the basic principle that it is sufficient for us to track a reduced number of points in order to identify motion is a sound principle, and it is a central foundation of our work. However, in contrast to the foregoing approaches, the points, as well as the tracking trajectories, are found with preservation of the contextual information, which shows the new positions of the objects and the direction in which they move (this will be detailed below).

Other studies that deal with tracking the motion of pedestrians employ various models of people. The models can be simple, as in the case of an abstract skeleton which can be derived from the silhouette of a person walking, or more detailed models as in [14]. These approaches have an advantage in a real analysis of walking, but they are excessively complicated and do not provide a solution for the simple need of reliable and persistent tracking in the presence of partial occlusion, changes in the direction of motion, etc.

An especially interesting study in this area is described in [15]. In this work the motion of numerous people crossing the streets at an intersection was tracked simultaneously. This approach seems highly stable, and it also contends well with instances of partial occlusion. On the other hand, a basic assumption underlying the algorithm is that the basic motion of pedestrians is stable and that they do not change the direction in which they move. In addition, there is an assumption that the camera is static.

A subject that seems totally different from tracking, but, as we shall see below, is closely linked to this work, is the issue of figure/ground segmentation using temporal averaging (when the camera is static or in motion). Early studies in this area were published already at the end of the seventies, e.g., [16, 17]. In this work the fundamental assumption was that the camera is static and that the moving objects do not change as they move and do not change the direction in which they move.

An extension of the method just described on the basis of integration of pictures of the differences along a number of frames using linear subtraction is described in [19]. For an object that is brighter than the background we obtain a positive difference at the object's new position, while negative differences are measured in all the previous regions where it was located; therefore, when preliminary information on the relative shading of the moving object is given, it is possible to isolate its new position. This idea is precisely the idea which we utilize in the present work, but in a different way: the purpose of the work just cited was to isolate the moving object from the background, while in our work the purpose is to track it. Actually, here we utilize basic information that was obtained during the process and was not utilized in the preceding algorithm: the region of negative differences left behind by the object shows the direction in which the object moves and its velocity. Moreover, this information remains reliable even if the object changes its shape and shading as it moves. We shall see below just how the same kind of computation can lead us to actual stable tracking and not just object segmentation.

How can these measurements be extended to cases in which the camera moves? In [21] the basic idea was to stabilize the background by assuming that this motion is the dominant motion in the area examined. In this work it is also shown how the motion of a single object relative to the background can be isolated in this way. In [20] this idea was extended to the isolation of several different motions from the background. An additional extension can be found in [22], which involved finding more complex parameters than displacement. Also, emphasis was placed on the importance of integration of the computation over a number of frames (a large temporal region). Using these methods it is possible to isolate each moving object in a picture without preliminary segmentation and without a requirement that each object maintains constant velocity. The problem is that this algorithm requires objects to be rigid and to undergo only $2D$ transformations.

Our work is based on the ideas presented above for the purpose of stabilizing the background picture or selected regions within it. Afterwards we utilize the principle of linear subtraction of the stabilized background picture from the present frame in order to obtain regions of positive and negative variation. Note that although the process of stabilizing the sampled region is based on spatial information and the assumption that most of the region is rigid, afterwards the computation is based on temporal information alone. This computation does not rely at all on local features and particularly not on contours, segmentation into areas, or any other local information that can be measured in a single frame. This enables us also to track objects for which it is very difficult to define local features, such as flowing water or the motion of camouflaged objects with shading similar to that of the background. Also, the algorithm does not make any assumption regarding the nature of the trajectories, in contrast to many of the algorithms described above; therefore, the motion of the objects need not be smooth or stable. As a result of all this, we obtain highly stable tracking even when the objects vary their velocity and the direction of motion, as well as when they change their shading or shape as they move.

3 Computation of the Disturbance Field

We want to treat cases in which the objects to be tracked undergo significant changes in shape upon transition from frame to frame, e.g., water flowing down a waterfall. Our claim is that in these cases we can still observe a dynamic sequence of abrupt changes in the picture, which successively appear and vanish. We call these changes "disturbances." The basic idea is that once such a disturbance is exposed, there is no longer any importance to its shape or its intensity, and the only thing of interest is its position and whether this position forms a continuous dynamical sequence with previous disturbances.

In defining a disturbance we do not want to rely on shape-related spatial information, such as occluding contours of objects, but only on temporal information stemming from changes between frames. This forces us to assume that the examined region is stabilized. This is similar to what happens in the human eye, where at every moment our glance is stabilized on a different region and it thereby becomes possible to quickly discern objects moving relative to the stabilized region. For simplicity, we shall begin with cases in which the camera is stabilized, and thus complete frames can be compared.

In this section we shall define a disturbance by means of temporal changes between the present frame and previous frames, and we shall obtain a disturbance field, in which every moving object creates a disturbance. The disturbance links the previous positions of the object to its new position. The basic structure of the disturbance does not depend on the changes in shading or shape that the object undergoes, but only on its motion.

3.1 Definition of a disturbance

Disturbance: an abrupt change in grey-levels that appears (and disappears) in a certain region and at a certain time.

The computation of a disturbance must be temporal and must be based only on a comparison between the present shading of a certain region and the mean past shading in that region. The use of averaging over a number of frames over time imparts stability to the system and enables blurring of the noise that does not stem from actual motion. The problem with ordinary averaging is that it is not sensitive to relatively slow changes that occur in the picture. By definition, a disturbance is abrupt; therefore, we expect the system to be able to adjust to slow changes and not to identify them as disturbances. For this reason, we preferred to use temporal averaging, which gives more weight to the last frames and thereby adjusts to slow changes (see Sec. 3.3).

3.2 Using linear subtraction

In many algorithms (e.g., [26]) the difference between consecutive frames is found as a first step when it is known that the camera has been stabilized, and then the result is raised to the second power or its absolute value is taken. This operation immediately emphasizes the regions of motion in a movie. For example, this information is used in [26] to quickly focus on the interesting regions in the movie without wasting efforts calculating static regions. This method suffers from several drawbacks, including: (1) there is symmetry between the previous position of the object and its new position; (2) There is no smooth trajectory connecting the peaks and pointing to a connection between them. The use of linear subtraction, on the other hand, emphasizes the fact that the site reached by the object undergoes a change in shading that is the opposite to the change occurring at the site abandoned by the object. The difference between linear and non-linear subtraction is illustrated in Fig. 2.

