On the Handling of Spatial and Temporal Scales in Feature Tracking

Lars Bretzner and Tony Lindeberg

Computational Vision and Active Perception Laboratory (CVAP)
Department of Numerical Analysis and Computing Science
KTH (Royal Institute of Technology)
S-100 44 Stockholm, Sweden

1 Introduction

Being able to track image structures over time is a useful and sometimes necessary capability for vision systems intended to interact with a dynamic world. There are several computer vision algorithms in which tracking arises as an important subproblem. Examples of this include:

- Fixation of a physical point/region in the world over time.
- Object recognition in a dynamically varying environment.
- Motion segmentation and structure from motion computations.

There is an extensive literature on tracking methods operating without *a priori* knowledge about the world or specific models. The work in this direction can be classified into three main categories: correlation based tracking, optical flow based tracking and feature tracking. As examples of feature tracking, Shapiro [1] detects and tracks corners individually in an algorithm originally aimed at applications such as videoconferencing. Smith and Brady [2] track a large set of corners and use the results in a flow-based segmentation algorithm. Zheng and Chellappa [3] track features while compensation for camera motion. Cipolla and Blake [4] estimate time-to-contact by using snakes to track deforming image features, and Koller et al. [5] track combined motion and grey-level boundaries in traffic surveillance. For edge based tracking approaches, see Faugeras [6].

The subject of this article is to consider the domain of feature tracking and to complement previous works by addressing the scale problems arising in this context. In most previous works, the analysis is performed at a single predetermined scale, and this may can cause severe problems if the size of image structures change over time due to expansions or contractions.

We will show how an explicit mechanism for automatic scale selection can be included in a feature tracker to handle tracking situations in which the size variations are large. Besides avoiding explicit setting of scale levels for feature detection, and thus overcoming some of the very fundamental limitations of processing image sequences at a single scale, it will be demonstrated how scale levels selected by a scale selection procedure are useful for adapting the window size for correlation and as a matching cue. It will also be illustrated how an appropriate choice of temporal scale can improve the performance.

