Making Good Features Track Better

Tiziano Tommasini[†], Andrea Fusiello[†],
([†]) Machine Vision Laboratory, Dept. Informatics
University of Udine, Italy
{tommasin,fusiello,roberto}@dimi.uniud.it

Emanuele Trucco[*] and Vito Roberto[†]
([*])Dept. of Computing and Electrical Eng.
Heriot-Watt University, UK
mtc@cee.hw.ac.uk

Abstract

This paper addresses robust feature tracking. We extend the well-known Shi-Tomasi-Kanade tracker by introducing an automatic scheme for rejecting spurious features. We employ a simple and efficient outlier rejection rule, called X84, and prove that its theoretical assumptions are satisfied in the feature tracking scenario. Experiments with real and synthetic images confirm that our algorithm makes good features track better; we show a quantitative example of the benefits introduced by the algorithm for the case of fundamental matrix estimation. The complete code of the robust tracker is available via ftp.

1. Introduction

Feature tracking is an important issue in computer vision, as many algorithms rely on the accurate computation of correspondences through a sequence of images [9, 13, 17]. When an image sequence is acquired and sampled at a sufficiently high time frequency, frame-to-frame disparities are small enough to make optical-flow techniques viable [1]. If frame-to-frame disparities are large (e.g., the images are taken from quite different viewpoints), stereo matching techniques [3] are used instead, often in combination with Kalman filtering [7, 10, 16]. *Robust tracking* means detecting automatically unreliable matches, or *outliers*, over an image sequence (see [8] for a survey of robust methods in computer vision). Recent examples of such robust algorithms include [15], which identifies tracking outliers while estimating the fundamental matrix, and [14], which adopts a RANSAC approach to eliminate outliers for estimating the trifocal tensor. Such approaches increase the computational cost of tracking significantly.

This paper concentrates on the well-known Shi-Tomasi-Kanade tracker, and proposes a robust version based on an efficient outlier rejection scheme. Building on results from [6], Tomasi and Kanade [12] introduced a feature tracker based on SSD matching and assuming translational frame-to-frame displacements. Subsequently, Shi and Tomasi [11] proposed an *affine model*, which proved adequate for region matching over longer time spans. Their system classified a tracked feature as *good* (reliable) or *bad* (unreliable) according to the residual of the match between the associated image region in the first and current frames; if the residual exceeded a user-defined threshold, the feature was rejected. Visual inspection of results demonstrated good discrimination between good and bad features, but the authors did not specify how to reject bad features *automatically*.

This is the problem that our paper solves. We extend the Shi-Tomasi-Kanade tracker (Section 2) by introducing an *automatic* scheme for rejecting spurious features. We employ a simple, efficient, model-free outlier rejection rule, called *X84*, and prove that its assumptions are satisfied in the feature tracking scenario (Section 3). Experiments with real and synthetic images confirm that our algorithm makes good features to track better, in the sense that outliers are located reliably (Section 4). We illustrate quantitatively the benefits introduced by the algorithm with the example of fundamental matrix estimation. he complete code of the robust tracker is available via ftp from: ftp://taras.dimi.uniud.it/pub/sources/rtrack.tar.gz.

2. The Shi-Tomasi-Kanade tracker

In this section the Shi-Tomasi-Kanade tracker [11, 12] will be briefly described. Consider an image sequence $I(\mathbf{x}, t)$, with $\mathbf{x} = [u, v]^\top$, the coordinates of an image point. If the time sampling frequency is sufficiently high, we can assume that small image regions are displaced but their intensities remain unchanged:

$$I(\mathbf{x}, t) = I(\delta(\mathbf{x}), t + \tau), \tag{1}$$

where $\delta(\cdot)$ is the *motion field*, specifying the *warping* that is applied to image points. The fast-sampling hypothesis allows us to approximate the motion with a translation, that is, $\delta(\mathbf{x}) = \mathbf{x} + \mathbf{d}$, where $\mathbf{d}$ is a displacement vector. The tracker's task is to compute $\mathbf{d}$ for a number of selected points for each pair of successive frames in the sequence.

