

Zoran Živković · Ferdinand van der Heijden

## Improving the selection of feature points for tracking

Received: 26 August 2003 / Accepted: 5 March 2004 / Published online: 4 June 2004  
© Springer-Verlag London Limited 2004

**Abstract** The problem considered in this paper is how to select the feature points (in practice, small image patches are used) in an image from an image sequence, such that they can be tracked adequately further through the sequence. Usually, the tracking is performed by some sort of local search method looking for a similar patch in the next image in the sequence. Therefore, it would be useful if we could estimate “the size of the convergence region” for each image patch. There is a smaller chance of error when calculating the displacement for an image patch with a large convergence region than for an image patch with a small convergence region. Consequently, the size of the convergence region can be used as a proper goodness measure for a feature point. For the standard Kanade-Lucas-Tomasi (KLT) tracking method, we propose a simple and fast way to approximate the convergence region for an image patch. In the experimental part, we test our hypothesis on a large set of real data.

**Keywords** Feature (interest) point selection · Motion estimation · Visual tracking · Optical flow · Convergence region · Robustness

### Introduction

The term “feature point” denotes a point in an image that is sufficiently different from its neighbours (for

example, L-corner, T-junction, a white dot on black background, etc.). A feature point has a well defined position and this is useful in many applications [17]. An important example is the simple “optical flow” problem [10, 2] where the task is to find, for a feature point from one image, the corresponding point in the next image in a sequence. Usually, it is assumed that some small neighbourhood is also moving together with the point and, therefore, a small image patch around the point can be considered. When the displacements are small, the Kanade-Lucas-Tomasi (KLT) algorithm [15, 22] is commonly used to search for a similar patch in the next image. Furthermore, it is often useful to track the feature points further through the sequence. The positions of the tracked feature points are used, for example, in people motion or in the “structure and motion” algorithms [22, 1, 9]. There are various ways for detecting the errors that occur during tracking. The task of “monitoring”, i.e. checking whether the points found from a sequence still look similar to the original feature point, is discussed in [21, 18] and elaborated further in [6, 11]. Furthermore, the false measurements can also be detected on a higher level of the processing chain, for example, when the measurements are combined into 3D structure and motion estimates (see [9]).

The problem considered in this paper is how to select the feature points from the initial image that are less likely to lead to false measurements (and are, therefore, suitable for tracking). Feature point selection strategies have been previously analysed and evaluated many times [17]. However, selection in the tracking context was not often analysed previously. In [23], there is a tracking evaluation experiment for several corner detectors. In [21], the Harris corner operator [8] is analysed in conjunction with the accuracy of the matching (summarised in Sect. 2). Standard feature point operators (usually corner detectors) give a numerical value, the so-called interest response (IR), at a pixel location based on the intensity values from the local image neighbourhood. The points with high IR are the possible feature point candidates. The IR of the

---

Z. Živković (✉) · F. van der Heijden (✉)  
Laboratory for Measurement and Instrumentation,  
University of Twente, PO Box 217, 7500AE Enschede,  
The Netherlands  
E-mail: zivkovic@science.uva.nl  
Tel.: +31-20-5257564  
Fax: +31-20-5257490  
E-mail: f.vanderheijden@utwente.nl  
Tel.: +31-53-4892790  
Fax: +31-53-4891067

*Present address:* Z. Živković  
University of Amsterdam, Kruislaan 403,  
1098SJ Amsterdam, The Netherlands

standard feature point detectors is related to the accuracy of the matching. However, tracking also involves some other factors. The practical tracking is performed by some sort of local search which might not converge to the correct solution. Furthermore, similar structures in the neighbourhood can lead to mismatching that is hard to detect. With larger movements in the image (low temporal sampling), we can expect the mismatching problem to occur often. We propose an additional goodness measure, “the size of the convergence region” (SCR), for the selected points, which can help to identify and discard the point candidates that are likely to be unreliable. In Sect. 3, for the KLT tracker, we propose a simple method for estimating the SCR for a feature point (initially presented in [25]). We show how this can improve the standard feature point detectors. We use two common, simple and fast corner detectors: the Harris corner operator (many times evaluated to be the best) and the recent, often used SUSAN corner detector [19] (which is based on quite different principles). For the selected corners, we estimate the SCR and show that the points with small SCR are usually the points that are erroneously tracked. For evaluation purposes, we use a large set of data with ground truth. In Sect. 5.2, we discuss how simple image blurring can also be used to avoid textured regions, which is similar to our approach, and compare this to our method. In this paper, we consider the KLT tracker. However, the main idea could be useful, if appropriately applied, for numerous other tracking/matching schemes.

