Strategies for Tracking Tokens in a Cluttered Scene

Zhengyou Zhang
INRIA Sophia-Antipolis, 2004 route des Lucioles
B.P. 93, F-06902 Sophia-Antipolis Cedex, France

Abstract

Tracking is an important approach to analyze long sequences of images in Computer Vision. Although it has extensively been studied in other domains such as in radar imagery, it was introduced only recently in Computer Vision, and is already recognized as an efficient approach to solving correspondence and motion problems. We describe in this paper some strategies for tracking with emphasis on practical importance. They include beam search for resolving multiple matches, support of existence for discarding false matches, and locking on reliable tokens and maximizing local rigidity for handling combinatorial explosion. We have implemented those strategies in a 3D line segment tracking algorithm and found them very useful.
Keywords: Token Tracking, Matching, Cluttered Scenes, Search Strategies

1 Introduction

Statistical data association techniques have been extensively studied in radar imagery for target tracking [1, 2]. Only recently they were introduced in Computer Vision. Early work on motion analysis in Computer Vision was mainly on the computation of motion for two frames obtained from two quite different positions [3, 4, 5]. One dominant difficulty is the establishment of feature correspondences between frames. Many techniques have been proposed which are mainly based on subgraph isomorphism, relational structure matching and tree searching. A number of constraints or heuristics, especially the rigidity assumption, have been incorporated. The correspondence problem is still found to be very difficult. Sooner, researchers realized that the problem would become much easier if, instead, using long sequences of images taken at short time interval. Indeed, as the time interval is small and object velocity is constrained by physical laws, the interframe displacements of objects are bounded, i.e., the correspondence of a token in the next instant must be in its neighborhood. Furthermore, objects usually move smoothly [6, 7], thus the motion coherence can be used to predict the occurrence of tokens in the future, which considerably reduces the search region. The statistical data association techniques for target tracking, originally developed for radar imagery, fit well in this framework, and are already recognized as an efficient approach to solving correspondence and motion problems [8, 9].

However, most of these techniques were originally developed for tracking a few and known targets, although recently progresses have been made to deal with large number of targets [10]. The theoretical base under these techniques is directly applicable to tracking problems in computer vision. A number of particularities, though, are required to be taken care. `[staffs deleted]` The interested reader is referred to [11, 12]. This paper is a continuation of our previous work and we concentrate on a couple of strategies we recently formulated for tracking tokens.

2 Notations and Terminology

We are interested in tracking geometric primitives including points, lines and curves. A group of geometric primitives such as vertex and attached edges is also of interest. We shall call them *tokens*. A token at time t_i is characterized by its position, orientation and kinematic parameters, which are captured in a vector called the *state vector* $\mathbf{x}_i$. An imaging system observes the token which is

BMVC 1993 doi:10.5244/C.7.21

represented by a vector called the *measurement* (or *observation*) *vector* $\mathbf{z}_i$. We call the observation a *scene token*.

The (right) *subscript* is used to denote the time instant, as in $\mathbf{x}_i$ and $\mathbf{z}_i$. At each instant, there are many tokens and scene tokens which will be distinguished to each other by a *left subscript*. For example, ${}_j\mathbf{z}_i$ is the jth scene token observed at time i. One or both subscripts will be omitted if this does not result in any ambiguity. The caret $\hat{}$ denotes the estimation or prediction. For example, $\hat{\mathbf{x}}_{k|k-1}$ denotes the prediction of the state at time k given measurements up to time $k-1$. P denotes the covariance matrix of a state vector and Λ denotes that of a measurement vector.

3 Problem Formulation

The dynamics of a token is assumed to be described by a difference equation

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \ , \tag{1}$$

where $\mathbf{f}_k(\cdot)$ is a vector function describing the transition of the state vector from t_k to t_{k+1} (the so-called *state transition function*), and $\mathbf{w}_k$ is the random disturbance of the dynamic system. In practice, the state transition function is determined by the underlying token kinematics assumed. Two commonly used kinematic models are:

a) Polynomial model: State variables evolve polynomially in time. In general, constant velocity or constant acceleration model is used [8].
b) General motion model: A token is assumed to undergo a motion with polynomial angular velocity and polynomial translational velocity [11, 13]. In practice, constant angular velocity and constant translational velocity or acceleration model is sufficient.