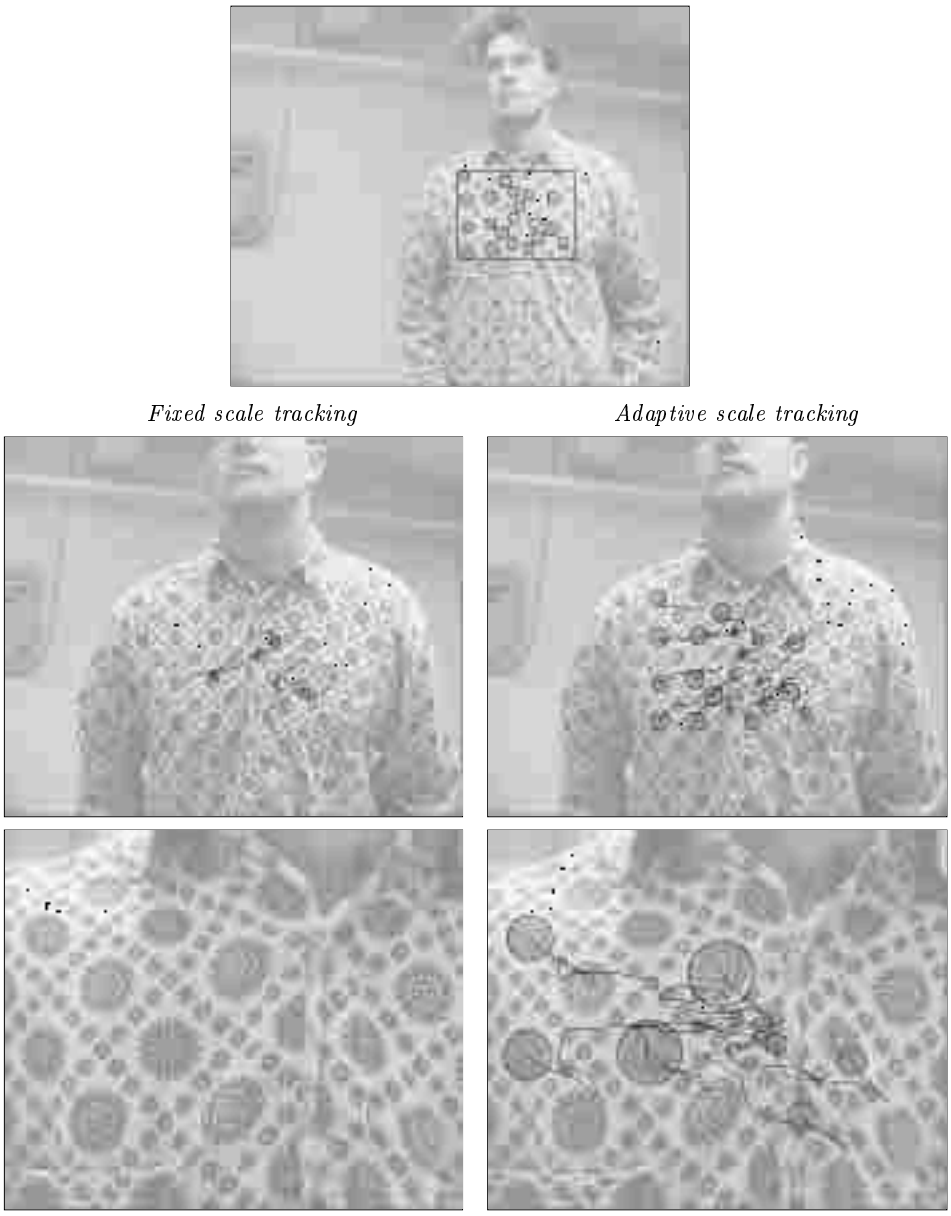

Fig. 1. Illustration of the importance of automatic scale selection when tracking image structures over time. A fixed scale tracker fails early when the size variations are large (left), whereas all visible blobs are correctly tracked up to the last image when a mechanism for adaptive scale selection has been included (right).

2 The need for automatic scale selection in tracking

Figure 1 shows a few snapshots from a sequence in which a person, dressed in a spotted shirt, approaches the camera. In a rectangular area in the initial frame, 20 blobs have been detected. Then, a comparison is given between a fixed scale and an adaptive scale tracker. Notably, the tracker using a fixed scale for blob detection and a fixed window size for hypothesis evaluation by correlation will fail after a number of frames and the features will be lost. The reason why this occurs, is simply the fact that the blobs no longer exist at the predetermined scale. The right column shows the result of incorporating a mechanism for adaptation of the scale levels to the local image structure (details will be given in later sections). As can be seen, all visible blobs are correctly tracked over the whole sequence.

A more general motivation to this study is that any feature detector suffers from localization errors due to, e.g, noise and motion blur. When detecting rigid body motion or recovering 3D structure from feature point correspondences in an image sequence, it is important that the motion in the scene is large compared to the localization errors of the feature detector. If the inter-frame motion is small, we therefore have to track features over a large number of frames to obtain accurate results. This requirement constitutes a key motivation for including a scale selection mechanism in the tracker, to obtain long feature trajectories of features as input to algorithms for motion estimation and recovery of 3D structure.

3 Feature detection with automatic scale selection

A natural framework to use when extracting features from image data is to define the image features from multi-scale differential invariants expressed in terms of Gaussian derivative operators [7, 8], or more specifically, as maxima or zero-crossings of such entities [9]. In this way, image features such as corners, blobs, edges and ridges can be computed at any level of scale.

A basic problem that arises for any such feature detector concerns how to determine at what scales the image features should be extracted, or if the feature detection is performed at several scales simultaneously, what image features should be regarded as significant. A framework addressing this problem has been developed in [10, 9]. In summary, one of the main results from this work is a general principle for scale selection, which states that scale levels for feature detection can be selected from the scales at which normalized differential invariants assume maxima over scales. In this section, we shall give a brief review of how this methodology applies to the detection of features such as blobs and corners. The image features so obtained, with their associated attributes resulting from the scale selection method, will then be used as basic primitives for the tracking procedure.

