

Towards global consistent pose estimation from images*

Stephan H.G. ten Hagen¹, Ben J.A. Kröse²

¹University of Amsterdam, Amsterdam, The Netherlands, stephanh@science.uva.nl

²University of Amsterdam, Amsterdam, The Netherlands, kroese@science.uva.nl

Abstract

We propose a method for making globally consistent pose estimates using vision. Lu and Milios described an approach where the links between poses were estimated using range scanner data. When using point correspondences the length of the links cannot be estimated and the approach of Lu and Milios has to be modified. First we use the nonlinear orientation part of the pose differences to obtain a reference trajectory. Then the reference trajectory is used to scale and orientate the linear spatial part of the pose differences, such that the positions can be estimated as well. We show the results of an experiment where we navigated a robot equipped with an omni-directional camera on our corridor.

1 Introduction

Global pose alignment schemes [8] are batch approaches that try to estimate past robot poses such that the sensor readings at all poses are consistent. Lu and Milios [8] use range scanner data from which ‘links’ between pairs of poses are estimated. These links are the relative differences in poses estimated by comparing the sensor readings at these poses. The absolute poses in the environment are derived from all relative pose differences.

Our mobile robot is equipped with an omni-directional vision system [2]. In order to build an appearance model for localization [7], a supervised set containing the images and their corresponding poses are required. It is a cumbersome process to obtain such set by hand and the odometry is noisy and will lead to large errors in the long run. In this paper we describe our attempts to estimate the robot poses from links estimated from vision data.

The relative orientation between two poses can be determined from two images, but the distance between two poses can only be estimated up to a scale factor. To estimate the scale, we can use a reference

trajectory (i.e. the odometry). In this paper we introduce a novel approach to obtain a reference trajectory using the relative orientations between poses. This approach is based on our work on *trajectory reconstruction* [5], where an odometry trajectory can be re-estimated if the pose at the beginning and the pose at the end is known. We use the visual link between the first and last pose to find the end pose for which the reconstructed trajectory agrees most with the measured relative orientations. Once we have this trajectory we use it to scale the position part of the links. Also we use this to rotate all links into one global orientation, such that the linear consistent pose alignment of Lu and Milios can be applied.

We start in section 2 by describing the Lu and Milios approach and point out the differences with our approach. In section 3 we briefly describe the estimation of the links from omni-directional images. Our approach to obtain and use the reference trajectory is described in section 4. In section 5 we present our experimental results.

2 Consistent pose estimation from links

2.1 Links from scan alignment

Lu and Milios introduced a globally consistent range scan alignment approach [8], in which relative spatial relations between local measurements are used to obtain a globally consistent environment mapping.

If we take X_i and X_j as two different poses of the robot, then a relative pose can be defined as

$$D_{ij} = X_i - X_j. \quad (1)$$

This is the linear situation and D_{ij} is called a link. We can write this compactly for all links as

$$\mathbf{D} = \mathbf{H}\mathbf{X}, \quad (2)$$

where \mathbf{D} is the vector with all possible links, \mathbf{X} is the vector of all poses. The \mathbf{H} is a matrix with only zeros and for each row a one and minus one at the appropriate column to form the differences (1) for each possible link.

*This research is supported by the Netherlands Organization for Scientific Research (NWO).

When $\bar{\mathbf{D}}$ represents all measurements of the links, then the Mahalanobis distance

$$W = (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X})^T \mathbf{C}^{-1} (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X}) \quad (3)$$

is minimized as a linear weighted least squares estimation to find \mathbf{X} . The \mathbf{C} is the covariance matrix of $\bar{\mathbf{D}}$.

The minimization of (3) will only work if all links are expressed in the same coordinate system. When the orientation of the robot is part of $\bar{\mathbf{D}}$, a nonlinearity is introduced in $\bar{\mathbf{D}}$ making it dependent on the global orientation. Lu and Milios consider the measured links as local linear approximation of the nonlinear difference in pose. They use the odometry as initial reference trajectory to “rotate” all links in the same direction”.

2.2 Our approach: links from vision

We estimate the links from point correspondences in images. The difference with the use of range scanner data as in [8], is that the length of the link is unknown. An other difference is that we do not have a clear description of the error covariance.

