Tracking of points in a calibrated and noisy image sequence

Domingo Mery[1], Felipe Ochoa[2] and René Vidal[3]

[1]Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Av. Vicuña Mackenna 4860(143), Santiago de Chile
Tel. (+562) 354-5820, Fax. (+562) 354-4444
e-mail: dmery@ing.puc.cl
http://www.ing.puc.cl/∼dmery

[2]Departamento de Ingeniería Informática, Universidad de Santiago de Chile
felipe_ochoa@entelchile.net

[3]Center of Imaging Science, Johns Hopkins University
rvidal@cis.jhu.edu

Abstract. In this paper an algorithm that performs the tracking of points in a calibrated and noisy image sequence is presented. The candidate points to be tracked must satisfy certain constraints that can be deduced from the multiple view geometry. The idea is to consider as noisy points those candidates which cannot be tracked in the sequence. The robustness of the algorithm has been verified on simulated data using different constraints. The methods are assessed in several cases where the number of noisy points and the noise in the measurement of the points to be tracked are varied. Using this study, it is possible to know the performance of the tracking method. An example that shows a perfect tracking of 8 points in a sequence of 10 images with 500 noisy points per image is shown.

Keywords: Tracking, computer vision, multiple view geometry, epipolar geometry, trifocal tensors.

1 Introduction

Tracking is posed as a matching problem between correspondences of features in an image sequence [1]. In this problem, corresponding features, i.e., features in different views that are representation of the same 3D entity, must be matched. Generally, the problem is solved in two steps. The first step is the extraction of features obtained from the (2D) images of the sequence, e.g. points, lines or curves. The second step is to find correspondence between them. The most difficult problem is the correspondence of features. Since many tracking algorithms have been developed to date, we present now, in order to make this topic more clear, the following dichotomies of tracking. We do not intend to make here a survey of all previous work. However, the references presented in this review give a good overview to the reader.

i) Tracking by short or long sequences: in short sequences, the tracking is performed using epipolar or trifocal constraints [2]. The use of a larger number of images allows significant smoothing to be achieved, where a long sequence of frames is taken so that there is only a small change between adjacent frames [1].

ii) Tracking in monocular or stereo frames: with one camera 2D information is used to establish the correspondence in the sequence [3], whereas stereo frames gives the possibility to infer 3D information [1].

iii) Tracking from geometric features (points, lines, contours) or from grey values features (regions, windows): feature based methods rely on extraction of discrete object features in successive images, the image coordinate of which are used to estimate the object motion or to find the trajectories of the objects in the sequence. Geometric feature tracking is pose as a matching problem between points [4], lines [5] or contour curves [6] constructed out of edges in the image. Grey value feature tracking uses correlation techniques between regions or windows in order to establish the matching in two frames [1].

iv) Tracking using a motion model or based on a dense motion field: object motion is computed based on some model of the evolution of 3D motion parameters (e.g. kinematics model with Kalman filter [1]). In the other hand, optical flow methods represent motion in the image plane as sampled continuous velocity fields that can be used to track features [7].

v) Single object motion or multiple object motion: in the first case there is a single moving object in the sequence [8], whereas in the second case there are multiple objects with independent motions [4].

vi) Rigid or deformable objects: when the objects are rigid, the 3D shape of the object is invariable over the time, i.e., the shape can be easily modelled and stays relatively constant over the sequence [8]. In contrast, there are cases where the region of interest changes shape throughout the sequence, corresponding to deformable objects [9].

In this paper we deal with the problem of tracking of points of a single rigid object in long calibrated image sequences with monocular frames, where the density of noisy points is high. In a *calibrated* image sequence the model 3D $\rightarrow$ 2D is a priori known because it was obtained in an off-line process called calibration [2]. The idea of the tracking algorithm is to consider as noisy points those candidates which cannot be tracked in the sequence. The tracking is performed using geometrical multiple view constraints. The original algorithm was developed in [10] for automated visual inspection analysing multiple views with simple geometric constraints. However, no detailed study of the performance was made in [10]. In our paper, i) we demonstrate that the consideration of the geometric constraints based on the Sample distance [2] achieves a better performance than the simple constraints; and ii) we show how robust is the tracking algorithm under several noisy conditions. The rest of the paper is organised as follows: Section 2 gives an overview of the tracking algorithm. In Section 3 the different criteria used to establish the correspondence between the points of the views are described. Section 4 shows the experiments and the results obtained on synthetic data. Finally, Section 5 gives concluding remarks.