Normalized derivatives. The scale-space representation L [11, 12] of a signal f is defined as the result of convolving f with Gaussian kernels $g(\cdot; t)$ for different values of the scale parameter t

$$L(\cdot;\ t) = g(\cdot;\ t) * f \tag{1}$$

In this representation, γ-parameterized, normalized derivatives [13] are defined by

$$\partial_\xi = t^{\gamma/2}\, \partial_x \tag{2}$$

where t is the variance of the Gaussian kernel. From this construction, a normalized differential invariant is then obtained by replacing all spatial derivatives by corresponding normalized derivatives according to (2). γ is chosen to be one for blobs, 0.875 for corners and 0.75 for ridges.

Corner detection with automatic scale selection. A common way to define a corner in a grey-level image is as a point at which both the curvature κ of a level curve and the gradient magnitude $|\nabla L|$ are high [14, 15]. From the product of κ and $|\nabla L|^3$, we obtain the following essentially affine invariant expression

$$\tilde{\kappa} = L_{yy} L_x^2 + L_{xx} L_y^2 - 2 L_x L_y L_{xy} \tag{3}$$

with its corresponding γ-parameterized, normalized differential invariant

$$\tilde{\kappa}_{\gamma-norm} = t^{2\gamma} \tilde{\kappa}. \tag{4}$$

In [16], a junction detector with automatic scale selection is formulated in terms of the detection of *scale-space maxima* of $\tilde{\kappa}^2_{norm}$, *i.e.*, by detecting points in scale-space where $\tilde{\kappa}^2_{norm}$ assumes maxima with respect to both scale and space. To improve the localization of the corner feature, a modified Förstner operator [17] is used to compute new localization estimates using scale information from the detection step.

Blob detection with automatic scale selection. A straightforward method for blob detection can be formulated in an similar way by detecting scale-space maxima of the square of the normalized Laplacian

$$\nabla^2_{norm} L = t\, (L_{xx} + L_{yy}). \tag{5}$$

Ridge detection with automatic scale selection. In [13] a ridge detector for automatic scale selection is formulated based on the difference between the squared principal curvatures L_{pp} and L_{qq} of the grey-level landscape

$$\mathcal{A} L_{\gamma-norm} = t^{2\gamma}\, |L_{pp}^2 - L_{qq}^2| = t^{2\gamma}\, ((L_{xx} - L_{yy})^2 + 4 L_{xy}^2) \tag{6}$$

In analogy with the corner detection method we can expect the scales where $\nabla^2_{norm} L$ and $\mathcal{A} L_{\gamma-norm}$ assume their maxima over scales to reflect the size of the corresponding image structures.

4 Multi-scale and multi-cue tracking and prediction

When tracking features over time, we can expect the position of the feature as well as the appearance of its surrounding grey-level pattern to vary. To relate features over time, we shall throughout this work make use of the common assumption about small motions between successive frames. The tracking scheme is based on a traditional predict-detect-match-update loop.

There are several ways to predict the position of a feature in the next frame based on its positions in previous frames. Whereas the Kalman filtering methodology has been commonly used in the computer vision literature, this approach suffers from a fundamental limitation if the motion direction suddenly has changed. If a feature moving in a certain direction has been tracked over a long period of time, then the built-in temporal smoothing of the feature trajectory in the Kalman filter, implies that the predictions will continue to be in essentially the same direction, although the actual direction of the motion has changed. It will hence be likely that the feature is lost at the discontinuity.

For this reason, we use a simpler first-order prediction, which uses the motion between the previous two successive frames as a prediction to the next frame.