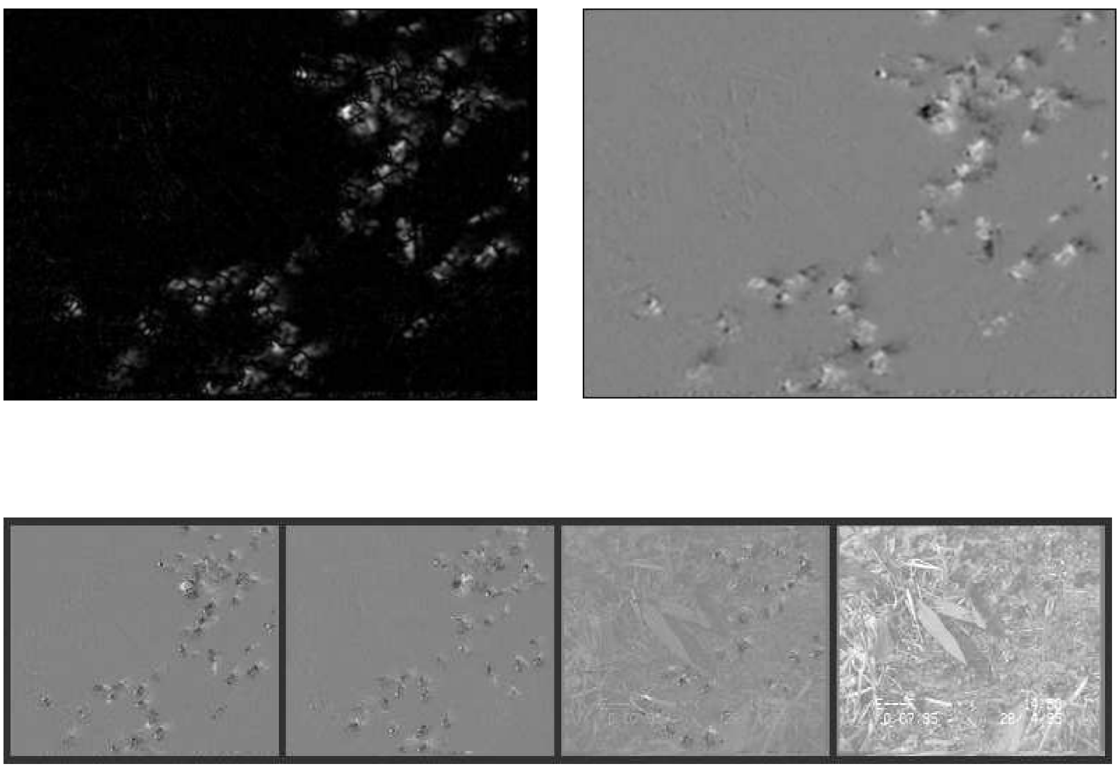


Figure 2: Ants sequence, difference results. *Top left:* non-linear difference: note the symmetry between the new and past locations, and the dark lines which separate them. *Top right:* linear difference: new locations are bright, while past locations are dark. Following the smooth path between them, we can easily identify motion direction. *Bottom:* the process of segmenting the ants from the stationary background, using temporal averaging, may require a few frames. In this example we used a high history weight factor $w = 0.7$. The frames shown, from left to right, are respectively: 10, 6, 3, 1.

The use of linear subtraction creates a new problem that did not previously exist. We find that an object which is dark relative to the background creates a disturbance pattern as it moves that is opposite to the pattern produced by an object brighter than the background. Thus, if we do not know the relative shading of the object, we still have the same problem of not being able to distinguish between the former position and the new position. On the other hand, if we know that the object is brighter than the background, then there is no problem: the maximum of the disturbance indicates the new position of the object (henceforth to be called the ”head” of the disturbance), while the minimum of the disturbance indicates the previous position of the object (to be called the ”tail” of the disturbance).

In most cases the relative shading of the object is maintained for a long period of time, so that the shading can be determined once at the beginning of the tracking , and thereafter it is only necessary to stick with that determination. There are many ways to determined the relative shading of an object. One possibility is to compare the disturbance with the region surrounding it (the background) in the original frame. Another approach is to compare the intensity of the disturbance at the head and the tail.

3.3 Using a history of multiple frames

Comparing only consecutive frames has several drawbacks:

1. The shape of the disturbance and its intensity are highly dependent on the velocity of the object.
2. There is great sensitivity to noise arising from the instability of the camera or changes in lighting.
3. The linkage between earlier positions of the object and its present position can be easily broken.

Excellent isolation of a moving object could be achieved, if we were given a background picture in advance. In this case ordinary subtraction of the background picture from the present frame would yield a result that would greatly diminish the problems raised above. Something close to the background picture could be obtained by averaging all the frames, which would require scanning the movie twice. However, in order to attain a real-time algorithm, at each stage we compare between the present frame and the average of the preceding frames. The computation becomes:

$$\begin{aligned} \text{new average} &= \frac{1}{n}(\text{new frame}) + \frac{\text{n}-1}{\text{n}}(\text{previous average}) \\ \text{disturbance map} &= (\text{new frame}) - (\text{previous average}) \end{aligned}$$

where n denotes the frame number.

There is a problem, though, in giving “old” frames similar weight as more recent frames in the computation of the background. We therefore employ temporal averaging, which gives greater weight to the last frames. The computation of the disturbance map in the final algorithm has the following form:

$$\begin{aligned} A_t &= (1-w)I_t + wA_{t-1} \\ \Delta_t &= I_t - A_{t-1} \end{aligned} \tag{1}$$

where A denotes the temporal average image, I the actual image after initial smoothing, Δ the disturbance field, and $0 < w < 1$ is a history weight factor.

We note that the average decays exponentially relative to the number of the frame (as w^n). Previous to the computation the new frame is smoothed, and after the computation the updated disturbance map obtained is smoothed.

The bottom row of Fig. 2 shows the disturbance field obtained for the motion of ants when $w = 0.5$. It can be seen that the disturbance is characterized by a bright head accompanied by a dark tail, which gradually disappears. Also, it can be seen that a period of time passes before the background stabilizes and disappears.

The various scene variables have the following effect on the algorithm:

Velocity: The velocity at which the object moves has a decisive influence on the intensity of the disturbance, despite the fact that the use of averaging diminishes this influence. For a static object we do not obtain any disturbance. When the motion of the object is very slow and there is great overlap between frames, the intensity of the disturbance is small. The intensity increases as the velocity of the object increases and the degree of overlap decreases. Once the velocity of the object is very high, there is no longer any overlap between consecutive frames, and the velocity of the object is no longer of any importance.

Object shape: The fact that a certain object changes its brightness and shape as it moves does not have a qualitative influence on the shape of the disturbance. In other words, the structure of the wave leading from the previous position of the object to its new position is maintained regardless of the shape of the tracked object.

Camouflaged objects: As we pointed out above, the fact that the computation is based only on temporal changes and not on spatial information enables us to detect camouflaged objects: objects whose texture is very similar to the texture of the background. This was already demonstrated on the example of the ant column, which, as demonstrated, could not be discerned by looking at a single frame from the original movie.

3.4 Summary: features of the disturbance field

We have hitherto dealt with the features of a single disturbance; however, when there are many moving objects, a complete field of disturbances is obtained. In this smooth field every disturbance acts as a locus of attraction, which attracts "tracking particles" found in the history of that disturbance. The field is obtained by linear subtraction of the temporal average of the previous frames from the last frame in the following manner:

1. The new temporal average is computed according to Eq. (1), after the new frame has been smoothed.
2. The disturbance field is computed using linear subtraction of the previous temporal average from the new frame, followed by smoothing of the disturbance field.

The only free parameter in this computation is the history factor w, which specifies the weight given to the previous frames.

The disturbance field is characterized by the following features:

1. In regions where there is no motion the field equals 0.
2. Every moving object creates a disturbance in the field in the form of a wave, which includes: one extremum at the present position of the object - to be called the "head" of the disturbance, and a "tail" having the opposite sign of the head - indicating the previous positions of the object.
3. There is a smooth monotonic path between the extremum at the tail of the disturbance to the extremum at its head.
4. The direction of the gradient at the point defining the previous position of the object points toward the head of the disturbance, which marks the new position of the object (the slope is typically greatest along the path connecting the two extrema).
5. The absolute value of the intensity of the disturbance is greater at the head of the disturbance than at the tail (the latter value at the tail of the disturbance is multiplied by $w < 1$).
6. The basic shape of the disturbance and the positions of the extrema are not influenced by changes in the shape of the object or its velocity.
7. The higher is the velocity of the object, the greater is the intensity of the disturbance (as long as there is overlap between consecutive frames).

4 Tracking Disturbances

In the previous section we defined a disturbance field and showed how every moving object in an image creates a disturbance that includes a "head" and a historical "tail". In this section we shall define "tracking particles," which are attracted to these disturbances. The form of every disturbance is such that it attracts only particles found in its tail, i.e., in its previous positions, and does not attract other particles even if they happen to be closer. In this way excellent separation between different trajectories is achieved, and highly stable tracking is obtained.

A disturbance head can appear as a local maximum or minimum, depending on the shading of the object relative to the background. A tracking particle must set the relative shading of the object that it tracks at the beginning of the tracking and maintain the value set during the tracking.

In most cases the disturbance field makes it possible to steadily and clearly distinguish between the trajectories of different objects. Nevertheless, there are cases in which incorrect tracking trajectories appear. Later on in this section we shall see how these cases can be identified and corrected. We will demonstrate tracking on movies of an ant column, waterfalls, wondering shoppers and fast moving cars.

4.1 Disturbances as loci of attraction for particles

In order to track disturbances, we shall utilize data structures called "tracking particles" or simply "particles". Every particle contains the following information:

1. Location in the picture
2. Tracking state (inactive, tracking, holding), and how much time it has been in the last state.
3. Is the object being tracked brighter or darker than the particle.
4. History of the objects' motion (e.g., previous positions, velocities).

The third point, describing the relative shading of the object, is a feature which is maintained for a long period of time; therefore, it is sufficient to determine it at the beginning of the tracking. In Sec. 3.2 we discussed several ways to do this. In the next section, which discusses the initialization of particles, we shall detail the approach taken here.