In fact, the polynomial model is a special case of the general motion model where the angular velocity is zero. One advantage of the polynomial model is that the transition function $\mathbf{f}_k(\cdot)$ is linear while we generally cannot write down a linear function using the latter model. However, the latter can more reasonably approximate a real motion than the former. The statistical property of the system noise term $\mathbf{w}_k$ cannot in general be known exactly. We model $\mathbf{w}_k$ as an independent Gaussian noise sequence with zero mean and known covariance, i.e., $E[\mathbf{w}_k] = 0$, and $E[\mathbf{w}_k\mathbf{w}_l^T] = Q_k\delta_{kl}$ for all k and l, where δ_{kl} is the Kronecker delta, which is 1 for $k = l$ and 0 otherwise.

The measurement equation describes the relation between measurements (observations) and state variables of the dynamic system, which can usually be expressed as

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k \ , \tag{2}$$

where $\mathbf{h}_k(\cdot)$ is a vector function called the *observation function* and $\mathbf{n}_k$ represents the random noise contained in the measurements. Measurements are obtained through some signal processing algorithm such as edge detection and 3D reconstruction in a stereo system. The statistical property of $\mathbf{n}_k$ is provided either by the signal processing algorithm if uncertainty is modeled or is guessed on the basis of the designer's experience and physical understanding of the signal processing algorithm. We model $\mathbf{n}_k$ as an independent Gaussian noise sequence with zero mean and known covariance, i.e., $E[\mathbf{n}_k] = 0$, and $E[\mathbf{n}_k\mathbf{n}_l^T] = R_k\delta_{kl}$ for all k and l.

Given a sequence of measurements $\{\mathbf{z}_k \mid k = 1..n\}$ of a token, we are ready to use the Kalman filter if $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$ are linear, or the extended Kalman filter otherwise, to estimate the state variable $\mathbf{x}_k$ of the token. The reader is referred to [14, 15] for the details of the Kalman filter and the extended Kalman filter.

4 Main Steps in Tracking

We shall sketch out in this section the tracking process. By tracking, we mean establishing at each instant a correspondence between tokens being tracked and scene tokens observed. As time goes on, some tokens move out of and some others come into the field of view. Thus we must also deal with the disappearance and appearance problems. The tracking problem becomes more difficult, because some tokens may be occluded by others (the so-called *occlusion problem*) or may not be detected due to temporary failure of the signal processing algorithm (which we refer as the *absence problem*). We shall address these issues in this and next sections.

4.1 Prediction-Matching-Update Loop

The tracking is performed in a prediction-matching-update loop. At time t ($t_{k-1} \leq t < t_k$), i.e., before data at t_k are available, we predict the occurrence at t_k each token being tracked. When data at t_k are available, we try to find for each token a scene token as its match in the neighborhood of its predicted position. When a match is found, the token parameters (state) are updated using either the Kalman filter or the extended Kalman filter (EKF). In the following, both the state transition and measurement observation functions are assumed nonlinear, and the EKF will be used. The discussions, however, are directly applicable to the linear case.

The prediction is done in two stages. First, the state and its error covariance are propagated to t_k according to Eq. (1), denoted by $\hat{\mathbf{x}}_{k|k-1}$ and $P_{k|k-1}$, respectively. We use the first order approximation to compute the prediction of the state error covariance if $\mathbf{f}_{k-1}(\cdot)$ is nonlinear. Second, the predicted position and its covariance matrix of the token are computed according to Eq. (2), denoted by $\hat{\mathbf{z}}$ and Λ_k, respectively. Here again we use the first order approximation if $\mathbf{h}_k(\cdot)$ is nonlinear.

Due to noise from multiple sources, it is very unlikely that a scene token observed at t_k has exactly $\hat{\mathbf{z}}_k$. Given n observed scene tokens at t_k $\{{}_j\mathbf{z}_k \mid j = 1, \ldots, n\}$ with covariance matrices $\{{}_jR_k \mid j = 1, \ldots, n\}$, we use the Mahalanobis distance to decide which scene token matches the token having the predicted measurement vector $\hat{\mathbf{z}}_k$ with covariance matrix Λ_k.

The (squared) Mahalanobis distance between the prediction and the ith scene token is defined as

$$ {}_id_k^M \triangleq {}_i\mathbf{r}_k^T \Lambda_{{}_i\mathbf{r}_k}^{-1} {}_i\mathbf{r}_k \ , \tag{3} $$

where ${}_i\mathbf{r}_k = {}_i\mathbf{z}_k - \hat{\mathbf{z}}_k$ and $\Lambda_{{}_i\mathbf{r}_k} = {}_iR_k + \Lambda_k$. We usually call ${}_i\mathbf{r}_k$ the *measurement residual*. The variable ${}_id_k^M$ is a scalar random variate following a χ^2 distribution with q degrees of freedom, where q is the dimension of the measurement vector. By looking up the χ^2 distribution table, we can choose an appropriate threshold ϵ by setting $\Pr(\chi_p^2 < \epsilon) = \alpha$, where α is typically equal to 95%. If ${}_id_k^M < \epsilon$, then the ith scene token is considered as a match of the token.