Within a neighbourhood of each predicted feature position, we detect new features using the above mentioned detection procedure with automatic scale selection. An important property of this method is that it automatically selects smaller regions of interest for small-size image structures, and larger search regions for larger size structures. This scale information is used for three main purposes:

- Setting the search regions for matching candidates.
- Setting the window size for correlation matching.
- Using scale stability as a matching condition.

In addition, for elongated structures, such as ridges, we define elliptic regions of interest by computing a second moment matrix for the distribution of directional derivatives. With t denoting the scale of an image feature detected according to section 3, this descriptor is defined as

$$\mu = \int \begin{pmatrix} L_x^2 & L_x L_y \\ L_x L_y & L_y^2 \end{pmatrix} g(\xi; t) d\xi \tag{7}$$

The correlation window size is determined by the largest of the semi-axes and the size of the search region is set to the spatial extent of the previous image feature, multiplied by a safety factor. Within this window, a certain number of candidate matches are selected. Then, an evaluation of these matching candidates is made according to the next section.

Based on the assumption of small inter-frame image motions, we use a multiple cue approach to the feature matching problem. Instead of evaluating the matching candidates using a correlation measure on a local grey-level patch only, as done in most feature tracking algorithms, we combine the correlation measure with significance stability, scale stability and proximity measures as defined below.

Patch similarity. Let $C(A,B)$ be the Gaussian-weighted intensity cross-correlation between two square image patches A and B, centered at each feature and with the size set from the detection scale. Then, this normalized measure is

$$S_{corr}(A,B) = \frac{C(A,B)}{\sqrt{C(A,A)\,C(B,B)}} \tag{8}$$

For anisotropic features, the shape of the Gaussian window is given by the second moment matrix as defined in (7).

Significance stability. The significance of an image feature detected according to section 3 is defined as the normalized response at the local scale-space maximum (either by (4) or (5)). With R_A and R_B denoting the significance measures of the corresponding features A and B, we measure similarity by relative differences according to

$$S_{sign} = |\log \frac{R_B}{R_A}| \tag{9}$$

Scale stability. To measure the relative scale variation between two features A and B, detected at scales t_A and t_B respectively, we define

$$S_{scale} = |\log \frac{t_B}{t_A}| \tag{10}$$

Proximity. We measure how well the position x_A of feature A corresponds to the position x_{pred} predicted from feature B (with detection scale t_B)

$$S_{pos} = \frac{\|x_A - x_{pred}\|}{\sqrt{t_B}} \tag{11}$$

The division by $\sqrt{t_B}$ compensates for poor feature localization at coarse scales.

Combined similarity measure. The combined multi-cue similarity measure is then

$$S_{comb} = c_{corr}S_{corr} + c_{sign}S_{sign} + c_{scale}S_{scale} + c_{pos}S_{pos}$$

The winner of the matching is the candidate having the highest similarity value above a fixed threshold. The threshold is necessary to suppress false matches when features disappear due to e.g occlusion. Each feature has been assigned a quality measure indicating how stable it is over time. If a feature is matched, this quality measure is increased and the position, the scale descriptor, the significance value and the grey-level patch of the feature are updated. If no match is found, the quality measure is decreased, the position is set to the predicted one and the scale descriptor, the significance value and the patch are left unchanged (see [18] for further details about the scheme).

Finally, for each frame, the feature set is parsed to detect feature merges and to remove features having too low quality values. When two features merge, their trajectories are terminated and a new trajectory is initiated. In this way, we obtain more reliable feature trajectories for further processing.

5 Experimental results

Corner tracking. Let us first demonstrate the performance of the algorithm when applied to an image sequence consisting of 60 frames. In this sequence, the camera moves in a fairly complex way relative to a static scene. The objects of interest on which the features (here corners) are detected are a telephone and a package on a table. From the junctions detected in the initial frame, a subset of 14 features were selected manually. Figure 2 shows the initial frame and the situation after 30, 50 and 60 frames. The size of the circles correspond to the detection scales of the features, black segments on the trajectories indicate matched positions while white segments show unmatched (predicted) positions.