After we set the relative shading of the object and placed the particle on it during initialization, we move on to the tracking stage based on the disturbance field. W.l.o.g. we shall henceforth assume that the shading of the object is bright relative to the background. In this case there is a negative change at the previous position of the object, and the tracking particle will find itself at the minimum at the tail of the new disturbance. The maximum at the head of the disturbance marks the new position of the object and is the location to which the particle must be attracted.

Fig. 3 gives a three-dimensional illustration of the disturbance field, shown in the top right picture in Fig. 2. It is easy to see from the figure that the steepest slope is achieved upon passage between the minimum, where the particle is located, to the maximum, which marks the new position of the object. Therefore, it would have been enough, in principle, to compute the gradient at the point where the particle is located and then to follow the direction of the gradient until the maximum is found. We preferred not to rely on a single-point computation of the gradient and, instead, to utilize the fact that there is always a smooth monotonic path between the two extrema. Therefore, all we have to do is to move along this monotonic trajectory.

In order to identify cases in which the object disappears, for example, as a result of occlusion, a very low threshold level is set so that if the extremum found is smaller in absolute value than this level, it is assumed that tracking is lost. In this case the particle switches to a holding state and remains at the last point where the object was detected. During the wait period, the particle continues to search for the object, as we shall see below. If renewed detection is not achieved within a certain time period, the particle goes into the inactive state and searches for a new object to track.

4.2 Initialization and revision of tracking particles

The particle initialization stage is designed to find one of the following:

1. a group of "good" objects to track, and the type of object (bright or dark);
2. new objects to track.

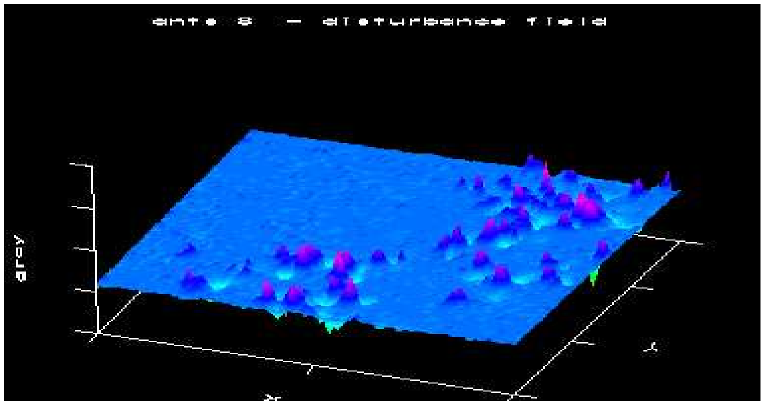


Figure 3: $3D$ visualization of the disturbance field from Fig. 2; note the smooth path between tails and heads.

In order to determine when a disturbance appears or disappears and in order to be able to distinguish between a disturbance and background noise, we must place a lower bound on the absolute value of a disturbance. This threshold can be very small, and it does not have any qualitative influence on the identification of disturbances. Nevertheless, the number of objects that can be tracked simultaneously is restricted by the number of free particles. Thus, if we have n inactive particles that are not in the tracking state, we can assign to them the n best (i.e., those with the highest absolute value) disturbances. In addition, we must determine the position of the head of the disturbance for each disturbance (or, equivalently, the relative shading of the object).

The entire process can be described as follows:

- First the system is allowed to stabilize for time t, so that the average will faithfully reflect the background. t is determined by the history factor w and Eq. (1): it is given a value such that $w^n < 0.1$. For example, $w = 0.5 \Rightarrow t = 4$.

- Next the grid of the disturbance pattern is scanned, and the absolute value of the intensity of each disturbance is measured in the surroundings of the grid points. The grid points are sorted in a list in decreasing order of intensity.

- We assign free particles to grid points in the list by scanning the grid points, starting from the highest value. An assigned particle is then allowed to be drawn to the optimal neighboring point. In other words, if the amplitude of the disturbance measured in the surroundings of the point in the grid is positive, a maximum is sought; otherwise, a minimum is sought.

- Each particle is now located either at the head of the disturbance or at the extremum at the tail of the disturbance. To determine the location of the head of the disturbance or the relative shading of the object, we utilize the fact that the intensity of a disturbance at the head is always greater than the intensity measured at the tail. (This is because the tail of the disturbance underwent averaging and, therefore, the extremum in it was multiplied by $w < 1$). Thus, we continue the search: if we are at a maximum we search for the minimum and vice versa. We compare the absolute values of the maximum and the minimum and select the larger of the two as the head of the disturbance. If the chosen head is the maximum – the object is bright, and if it is the minimum – the object is dark.

- Since the process just described is also intended to revise the list of particles and not just to initialize them, the disturbance that we found may already be "covered" by another particle (this can also result from the fact that the disturbance found is a source of attraction for several points in the grid); therefore we must compare the position that we found with the positions of the other active particles already in the tracking state. (Since the disturbances are smooth and include a single extremum, it is sufficient.)

Should the disturbance that we discovered be "vacant," we position the particle there and switch it to the tracking state. Also, we store the type of disturbance that we found in it. Conversely, if the disturbance is "occupied" by another particle, we move the particle to the next point in the list of grid points and repeat the process.

- We continue until all the particles that were in the inactive state are used up or until the intensity of the disturbance that we have reached is less than a pre-defined threshold.

The above process can be actually performed very fast, since we do not examine the entire frame but only a certain number of selected points. In addition, we utilize the smoothness of the disturbance field in order to ensure that we do not remain with an uncovered disturbance. The above process repeats itself in each frame after the positions of the active particles are revised. In this way it is possible to discover new disturbances that were not identified previously.

4.3 Summary of the algorithm

1. Read a new frame and revise the disturbance map according to Eq. (1).
2. Move particles in the tracking state toward the heads of the disturbances (as described in Sec. 4.1).
3. Assign inactive particles to new disturbances that are not yet covered (as described in Sec. 4.2).
4. (Optional) Correct the tracking trajectories (see Sec. 4.5).

The parameters in the algorithm are:

1. The threshold level for identifying a new disturbance.
2. The threshold level for identifying the disappearance of a disturbance (less or equal to the former level).
3. The number of tracking particles.
4. The history factor w.

In all the experiments to be described, the same pre-defined threshold levels and parameter values were used; the parameters were not tuned to accomplish better performance per each example.

4.4 Results

Fig. 4 shows the result on the trajectories of ants: in the top picture, each x indicates the position of a particle, corresponding to an individual ant; the next 2 pictures show the trajectories of the disturbances (ants) after 10 frames, and after 30 frames. Apart from the fact that the shading of the ants is very similar to the background and it is very difficult to discern them when they are not in motion, we are dealing here with a very complex scene: the ants are avoiding obstacles, climbing on branches, or passing beneath leaves, being partially or fully occluded. They bump into one another and intermittently change their angle and shape. Despite all this, stable tracking over a long period of time is attained, and it appears that the trajectories are mostly accurate except for isolated points.

Fig. 5 shows the results that were obtained for various samples of waterfalls and water flowing down a stream. In these cases the changes from frame to frame are enormous. Nevertheless, a good sense of tracking of the flow is attained, and a person viewing the movie presenting the motion of the particles has the impression that the particles are swept along with the water flow.