A naive matching algorithm yields a linear complexity in the number of scene tokens to match one token being tracked, i.e., $O(n)$. However, the matching process may be slow, especially when there is a large number of scene tokens. This is because the computation of the Mahalanobis distance involves a matrix inversion and is relatively expensive. Many techniques exist to speed up the matching process. One of them is the bucketing technique, which allow us to access directly a subset of scene tokens which are in the neighborhood of the prediction. See [13] for details. Another technique is proposed by Orr et al. [16], which uses the inequality

$$ \mathbf{r}^T \Lambda_{\mathbf{r}}^{-1} \mathbf{r} \geq \frac{\mathbf{r}^T \mathbf{r}}{\mathrm{trace}(\Lambda_{\mathbf{r}})} \ . \tag{4} $$

Thus we can first compute the simplified distance $d' = \mathbf{r}^T\mathbf{r}/\text{trace}(\Lambda_\mathbf{r})$, which is computationally much simpler than Eq. (3). If $d' \geq \epsilon$, so will be ${}_id_k^M$, then the computation of Eq. (3) is not necessary. This avoids the necessity of performing a matrix inverse for every test.

Once a match is found, the (extended) Kalman filter is used to update the token parameters. The updated state and covariance matrix are denoted by $\hat{\mathbf{x}}_k$ and P_k, respectively.

4.2 Initialization

At time t_1, each scene token is used to initialize a token. As described in Sect. 2, the state of a token is composed of its position, orientation and kinematic parameters. The position and orientation parameters of a scene token are assigned to those of its corresponding token. The initialization of the kinematic parameters depends upon the a priori information. If such information is not available, it is reasonable to initialize them to zero, because we are considering a dense sequence and that the interframe motion is small. However, in the state covariance matrix, we should set the diagonal elements corresponding to the kinematic parameters to a fairly big number and the off-diagonal ones to zero, in order to reflect the fact that we know nothing about the kinematics of the token.

4.3 Appearance

Because some new tokens enter into the field of view, their corresponding scene tokens in the current frame cannot be matched with any token being tracked. In this case, each such scene token is used to initialize a new token as described in the previous subsection, which starts the same process as the others.

5 Beam Search and Support of Existence

5.1 Different Cases in Matching and Beam Search Strategy

In using the criterion of the Mahalanobis distance, three cases occur in matching a token:

(i) Unique match: only one scene token is identified as a match of the token.
(ii) No match: no scene token is identified as a match of the token.
(iii) Multiple matches: several scene tokens are identified as plausible matches of the token.

If there is only one match, then there is a high probability that the scene token is the observation of the token being tracked. Thus we just update the token's state by incorporating the scene token.

"No-match" may occur due to a number of reasons. `This paragraph is deleted due to space limitation!`

"Multiple-matches" occurs especially when a token is very uncertain (for example, during the first instants after initialization) or when several scene tokens are near to each other. One (maybe the most common) strategy is *best-first search*, that is, to choose the nearest scene token as in [8] and to discard the other possibilities. This method is efficient but not robust. It may lead to unpredictable results, because the closest scene token is not always the correct match. Another possible strategy is to replace all scene tokens satisfy the criterion of the Mahalanobis distance by a virtual one with a modified probability distribution. This is the idea of the JPDAF method proposed by Bar-Shalom and Fortmann [1]. However, this method introduces a bias in the state estimate because it merges several physically distinct scene tokens as a single one to update the token's state.

A more efficient approach is to exploit the *beam-search strategy*. That is, instead of choosing the nearest scene token, several (2, in our implementation), if any, nearest ones are used. This approach is similar to the *track-splitting filter* in the literature [1]. Different from the JPDAF method, we split the token being tracked into several, as many as the scene tokens found in the search region. Each

split token updates its state with one of the scene tokens chosen. We leave the forthcoming observations to decide which match is correct. The token resulted from the correct match will be confirmed by forthcoming scene tokens, while those resulted from incorrect matches will in general not. Thus the multiple-matches problem is handled gracefully. However, the algorithm is potentially exponential. Some strategy needs to be developed to discard the false tokens.

5.2 Support of Existence

As described in the previous section, our idea of matching is to keep open the possibility of accepting several or no matches for any given token. However, such strategy may lead to a computational explosion. To avoid this we must discard tokens resulted from false matches. We compute for each token a number that we call its *support of existence* which measures the adequateness of the token with the measurements. We have already introduced this measure in our previous work. The reader is referred to [12].