Note how the detection scale adapts to the image structure when we approach the objects; tracked sharp corners are still detected at fine scales while blunt corners are detected at coarser scales. These blunt corners are lost if similar tracking is performed at a fixed scale.

Blob tracking. Besides the ability to track blobs under large size variations, as shown already in figure 1, let us illustrate how stability of feature trajectories over a large number of frames is likely to give us trajectories corresponding to reliable and stable physical scene points or regions of interest on objects. In other words, responses due to noise, illumination and temporary occlusions can be suppressed. Figure 3 shows 4 frames from a sequence in which the 10 most significant blobs have been tracked in a region around the face of the subject. The subject first approaches the camera and then moves back to his initial position. The figure shows the initial frame and the situation after 20, 45 and 90 frames. As can be seen, the remaining four features correspond to the nostrils and the eyes. This ability to register stable image structures over time is clearly a desirable quality in many computer vision applications. And as we stress again, for general scenes with large expansions or contractions, a scale selection mechanism is essential to allow for such registrations.

Ridge tracking. Figure 4 shows a few frames from a sequence with hand gestures, in which the two most significant ridges, corresponding to the thumb and the forefinger, have been detected in the initial frame. The ridge operator is anisotropic and the features are represented by elliptical regions given by the second moment matrix according to (7). The figure shows the situation after tracking over 40, 110 and 150 frames. Again, the scale selection mechanism captures the large size variations in the scene.

A more extensive experimental evaluation of the performance of the tracker is presented in [18].

6 The influence of temporal scale

So far we have shown how the performance of a feature tracker can be improved by automatic selection of spatial scales. To illustrate the importance of temporal

