In Pattern Analysis and Applications, vol.7, no.2, 2004, Copyright Springer-Verlag London Limited.

Improving the selection of feature points for tracking

Zoran Živković (corresponding author)

is with the Informatics Institute,

University of Amsterdam, Kruislaan 403, 1098SJ Amsterdam, the Netherlands.

Phone: +31-20-5257564, Fax:+31-20-5257490, E-mail: zivkovic@science.uva.nl

(the work was done while the author was with the Laboratory for Measurement

and Instrumentation, University of Twente, Enschede, the Netherlands)

Ferdinand van der Heijden

is with the Laboratory for Measurement and Instrumentation,

University of Twente, PO BOX 217, 7500AE Enschede, the Netherlands.

Phone: +31-53-4892790, Fax:+31-53-4891067, E-mail: f.vanderheijden@utwente.nl

July 26, 2004

Abstract

The problem considered here 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 well further through the sequence. Usually, tracking is performed by some sort of local search methods searching for a similar patch in the next image from the sequence. Therefore, it would be useful if we could estimate 'the size of the convergence region' for each image patch. It is less likely to erroneously calculate 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 method 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

Originality and contribution:

The term ”feature point” denotes a point in an image that is sufficiently different from its neighbors (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 from a sequence. When the displacements are small, the Kanade-Lucas-Tomasi (KLT) algorithm is often used for tracking. There are various techniques 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 tacking 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 analyze how blurring the images can improve feature selection as well. 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.

1 Introduction

The term ”feature point” denotes a point in an image that is sufficiently different from its neighbors (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 from a sequence. Usually it is assumed that some small neighborhood 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 KLT algorithm [15, 22] is commonly used to search for a similar patch from 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 [3, 20] 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 from a sequence that are found still look similar to the original feature point, is discussed in [21, 18] and further elaborated 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 (therefore suitable for tracking). Feature point selection strategies are analyzed and evaluated many times [17]. However, the selection in the tracking context was not often analyzed previously. In [23] there is a tracking evaluation experiment for a few corner detectors. In [21] the Harris corner operator [8] is analyzed in connection with the accuracy of the matching (summarized in section 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 neighborhood. The points with high IR are the possible feature point candidates. The IR of the standard feature point detectors is related to the accuracy of the matching. However, tracking involves also 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 neighborhood 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 section 3, for the KLT tracker we propose a simple method for estimating the SCR for a feature point (initially presented in []). 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 the best) and the recently 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 we use a large set of data with ground truth. In section 5.2, we discuss how simple image blurring can also be used to avoid textured regions similarly to our approach and compare this to our method. In this paper we considered the KLT tracker. However, the main idea could be useful, if appropriately applied, for numerous other tracking/matching schemes.

2 Image Motion

The simplest and 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 minimizes the sum of squared differences [10, 2]:

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

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

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

$$Z\vec{d} = \overrightarrow{e}, \text{ with} \tag{2}$$

$$Z = \iint_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} d\vec{x}_{im} \tag{3}$$

$$\overrightarrow{e} = \iint_W (I_0 - I_1) \begin{bmatrix} g_x & g_y \end{bmatrix}^T d\vec{x}_{im} \tag{4}$$

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

The Lucas-Kanade procedure [15, 14] minimizes (1) iteratively. The solution of the linearized system (2) is used to warp the new image I_1 and the procedure is repeated. This can be written as:

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

where $\vec{d}(k)$ presents the estimated displacement at the k-th iteration. Equation (4) with the image I_1 warped using $\vec{d}(k)$ gives us $\overrightarrow{e}(k)$ (linear interpolation is usually used). The described algorithm is the *Gauss-Newton minimization 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 λ_1, λ_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 λ we have:

$$IR_{Harris} = \min(\lambda_1, \lambda_2) > \lambda \tag{6}$$

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 α. Described point selection and tracking is known as KLT-tracker.

3 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 minimized 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 (6). Here we introduce the notion of 'the convergence region' for a selected point, which is more global in nature.