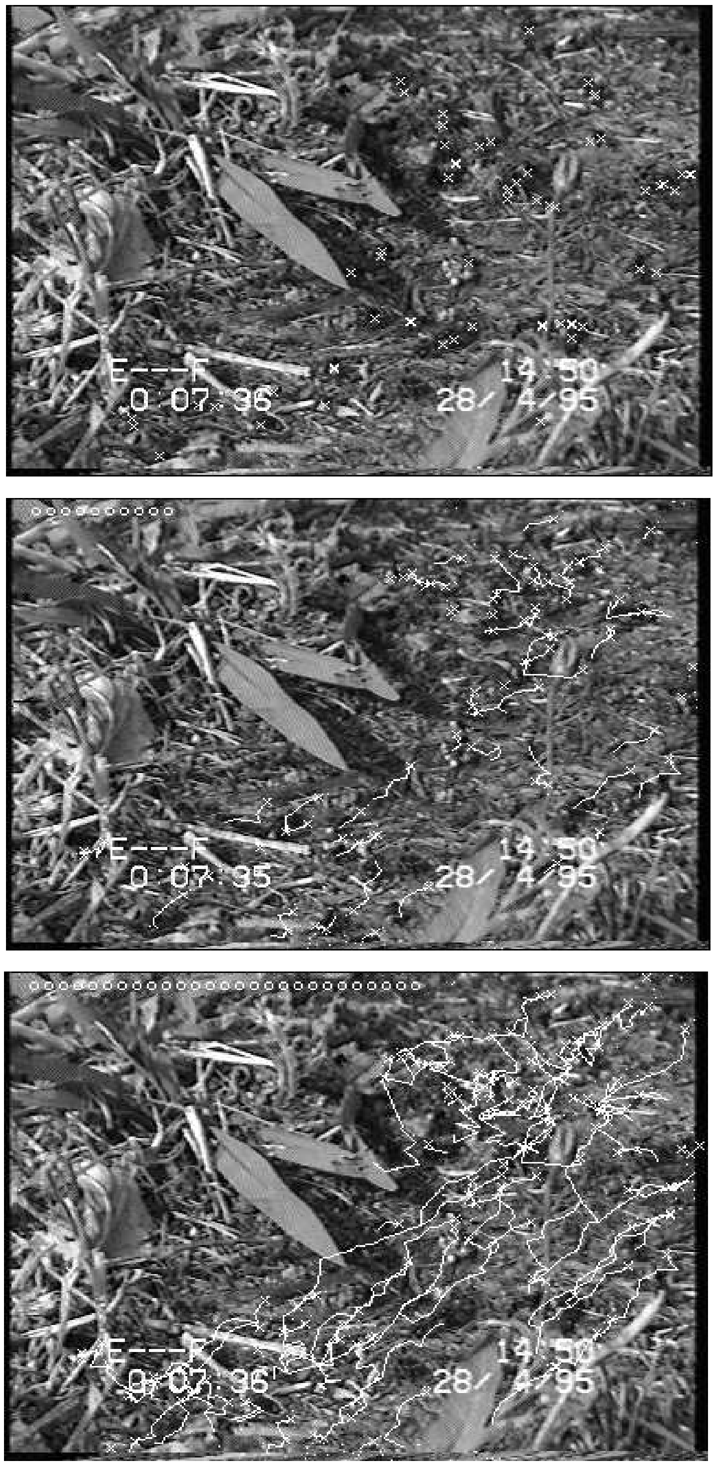


Figure 4: Results with the ant sequence. *Top:* ants detection: each disturbance attracts one particle. *Bottom:* the trajectories of the ants after 10 and 30 frames; note that the trajectories are smooth despite the complexity of the motion.

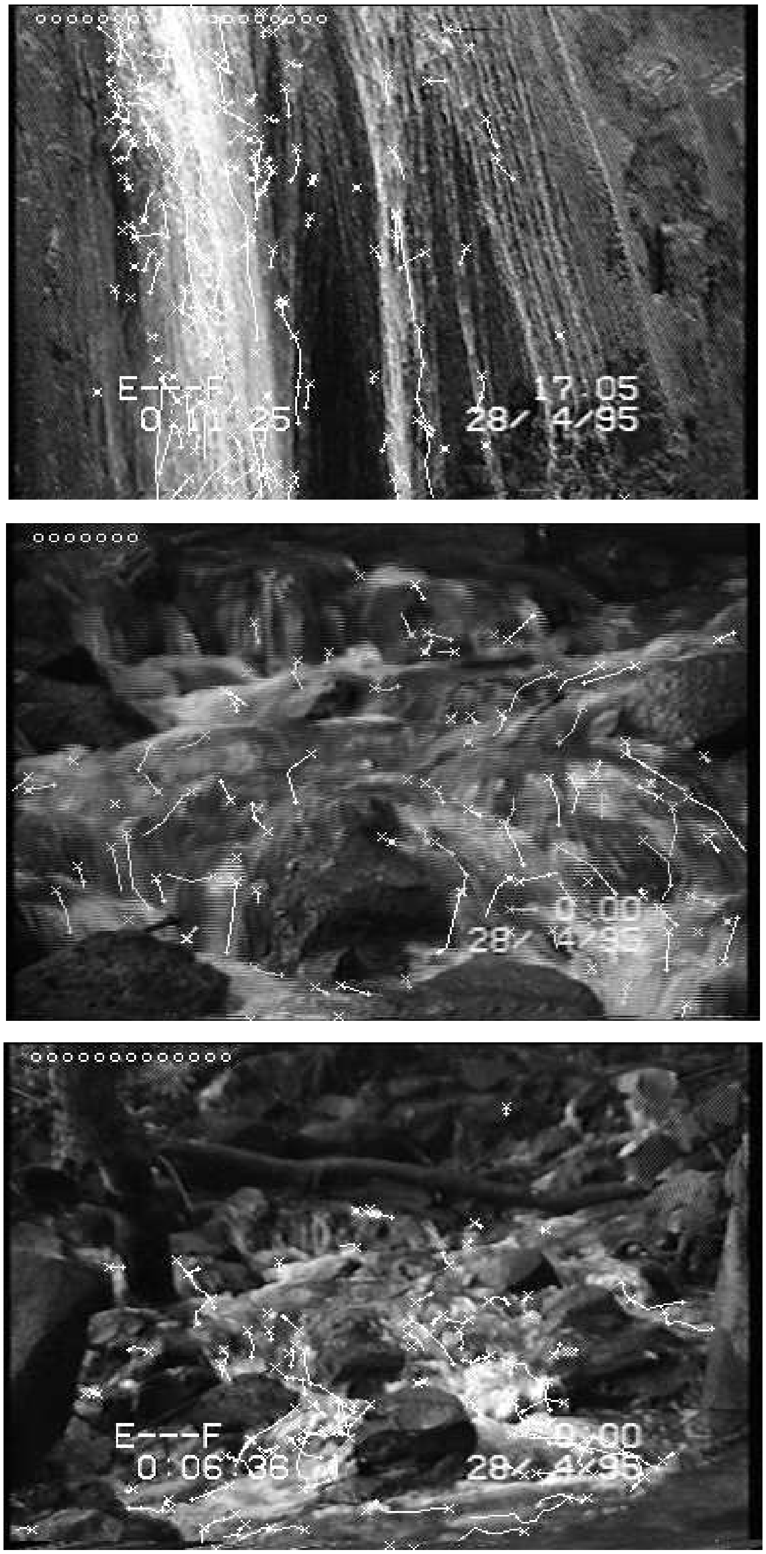


Figure 5: Sequences of waterfalls and river fbws: *Top:* the computed trajectories super-imposed on one frame from the waterfall sequence. *Middle:* the computed trajectories super-imposed on one frame from a complex rapids sequence; the trajectories follow the fbw of the water correctly (when looking at a movie of the particles, they appear to be carried downstream by the water). *Bottom:* similarly for another rapids sequence.

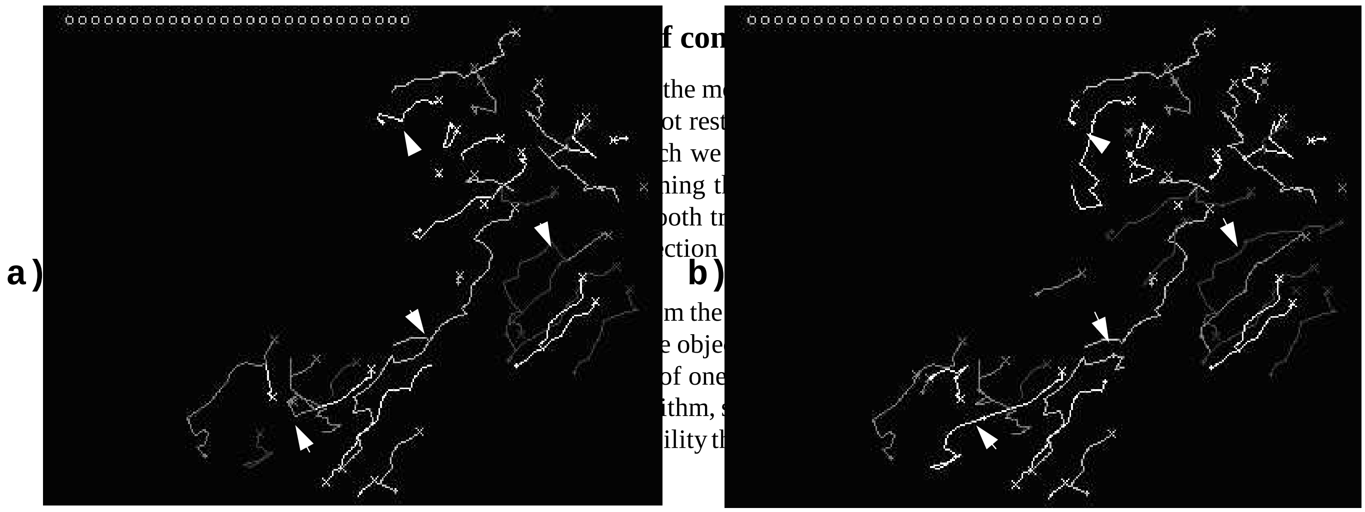


Figure 6: A few examples from the trajectories of the ants sequence, where arrows indicate probable errors, before (left) and after (right) correction.