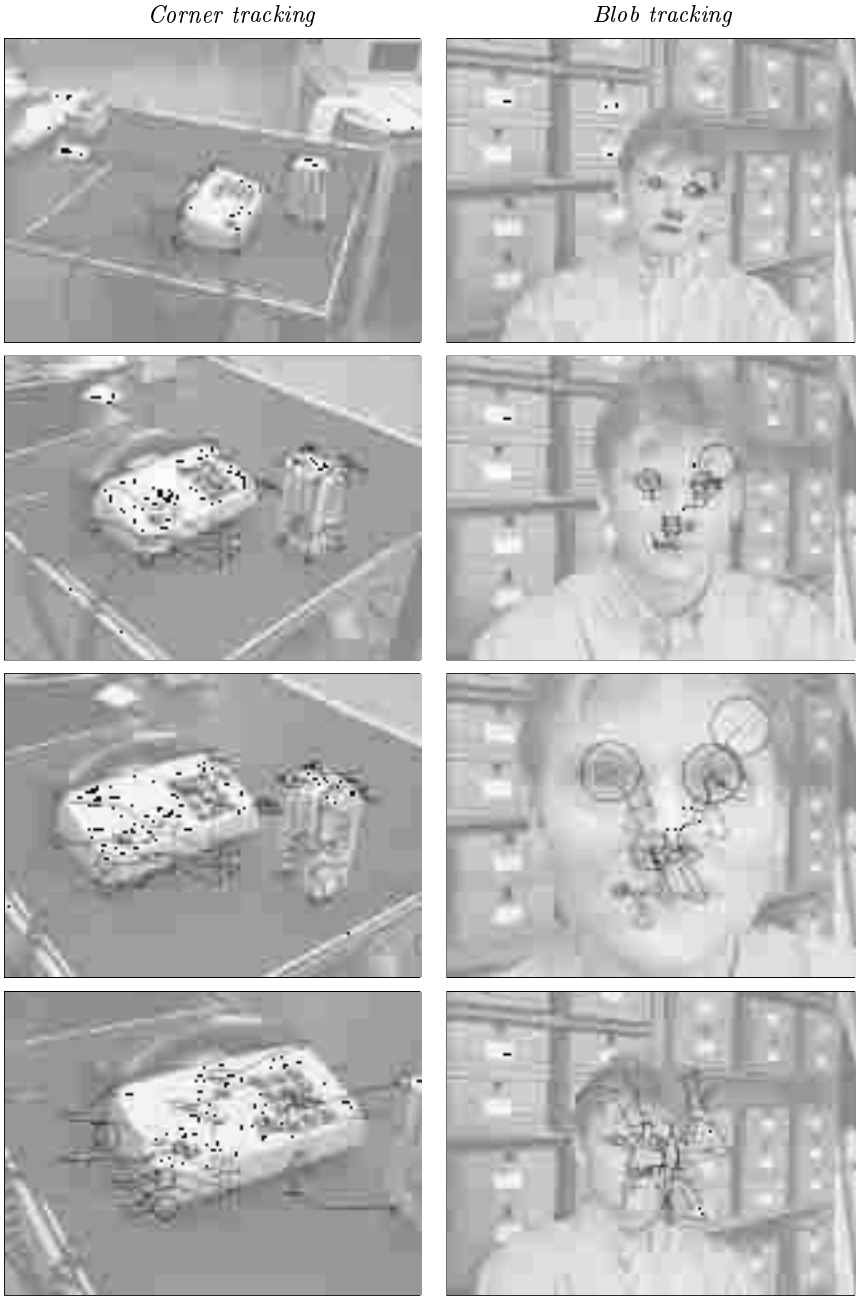

Fig. 2. (Left column) Corner tracking using automatic scale selection, the initial frame with 14 selected corners and the situation after 30, 50 and 60 frames. Sharp corners are detected at fine scales while blunt corners are detected at coarser scales. As can be seen, all visible corners are correctly tracked.

Fig. 3. (Right column) Blob tracking using automatic scale selection, the initial frame and the situation after 20, 45 and 90 frames. Note that blob responses that are unstable over time are suppressed.