As the image motion model is not perfect, and because of image noise, Eq. (1) is not satisfied exactly. The problem is then finding the displacement $\hat{\mathbf{d}}$ which minimizes the SSD residual:

$$\epsilon = \sum_{\mathcal{W}} \left[I(\mathbf{x} + \mathbf{d}, t + \tau) - I(\mathbf{x}, t) \right]^2 \tag{2}$$

where $\mathcal{W}$ is a small image window centered on the point for which $\mathbf{d}$ is computed. By plugging the first-order Taylor expansion of $I(\mathbf{x} + \mathbf{d}, t + \tau)$ into (2), and imposing that the derivatives with respect to $\mathbf{d}$ are zero, we obtain the linear system

$$G\mathbf{d} = \mathbf{e}, \tag{3}$$

where

$$G = \sum_{\mathcal{W}} \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix}, \quad \mathbf{e} = -\tau \sum_{\mathcal{W}} I_t \left[I_u \ I_v \right]^\top,$$

with $[I_u \ I_v] = \nabla I = [\partial I / \partial u \ \partial I / \partial v]$ and $I_t = \partial I / \partial u$. The tracker is based on Eq. (3): given a pair of successive frames, $\hat{\mathbf{d}}$ is the solution of (3), that is, $\hat{\mathbf{d}} = G^{-1}\mathbf{e}$, and is used to predict a new (registered) frame. The procedure is iterated according to a Newton-Raphson scheme, until convergence of the displacement estimates.

2.1 Feature extraction

In this framework, a feature can be tracked reliably if a numerically stable solution to Eq. (3) can be found, which requires that G is well-conditioned and its entries are well above the noise level. In practice, since the larger eigenvalue is bounded by the maximum allowable pixel value, the requirement is that the smaller eigenvalue is sufficiently large. Calling λ_1 and λ_2 the eigenvalues of G, we accept the corresponding feature if $\min(\lambda_1, \lambda_2) > \lambda$, where λ is a user-defined threshold.[11].

2.2 Affine Model

The translational model cannot account for certain transformations of the feature window, for instance rotation, scaling, and shear. An *affine motion field* is a more accurate model [11], that is,

$$\delta(\mathbf{x}) = A\mathbf{x} + \mathbf{d}, \tag{4}$$

where $\mathbf{d}$ is the displacement, and A is a 2×2 matrix accounting for affine warping, and can be written as $A = \mathbf{1} + D$, with $D = [d_{ij}]$ a deformation matrix and $\mathbf{1}$ the identity matrix. Similarly to the translational case, one estimates the motion parameters, D and $\mathbf{d}$, by minimising the residual

$$\epsilon = \sum_{\mathcal{W}} \left[I(A\mathbf{x} + \mathbf{d}, t + \tau) - I(\mathbf{x}, t) \right]^2. \tag{5}$$

By plugging the first-order Taylor expansion of $I(A\mathbf{x} + \mathbf{d}, t + \tau)$ into (5), and imposing that the derivatives with respect to D and $\mathbf{d}$ are zero, we obtain the linear system

$$T\mathbf{z} = \mathbf{a}, \tag{6}$$

in which $\mathbf{z} = [d_{11} \ d_{12} \ d_{21} \ d_{22} \ d_1 \ d_2]^\top$ contains the unknown motion parameters, and

$$\mathbf{a} = -\tau \sum_{\mathcal{W}} I_t \left[uI_u \ uI_v \ vI_u \ vI_v \ I_u \ I_v \right]^\top,$$

with

$$T = \sum_{\mathcal{W}} \begin{bmatrix} U & V \\ V^\top & G \end{bmatrix},$$

$$U = \begin{bmatrix} u^2 I_u^2 & u^2 I_u I_v & uv I_u^2 & uv I_u I_v \\ u^2 I_u I_v & u^2 I_v^2 & uv I_u I_v & uv I_v^2 \\ uv I_u^2 & uv I_u I_v & v^2 I_u^2 & v^2 I_u I_v \\ uv I_u I_v & uv I_v^2 & v^2 I_u I_v & v^2 I_v^2 \end{bmatrix},$$

$$V^\top = \begin{bmatrix} uI_u^2 & uI_u I_v & vI_u^2 & vI_u I_v \\ uI_u I_v & uI_v^2 & vI_u I_v & vI_v^2 \end{bmatrix}.$$

Again, Eq. (6) is solved for $\mathbf{z}$ using a Newton-Raphson iterative scheme. If frame-to-frame affine deformations are negligible, the pure translation model is preferable (the matrix A is assumed to be the identity). The affine model is used for comparing features between the first and the current frame in order to monitor the quality of tracking.