The iteration equation (5) can be rewritten as:

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

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

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

The convergence region is the domain where for each initial displacement $\vec{x}(0)$ 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:

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

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 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 $\dot{V}(\vec{x}_c) \geqslant 0$, 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 figure 1 we present an illustrative example. We selected 30 'corner-like' feature points (7×7 pixels 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 nonlinear system analysis methods [24] and in this sense $V(\vec{x})$ corresponds to *the Lyapunov function*. The function V and the derivative $\dot{V}$ are highly non-linear and depend on the local neighborhood of a feature point. An example is given in figure 2. The function $\dot{V}$ is presented using a 0.5 pixel grid. The circle presents the estimated convergence region.

4 Implementation

In practical implementation, for each feature point we compute $\dot{V}(\vec{x})$ for some discrete displacements around the feature point till we find the first $\dot{V}(\vec{x}) \geqslant 0$:

Input: I_0, g_x, g_y, W , Z^{-1}, SS (an array of 2D displacements $\vec{d}$ with non-decreasing $\left\|\vec{d}\right\|$- we use 8 points (angular sampling every 45^o) on concentric circles with radiuses increasing in 0.5 pixel steps starting from initial 0.5 pixel radius)

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

   else $\{i = i + 1$; go to $1\}$

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

The computational cost for a feature point is comparable with 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 8(because of every 45^o)$\cdot 2$(because of 0.5 pixel sampling steps)$\cdot$ average$\|\vec{x}_c\|$. Increasing the number of the angular or the radial sampling steps does not lead to significant changes in the results we present in the next section while decreasing the number of sampling steps degrades the results. Further, in our experiments the algorithm was modified to stop when we find $\dot{V} \geqslant 0$ for the third time (step 4 from above is modified) and for IR_{SCR} we used the average of the three distances $\|\vec{x}_c\|$. This leads to some improvement since isolated points having $\dot{V} \geqslant 0$ are suppressed.

5 Experiments

The initial frames from the image sequences we used are presented in figure 5. The sequences are from the CMU VASC Image Database. The sequence "marbled-block" 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 2143 '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 (7x7 pixel patches) for some short time. This was used as the ground truth. To generate more difficult situations and some errors we start again from the initial frame and use the KLT tracker with fixed 20 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 per sequence for at least 20% of the feature points the displacement is erroneously calculated.

5.1 Improving the Harris and SUSAN corner selection

First we selected 'corner-like' points having $IR_{Harris} > 0.05$. From the initial 2143 points 754 lead to false measurements. We select the same number of feature points using the SUSAN corner detector and get 876 'bad' points. The worse performance of the SUSAN detector in the tracking context is in correspondence with [23]. In our experiments we use the 3×3 Sobel operator for the image derivatives. For the SUSAN corner detector we use usual 3.5 pixels radius circular neighborhood for the feature points (giving a mask W containing 37 pixels). If I_C is the intensity value at the center pixel the response function is $IR_{SUSAN} = 37/2 - \sum_W \exp(-(I(\vec{x}_{im}) - I_C)/t)^6)$. 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 a minimum of 15 pixel

distance from each other.

During the selection we need to set a threshold and discard the features point candidates having IR below the threshold. If we plot the results for different thresholds we get a 'receiver operator characteristic' (ROC) curve that shows the discriminative power of the IR. For our data set that contains 2143 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. Relative values are used - we divide by the total number of the well tracked and the 'bad' ones respectively. The ROC curves in figure 3 show the large improvements when the new IR_{SCR} is used. A widely accepted comparison method is to use the area under the ROC curve (AUC) [4]. Summary for all experiments is given in table 1. We also calculate the standard error for the AUC measure using a Gaussian distribution assumption as in [4].

5.2 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 section 3 (figures 1 and 2). Another simple method for detecting and discarding this kind of points is to blur the image (convolve with a Gaussian kernel with standard deviation σ) so that only 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 $IR_{Harris(\sigma)}$ (Harris with blurring). The improvement with this additional measure for different σ is presented in figure 4 by plotting the ROC curves for our data set. The optimal result for our data set was

achieved using $\sigma = 2.5$ and it is similar to the result using SCR. Table 1 presents a summary of the results.