In other cases, such as waterfalls, there are generally not enough particles to cover every meaningful disturbance, and the criterion for an incorrect trajectory must be defined in terms of threshold values: for example, the maximum distance that an object can traverse or the maximum angle between the previous direction of motion of the object and its last direction of motion (smooth trajectories). Fig. 7 presents an example from the waterfall sequence. It can be seen that the trajectories are not always smooth and that there are some sites of twisting and loops that are not consistent with the real motion of the current.

To handle these situations, in every case of a collision or every case in which the smoothness or speed threshold is exceeded, the last particle is sent back and switched to the waiting state. When collision is identified, the assumption is that when the objects continue to move away from one another, we will again be able to distinguish between the disturbances and to correctly renew the tracking. The problem is that while in the previous cases the particle was always at the minimum (or the maximum) of the tail of the disturbance, after a waiting period it will be pushed further

Figure 7: Results of the rapids sequence before smoothing (left) and after smoothing (right).

back in the tail. Therefore we must refine the search for the head of the disturbance by the particle: if a particle searches for a positive head, it must first move to the minimum in the tail and only then continue toward the maximum, and vice versa. (Of course, this refinement does no harm under "normal" circumstances where the particle is already located at the minimum.)

Fig. 6 shows the trajectories of the ants before and after the algorithm was enhanced in the manner described above. The trajectories were corrected at the problematic points, and they appear smoother and conform to the actual motion of the ants. Similarly, Fig. 7 shows the waterfall trajectories before and after thresholding for smooth trajectories. Although the result obtained no longer has the twists obtained by the original algorithm, there are many cases in which the trajectories are now shorter than before. This stems from the fact that the velocity of the current was large and in many cases it was not possible to find a continuous extension for a trajectory after the waiting period.

5 Computation of the Disturbance Field when the Camera Moves

The computation of the disturbance field until now was based only on temporal changes. This requires that the region under consideration be stabilized. When the camera moves, we perform affine registration of the average for each new frame, and then we compare and revise the average for the ensuing frames. Since the comparison to the average drastically lowers the sensitivity of the computation to noise, better results are also obtained in cases in which the registration is not perfect.

In extreme cases of a complex three-dimensional scene, it is impossible to obtain good stabilization using two-dimensional registration performed on the entire frame. In these cases, there is a possibility, in principle, of improving the results by dividing the frame into windows and performing the registration and averaging for each window separately.

In this section we discuss how to incorporate affine stabilization into our algorithm, and illustrate the results on a few examples.

5.1 Stabilization by $2D$ affine registration

The frame stabilization is a process that is based entirely on spatial information, and its purpose is to achieve optimal overlap between a pair of frames on the basis of a limited number of parameters. In our case the frames that we want to compare are the new frame and the picture of the average of the previous frames, which reflects the background.

If we would perform registration of the *last frame* to the background frame, as is normally done, we would very quickly reach a situation in which there are almost no areas of overlap between the new frames and the average due to the motion of the camera and its increased distance from the first frame. For this reason we employ the opposite approach of repeated registration of the *background* to the last frame. In this way maximum overlap between the average and the new frame is always maintained.

Note that registration cannot be perfect in principle, as it is a two-dimensional correction to a three-dimensional change, and the above errors accumulate from frame to frame. However, in our case the averaging is temporally weighted, which gives greater weight to the last frames (in which the errors are relatively small) than to the more distant frames (in which the errors are large), cf. [22].

The stabilization process is based on a search for affine correspondence between the frames. It starts out from the optical flow equation [23]:

$$pI_x + qI_y + I_t = 0$$

where q and p denote the displacement of every point along the x and y axes, and I_x, I_y, I_t denote the derivatives of the brightness of the point with respect to x, y, t respectively. If $2D$ affine correspondence exists between the frames (i.e., if overlap can be achieved by a $2D$ rotation, translation and anisotropic stretching), 6 parameters $a, \ldots, f$ can be found which satisfy the conditions

$$\begin{aligned} p(x,y,t) &= a + bx + cy \\ q(x,y,t) &= d + ex + fy. \end{aligned}$$

The stabilization algorithm seeks the parameters which would minimize the error of the optical flow over the entire frame, i.e., which would minimize the quantity

$$\mathrm{Err}(\mathrm{p},\mathrm{q}) = \sum_{\mathrm{x},\mathrm{y}} (\mathrm{pI_x} + \mathrm{qI_y} + \mathrm{I_t})^2$$

The minimum is found by differentiating the equation with respect to each of the parameters and setting the expression obtained equal to 0. Six linear equations with the six unknowns $a, \ldots, f$ are obtained.

In [21] it was proved that this computation always converges to a result that correctly reflects the most dominant motion in the scene. In most cases it is the motion of the background. Therefore, this stabilization method can also be applied to scenes that include the independent motion of many objects relative to the background.

After performing the registration and revising the disturbance map, registration of the particles should be performed according to the parameters found. In this way we maintain the stability of the particles positions relative to the background. Only after this step can the particles be displaced according to the new disturbance map.

Fig. 8 shows 10 pictures from a sequence showing a person walking while being followed by the camera. In this example the registration is very difficult to perform, as the background is three-dimensional, complex, and very close to the image of the person walking. Nevertheless, it can be seen that the tracking particles cling to the image of the person and not to the background. The tracking trajectories remain continuous up to the last frame, which is frame No. 80 in the original sequence.

Fig. 9 shows 4 pictures from a 60 frames movie taken from the second floor of a shopping mall while the camera was moving downward. Three figures moving along the entire sequence, for which individual stable continuous tracking trajectories were obtained, can be discerned (see arrows in first frame and trajectories in last frame). Stable tracking was not achieved for the fourth figure, the soldier moving along the lower right-hand side of the picture, possibly because his direction of motion was similar to the camera's motion. In this example we used tracking particles which had dynamic size, i.e., the particle size varied during tracking in order to optimally fit itself to the size of the disturbance been tracked.

Fig. 10 shows pictures from a sequence of a moving car. In this case we have an especially noisy scene, in which, in addition to the non-stabilized camera, there is a large tree that waves in the wind in the front. Although the movement

Figure 8: Ten frames from a sequence in which the camera was moving from left to right. This is a very difficult sequence for the stabilization process, because the background is very close to the moving person and it includes complex $3D$ information, In addition, the person moves very slowly (there are 80 frames between the first and last picture), thus the disturbances field is rather weak. Still, the tracking is accurate and persistent.

of the tree makes inter-frame registration difficult, it can be seen that the disturbance field is strong mainly in the region of the car. This stems from the fact that the intensity of the disturbances is proportional to the velocity of the moving objects, so that only relatively small disturbances are obtained around the tree. Note that the tracking trajectories appear only behind the car and not in other areas of motion.

Finally, Fig. 11 shows pictures from a sequence of 60 frames showing vehicles going through a traffic intersection. Many of the vehicles make a left turn accompanied by a change in their two-dimensional projection. Nevertheless, the tracking remains stable along the entire sequence (see trajectories in last frame).