## Image motion

The simplest and most often used approach for calculating the movement of a small image patch from an image,  $I_0$ , is to search the next image,  $I_1$ , for a patch that minimises the sum of squared differences [10, 2]:

$$J(\vec{d}) = \iint_W \left[ I_1(\vec{x}_{\text{im}}) - I_0(\vec{x}_{\text{im}} + \vec{d}) \right]^2 d\vec{x}_{\text{im}} \quad (1)$$

where  $W$  is the window of the feature (interest) point under consideration,  $\vec{x}_{\text{im}} = [x_{\text{im}} \ y_{\text{im}}]^T$  presents the 2D position in the image plane and  $\vec{d}$  is the displacement between the two frames. In practice, the integration simply denotes the summing over all of the image pixels within the patch.

If we use a truncated Taylor expansion approximation in Eq. 1, we can find a  $\vec{d}$  that minimises the sum of squared differences by solving:

$$Z\vec{d} = \vec{e}, \text{ with} \quad (2)$$

$$Z = \iint_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} d\vec{x}_{\text{im}} \text{ and} \quad (3)$$

$$\vec{e} = \iint_W (I_0 - I_1) [g_x \ g_y]^T d\vec{x}_{\text{im}} \quad (4)$$

Here,  $g_x(\vec{x}_{\text{im}})$  and  $g_y(\vec{x}_{\text{im}})$  are the derivatives of  $I_0$  in the  $x_{\text{im}}$  and  $y_{\text{im}}$  directions at the image point  $\vec{x}_{\text{im}}$ . The dependence on  $\vec{x}_{\text{im}}$  is left out for simplicity.

The Lucas-Kanade procedure [15, 14] minimises Eq. 1 iteratively. The solution of the linearised system in Eq. 2 is used to warp the new image,  $I_1$ , and the procedure is repeated. This can be written as:

$$\vec{d}(k+1) = \vec{d}(k) + Z^{-1}\vec{e}(k), \text{ with } \vec{d}(0) = 0 \quad (5)$$

where  $\vec{d}(k)$  represents the estimated displacement at the  $k$ -th iteration. Equation 4 with the image  $I_1$  warped using  $\vec{d}(k)$  gives us  $\vec{e}(k)$  (linear interpolation is usually used). The algorithm described is the *Gauss-Newton minimisation procedure* (see [5], Chapter 6).

The image derivatives and the matrix  $Z$  are calculated only once [7]. System 2 is solved in each iteration using the same matrix  $Z$ . Therefore, the matrix  $Z$  should be both above the noise level and well conditioned. This means that the eigenvalues,  $\lambda_1$  and  $\lambda_2$ , of  $Z$  should be large and they should not differ by several orders of magnitude. Since the pixels have a maximum value, the greater eigenvalue is bounded. In conclusion, an image patch can be accepted if, for some predefined  $\lambda$ , we have:

$$\text{IR}_{\text{Harris}} = \min(\lambda_1, \lambda_2) > \lambda \quad (6)$$

We use the presented formulation as in [21]. The approximation,  $|Z| - \alpha \text{trace}(Z)^2$ , from [8] is avoided because of the additional parameter,  $\alpha$ . The described point selection and tracking is known as the KLT-tracker.

## Estimating the convergence region

We denote the true displacement by  $\vec{d}^*$  and define  $\vec{x}(k) = \vec{d}^* - \vec{d}(k)$ . In the ideal case (no noise and no deformations), the minimised function 1 can be locally approximated by  $J(\vec{x}) \approx \vec{x}^T Z \vec{x}$ . This is another way to interpret the Harris operator given by Eq. 6. Here, we introduce the notion of “the convergence region” for a selected point, which is more global in nature.

The iteration in Eq. 5 can be rewritten as:

$$\vec{x}(k+1) = \vec{x}(k) - Z^{-1}\vec{e}, \text{ with } \vec{x}(0) = \vec{d}^* \quad (7)$$

First, we define  $V(\vec{x}) = \|\vec{x}\|$  and successful tracking would mean that:

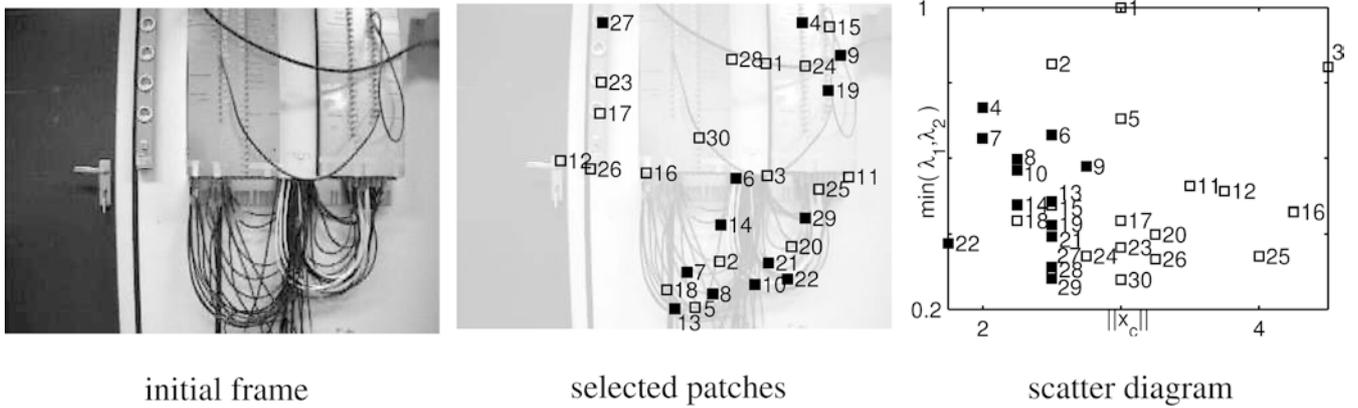
$$V(\vec{x}(k)) = \|\vec{x}(k)\| \rightarrow 0 \text{ for } k \rightarrow \infty \quad (8)$$

The convergence region is the domain where, for each initial displacement  $\vec{x}$ , the tracking process converges. The size of this region would be an appropriate criterion to define how well the feature point could be tracked.

Suppose that we can find a domain,  $S$ , with the following properties:

$$\forall \vec{x}(k) \in S, \dot{V}(\vec{x}(k)) < 0 \text{ and } \vec{x}(k+1) \in S \quad (9)$$

with  $\dot{V}(\vec{x}(k)) = V(\vec{x}(k+1)) - V(\vec{x}(k))$ . Convergence is guaranteed within  $S$  since what we state is simply that



**Fig. 1** An illustrative experiment; 30 points are selected and tracked.  $IR_{Harris}$  and the new  $IR_{SCR}$  for the points is presented in the *scatter diagram* (the *black squares* are the erroneously tracked points)

we want to always move closer to the solution. Our function,  $V(\vec{x})$ , is symmetric and monotonously increasing with  $\|\vec{x}\|$ . If we find the point  $\vec{x}_C$  closest to the origin for which  $\dot{V}(\vec{x}_C) \geq 0$ , then the region  $\|\vec{x}\| < \|\vec{x}_C\|$  will have the mentioned properties. The distance  $\|\vec{x}_C\|$  can be used to describe the size of the estimated convergence region and, consequently, it is a proper feature point goodness measure, denoted further as  $IR_{SCR}$ .

In Fig. 1, we present an illustrative example. We selected 30 “corner-like” feature points (7×7-pixel image patches). After a circular camera movement, some of the feature points were erroneously tracked (black boxes). From the scatter diagram, we observe that the radius of the estimated convergence region ( $x$ -axis,  $IR_{SCR}$  in pixels) discriminates the well tracked and the lost feature points. We also see that the smaller eigenvalue does not carry this information ( $y$ -axis, relative  $IR_{Harris}$  value with respect to the largest).

The theory presented here is inspired by the non-linear system analysis methods [24] and, in this sense,  $V(\vec{x})$  corresponds to the *Lyapunov function*. The function,  $V$ , and its derivative,  $\dot{V}$ , are highly non-linear and depend on the local neighbourhood of a feature point. An example is given in Fig. 2. The function  $\dot{V}$  is pre-

sented using a 0.5-pixel grid. The circle presents the estimated convergence region.