3. Robust Monitoring

To monitor the quality of the features tracked, the tracker checks the residuals between the first and the current frame: high residuals indicate bad features which must be rejected. Following [11], we adopt the affine model, as a pure translational model would not work well with long sequences: too many good features are likely to undergo significant rotation, scaling or shearing, and would be incorrectly discarded. Non-affine warping, which will yield high residuals, is caused by occlusions, perspective distorsions and strong intensity changes (e.g. specular reflections). This section introduces our method for selecting a robust rejection threshold *automatically*.

3.1 Distribution of the residuals

We begin by establishing which distribution is to be expected for the residuals when comparing good features, i.e, almost identical regions. We assume that the intensity $I(\delta(\mathbf{x}), t)$ of each pixel in the current-frame region is equal to the intensity of the corresponding pixel in the first frame $I(\mathbf{x}, 0)$ plus some Gaussian noise $n \simeq \eta(0, 1)$. Hence

$$I(\delta(\mathbf{x}), t) - I(\mathbf{x}, 0) \simeq \eta(0, 1).$$

Since the square of a Gaussian random variable has a chi-square distribution, we obtain

$$[I(\delta(\mathbf{x}), t) - I(\mathbf{x}, 0)]^2 \simeq \chi^2(1).$$

The sum of n chi-square random variables with one degree of freedom is distributed as a chi-square with n degrees of freedom (as it is easy to see by considering the moment-generating functions). Therefore, the residual computed according to (2) over a $N \times N$ window $\mathcal{W}$ is distributed as a chi-square with N^2 degrees of freedom:

$$\epsilon = \sum_{\mathcal{W}} [I(\delta(\mathbf{x}), t) - I(\mathbf{x}, 0)]^2 \simeq \chi^2(N^2). \qquad (7)$$

As the number of degrees of freedom increases, the chi-square distribution approaches a Gaussian, which is in fact used to approximate the chi-square with more than 30 degrees of freedom. Therefore, since the window $\mathcal{W}$ associated to each feature is at least 7×7, we can safely assume a Gaussian distribution of the residual for the good features: $\epsilon \simeq \eta(N^2, 2N^2)$.

3.2 The X84 rejection rule

When the two regions over which we compute the residual are bad features (they are not warped by an affine transformation), the residual is not a sample from the Gaussian distribution of good features: it is an outlier. Hence, the detection of bad features reduces to a problem of outlier detection, which is equivalent to the problem of estimating the mean and variance of the corrupted Gaussian distribution. To do this, we employ a simple but effective model-free rejection rule, X84 [5], which achieves robustness by employing median and median deviation instead of the usual mean and standard deviation. This rule prescribes to reject values which are more than k Median Absolute Deviations (MADs) away from the median:

$$\mathrm{MAD} = \operatorname*{med}_i \{ |\epsilon_i - \operatorname*{med}_j \epsilon_j| \}. \qquad (8)$$

In our case, ϵ_i are the tracking residuals between the i-th feature in the last frame and the same feature in the first frame. A value of $k{=}5.2$, under the hypothesis of Gaussian distribution, is adequate in practice, as it corresponds to about 3.5 standard deviations, and the range $[\mu{-}3.5\sigma, \mu{+}3.5\sigma]$ contains more than the 99.9% of a Gaussian distribution [5]. The rejection rule X84 has a breakdown point of 50%: any majority of the data can overrule any minority.

3.3 Photometric normalisation

Our robust implementation of the Shi-Tomasi-Kanade tracker incorporates also a *normalize*d SSD matcher for residual computation. This limits the effects of intensity changes between frames, by subtracting the average grey level in each of the two regions considered:

$$\epsilon = \sum_{\mathcal{W}} [(J(A\mathbf{x} + \mathbf{d}) - \bar{J}) - (I(\mathbf{x}) - \bar{I})]^2, \qquad (9)$$

where $J(\cdot){=}I(\cdot, t+1)$, $I(\cdot){=}I(\cdot, t)$, and $\bar{J}$, $\bar{I}$ are the average grey levels in the two regions considered. A more elaborate normalization is described in [2]; [4] reports a modification of the Shi-Tomasi-Kanade tracker based on explicit photometric models.

4. Experimental results

We evaluated our tracker in a series of experiments, of which we report some.

Platform (Fig. 1, 256×256 pixels). A 20-frame synthetic sequence, created at the Heriot-Watt Computer Vision Laboratory, simulating a camera rotating in space while observing a subsea platform sitting on the seabed (real seabed texture-mapped onto a plane).