6 Comparison with Other Methods

As we have already indicated in the introduction, the conventional methods for tracking nonrigid objects are not suitable for handling the examples presented here. The basic assumption is almost always that the shape of the object changes slowly (small deformation), as in the methods based on the occluding contours of objects [1], or methods based on extreme points in the object [4]. Clearly, these approaches are not suitable to handle flowing water (what are the "objects" in this case?), or camouflaged ants where occluding contours cannot be discerned reliably. Also, an approach like that described in [8], in which the minimum of the Hausdorff distance between prominent points sampled from the frame is sought, requires that the shape of the object change slowly between consecutive frames, and thus it does not meet our needs. The only methods that are not based on the assumption of slow changes, such as the method described in [7], require a geometric model that is confined to very specific cases. Typically, however, we do not have a general geometric model for flowing water or the motion of ants.

6.1 Comparison to the optical flow

The methods that could potentially be successful are the conventional methods for computing a motion field on the basis of brightness changes. These algorithms start out from the erroneous assumption (erroneous in our case) that all the changes in shading are caused only by motion and are not caused by any other factor, particularly not by changes in shape; at the same time, however, they are general enough to deal with diverse kinds of motions and do not require

Figure 9: Four frames from a long sequence, taken with a downward moving camera while the people in the scene are moving in different directions. There are 60 frames between the first picture (top right) and the last one (bottom left). Good and stable tracking of 3 moving figures is demonstrated (see arrows in first frame, and long trajectories in the last frame).

the isolation of the objects being tracked or the knowledge of their shape. In the following experiments we used the widely available algorithm of Lucas and Kanade [24], which is based on the optical flow equation [23] (see Sec. 5.1):

$$I_x U + I_y V + I_t = 0.$$

Our algorithm exploits temporal continuity. Conversely, the optical flow computation is instantaneous, and relies only on spatial information within a frame, which is the gradient (I_x, I_y, I_t) at the point. This computation is restricted to small motions (up to one grid-point displacement). In order to increase the range of displacements, a multi-scale approach using a Gaussian pyramid is typically employed. The use of a pyramid is based on the assumption that there is continuous transition between different scales regarding the displacement in the region being examined. The use of several iterations at every scale of the pyramid is based on the expectation of continuous convergence toward the optimal fit.

These assumptions do not always hold. In particular, in the case of the ants most of the information is found at fine scales (high frequencies). When we switch to coarser scales, we can no longer discern the ants or their motion. Therefore, we performed the experiments using many scales and allowing a large number of iterations at each scale. Still, the results obtained did not resemble the motion of the ants at all.

To accomplish a fair comparison, we advanced the hypothesis that perhaps our algorithm benefited from the use of two stages: first the disturbances (or, in effect, the positions of the ants) are obtained, and then the motion is computed. In order to help the optical flow algorithm as much as possible, we used our algorithm to identify the regions in which

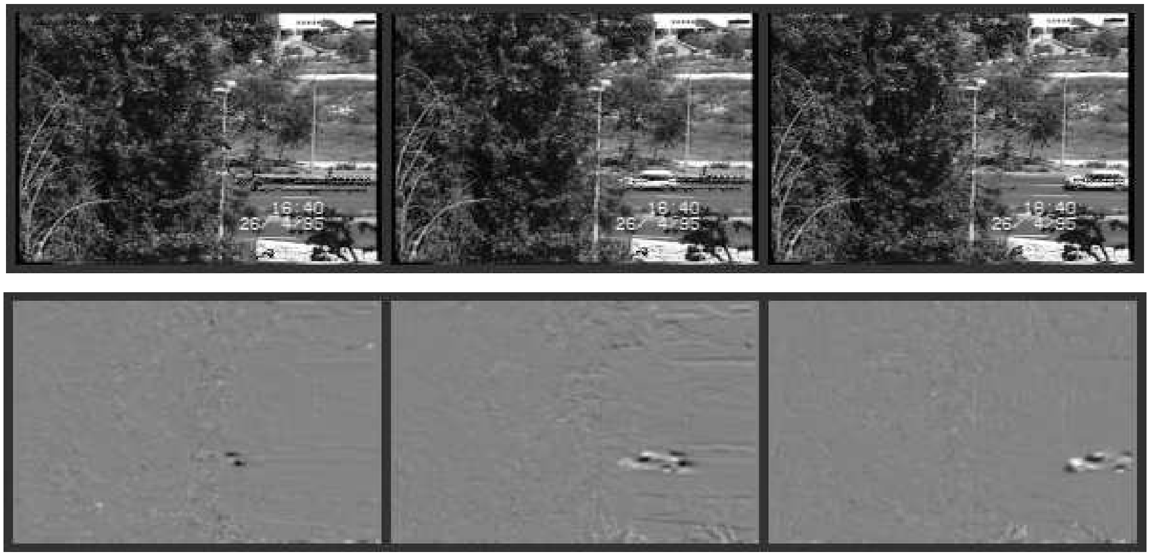


Figure 10: A non-stabilized sequence of a complex non-rigid scene: the tracking results (top) and the disturbance field (bottom); note that the disturbance is much stronger around the moving car as compared to the shaking tree. The movie was taken by a moving camera, while the big tree in front wave in the wind. This leads to fairly poor stabilization, but the tracking remains fairly accurate.

the ants are found and to mask all the rest (the background). We incorporated the masks obtained into the algorithm described in [24], to ensure that the computation is performed only on the regions of the ants and not the background.

The optical flow results can be seen in Fig. 12-I: The top picture shows the vertical component of the optical flow, which is nonzero only around ant locations. The next picture shows the flow vectors super-imposed on the original picture; as can be seen, the size of the flow lines does not provide a “good” representation of the motion of the ants, and in most of the regions where ants were discovered the displacements obtained are rather small. The last (bottom) picture shows the tracking trajectories that were obtained after 30 frames on the basis of the optical flow. Clearly the results do not reflect the real motion of the ants (compare with final results in Fig. 4).

We are not quite sure why the optical flow algorithm failed so miserably on this example. Possibly because at the fine scales at which the relevant information is attained the distances traversed by the ants are too large for the iterations to converge to the correct position. Or maybe because the ants change their direction and shape all the time, in violation of the basic assumption of the algorithm, according to which changes in shading are caused by motion.

Fig. 12-II shows the optical flow computation on one of the waterfalls movies: Here the optical flow algorithm does better, since the measured motion seems to be maintained across scales. Indeed, the direction of the optical flow seems to match the actual motion (see top 2 pictures, which correspond to two consecutive frames in the movie). At the same time, the length of the flow vectors varies a lot, and there are large variations even between consecutive frames.

The bottom picture in Fig. 12-II shows the results of tracking through 30 frames: the final result appears good, but closer inspection of a movie showing the motion of the tracking particles reveals that the speed (corresponding to the length of the flow vectors) does not correspond at all to the rate of flow of the water, and that the particles in fact move at a much slower rate than the water flow. In other words, in this case the optical flow correctly reflects the direction of flow, but not the flow velocity. Thus in this example where the moving targets change their shape, correct information regarding the direction of motion is obtained locally, but the magnitude of the motion cannot be extracted directly.

A final note of comparison: there was a difference of at least an order of magnitude between the run time of our algorithm (a few frames per second) to that of the multi-scale optical flow algorithm (a few minutes per frame).