### Implementation

In the practical implementation, for each feature point, we compute  $\dot{V}(\vec{x})$  for some discrete displacements around the feature point until we find the first  $\dot{V}(\vec{x}) \geq 0$ . We can use the following algorithm:

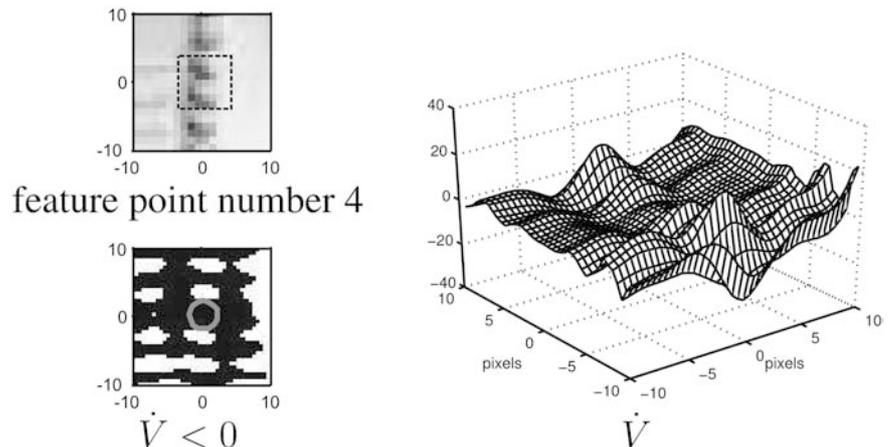
**Input:**  $I_0, g_x, g_y, W, Z^{-1}, SS$  (an array of 2D displacements,  $\vec{d}$ , with non-decreasing  $\|\vec{d}\|$ ; we use eight points (angular sampling every 45°) on concentric circles with radii increasing in 0.5-pixel steps starting from the initial 0.5-pixel radius)

1.  $\vec{x}(0) = (\vec{d}^* =) SS(i)$  (initially  $i=0$ )
2. Calculate  $\vec{e}$  (window  $W$  from  $I_1$  simulated using  $W$  shifted for  $\vec{d}^*$  from  $I_0$ )
3.  $\vec{x}(1) = \vec{x}(0) - Z^{-1}\vec{e}$  (one Lucas-Kanade iteration step)
4. If  $\|\vec{x}(1)\| \geq \|\vec{x}(0)\|$  (equivalent to  $\dot{V} \geq 0$ ) return  $\|\vec{x}_C\| = \|\vec{x}(0)\|$  else  $\{i=i+1; \text{go to 1}\}$

**Output:**  $IR_{SCR} = \|\vec{x}_C\|$

The computational cost for a feature point is comparable to the computations needed for calculating the movement of the point. In our case, the average number of iterations (that are similar to the KLT iterations) is

**Fig. 2** Feature point 4 (zoomed in), function  $\dot{V}$  in the neighbourhood of the point and the estimated SCR (smallest circular area where  $\dot{V} < 0$ )



$8 \times 2 \times \text{average } \|\vec{x}_C\|$ . The 8 is due to the  $45^\circ$  angular sampling step and the 2 is because of the 0.5-pixel sampling step. Increasing the number of angular or radial sampling steps does not lead to significant changes in the results we present in the next section (Sect. 5) while decreasing the number of sampling steps degrades the results. Furthermore, in our experiments, the algorithm was modified to stop when we find  $\dot{V} \geq 0$  for the third time (step 4 from above is modified) and for  $\text{IR}_{\text{SCR}}$ , we used the average of the three distances,  $\|\vec{x}_C\|$ . This leads to some improvement since isolated points having  $\dot{V} \geq 0$  are suppressed.

## Experiments

The initial frames from the image sequences we used are presented in Fig. 3. The sequences are from the CMU VASC Image Database. The “marbled-block” sequence used in [16] is added (complex motion; both camera and an object are moving). The sequences exhibit a variety of camera movements, object textures and scene depth variations. We used 2,143 “corner-like” points for tracking. The chosen sequences have very small displacements between the consecutive frames, therefore, it was possible to track the feature points ( $7 \times 7$ -pixel patches) for some short time. This was used as the ground truth. To generate more difficult situations and introduce some errors, we start again from the initial frame and use the KLT tracker with 20 fixed iterations to calculate the displacements between the initial and  $i$ -th frame in the sequence (skipping the frames in between). We choose  $i$  so that, for each sequence, the displacement is erroneously calculated for at least 20% of the feature points.

### Improving the Harris and SUSAN corner selection

First, we selected “corner-like” points having  $\text{IR}_{\text{Harris}} > 0.05$ . From the initial 2,143 points, 754 lead to false measurements. We selected the same number of feature points using the SUSAN corner detector and got 876 “bad” points. The worse performance of the SUSAN detector in the tracking context is in agreement with [23]. In our experiments, we use the  $3 \times 3$  Sobel operator for the image derivatives. For the SUSAN corner detector, we use the usual 3.5-pixel-radius circular neighbourhood for the feature points (giving a mask,  $W$ , containing 37 pixels). If  $I_C$  is the intensity value at the centre pixel, the response function is  $\text{IR}_{\text{SUSAN}} = 37/2 - \sum_W \exp(-(|\vec{x}_{im} - I_C|/t)^6)$ . Any negative values are discarded. For additional details, see [19]. For our data, we have empirically chosen  $t = 15$ . For both SUSAN and Harris detectors, the feature points are the local maxima but constrained to be at least a distance of 15 pixels from each other.