Hotel (Fig. 2, 480×512 pixels). The well-known Hotel sequence from the CMU VASC Image Database (59 frames). A static scene observed by a moving camera rotating and translating.

Stairs (Fig. 5, 512×768 pixels). A 60-frame sequence of a white staircase sitting on a metal base and translating in space, acquired by a static camera. The base is the platform of a translation stage operated by a step-by-step motor under computer control.

Artichoke (Fig. 6, 480×512 pixels). A 99-frame sequence, the most complex one shown here (see later on). The camera is translating in front of the static scene. This sequence was used by [13].

4.1 Experiment discussion

Platform is the only synthetic sequence shown here. No features become occluded, but notice the strong effects of the coarse spatial resolution on straight lines. We plotted the residuals of all features against the frame number (Fig. 3). All features stay under the threshold computed automatically by X84, apart from one which is corrupted by the interference of the background. In *Stairs*, some of the features picked up in the first frame are specular reflections from the metal platform, the intensity of which changes constantly during motion. The residuals for such features become therefore very high (Fig. 7). All these features are rejected correctly. Only one good feature is dropped erroneously (the bottom left corner of the internal triangle), because of the strong intensity change of the inside of the block. In the *Hotel* sequence (Fig. 4), all good features but one are preserved. The one incorrect rejection (bottom centre, corner

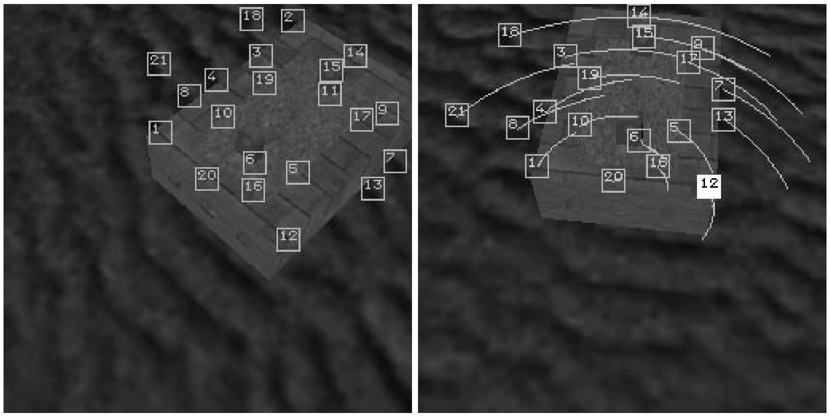

Fig. 1. First (left) and last frame of the *Platform* sequence. In the last frame, filled windows indicate features rejected by the robust tracker.