Although the performance of the $IR_{Harris(\sigma)}$ for correctly chosen σ is similar to the IR_{SCR}, they are inherently different measures. Figure 4 presents further improvement for empirical combination $IR_{SCR} + \log(IR_{Harris(\sigma=2.5)})$. The improvement can be considered as a proof that the two measures describe different effects. Finding the optimal combination is beyond the scope of this paper. Further, the SCR measure has no parameters and it is clearly related to tracking while blurring is rather ad-hoc and we need to choose appropriate σ. Another ad-hoc idea would be to select 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 it is not considered here.

5.3 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 has the resulting similarity measure (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 figure 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 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 figure 6 we present also 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 significant improvement already after the first frame.

6 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 (2) assures low sensitivity to the noise but only if the tracking converges. We analyzed in this paper the problems with convergence of the tracker. Our new goodness measure is en 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.

References

[1] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 17(6), pages 562-575, 1995.

[2] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3), pages 433-467, September 1995.

[3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. *In Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 495-501, 1997.

[4] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), pages 1145-1159, 1997.

[5] R. Fletcher. *Practical Methods of Optimization*. J. Wiley, 1987.

[6] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2(4), pages 312-320, 1999.

[7] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10), pages 1025-1039, 1998.

[8] C. Harris and M. Stephens. A combined corner and edge detector. *In Proceedings of 4th Alvey Vision Conference*, pages 147-151, 1988.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow: a retrospective. *Artificial Intelligence*, 59(1-2), pages 81-87, January 1993.

[11] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. *In proceedings of ICCV*, pages 684-689, 2001.

[12] A. Kuijper and L.M.J. Florack. Understanding and modeling the evolution of critical points under gaussian blurring. *In Proceedings of the 7th European Conference on Computer Vision*, pages 143-157, 2002.

[13] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2), pages 77-116, 1998.

[14] B.D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD Thesis, Dept. of Computer Science, Carnegie-Mellon University, 1984.

[15] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *In Proceedings IJCAI81*, pages 674-679, 1981.

[16] Michael Otte and Hans-Hellmut Nagel. Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artificial Intelligence*, 78(1/2), pages 5-43, November 1995.

[17] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2), pages 151-172, 2000.

[18] J. Shi and C. Tomasi. Good features to track. *In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593-600, 1994.

[19] S.M. Smith and J.M. Brady. SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1), pages 45-78, 1997.

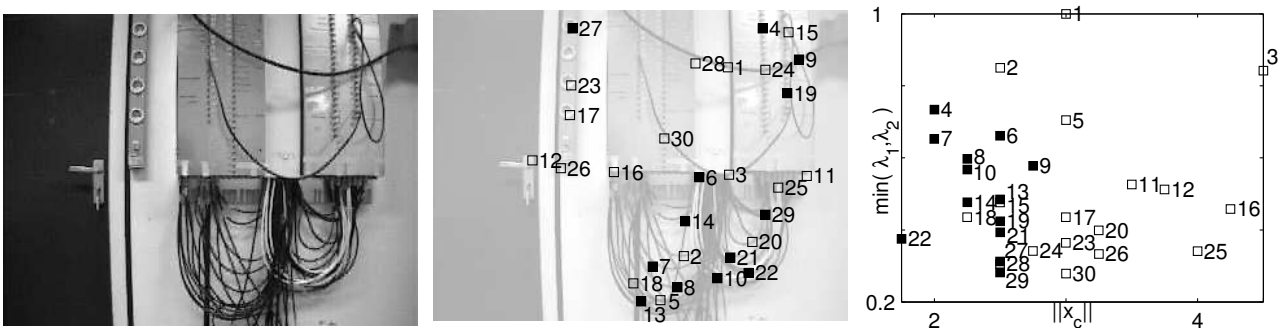
[20] Yang Song, Luis Concalves, and Pietro Perona. Unsupervised learning of human motion. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25(7), pages 814-827, July 2003.

[21] C. Tomasi and T. Kanade. Detection and tracking of point features. *Carnegie Mellon University Technical Report CMU-CS-91-132*, 1991.

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2), pages 137-154, 1992.

[23] M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16, pages 75-87, 1998.