We split the measured link into a displacement vector $\bar{D}_{d,ij}^l$ (of arbitrary length) and a difference in orientation $\bar{D}_{\phi,ij}$, where the superscript l denotes that the link is expressed in the local robot coordinate system. To be able to use (3) we need a reference trajectory from which we can compute the links $D_{d,ij}^r$ and $D_{\phi,ij}^r$. The links from the reference trajectory can be used to obtain the scale and global orientation of $\bar{D}_{d,ij}^l$. In [8] the global orientation is derived from odometry.

The novel idea presented in this paper is to use the orientation links to get the initial reference trajectory. So we use $\bar{D}_{\phi,ij}$ to obtain the global orientation and do not need to estimate the orientation by minimizing (3). We use the trajectory to rotate and scale the $\bar{D}_{d,ij}^l$ and minimize (3) *only* for the position. Because the linear and nonlinear part are separated, we do not need a local linear approximation as in [8]!

Note that we did not implement the minimization of (3) using the network of links representation proposed in [8]. Very accurate results were obtained by treating (3) as one weighted least squares estimate and solving this using QR decomposition, in which no matrix had to be inverted.

3 Vision and estimation

We used existing methods for the vision and estimation. So we will only give a overview of the architecture used to estimate the links from the images.

3.1 The vision

The vision starts with images taken using an omnidirectional camera and mapped to a cylinder to form gray-scaled panoramic images (see [2] for details). To obtain interesting features at different levels of detail, we generated two extra images. One twice as small and one four times as small as the original image. The size is reduced using the Gaussian kernels from [3]. This results in three images for each pose.

In these three images we detected the features that are good to track using the `klt` software from [1]. This implements a Kanade-Lucas-Tomasi tracker, but we only used the feature detection part. We assume that some good features at one pose will also be considered good features in the proximity of that pose. In contrast with the tracking application, our approach is an off-line approach.

The quality of a feature is expressed using the measure from [9] that is based on the minimal eigenvalue of a 2×2 matrix, computed by comparing two shifted windows in the image with each other. If the minimal eigenvalue for a certain window compared to all other shifted windows is above a certain threshold, we keep the center of that window as a good feature. The result is a set of good features for the three images at each pose.

Now we have to find correspondences between the sets of features in all images. We first remove feature pairs that cannot be corresponding pairs because of their location in the images. Features in the top part of the images represent points that were higher than the camera and can therefore not correspond to features in the bottom part of the images. Then we start looking at feature similarities. We do this by taking a window around a feature in one image and compute the summed squared difference in pixel values around all features in the other images. We keep the pairs of features for which the relative summed squared difference is below a certain threshold. We use the same size of window for each of the three images, so effectively the large images focuses on the details while the small images focuses on larger objects. The result is a set of corresponding features between all poses.

3.2 Estimating the visual links

The difference between poses can be described by a motion given by a rotation and translation. We will use a motion estimation technique from point correspondences based on epipolar geometry. We have a calibrated camera so we can map the features of the three images of a pose to a unit sphere around the focal point of the camera.

The epipolar geometry sets constraints to the dis-

placements of points in images when moving the camera [4]. These constraints can be expressed using the essential matrix E :

$$\mathbf{u}^T E \mathbf{v} = 0. \quad (4)$$

Here \mathbf{u} and \mathbf{v} are features on the unit sphere for two different poses.

We estimate the essential matrix using the M-estimator described in Torr and Murray [10]. This is an approach to estimate the fundamental matrix (uncalibrated essential matrix), and they set the smallest singular value to zero to make sure the resulting matrix has rank 2. We had to modify this because the two non-zero singular values of an essential matrix have to be equal. We replaced the two non-zero singular values by their average.

The rotation matrix R and translation matrix S should be derived from the estimated essential matrix such that $RS = E$. We first normalize E and then apply the decomposition method described by Horn [6]. The results are four possible combinations of R and S and we have to select one of these. The selection is based on reconstruction of features into the 3D environment. At the two poses the features are projected onto the unit spheres and we can compute a line from the focal point through the feature on the unit sphere. For each combination of R and S we compute the point where these lines intersect. For the correct combination of R and S the distance of the intersection should be positive. Because of noise the lines do not have to intersect, so instead we compute the point where they are at the closest distance. A consequence is that not every feature will select the same combination of R and S and therefore we select that combination of R and S that is selected most often by the corresponding features.