2 Tracking algorithm

After the candidate points in each image of the sequence are segmented, the coordinates of the points are normalised using the Cholesky factorisation [2]. Then the attempt is made to track them in the sequence in order to separate the noise from the object points. The tracking algorithm consists of three steps: *matching in two views*, *tracking in three views* and *tracking in four views*.

2.1 Matching in two views

Matching requires the position of each detected point. In this work, $\mathsf{a} = (a, p)$ will denote the identified point a in image p. It is assumed that the image sequence has N images ($1 \leq p \leq N$) and n_p points were identified in image p ($1 \leq a \leq n_p$). The position of point $\mathsf{a} = (a, p)$ is arranged in a *position vector* $\mathbf{m}_p^a$. One obtains then the position vector $\mathbf{m}_p^a = (x_p^a\ , y_p^a)$. This step matches two points (of two views), point $\mathsf{a} = (a, p)$ with point $\mathsf{b} = (b, q)$, for $p \neq q$, if they fulfil all following *matching conditions*:

• **Constraint in two views:** $\mathbf{m}_p^a$ and $\mathbf{m}_q^b$ must satisfy the constraint of correspondence between two views, i.e., the epipolar constraint.

• **Correct location in 3D:** the 3D point reconstructed from the position of the points must belong to the space occupied by the object. From $\mathbf{m}_p^a$ and $\mathbf{m}_q^b$ the corresponding 3D point $\hat{\mathbf{M}}$ is estimated. It is necessary to examine if $\hat{\mathbf{M}}$ resides in the volume of the object, the dimensions of which are usually known a priori.

In addition, a *similarity condition* can be used, if certain features of the candidate points and their neighbourhood are available. The matching is established if the points are similar enough.

The matching conditions in both identified points $\mathsf{a} = (a, p)$ and $\mathsf{b} = (b, q)$ are evaluated in 3 consecutive frames, for $p = 1, ..., N-3$; $q = p+1, ..., p+3$; $a = 1, ..., n_p$ and $b = 1, ..., n_q$. If a point is not matched with any other one, it will be considered as noise. Multiple matching, i.e., a point that is matched with more than one point, is allowed. Using this method, problems like non-segmented points or occluded points in the sequence, can be solved by the tracking if a point is not identified in consecutive views.

2.2 Tracking in three views

A match between two points a and b will be denoted by $\mathsf{a} \leftrightarrow \mathsf{b}$ or $(a, p) \leftrightarrow (b, q)$. A $N_2 \times 4$ matrix $\mathsf{A} = [\mathsf{a}_k\ \mathsf{b}_k] = [a_k\ p_k\ b_k\ q_k]$, $k = 1, ..., N_2$, is defined, where N_2 is the number of all matches determined in Section 2.1.In the tracking problem, it is required to find *trajectories* of points in different views. To establish the correspondence of points in three images one seeks all possible links of three points in matrix A that satisfy the condition of correspondence in three views.

The procedure is as follows: first, one looks for all two rows i and j of A ($i, j = 1, ..., N_2$ and $i \neq j$) that satisfy $\mathsf{b}_i = \mathsf{a}_j$. Supposing rows i and j fulfil this condition, in other words, the last two elements of row i are equal two the first

two elements of row j, e.g. $\mathsf{A}_i = [a\ p\ b\ q]$ and $\mathsf{A}_j = [b\ q\ c\ r]$, one finds three points $(a,p) \leftrightarrow (b,q) \leftrightarrow (c,r)$ with coordinates $\mathbf{m}_p^a$, $\mathbf{m}_q^b$ and $\mathbf{m}_r^c$ respectively that could be corresponding points. Finally, to examine if they really correspond to each other, the condition of correspondence between three views must be evaluated. If the condition is fulfilled, then it is assumed that the points are corresponding. The non tracked points are eliminated, while the N_3 linked triplets are arranged in a new matrix B.