During the selection, we need to set a threshold and discard the feature point candidates having IR below the threshold. If we plot the results for different thresholds,

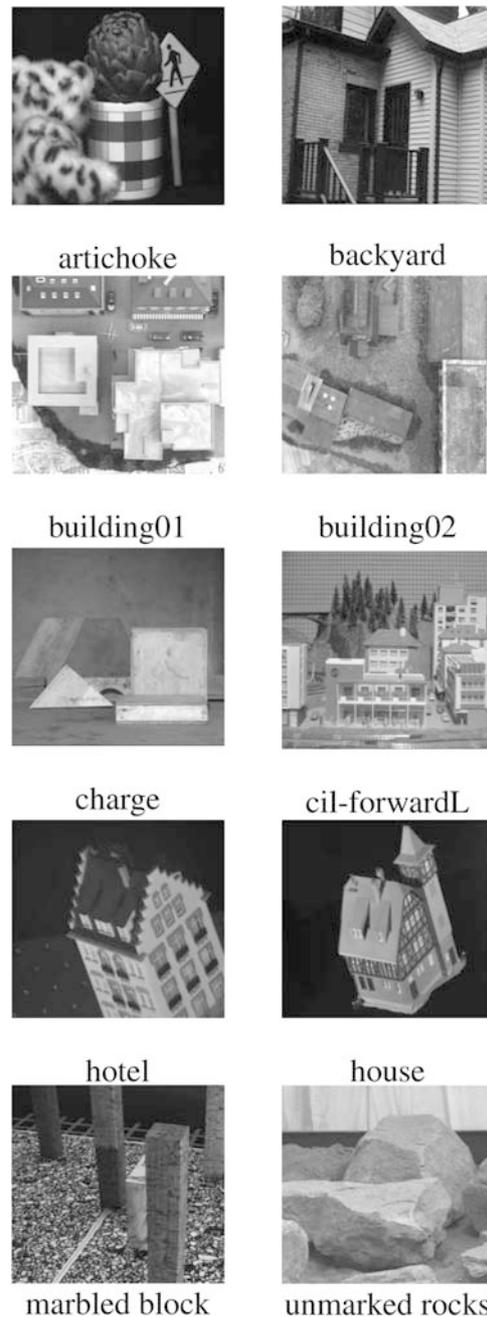
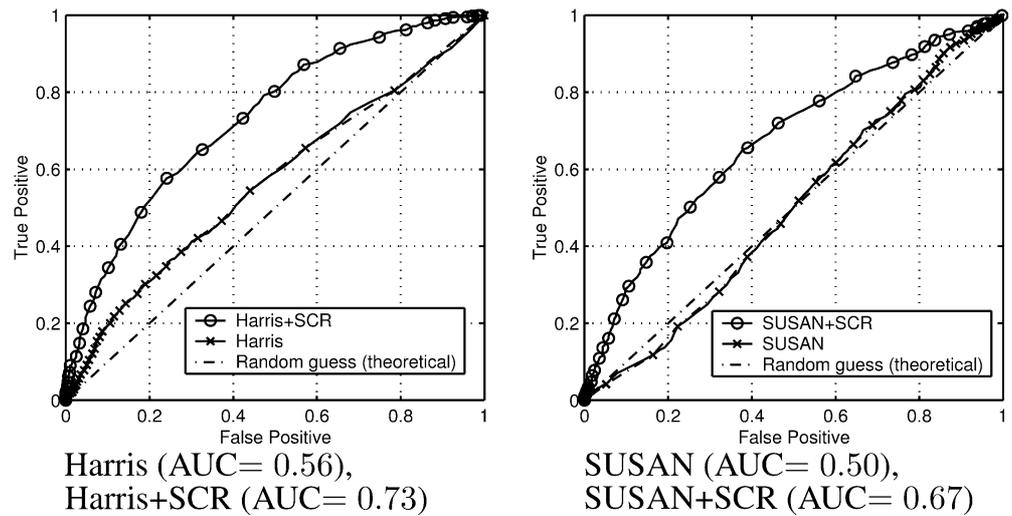