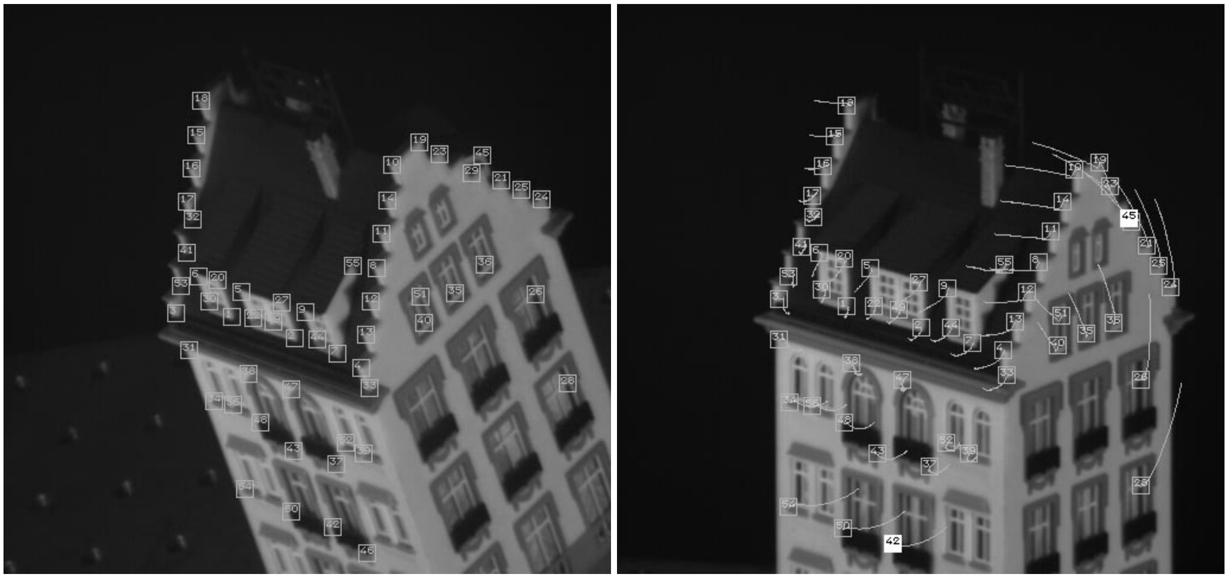

Fig. 2. First (left) and last frame of the *Hotel* sequence. In the last frame, filled windows indicate features rejected by the robust tracker.

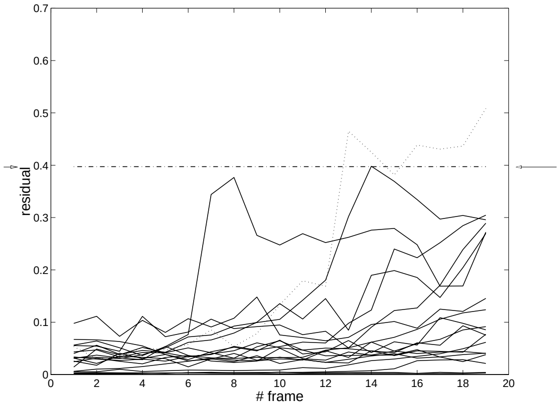

Fig. 3. Residuals magnitude against frame number for *Platform*. The arrows indicate the threshold set automatically by X84 (0.397189).

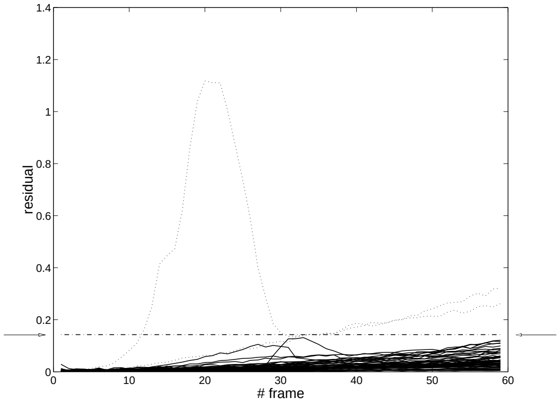

Fig. 4. Residuals magnitude against frame number for *Hotel*. The arrows indicate the threshold set automatically by X84 (0.142806).

	Artichoke	Hotel	Stairs	Platform
All	1.40	0.59	0.66	1.49
X84	0.19	0.59	0.15	1.49

Table 1. RMS distance of points from epipolar lines. The first row gives the distance using all the features tracked (non-robust tracker), the second using only the features kept by X84 (robust tracker).

of right balcony) is due to the warping caused by the camera motion, in this case too large to be accommodated by the affine model. The only spurious feature present (on the right-hand side of the stepped-house front) is rejected correctly. All features involved in occlusions in the *Artichoke* sequence (Fig. 8) are identified and rejected correctly. Four good features out of 54 are also rejected (on the signpost on the right) owing to a marked contrast change in time between the pedestrian figure and the signpost background.

4.2 Quantifying improvement: an example

To illustrate quantitatively the benefits of our robust tracker, we used the feature tracked by robust and non-robust versions of the tracker to compute the fundamental matrix between the first and last frame of each sequence, then computed the RMS distance of the tracked points from the corresponding epipolar lines, using Zhang's code [17] (if the epipolar geometry is estimated exactly, all points should lie on epipolar lines). The results are shown in Table 1. In all cases, the robust tracker brings a decrease in the RMS distance. Notice the limited decrease and high residual for *Platform*; this is due to the significant spatial quantization and smaller resolution, which worsen the accuracy of feature localization.

5. Conclusions

We have presented a robust extension of the Shi-Tomasi-Kanade tracker, based on the X84 outlier rejection rule. The computational cost is much less than that of schemes based on robust regression and random sampling like RANSAC or LMedSq [8, 14], yet experiments indicate excellent reliability in the presence of non-affine feature warping (most right features preserved, all wrong features rejected). Our experiments have also pointed out the pronounced sensitivity of the Shi-Tomasi-Kanade tracker to illumination changes. We believe that our robust tracker can be useful to the large community of researchers needing efficient and reliable trackers.