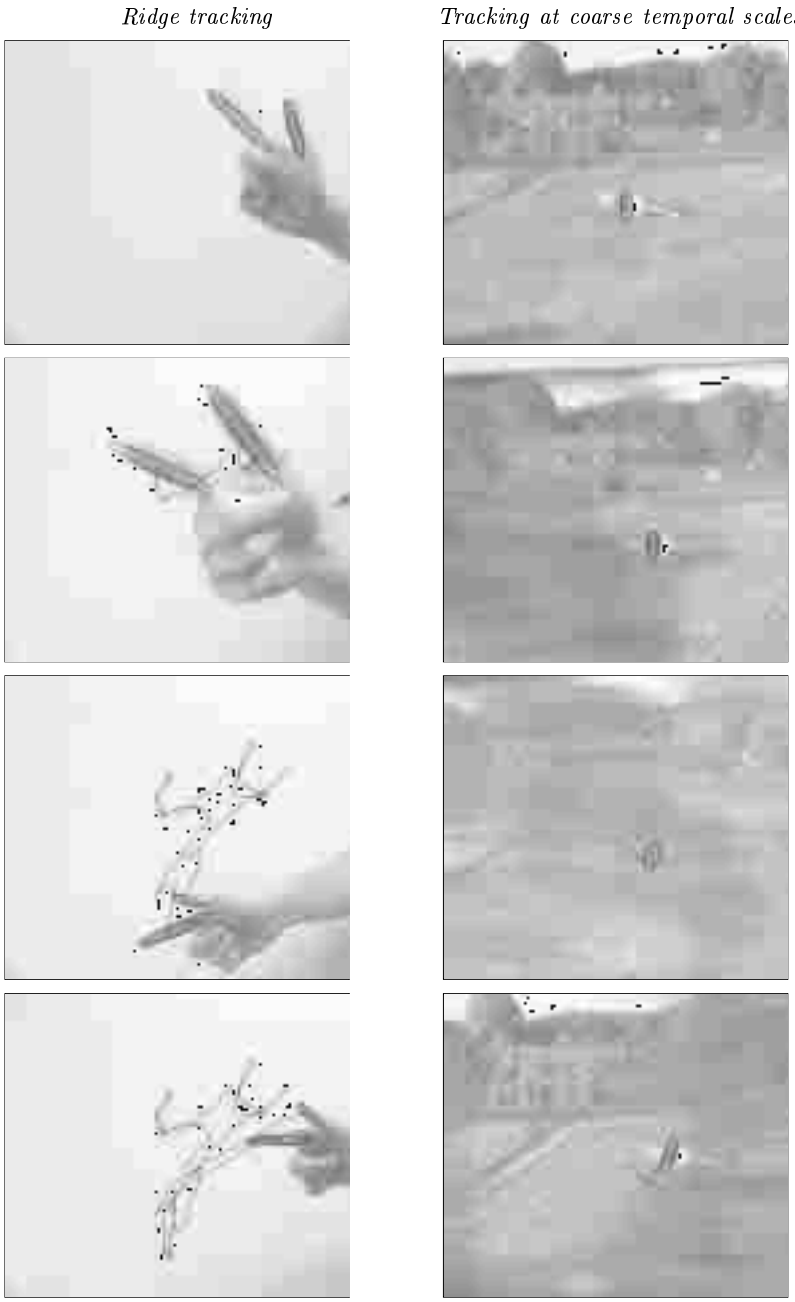

Fig. 4. (Left column) Ridge tracking of the two most significant ridges in a hand sequence. The images show the situation after 40, 110 and 150 frames.

Fig. 5. (Right column) The result from feature tracking at coarse temporal scales in a scene with transparent motion. The illustration shows a blob tracked in the time-filtered sequence superimposed on the original sequence.

scales in this context, let us consider an example with transparent motion. A well-known problem in computer vision concerns how to maintain tracking in situations when objects are (partly) occluded. In transparent motion scenes, we can have distant slowly moving objects of interest which are occluded by nearby fast moving structures. Under such circumstances, a normal feature tracker will inevitably fail. However, if we apply the tracker at coarser temporal scales in a spatio-temporal scale-space representation [19], the performance can be improved substantially if we select an appropriate temporal scale.

Figure 5 shows an example of this where we track a distant object sen through fast moving vegetation in the foreground. In many frames of the sequence, the object is hardly visible at all and hence impossible to detect if the analysis is restricted to individual frames only. The figure shows the tracked blob in every 10th frame of the time-filtered sequence superimposed on the original sequence. The tracking is only succesful if it is performed at a suitable temporal scale and using automatic selection of spatial scales. Here, the temporal scale is selected manually, and the example clearly illustrates the need for a mechanism for automatic selection of temporal scales in this kind of tracking situations. A basic framework for approaching this problem is presented in a companion paper, see [20]. So far, however, this approach has not been algorithmically integrated with the feature tracker, and we are currently working on the topic.

7 Summary

We have presented a framework for feature tracking in which a mechanism for automatic scale selection has been built into the feature detection stage and the additional attributes of the image features obtained from the scale selection module are used for guiding the other processing steps in the tracking procedure.

We have argued that such a mechanism is essential for any feature tracking procedure intended to operate in a complex environment, in order to adapt the scale of processing to the size variations that may occur in the image data as well as over time. If we attempt to track features by processing the image data at one single scale only, we can hardly expect to be able to follow the features over large size variations. This property is a basic consequence of the inherent multi-scale nature of image structures, which means that a given object may appear in different ways depending on the scale of observation.

Specifically, based on a previously developed feature detection framework with automatic scale selection, we have presented a scheme for tracking corners, blobs and ridges over time in which:

- the image features at any time moment are detected using a feature detection method with automatic scale selection, and
- this information is used for
  - guiding the detection and selection of new feature candidates,
  - providing context information for the matching procedure,
  - formulating a similarity measure for matching features over time.

Besides avoiding explicit selection of scale levels for feature detection, the feature detection procedure with automatic scale selection allows us to track image features over large size variations. As demonstrated, this mechanism can substantially improve the performance relative to a fixed-scale feature tracker.