2.3 Tracking in four views

A $N_3 \times 6$ matrix $\mathsf{B} = [\mathsf{a}_k\ \mathsf{b}_k\ \mathsf{c}_k] = [a_k\ p_k\ b_k\ q_k\ c_k\ r_k]$, $k = 1, ..., N_3$, is defined, where N_3 is the number of all triplets determined in Section 2.2. Now, the attempt is made two find two triplets in B that correspond to the same 3D point. It is well known that given four points (in four views), if the first three are corresponding points and the last three are corresponding points too, then all of them are corresponding points. In this case, it is not necessary to evaluate a condition of correspondence between four views, because it is redundant. That means, to seek quadruplets that satisfy the condition of correspondence in four views, it is necessary to look for all rows i and j of B for $(i, j = 1, ...N_3$ and $i \neq j)$ that satisfy $\mathsf{b}_i = \mathsf{a}_j$ and $\mathsf{c}_i = \mathsf{b}_j$. Supposing rows i and j fulfil this condition, e.g. $\mathsf{B}_i = [a\ p\ b\ q\ c\ r]$ and $\mathsf{B}_j = [b\ q\ c\ r\ d\ s]$, four corresponding points $(a,p) \leftrightarrow (b,q) \leftrightarrow (c,r) \leftrightarrow (d,s)$ (with coordinates $\mathbf{m}_p^a$, $\mathbf{m}_q^b$, $\mathbf{m}_r^c$ and $\mathbf{m}_s^d$ respectively) are found. The N_4 detected quadruplets are placed in a new matrix C. Finally, a tracking in more views can be achieved by linking quadruplets of matrix C having common elements.

3 Correspondence and 3D reconstruction criteria

In this investigation different criteria of correspondence of points between two and three views and 3D reconstruction from two views are used. Normally, the epipolar constraint, the trilinear constraint and the triangulation approach are used for these tasks. However, a typical observation consists of a noisy point correspondence which does not in general satisfy the correspondence constraints [2]. For this reason, other criteria should be used. In this Section, we present alternative criteria that take into account the noise in the imaged points.

3.1 Correspondence in two views

Two methods are used to investigate if $\mathbf{m}_p^a$ and $\mathbf{m}_q^b$ can be corresponding points.
• **Simple:** the constraint is fulfilled if the perpendicular Euclidean distance from the epipolar line of the point $\mathbf{m}_p^a$ to the point $\mathbf{m}_q^b$ is smaller than ε [2].
• **Sampsom:** the constraint is fulfilled if the first-order geometric error of the epipolar constraint is smaller than ε [2].

3.2 Correspondence in three views

Two methods are used to investigate if $\mathbf{m}_p^a$, $\mathbf{m}_q^b$ and $\mathbf{m}_r^c$ satisfy the correspondence in three views:
• **Simple:** The position of third point is estimated from $\mathbf{m}_p^a$ and $\mathbf{m}_q^b$ using a reprojection approach based on trifocal tensors. The constraint is fulfilled if the Euclidan distance between estimated point and $\mathbf{m}_r^c$ is smaller than ε [10].
• **Sampsom:** the constraint is fulfilled if the first-order geometric error of the trilinear constraints is smaller than ε [2].

3.3 3D Reconstruction

Two methods are used to estimate a 3D point $\hat{\mathbf{M}}$ that may have produced the imaged points $\mathbf{m}_p^a$ and $\mathbf{m}_q^b$.
• **Simple:** Point $\hat{\mathbf{M}}$ is estimated using normalised projection matrices and a projective 3D transformation [11].
• **Sampsom:** Point $\hat{\mathbf{M}}$ is estimated using the triangulation of corrected points $\hat{\mathbf{m}}_p^a$ and $\hat{\mathbf{m}}_q^b$ that are computed from the first-order geometric correction of $\mathbf{m}_p^a$ and $\mathbf{m}_q^b$ [2].

4 Experiments and Results

In this Section we present the results obtained recently using synthetic data[1]. The data was simulated as follows: A cube was defined and located in N different positions. For each position the 8 vertices of the cube were projected using a perspective transformation into an image of $M \times M$ pixels. Thus, a sequence of N binary images of $M \times M$ pixels was obtained. Additionally, N_p points were randomly superimposed in each image. Fig. 1 illustrates one of these images with

[1] A real experiment is available in `http://www.ing.puc.cl/~dmery/sequences.htm`.

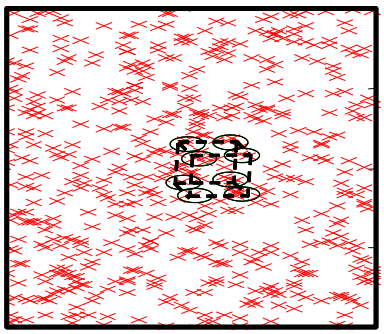

Fig. 1. Simulation of a projected cube into an image with $N_p = 500$ random points.