Fig. 3 Image sequences used in our experiments

we get a “receiver operator characteristic” (ROC) curve that shows the discriminative power of the IR. For our data set that contains 2,143 feature points with the ground truth, we plot the empirical ROC curves (linear interpolation is used between the points on the curve). A feature point belongs to the true-positives if it was selected and it was well tracked. The false-positives are the points selected but lost during tracking. Relative values are used; we divide by the total number of the well tracked and the “bad” ones, respectively. The ROC curves in Fig. 4 show the large improvements when the new  $\text{IR}_{\text{SCR}}$  is used. A widely accepted comparison method is to use the

**Fig. 4** Improving Harris and SUSAN corner detectors using SCR. The ROC curves and the areas under the ROC curves (AUC) are reported



**Table 1** A summary of the areas under the ROC curve and the standard errors

Corner selection + additional check	AUC (standard error)
Harris + (SCR + $\log(\text{Harris}(s=2.5))$ )	0.77 (0.010)
Harris + SCR	0.73 (0.011)
Harris + Harris( $s=2.5$ )	0.72 (0.011)
Harris + Harris( $s=3.5$ )	0.70 (0.011)
SUSAN + SCR	0.67 (0.011)
Harris + Harris( $s=1.0$ )	0.63 (0.012)
Harris( $s=1.5$ )	0.58 (0.013)
Harris	0.56 (0.013)
SUSAN	0.50 (0.013)

area under the ROC curve (AUC) [4]. A summary of all of the experiments is given in Table 1. We also calculate the standard error for the AUC measure by using a Gaussian distribution assumption, as in [4].

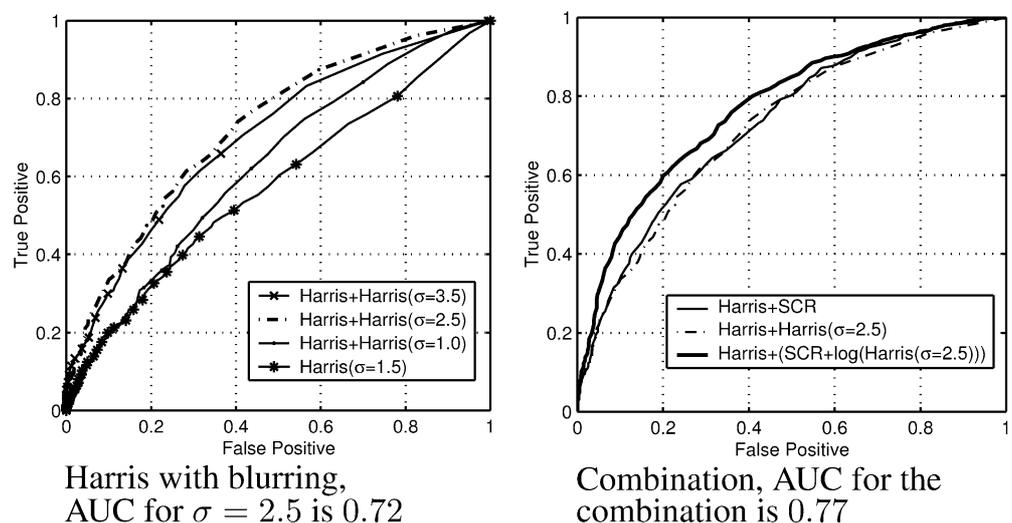
#### Feature point selection and blurring

Tracking errors can occur when points are selected in a textured area of an image. This is properly detected

using the SCR. See, for example, the experiment from Sect. 3 (Figs. 1 and 2). Another simple method for detecting and discarding these kinds of points is to blur the image (convolve with a Gaussian kernel with standard deviation  $\sigma$ ) so that only the isolated strong corners remain. Note that the positions of the detected corners (local maxima) are different in a blurred image [12], therefore, the corners need to be detected in the original image where the tracking is actually done. Then, we can blur the image and, for each of the initially detected points, we can define an additional goodness measure,  $\text{IR}_{\text{Harris}(\sigma)}$  (Harris with blurring). The improvement with this additional measure for different  $\sigma$ 's is presented in Fig. 5 by plotting the ROC curves for our data set. The optimal result for our data set was achieved by using  $\sigma=2.5$ , and it is similar to the result when using SCR. Table 1 presents a summary of the results.