From the selected matrices R and S we compute $\bar{D}_{\phi,ij}$ and $\bar{D}_{d,ij}^l$.

3.3 The reliability of the links

Lu and Milios take a closed expression for the uncertainty in their range scan measurements. In our case we do not directly measure the links. We have a vision pipeline consisting of many different computational steps all introducing their own uncertainties.

The first errors are introduced by the false correspondences. They may lead to errors in estimating the essential matrix. The residual of the estimation is the violation against (4), which can be computed for every point using $\mathbf{u}^T E \mathbf{v}$. Points that have a high residue are probably outliers and we remove all features for which the residue is more than twice the median of the residue of all features. This will remove the most obvious outliers but there is no guarantee

that all false correspondences are removed. We then estimate the essential matrix again using the reduced corresponding feature set.

Another source of errors is the selection of R and S . In an ideal situation the computation of the intersections is no problem at all. But noisy features and possible false correspondences lead to different selection choices. If most features agree on the selection then the estimation was reliable, but it is also possible to have two choices that have high support. Note that the consequences of an erroneous selection is quite dramatic, because it will lead to a completely different link.

Another indicator of the reliability is given by the knowledge that the robot moved on a 2D plane. The rotation can be expressed as $R = R_x R_y R_z$ and the translation as $S = S_x + S_y + S_z$. We can compare R with R_z and S with $S_x + S_y$ to judge the quality of the estimation. A large $S_z = S - S_x - S_y$ indicates that the robot moved up or down, so this is an indication of a false correspondence or a wrong selection.

We can conclude that the uncertainty cannot be expressed using covariance matrices. We only have heuristic indications of the reliability of the estimated links.

4 The reference trajectory

We will introduce the method to obtain the reference trajectory using the relative orientations. Then we describe how we use the reference trajectory.

4.1 Obtaining the reference trajectory

Trajectory reconstruction as described in [5] is a method to estimate past pose values given a known begin and end pose. It uses dead-reckoning from the begin pose in the same way as odometry, but it also uses dead-reckoning from the end pose. A trajectory can be reconstructed by combining both dead-reckoning trajectories. In [5] a Kalman filter approach is introduced that gives reliable pose estimates for long trajectories. Besides the begin and end pose it requires the sequence of actions and the motion model of the robot.

In this paper we reverse the situation by taking the end pose as the unknown variable. For any chosen end pose we can evaluate how well the *orientation* links of the reconstructed trajectory matches with that of the measurements. We have an estimation of the link between the first and last pose, without the scale of the spatial displacement. Let r_e be the scale of the spatial displacement of the end pose. We choose an r_e and compute the orientation links $D_{\phi,ij}^r$ from the reconstructed trajectory. The superscript r indicates links from a reconstructed trajectory. Then

we define¹ $\Delta D_{r_e,ij} = D_{\phi,ij}^r - \bar{D}_{\phi,ij}$ as the error for r_e and orientation link ij .

We estimate the spatial displacement r_e^* according to

$$r_e^* = \operatorname{argmin}_{r_e} \sum_{ij} \|\Delta D_{r_e,ij}\|^2. \quad (5)$$

Because there is no defined gradient for $\Delta D_{r_e,ij}$ only non gradient based optimization methods can be used.

We have to be careful when minimizing (5). The measured links depend on R selected after the Horn decomposition. Wrongly selected R matrices will lead to high values of $\Delta D_{r_e,ij}$, and so they influence the minimization of (5) strongly. To prevent this we only used links for which R and S were selected with absolute majority. We further rejected all links for which $|\Delta D_{r_e,ij}| > \frac{1}{4}\pi$, for the reconstruction computed using $r_e = 0$. We do not compute this for each value of r_e to make sure that the minimum value of (5) is only determined by the errors in orientation links and not by the rejection policy.