Since the scale levels reflect the spatial extent of the image structures, we can also use this context information for avoiding explicit settings of distance thresholds and predefined window sizes for matching. Moreover, by including the scale, position and significance information associated with the image features from the scale selection procedure into a multi-cue similarity measure, we showed how we in this way can improve the reliability of the low-level matching procedure.

Of course, there are inherent limitations in tracking each feature individually as done in this work, and there are always a number of situations where the tracking algorithm will fail. Typically, this occurs because of rapid changes in the local grey-level pattern around the feature, corresponding to violations of the assumption about small inter-frame motions.

A notable conclusion that can be made in this context, is that despite these limitations, we have shown by examples that the resulting tracking procedure is able to track most of the visible features that can be followed over time in the sequences presented in this article. By this we argue that the type of framework presented here provides an important step towards overcoming some of the limitations in previous feature tracking algorithms.

We have also shown an example of a situation where temporal filtering highly improves the tracking performance. We are convinced that in a integrated tracking system, working in a spatio-temporal scale-space, mechanisms for automatic selection of both spatial and temporal scales are necessary.

References

1. L. S. Shapiro, *Affine analysis of image sequences*. Cambridge, England: Cambridge University Press, 1995.
2. S. M. Smith and J. M. Brady, "Asset-2: Real-time motion segmentation and shape tracking," *IEEE-PAMI*, vol. 17, no. 8, pp. 814–820, 1995.
3. Q. Zheng and R. Chellappa, "Automatic feature point extraction and tracking in image sequences for arbitrary camera motion," *IJCV*, vol. 15, no. 1, pp. 31–76, 1995.
4. R. Cipolla and A. Blake, "Surface orientation and time to contact from image divergence and deformation," in *2nd ECCV*, pp. 187–202, Springer-Verlag, 1992.
5. D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *3th ECCV*, pp. 189–196, 1994.
6. O. Faugeras, *Three-dimensional computer vision*. Cambridge, Massachusetts: MIT Press, 1993.
7. J. J. Koenderink and A. J. van Doorn, "Generic neighborhood operators," *IEEE-PAMI*, vol. 14, no. 6, pp. 597–605, 1992.
8. L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "Scale and the differential structure of images," *IVC*, vol. 10, no. 6, pp. 376–388, 1992.

9. T. Lindeberg, *Scale-Space Theory in Computer Vision*. Netherlands: Kluwer, 1994.
10. T. Lindeberg, "On scale selection for differential operators," in *8th SCIA*, pp. 857–866, 1993.
11. A. P. Witkin, "Scale-space filtering," in *8th IJCAI*, pp. 1019–1022, 1983.
12. J. J. Koenderink, "The structure of images," *Biol. Cyb.*, vol. 50, pp. 363–370, 1984.
13. T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 1996*, (San Francisco, California), pp. 465–470, IEEE Computer Society Press, June 1996.
14. L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, vol. 1, no. 2, pp. 95–102, 1982.
15. J. J. Koenderink and W. Richards, "Two-dimensional curvature operators," *JOSA*, vol. 5:7, pp. 1136–1141, 1988.
16. T. Lindeberg, "Junction detection with automatic selection of detection scales and localization scales," in *Proc. 1st ICIP*, vol. I, pp. 924–928, IEEE Comp. Soc. Press, 1994.
17. M. A. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centers of circular features," in *ISPRS*, 1987.
18. L. Bretzner and T. Lindeberg, "Feature tracking with automatic selection of spatial scales," Tech. Rep. ISRN KTH/NA/P--96/21--SE, KTH, 1996.
19. T. Lindeberg and D. Fagerström, "Scale-space with causal time direction," in *4th ECCV*, vol. 1064, (Cambridge, UK), pp. 229–240, 1996.
20. T. Lindeberg, "On automatic selection of temporal scales in time-causal scale-space," in *1st Intl. Conf. on Scale-Space Theory in Comp. Vis.*, (Utrecht), 1997. (to appear).