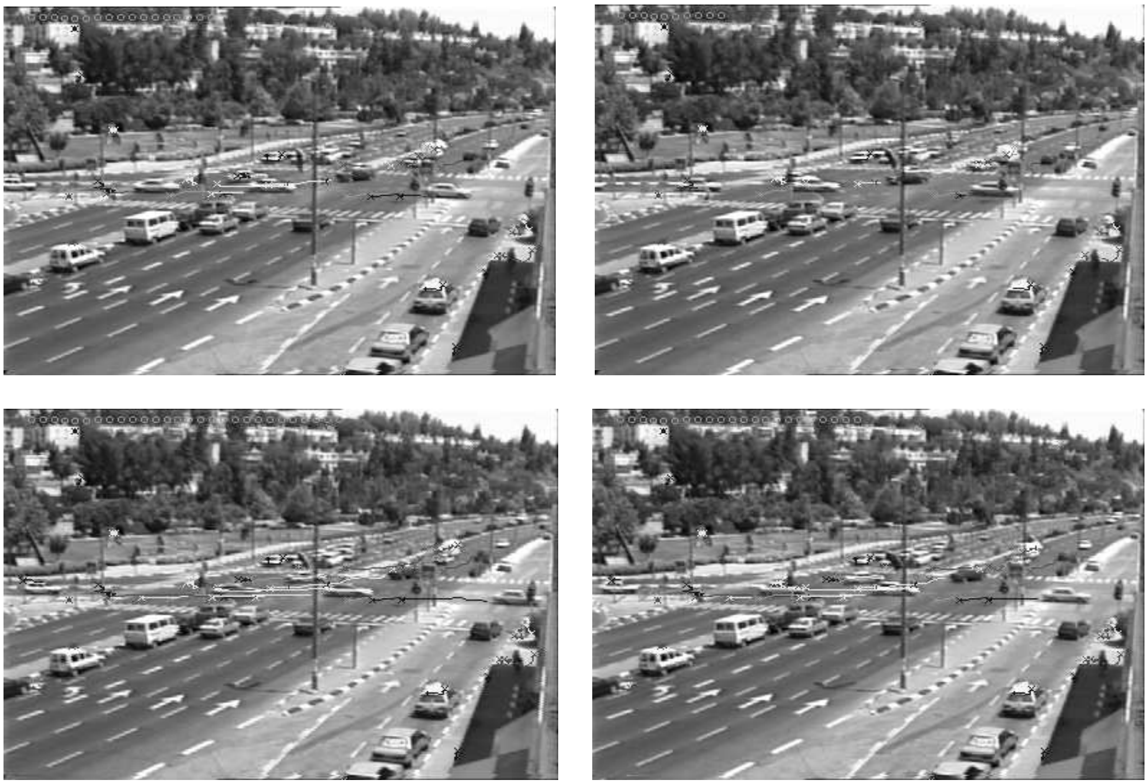


Figure 11: 4 frames from another long sequence, taken by a non-stabilized camera in a traffic intersection. The cars which turned to the left changed their $2D$ projection non-rigidly. There are 60 frames between the first picture (top right) and the last one (bottom left).

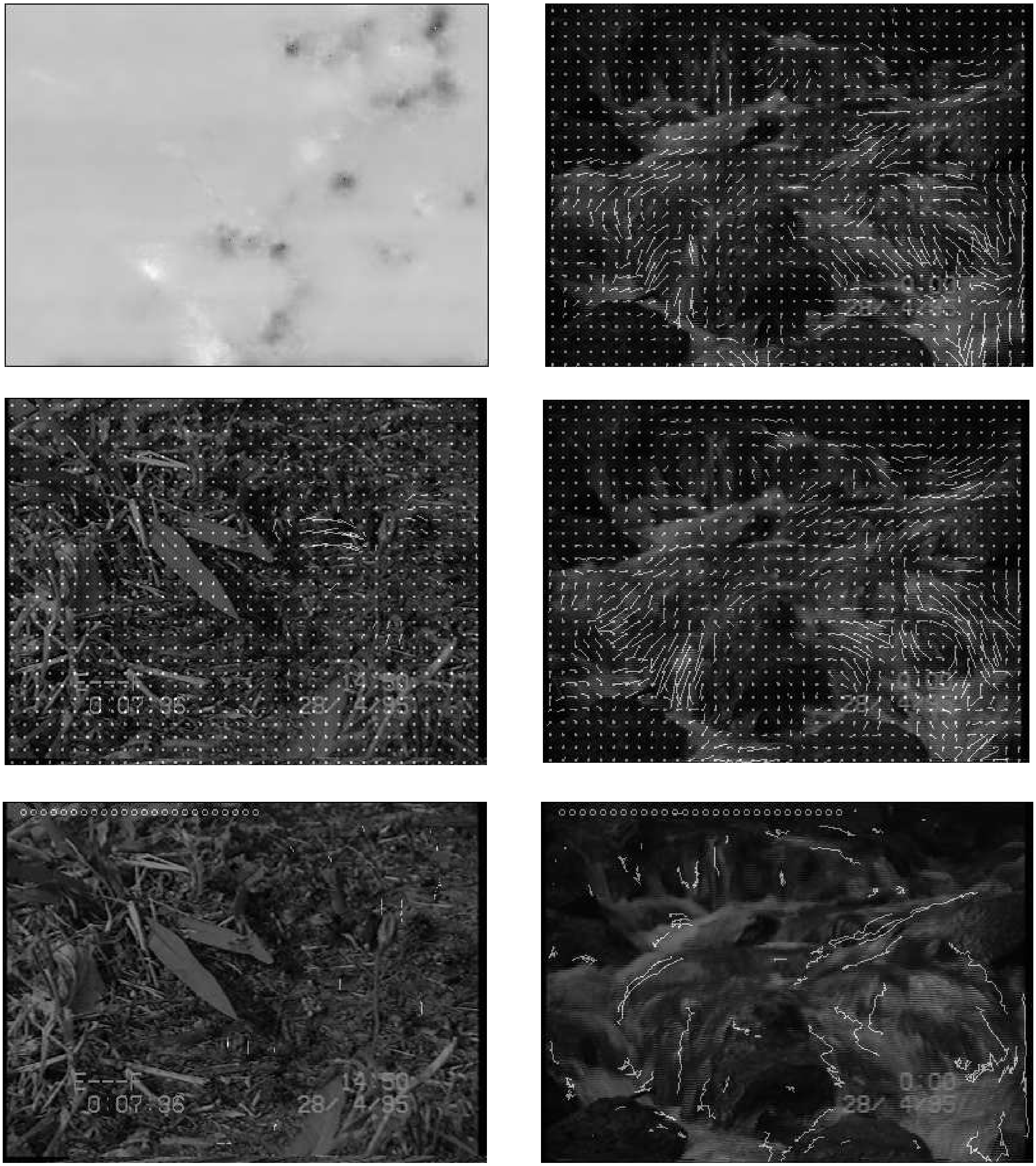
6.2 Comparison with point matching

We compared our algorithm with algorithms that find trajectories based on inter-frame point matching. The basic assumption of such algorithms is that interest points at each frame are provided by an external process (which in many cases includes manual identification or marked points), and the task is to match and follow the points in time.

For comparison, we used the algorithm described in [12], which was designed to identify and track the motion of vesicles. The vesicles are interesting objects to track: they move at different velocities, some upward and some downward; they appear and vanish, stop, or change direction. The algorithm described in [12] assumes that a list of positions of the vesicles is given in each frame in the sequence, and its purpose is to find trajectories that will connect the points is the best way from the standpoint of least energy (which is a function of inter-point distances in this case).

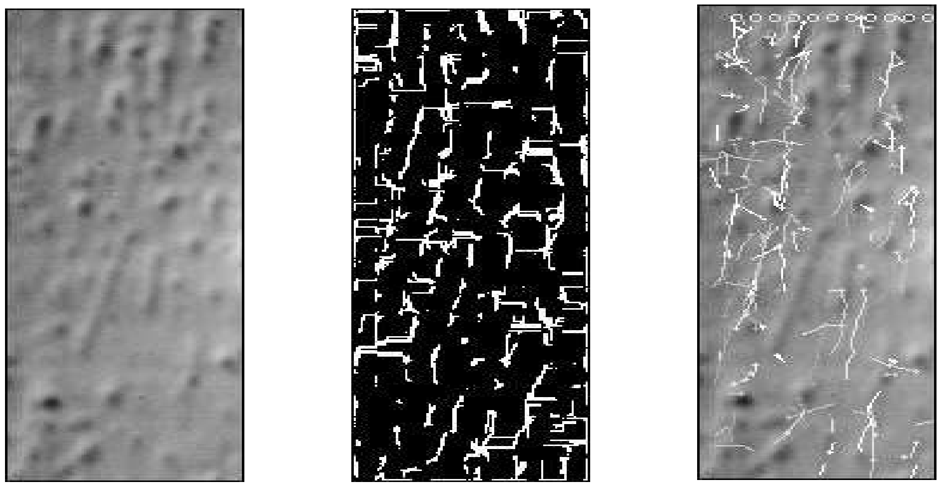
Fig. 13a shows one of the frames from the original movie. The output of [12] is shown in Fig. 13b. As far as the eye can see, the results are not so good. For example, we see many cases of transverse motion, which is not observed in the original movie. Moreover, only slow vesicles are tracked, whereas the rapidly moving vesicles are completely ignored. This solution is typically preferred by point matching algorithm. Fig. 13c shows the results obtained using our algorithm. When seen as a movie, the results appear smoother and in better agreement with the visually observed motion. Note that the particles preferred to cling to the rapidly moving vesicles, since the faster the object moves - the stronger is its disturbance.