To be less dependent on the accuracy of the estimated link between begin and end, we should not only optimize for r_e but for the complete end pose (position and orientation). Then minimization of (5) will give a good initial guess of the end pose. The reconstruction using the best end pose will be used as reference trajectory.

4.2 Using the reference trajectory

The reference trajectory already gives estimations of all poses. It only used the orientation links and the estimation between the begin and end pose. We also have available estimations of the local displacement vectors $\bar{D}_{d,ij}^l$ between all poses. In order to use these vectors we have to scale and rotate them based on the reference trajectory.

The reference trajectory can be used to compute all links $D_{d,ij}^r$. The length of these links are known so we can use this to scale $\bar{D}_{d,ij}^l$. The reference trajectory also gives the orientation ϕ_i^r of each pose expressed in the same global coordinate system. We can use this to rotate all local displacement vectors. So we use

$$\bar{D}_{ij} = \|D_{d,ij}^r\|_2 \begin{bmatrix} \cos(\phi_i^r) & -\sin(\phi_i^r) \\ \sin(\phi_i^r) & \cos(\phi_i^r) \end{bmatrix} \bar{D}_{d,ij}^l \quad (6)$$

to create a global displacement link \bar{D}_{ij} from a local unscaled displacement link $\bar{D}_{d,ij}^l$. We can express this for all \bar{D}_{ij} as in (2) and solve (3) to obtain the estimates of the pose positions.

¹Or $\Delta D_{r_e,ij} = \bar{D}_{\phi,ij} - D_{\phi,ij}^r$ in case $|\Delta D_{r_e,ij}| > \pi$.

To solve (3) we also need the covariance of the links. We do not have this but we can take a unity matrix and multiply this with an heuristic expressing the uncertainty. In section 3.3 we described some measures to express the reliability. We can take the residue of the essential matrix estimation, or we can use deviation with the 2D motion. Since we have a reference trajectory we can use it to see how well the estimated links fit in the global framework. This results in $\|\bar{D}_{ij} - D_{d,ij}^r\|_2$ as an extra indication of the certainty.

5 Experiment

5.1 Setup

We generated a dataset by taking a Nomad Scout II and navigate it through our corridors using a joystick. The robot was equipped with an omnidirection camera as described in [2]. We started in our robot lab and moved the robot out on the corridors and we ended the trajectory by entering the robot lab again. The length of the trajectory was approximately 25 meters. During the run 93 images were taken, corresponding to 93 different poses.

5.2 Vision and estimation

We used a 5 by 5 window for the feature selection in the images. The 93 poses result in $\frac{1}{2} \times 92 \times 93 = 4278$ possible links. For all 4278 possibilities the features in the images were compared to create a set of corresponding features for each link. We also used a 5 by 5 window to compare the features. We used the corresponding features to estimate the essential matrix and removed some outliers based on the residue of the estimation. Figure 1 shows the corresponding features between the first and last pose.

We only considered links for which we have at least 8 corresponding features. After processing all 4278 combinations of poses only 1520 links had more than 8 corresponding features. This still leaves an average of 13 links per pose. The maximum number of links for one pose was 87.

It was infeasible to manually verify all 1520 links. We only checked a few and some were very good and some were rather bad. We have to rely on the proper weighting or use some heuristics to remove the influence of the bad links.

5.3 Obtaining the reference trajectory

To reconstruct the trajectory we considered the begin and end pose. The estimated links seemed reasonable when considering the begin and end pose. The end link was $\bar{D}_{d,0-92}^T = [-0.9760 \quad 0.2238]^T$ and $\bar{D}_{\phi,0-92} = -3.1353$. We varied r_e from zero onwards and computed the error in orientation links.