Acknowledgements

This work was supported by a British Council-MURST/CRUI grant and by the EU programme SOCRATES. Remarks made by anonymous referees helped improving the paper.

References

[1] J. L. Barron, D. J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. Comp. Vis.*, 12(1):43–77, 1994.

[2] I. J. Cox, S. Roy, and S. L. Hingorani. Dynamic histogram warping of image pairs for constant image brightness. In *ICIP*, pages 366–369, 1995.

[3] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *CVPR*, pages 858–863, 1997.

[4] G. D. Hager and P. N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *CVPR*, pages 403–410, 1996.

[5] F. R. Hampel, P. J. Rousseeuw, E. Ronchetti, and W. Stahel. *Robust Statistics: the Approach Based on Influence Functions*. John Wiley & Sons, 1986.

[6] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[7] L. Matthies, T. Kanade, and R. Szelisky. Kalman filter based algorithms for estimating depth from image sequences. *Int. J. Comp. Vis.*, 3:209–236, 1989.

[8] P. Meer, D. Mintz, D. Y. Kim, and A. Rosenfeld. Robust regression methods in computer vision: a review. *Int. J. Comp. Vis.*, 6:59–70, 1991.

[9] L. Robert, C. Zeller, O. Faugeras, and M. Hébert. Applications of non-metric vision to some visually-guided robotics tasks. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter 5, pages 89–134. Lawrence Erlbaum Associates, 1997.

[10] L. S. Shapiro, H. Wang, and J. M. Brady. A matching and tracking strategy for indipendently moving objects. In *BMVC*, pages 306–315, 1992.

[11] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, June 1994.

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

[13] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography – a factorization method. *Int. J. Comp. Vis.*, 9(2):137–154, 1992.

[14] P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In *BMVC*, pages 655–664, 1996.

[15] P. H. S. Torr, A. Zisserman, and S. Maybank. Robust detection of degeneracy. In *ICCV*, pages 1037–1044, 1995.

[16] E. Trucco, V. Roberto, S. Tinonin, and M. Corbatto. SSD disparity estimation for dynamic stereo. In *BMVC*, pages 342–352, 1996.

[17] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, INRIA Sophia-Antipolis, France, July 1996.

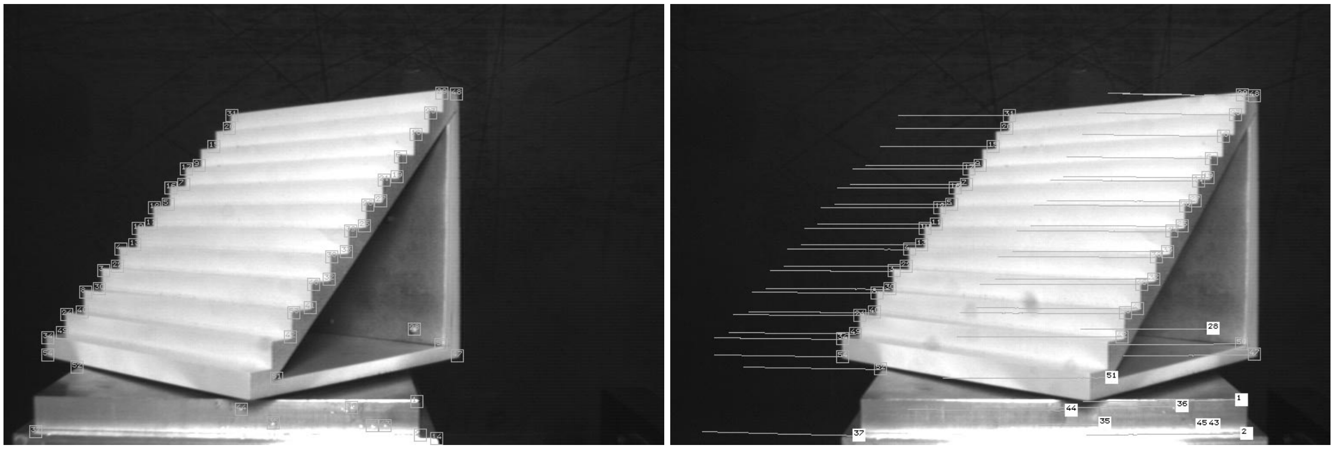

Fig. 5. First (left) and last frame of the *Stairs* sequence. In the last frame, filled windows indicate features rejected by the robust tracker.