[24] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, 1993.



initial frame selected patches scatter diagram

Figure 1: An illustrative experiment - 30 points are selected and tracked, the IR_{Harris} and the new IR_{SCR} for the points is presented in the scatter diagram (the black squares are the erroneously tracked points)

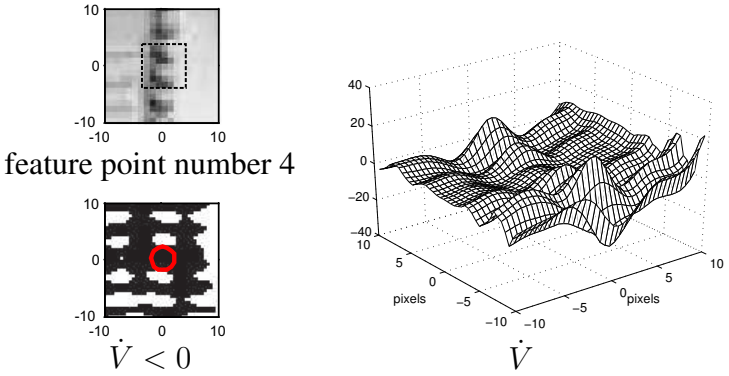
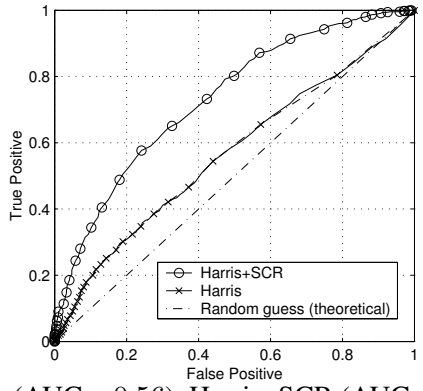
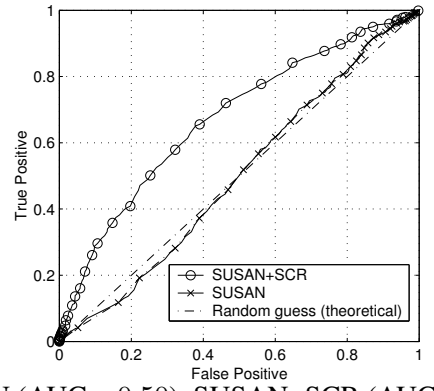


Figure 2: The feature point 4 (zoom in), function $\dot{V}$ in the neighborhood of the point and the estimated SCR (smallest circular area where $\dot{V} < 0$)

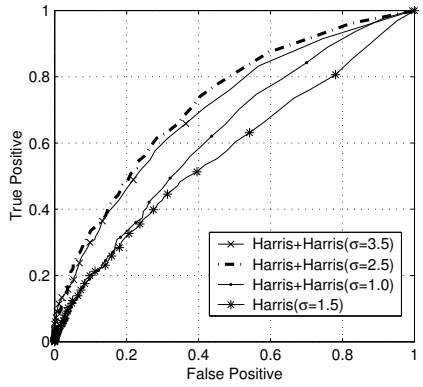


Harris (AUC= 0.56), Harris+SCR (AUC= 0.73)

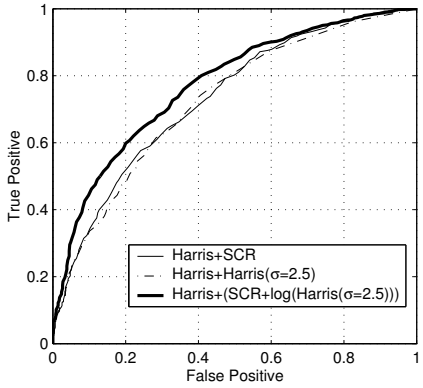


SUSAN (AUC= 0.50), SUSAN+SCR (AUC= 0.67)

Figure 3: Improving Harris and SUSAN corner detectors using SCR, ROC curves and the areas under the ROC curves (AUC) are reported.



Harris with blurring, AUC for $\sigma = 2.5$ is 0.72



Combination, AUC for the combination is 0.77

Figure 4: Improving feature selection with additional image blurring, ROC curves and the areas under the ROC curves (AUC) are reported.