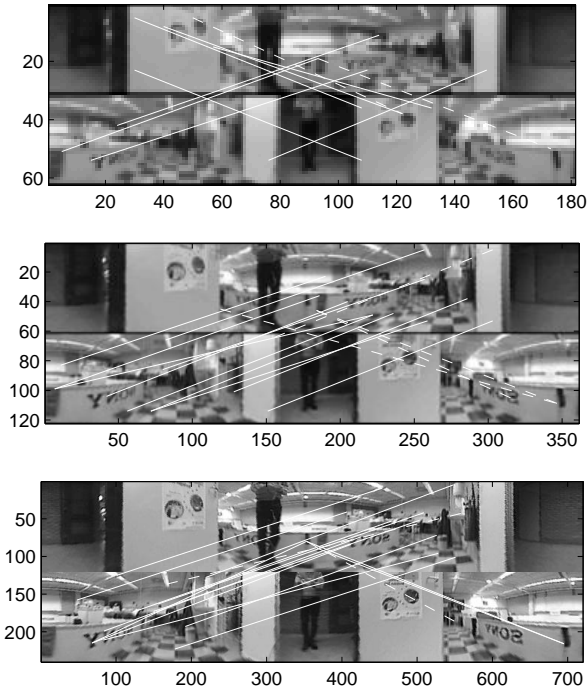


Figure 1: The small, medium and large image of the first and last pose. The lines indicate the corresponding pairs in the images. The dashed lines are the corresponding pairs that were removed because their residual of the epipolar constraint were too high.

Here we removed the obvious errors by only considering links for which the R was selected based on the absolute majority of the features. Also we removed the links for which $|\Delta_{r_e,ij}| > \frac{1}{4}\pi$. In figure 2 the error is shown for r_e varying from 0 to 0.4. In this case we could reduce the amount of time searching because we knew that the distance was not too large.

In figure 2 we see that the minimum value is found for $r_e = 0.14$. With this value we reconstructed a trajectory to form the reference trajectory. In figure 3 we show the resulting reference trajectory compared with the odometry.

We had to use the odometry because the true values of the poses were not known. This is because we did not perform the experiment in a controlled environment, but rather moved the robot around on the corridor using a joystick. The robot started in the origin moving in the direction of the positive x-axis. Then it made a turn to the right and the rest of the trajectory is clear from figure 3. We see that the end pose of the odometry is different and there is some deviation at the second half of the trajectory. Unfortunately we do not know the cause of this difference. It may be that the odometric estimates become more unreliable over time and that the reference trajectory

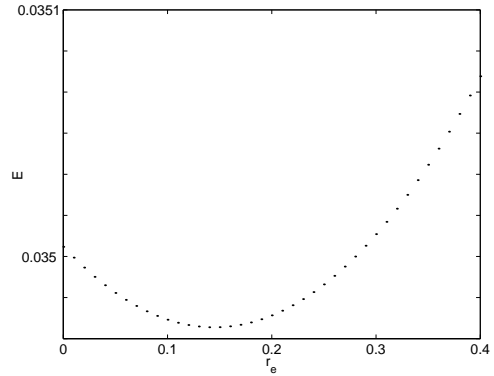


Figure 2: The error in orientation link between the reference trajectory and the measurements as function of the displacement of the end pose r_e .

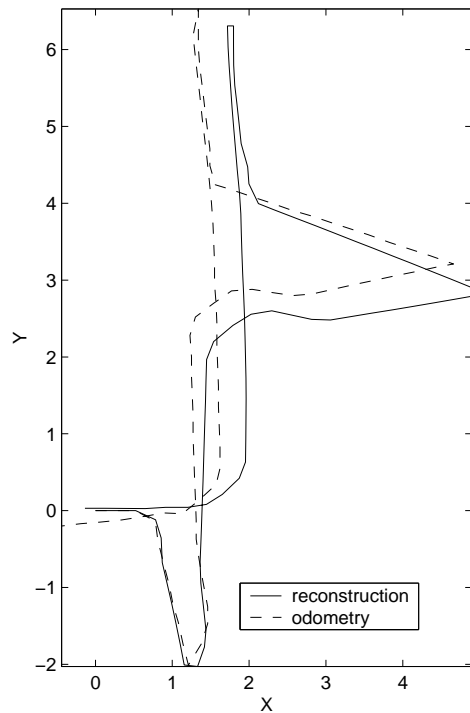


Figure 3: The reference trajectory obtained using trajectory reconstruction.

is more correct. It can also be that there was a slight error in the link between begin and end pose leading to errors in the reconstructed trajectory.