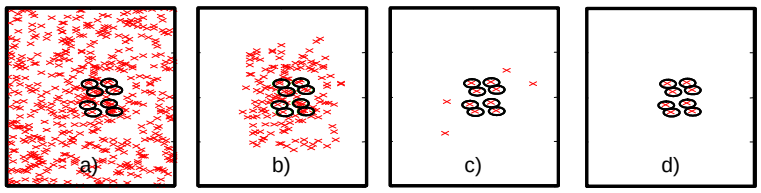

Fig. 2. Tracking of the vertices in a sequence of $N = 10$ images with $N_p = 500$ noisy points per image: a) original image of the sequence, and points after b) matching in two views, c) tracking in three views, and d) tracking in four views.

$N_p = 500$ and $M = 400$, where the doted lines and the circles were intentionally added to distinguish the projection of the cube. In order to simulate a more realistic projection, the coordinates of the projected vertices were altered adding normal distributed noise with standard deviation σ and mean zero.

The effectiveness of the tracking algorithm for 500 noisy points is illustrated in Fig. 2. Only one image (see Fig. 2a) of a whole sequence of 10 images is presented. This image corresponds to Fig. 1. After matching in two views, the remaining points are shown in Fig. 2b. The other points were eliminated because they were not matched. The dimensions of the cube were $2 \times 2 \times 2$, and the space used in the correct location in 3D criterion was $8 \times 8 \times 8$ centred in the cube, i.e., 64 times bigger than the cube. For this reason, the points that satisfied the matching conditions are approximately located in an area of $4h \times 4h$, where h is the distance between two adjacent projected vertices. After tracking in three and four views (see Fig. 2c and 2d), all noisy points were eliminated without discriminating the points of the cube.

The performance of the algorithm was evaluated for different conditions varying the number of noisy points ($N_p = 10, 25, 50, 100, 200, 500$); the standard deviation of the projected points ($\sigma = 0, 0.05, 0.1, 0.2, 0.35, 0.5$); the tolerance used in correspondence constraints ($\varepsilon = 0.0005, 0.001, 0.005, 0.01$); and the correspondence and 3D reconstruction criteria (*simple* or *Sampson*). In these experiments, the size of the images was 400×400 pixels, and the number of images in the sequence was 10. Each situation was simulated 10 times and the average of the following variables was computed: the number of true positives (TP), i.e., points of the cube correctly tracked (ideally, $TP = 8$); the number of false positives (FP), i.e., tracked points that do not correspond to the vertices of the cube (ideally, $FP = 0$); the computational time (t) required for the tracking process. The simulation environment was programmed in MATLAB using a PC based on a CPU Pentium 4, 2.6 GHz, 512 MB RAM, and operating system Microsoft Windows XP.

In order to evaluate the performance of the tracking, the true positive rate ($TPR = TP/8$) and the false positive rate ($FPR = FP/N_p$) were computed for different noise conditions (N_p, σ) varying parameter ε in the geometric constraints. Ideally, $TPR = 1$ and $FPR = 0$, i.e., all object points are tracked with-

Table 1. Performance (TPR^*, FPR^*) for different noise conditions (N_p, σ).

σ/M	method	$Np = 10$	$Np = 25$	$Np = 50$	$Np = 100$	$Np = 200$	$Np = 500$	average
0.000000	simple	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)
		$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	
	Sampson	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)
		$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	
0.000125	simple	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)
		$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	
	Sampson	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)
		$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.001$	
0.000250	simple	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(0.97,0.00)	(0.97,0.00)	(0.99,0.00)
		$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.001$	$\varepsilon^* = 0.001$	
	Sampson	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)
		$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	
0.000500	simple	(1.00,0.00)	(1.00,0.00)	(0.99,0.00)	(1.00,0.00)	(1.00,0.00)	(0.95,0.00)	(0.99,0.00)
		$\varepsilon^* = 0.005$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	
	Sampson	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)
		$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.005$	
0.000875	simple	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(0.97,0.00)	(0.98,0.00)	(0.99,0.00)
		$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 0.005$	
	Sampson	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(1.00,0.00)	(0.92,0.00)	(0.99,0.00)
		$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.005$	
0.001250	simple	(1.00,0.00)	(0.96,0.01)	(0.97,0.00)	(0.95,0.01)	(0.95,0.00)	(0.88,0.00)	(0.95,0.00)
		$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.005$	
	Sampson	(0.93,0.00)	(0.97,0.00)	(1.00,0.00)	(0.99,0.00)	(1.00,0.00)	(0.91,0.00)	(0.97,0.00)
		$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	$\varepsilon^* = 0.010$	