To further test the idea of using point matching for tracking, we used our disturbance field to automatically identify moving objects and determine their positions. In this way a list of points indicating the positions of the objects can be obtained from every frame, and all that remains is to match the lists. (Note that the problem is somewhat more



I II

Figure 12: Tracking using Lucas & Kanade Optical fbw. *I:* The ants sequence: top - the fbw differs from 0 only around ant locations, but (middle) the vector fi eld is very small at those exact locations; bottom - the results of tracking through 30 frames using the optical fbw given above, which look very poor (compare to Fig. 4). *II:* the rapdis sequence: Top - two consecutive fbw images; the direction of the fbw fi eld appears correct, but its magnitude changes a lot from frame to frame. Bottom - the results of tracking through 30 frames, based on the optical fbw from above.



a) b) c)

Figure 13: Tracking vesicles: *a)* One frame from the Vesicles sequence. *b):* Results based on matching between points (taken from [12]). *c):* Results using our tracking algorithm: the trajectories seem to be smooth and in the right direction. Note that while the point matching algorithm prefers to track the slower vesicles, our algorithm prefers the faster ones, since they create stronger disturbances.

complicated, because points can appear, disappear, merge, or separate). After many attempts in this direction, we arrived at results which were poorer than the original results obtained by our algorithm. The reason seems to be that the minimum distance requirement is never correct and places an undesirable restriction on the generality of the algorithm.

6.3 Discussion of results

In our experiments our algorithm was very reliable and performed better than other algorithms on sequences with many independently moving objects or complex patterns of motion. The reason for the superior performance seems to be the use for tracking of a shape invariant property, whose character is only weakly influenced by the shape of the moving objects. For every moving object, regardless of its shape, we obtain a wave-like disturbance with a head at the present position of the object. As soon as we have such a stable model, we can perform a straightforward search for the head of the disturbance - or the present location of the object.

In addition to the quality of the results, there is an important difference in the computation complexity and the time complexity between the algorithm proposed here and algorithms which compute the optical flow: while a multi-scale computation of the optical flow between a pair of frames generally takes several minutes, our experimental system can track 100 different objects simultaneously at the average rate of 5 frames per second. This was accomplished without having invested any effort in optimizing performance (on an Indigo II), so that it is undoubtedly possible to bring the system up to real-time video rate on standard computers.

Our tracking algorithm uses several free parameters, for example, the history factor w, as well as threshold levels for identifying the appearance and disappearance of a disturbance. Nevertheless, our initial experiments showed robustness to the choice of the parameters: different parameter values did not lead to significantly different results. Thus, in all the different experiments that we have subsequently performed on different scenes and reported above, we retained the same parameter values without change. Parameters were *not* tuned to obtain optimal results per each sequence.

What about the cases in which regions in the picture perform really large motions upon passage from frame to frame (i.e., when the optical flow assumption fails)? In these cases we cannot directly obtain reliable spatio-temporal information indicating the direction of motion; there is no escape from relying on other criteria, such as trajectory smoothness or velocity stability, as well as other parameters of motion. In these cases our algorithm can serve as a preliminary step of identifying and extracting selected interest points from a sequence of frames, to be followed possibly by point matching.

References

[1] A. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Modes", *International Journal of computer vision*, vol 1(3), pp. 321-331, 1988.

[2] N. Ueda, K. Mase, "Tracking Moving Contours Using Energy Minimizing Elasitic Contour Models", *ECCV 92, European Conference on computer Vision*, pp. 453-457, 1992.

[3] D. Geiger, A. Gupta, L. Costa and J. Vlontzos, "Dynamic Programing for Detecting, Tracking and Matching Deformable Contours", *IEEE Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3, pp. 294-302, 1995.

[4] I. Cohen, N. Ayache and P. Sulger, "Tracking Points on Deformable Objects, Using Curvature Information", *ECCV 92, Eurpean Conference on Computer Vision*, pp. 458-465, 1992.

[5] J. S. Duncan, R. L. Owen, L. H. Staib and P. Anandan, "Measurement of Non-Rigid Motion Using Contour Shape Descriptor", *IEEE Computer vision and pattern recognition*, pp. 318-324, 1991.

[6] A. M. Baumberg, D. C. Hogg, "An Efficient Method for Contour Tracking using Active Shape Models", *University of Leeds*, Report 94.11. 1994.

[7] J. P. Berroir, I. Herlin and I. Cohen,. "Non-Rigid Motion Without Relying on Local Features: A Surface Model Applied to Vortex Tracking", *INRIA Repport*, No. 2684, 1995.

[8] D. P. Huttenlocher, J. J. Noh and W. J. Rucklidge, "Tracking Non-Rigid Objects in Complex Scenes", *ICCV 93*, pp. 93-101, 1993.

[9] R. F. Rashid, "Lights: A System for Interpretation of moving light Display", *PhD thesis*, University of Rochester, 1980.

[10] M. J. Fletcher, K. Warwick and R. J. Mitchell, "The Application of a Hybrid Tracking Algorithm to Motion Analysis", *IEEE computer vision and pattern recognition*, pp. 84-89, 1991.

[11] R. F. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Enginiring.*, Vol. 82, No. 1 pp. 35-45, 1960.

[12] H. Applebaum, M. Werman, "The detection of motion in noisy environments using a 3D f* algorithm", *PhD thesis*, Hebrew University, 1995.

[13] I. K. Sethi, R. Jain, "Finding Trajectories of Feature Points in Monocular Images Sequance", *IEEE Trans. on Pattern Analysis and Machine Intell.*, Vol. 9 No. 1 pp. 56-73, 1987.

[14] J. O'Rourka J., N. L. Badler, "Model-Based Image Analysis of Human Motion Using Constraint Propagation", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, No. 6 pp. 522-536, 1980.

[15] A. Shio, J. Sklansky, "Segmentation of People In Motion", *IEEE Workshop on Visual Motion*, pp. 325-332, 1991.

[16] H. H. Nagel, "Formation of an Object Concept by Analysis of Systematic Time Variation in the Optically Perceptible Environment", *Computer Graphics and Image Processing*, Vol. 7, pp. 149-194, 1978.

[17] H. H. Nagel, J. Ramesh, "On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 2, 1979.

[18] Y. Z. Hsu, H. H. Nagel and G. Rekers, “New Likelihood Test Methods for Change Detection in Image Sequences”, *Computer Vision, Graphic, and Image Processing*, 26, pp. 73-106, 1984.

[19] R. C. Gonzalez, R. E. Woods, “Digital Image Processing”, pp. 465-478.

[20] J. R. Bergen, P. J. Burt, K. Hanna, R. Hingorani, P. Jeanne and S. Peleg, “Dyhamic Multiple-Motion Computation”, *Proceeding of Israeli Conference on Artificial Intelligence and Computer Vision*, pp. 147-156, 1991.

[21] P. J. Burt, R. Hingorani and R. J. Kolezynski, “Mechanism for isolating component patterns in the sequential analysis of multiple motion”, *IEEE Workshop on Visual Motion*, pp. 187-193, 1991.

[22] M. Irani, B. Rousso and S. Peleg, “Computing Occluding and Transparent Motion”, *International Journal of Computer Vision*, 1993.

[23] B. K. P. Horn, B.G. Schunck, “Determining Optical Flow”, *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.

[24] B. D. Lucas, T. Kanade. “An Iterative Image Registration Techniqe with an Application to Stereo Vision”, *Image Understanding workshop*, pp. 121-130, 1981.

[25] R. B. Podana, “Temporal Texture and Active Recognition”, *PhD Thesis*, University of Rochester, 1994.

[26] P. J. Burt, “Smart Sensing within a Pyramid Vision Machine”, *Proceeding of the IEEE*, vol. 76 No. 8, pp. 1006-1015, 1988.

[27] D. J. Heeger, “Optical Flow from Spatiotemporal filter”, *IEEE symp. of computer Vision*, pp. 181-190, 1987.

[28] Y. Rosenberg, M. Werman, “Tracking a Moving Object Using a Video Camera”, *Master Thesis*, Hebrew University of Jerusalem, Israel 1995.