Although the performance of  $\text{IR}_{\text{Harris}(\sigma)}$  for correctly chosen  $\sigma$  is similar to  $\text{IR}_{\text{SCR}}$ , they are inherently different measures. Figure 5 presents further improvement for the empirical combination,  $\text{IR}_{\text{SCR}} + \log(\text{IR}_{\text{Harris}(\sigma=2.5)})$ . The improvement can be considered as proof that the two

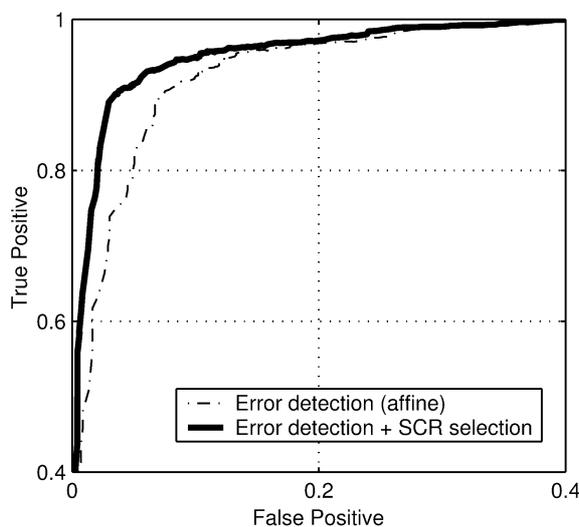
**Fig. 5** Improving feature selection with additional image blurring. The ROC curves and the areas under the ROC curves (AUC) are reported



measures describe different effects. Finding the optimal combination is beyond the scope of this paper. Furthermore, the SCR measure has no parameters and it is clearly related to tracking, while blurring is rather ad-hoc and we need to choose an appropriate  $\sigma$ . Another ad-hoc idea would be to selectively scale for each point, as discussed in [13]. The corners at larger scales are usually isolated corners. This ad-hoc procedure would be very slow and is not considered here.

### Error detection and improved feature point selection

Finally in this section, we show the influence of the improved point selection on practical KLT tracking. During tracking, the points are “monitored” in order to detect possible tracking errors. If the KLT tracker has found a point that is not similar enough to the initially selected point, we assume that an error has occurred and stop tracking this point. The erroneously tracked points are the points that have the resulting similarity measure (Eq. 1) above a certain error detection threshold. In [18], the affine transformation of the initial feature point appearance is used to improve the comparison. In Fig. 6, we show the error detection ROC curve for the affine comparison after the first new frame of the image sequences. We observe that, already after one frame, the detection is not perfect. There are erroneously tracked points that are difficult to detect because the KLT search finds some other similar structures. If the similar structures are close to the correct feature point position, even some higher level error detection is likely to fail (for example, some “smoothness” constraints or the 3D scene constraints used in the “structure and motion” algorithms [9]). In Fig. 6, we



**Fig. 6** Error detection (affine) and influence of the improved feature point selection (SCR), error detection AUC=0.948 (standard error 0.004), error detection with improved point selection AUC=0.972 (standard error 0.003)

also present the ROC curve for the KLT tracker with the improved point selection. The ROC curve presents a summary for all possible values of the error detection threshold and all possible values of the threshold for the improved feature selection using  $IR_{SCR}$ . We observe a significant improvement already after the first frame.

## Conclusions

The problem of estimating the motion of a feature point has two aspects: the accuracy of the result and the convergence of the tracker. The accuracy is well addressed by the standard feature point detectors. The corner-like points can be accurately matched; the Harris corner operator is a nice example. A well conditioned matrix  $Z$  from Eq. 2 assures low sensitivity to the noise but only if the tracking converges. We analysed in this paper the problems with convergence of the tracker. Our new goodness measure is an estimate of the convergence region. The new measure can be used as an additional check to improve the selection of the points for tracking. Significant improvements are possible, as we demonstrated on a large data set.

## Originality and contributions

The term “feature point” denotes a point in an image that is sufficiently different from its neighbours (L-corner, T-junction, a white dot on black background, etc.). The position of a feature point is well defined and this is useful for the tracking/matching problem where the task is to find, for a feature point from one image, the corresponding point in the next image in a sequence. When the displacements are small, the Kanade-Lucas-Tomasi (KLT) algorithm is often used for tracking. There are various techniques that can be applied to detect the errors that occur during tracking. This paper presents an analysis of the problem of initial feature point selection in the tracking context. We point out that the standard feature point criteria are more concerned with how accurate the feature point tracking will be, rather than how robust the tracking will be. We present a simple method for improving the initial selection of the feature points to reduce the number of possible errors during tracking and thereby ease the demand on the algorithms that further process the positions of the tracked points (for example, RANSAC in the “structure and motion” algorithms, etc.). The method is evaluated and compared to two standard feature point selection methods on a large set of real data. We also analyse how blurring the images can improve feature selection. We compare this scale-space approach to our method. The results are presented for the common KLT tracker but the main idea could be useful, if appropriately applied, for numerous other tracking/matching schemes.