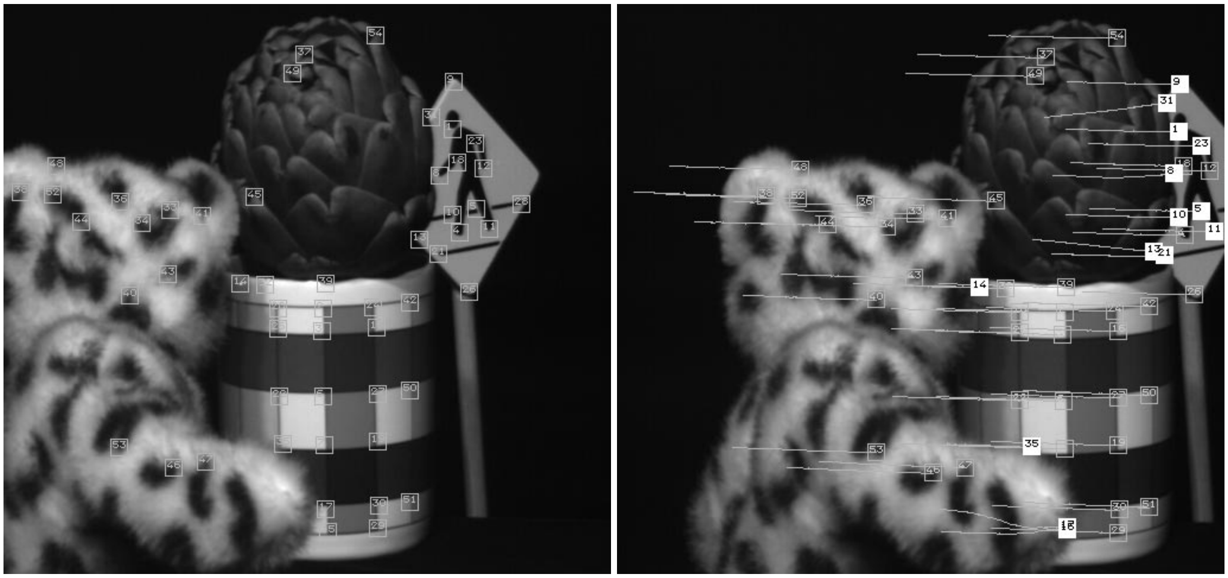

Fig. 6. First (left) and last frame of the *Artichoke* sequence. In the last frame, filled windows indicate features rejected by the robust tracker.

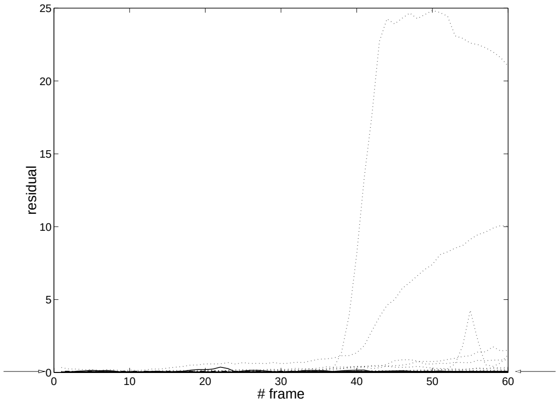

Fig. 7. Residuals magnitude against frame number for *Stairs*. The arrows indicate the threshold set automatically by X84 (0.081363) .

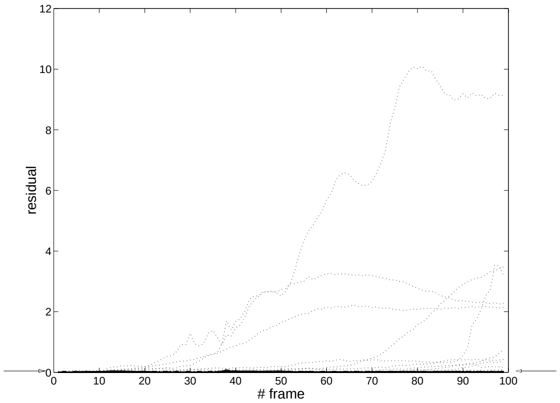

Fig. 8. Residuals magnitude against frame number for *Artichoke*. The arrows indicate the threshold set automatically by X84 (0.034511).