5.4 Using the reference trajectory

We used the reference trajectory to apply (6) to all 1520 estimated links. Note that we can remove the rows in the \mathbf{H} matrix in (2) and the rows and columns in the \mathbf{C} matrix in (3) if the corresponding links are not available. So missing links are not a problem, unless it leaves a pose with only one links. Poses with one link cannot be estimated and the corresponding

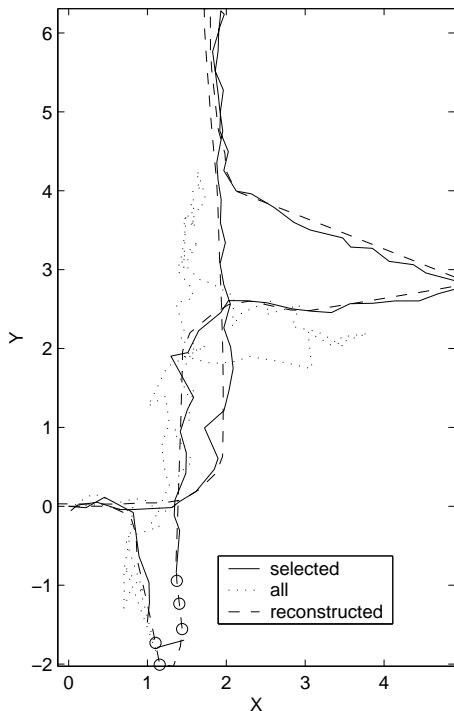


Figure 4: The resulting position estimates.

columns of \mathbf{H} should be removed. Then (3) can be applied to get an estimate of the spatial position.

For the covariances we used many different weighting schemes based on different combinations of quality measures mentioned in section 4.2. Unfortunately none of the measures resulted in reasonable estimations, except for the one based on the reference trajectory. In figure 4 the dotted trajectory is when all 1520 links are used to estimate the poses. The shape of the trajectory seems correct but too many links pull the pose estimates towards the origin.

We did the same estimation where we selected only the 303 links for which $\|\bar{D}_{ij} - D_{d,ij}^r\|_2 < 0.5$. These were mostly the shorter links. There was no significant difference between the heuristic reliability measures for these 303 selected links and for those that were rejected. The trajectory is also shown in figure 4, where the circles indicate the poses with less than two links that were not estimated. We see that the result is very good compared to the reference trajectory. That the estimated trajectory looks a bit more jagged is not because it is more noisy. It is because a reconstructed trajectory can only be smooth since it is based on a combination of two noise free dead-reckoning trajectories. Locally the estimated trajectory can be more correct than the reference trajectory that is only based on the begin and end

pose. To investigate whether the estimated poses are more correct we have to repeat the same experiment in a controlled environment where the true poses are known.

6 Conclusion

We introduced a novel approach for global consistent pose estimation for visual navigation. It is a batch approach dealing with the local orientation and the absence of scale in visual pose difference estimates. The global consistency is obtained by first using the orientation differences to create a reference trajectory and then use this trajectory to scale and rotate the position differences. The experiment on a real robot showed that our approach can be applied in a realistic environment. Experiments in a controlled environment are still needed to quantify the performance and accuracy of our approach.

References

- [1] S. Birchfield. KLT: an implementation of the Kanade-Lucas-Tomasi feature tracker. Source code: <http://robotics.stanford.edu/~birch/klt/>, 1997.
- [2] R. Bunschoten and B.J.A. Kröse. 3-D scene reconstruction from cylindrical panoramic images. In *Proceedings of the 9th International Symposium on Intelligent Robotic Systems (SIRS'2001)*, pages 199–205, 2001.
- [3] P. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communication, COM*, 31(532-540), 1985.
- [4] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Massachusetts and London, England, 1993.
- [5] S.H.G. ten Hagen and B.J.A. Kröse. Trajectory reconstruction for self-localization and map building. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2002.
- [6] B.K.P. Horn. Recovering baseline and orientation from essential matrix. <http://www.ai.mit.edu/people/bkph/publications.html>, 1990.
- [7] B.J.A. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, April 2001.
- [8] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 1997.
- [9] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, 1994.
- [10] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. Journal of Computer Vision*, 1997.