---

## About the authors

Zoran Živković graduated in 1999 at the University of Belgrade, Serbia, and received his PhD in 2003 from the University of Twente, The Netherlands. He worked as research assistant in the Group for Measurement and Instrumentation, University of Twente from 1999. Since 2003, he has worked as a post-doctoral researcher at the Intelligent Autonomous Systems Group, University of Amsterdam, Amsterdam, The Netherlands. His current research interests are computer vision and artificial intelligence.

Ferdinand van der Heijden received his MSc degree in 1981 and his PhD in 1992, both from the University of Twente, The Netherlands. In 1982, he joined the Group for Measurement and Instrumentation, Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, where he is currently working as a faculty member. His current research interests are in image-based measurement systems, including video tracking and optimal estimation.

---

## References

1. Azarbayejani A, Pentland A (1995) Recursive estimation of motion, structure, and focal length. *IEEE T Pattern Anal* 17(6):562–575
2. Beauchemin SS, Barron JL (1995) The computation of optical flow. *ACM Comput Surv* 27(3):433–467
3. Beymer D, McLauchlan P, Coifman B, Malik J (1997) A real-time computer vision system for measuring traffic parameters. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, San Juan, Puerto Rico, June 1997, pp 495–501
4. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn* 30(7):1145–1159
5. Fletcher R (1987) *Practical methods of optimization*. John Wiley and Sons, New York
6. Fusiello A, Trucco E, Tommasini T, Roberto V (1999) Improving feature tracking with robust statistics. *Pattern Anal Appl* 2(4):312–320
7. Hager GD, Belhumeur PN (1998) Efficient region tracking with parametric models of geometry and illumination. *IEEE T Pattern Anal* 20(10):1025–1039
8. Harris C, Stephens M (1988) A combined corner and edge detector. In: *Proceedings of the 4th Alvey vision conference*, Manchester, England, August/September 1998, pp 147–151
9. Hartley R, Zisserman A (2000) *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge
10. Horn BKP, Schunck BG (1993) Determining optical flow: A retrospective. *Artif Intell* 59(1–2):81–87
11. Jin H, Favaro P, Soatto S (2001) Real-time feature tracking and outlier rejection with changes in illumination. In: *Proceedings of the 8th IEEE international conference on computer vision*, Vancouver, Canada, July 2001, pp 684–689
12. Kuijper A, Florack LMJ (2002) Understanding and modeling the evolution of critical points under Gaussian blurring. In: *Proceedings of the 7th European conference on computer vision*, Copenhagen, Denmark, May/June 2002, pp 143–157
13. Lindeberg T (1984) Feature detection with automatic scale selection. *Int J Computer Vision* 30(2):77–116
14. Lucas BD (1984) Generalized image matching by the method of differences. PhD thesis, Dept. of Computer Science, Carnegie-Mellon University
15. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: *Proceedings of the international joint conference on artificial intelligence*, Vancouver, Canada, August 1981, pp 674–679
16. Otte M, Nagel HH (1995) Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artif Intell* 78(1/2):5–43
17. Schmid C, Mohr R, Bauckhage C (2000) Evaluation of interest point detectors. *Int J Computer Vision* 37(2):151–172
18. Shi J, Tomasi C (1994) Good features to track. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Seattle, Washington, June 1994, pp 593–600
19. Smith SM, Brady JM (1997) SUSAN – A new approach to low level image processing. *Int J Computer Vision* 23(1):45–78
20. Song Y, Concalves L, Perona P (2003) Unsupervised learning of human motion. *IEEE T Pattern Anal* 25(7):814–827
21. Tomasi C, Kanade T (1991) Detection and tracking of point features. Carnegie Mellon University Technical Report CMU-CS-91-132
22. Tomasi C, Kanade T (1992) Shape and motion from image streams under orthography: A factorization approach. *Int J Computer Vision* 9(2):137–154
23. Trajkovic M, Hedley M (1998) Fast corner detection. *Image Vision Comput* 16:75–87
24. Vidyasagar M (1993) *Nonlinear systems analysis*. Prentice Hall, Upper Saddle River, New Jersey
25. Živković Z, van der Heijden F (2002) Better features to track by estimating the tracking convergence region. In: *Proceedings of the international conference on pattern recognition*, Quebec, Canada, August 2002