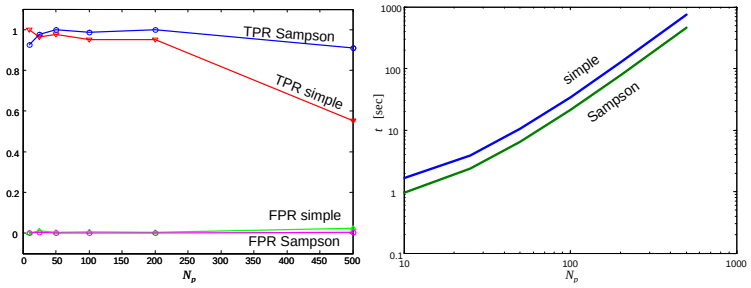

Fig. 3. Evaluation of the methods depending on the number of noisy points N_p: a) TPR and FPR at $\varepsilon = 0.01$ and $\sigma/M = 0.00125$; and b) computational.

out flagging false alarms. The best operation point (TPR^*, FPR^*) at $\varepsilon = \varepsilon^*$ is chosen as the point with the smallest Euclidean distance to the ideal point $(1, 0)$. The obtained results are summarised in Table 1, where the performance was calculated for different (N_p, σ) combinations. As shown in Table 1, parameter ε^*, i.e., the parameter ε that gives the best operation point, depends strongly on the inherent noise of the projected points (σ) and the number of noisy points (N_p). The larger is σ, the larger should be set ε if one wants to track all object points. However, the larger is ε, the larger the false alarm rate when N_p is large, because several noisy points can be tracked. As example, Fig. 3a shows TPR and FPR for $\sigma/M = 0.001250$ and $\varepsilon = 0.01$. In addition, the computational time is presented in Fig. 3b. We observe that the performance and the computational time of the Sampson criteria is better. On the other hand, the results of average column Table 1 show the high robustness of the tracking algorithm in both discrimining noisy points and detecting object points.

5 Conclusions

In this paper an algorithm that performs the tracking of points in a calibrated and noisy image sequence was presented. The idea is to consider as noisy points those candidates which cannot be tracked in the sequence. The tracked points satisfy certain constraints of the multiple view geometry. The robustness of the algorithm has been verified on simulated data using two different constraints: simple and Sampson. The methods were assessed in several cases where the number of noisy points and the noise in the measurement of the points to be tracked are varied. The obtained results are: the robustness of the algorithm is very high, and the best performance was achieved using the Sampson distance.

Acknowledgments

This work was supported by FONDECYT – Chile under grant no. 1040210.

References

1. Zhang, Z., Faugera, O.: Estimation of displacements from two 3D frames obtained from stereo. IEEE Trans. Pattern Analysis and Machine Intelligence **14** (1992) 1141–1156
2. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
3. Shirai, Y.: Estimation of 3-D pose and shape from a monocular image sequence and realtime human tracking. In: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Canada (1997) 130–140
4. Chetverikov, D., Verestóy, J.: Tracking feature points: a new algorithm. In: Proceedings of the 14th International Conference on Pattern Recognition, ICPR-1998, Brisbane, Australia (1998) 2:1436–1438
5. P.E. López-de Teruel, A.R., García, J.: A parallel algorithm for tracking of segments in noisy edge images. In: Proceedings of the 15th International Conference on Pattern Recognition, ICPR-2000, Barcelona (2000) 1051–1055
6. Freedman, D.: Effective tracking through tree-search. IEEE Trans. Pattern Analysis and Machine Intelligence **25** (2003) 604–615
7. Mae, Y., Shirai: Tracking moving object in 3-d space based on optical flow and edges. In: Proceedings of the 14th International Conference on Pattern Recognition, ICPR-1998, Brisbane, Australia (1998) 2:1439–1444
8. Broida, T., Chellapa, R.: Estimating the kinematics and structure of a rigid object from a sequence of monocular images. IEEE Trans. Pattern Analysis and Machine Intelligence **13** (1991) 497–513
9. Mansouri, A.: Region tracking via level set PDEs without motion computation. IEEE Trans. Pattern Analysis and Machine Intelligence **24** (2002) 947–961
10. Mery, D., Filbert, D.: Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. IEEE Trans. Robotics and Automation **18** (2002) 890–901
11. Hartley, R.: A linear method for reconstruction from lines and points. In: 5th International Conference on Computer Vision (ICCV-95), Cambridge, MA (1995) 882–887