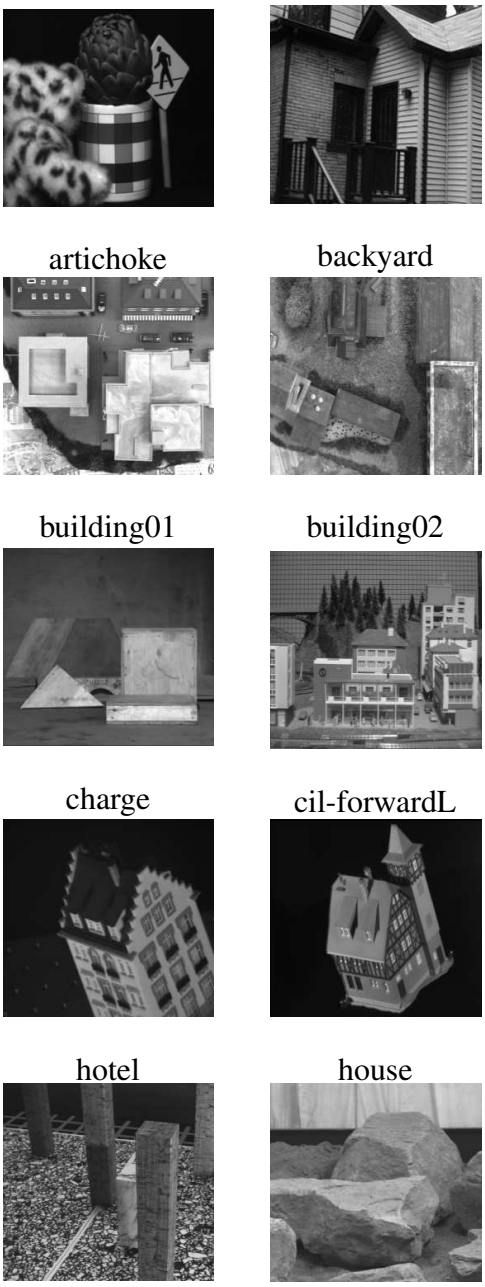


Figure 5: Image sequences

Corner Selection + Additional check	AUC (Standard Error)
Harris + (SCR + log(Harris($\sigma = 2.5$)))	$0.77(0.010)$
Harris + SCR	$0.73(0.011)$
Harris + Harris($\sigma = 2.5$)	$0.72(0.011)$
Harris + Harris($\sigma = 3.5$)	$0.70(0.011)$
SUSAN + SCR	$0.67(0.012)$
Harris + Harris($\sigma = 1.0$)	$0.63(0.012)$
Harris($\sigma = 1.5$)	$0.58(0.013)$
Harris	$0.56(0.013)$
SUSAN	$0.50(0.013)$

Table 1: Area under ROC curve and standard errors, a summary

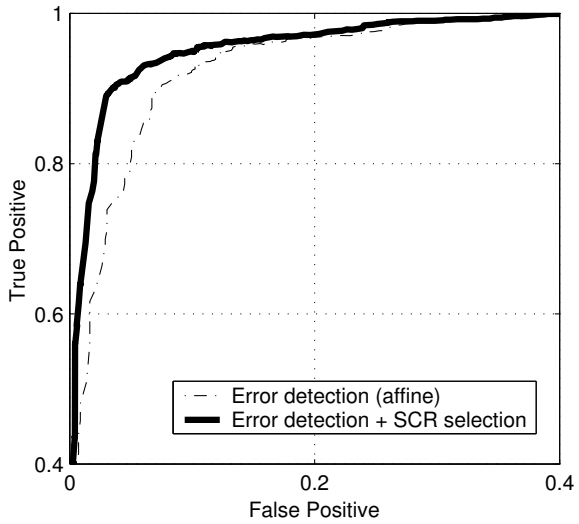


Figure 6: Error detection (affine) and influence of the improved feature point selection (SCR), error detection AUC=0.948(standard error 0.004), error detection with improved point selection AUC=0.972(standard error 0.003).