5.3 Discarding Redundant Tokens

In the beam-search approach, a token can be split, each being updated using a scene token satisfying the Mahalanobis distance. On the other hand, a scene token can be used to update several tokens being tracked. This occurs, for example, when a token splits into two (e.g., two fractions of a line segment is observed) and then both new tokens are updated with identical subsequent scene tokens. This implies that the state estimates of two or more tokens tokens may be similar, and it is likely that they represent the same token. We can thus just retain one token and discard the Redundant ones.

6 Trying to Resolve Ambiguity as Early as Possible

Use of the support of existence does prevent the algorithm from a computational explosion. However, it is not efficient enough because we need to process a token resulted from previous false matches during four or more frames before it is discarded. It is of much benefit if we can resolve match ambiguities as early as possible. This section describes two strategies which reduce the match ambiguity and thus reduce the number of tokens to be processed.

6.1 Locking on a Reliable Token

Besides potentially computational explosion, one major drawback of beam-search approach is due to the fact that a scene token can be shared by several different tokens being tracked. Thus it is possible that this approach generates tokens which are not mutually exclusive, nor consistent with each other. The former was already discussed in Sect. 5.3 (discarding redundant tokens). The latter is sometimes a desired feature. If we have not enough information, it is wise to leave the forthcoming observations to resolve the ambiguity.

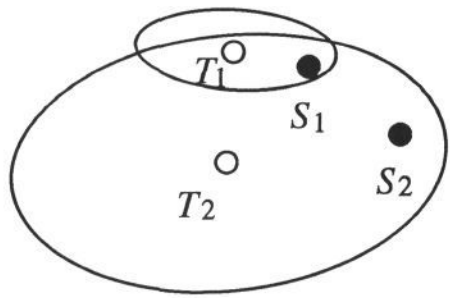


Fig. 1. A scene token can be locked by a secure token. o: tokens; •: scene tokens

However, in the situation as shown in Fig. 1, we can exploit a strategy, which we call the "locking-on-a-token", to obtain a better performance. Here two tokens share one of the measurement (S_1), and one token (T_1) has much less uncertainty than the other one (T_2). When a measurement (scene token) is validated by a secure token, whose state parameters are precise enough, the pairing is almost unambiguous. This measurement is said locked on by the token, and all other tokens search for their correspondences as if this measurement did not exist. In

the situation as shown in Fig. 1, the scene token S_1 is locked on by the token T_1, and then the scene token S_2 is uniquely paired to the token T_2.

There are at least three ways to implement this strategy:

1. Comparing the uncertainty measures. The trace of the covariance matrix roughly measures its magnitude. If $\text{trace}(\text{Cov}(T_1)) \ll \text{trace}(\text{Cov}(T_2))$ (e.g., $\text{trace}(\text{Cov}(T_1)) < \frac{1}{3}\text{trace}(\text{Cov}(T_2))$), then T_1 can lock on the shared scene token.
2. Counting the number of appearances. If during the past N (say, 5) frames, the number of appearance of T_1 (denoted by N_1) is much bigger than that of T_2 (denoted by N_2), e.g., $N_1 \geq 4$ and $N_2 \leq 2$, then T_1 can lock on the shared scene token.
3. Comparing the support of existence l_k (see Sect. 5.2). A secure token implies that it has a high coincidence with the measurements, that is, it should have a low value of l_k (it has a high support for the existence). If the l_k of the token T_1 is less than a small threshold τ, then T_1 can lock on the shared scene token.

The third method has been implemented because the value of l_k is readily available.

6.2 Maximizing the Rigidity

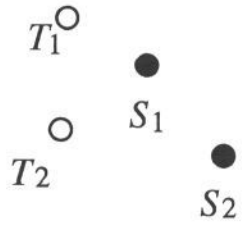


Fig. 2. Combining rigidity and motion continuity to disambiguate matches. o: tokens; •: scene tokens

Rigidity assumption has been used in most matching algorithms, especially in short-sequence motion analysis. Psychological study shows that, among many possible interpretations of any change between two successive frames, the human visual system only accepts a few, often only one, which are consistent with the rigidity assumption [3]. In long-sequence motion analysis like the problem studied in this paper, rigidity assumption is not exploited because the motion continuity or coherence is usually strong enough to resolve matching ambiguities. Here, we combine the rigidity and motion continuity to reduce the ambiguities.

Given a situation as shown in Fig. 2, where two tokens (T_1 and T_2) share the same measurements (S_1 and S_2). If we split tokens, we will obtain four tokens. If the relationship between S_1 and S_2 is not rigid compared with that between T_1 and T_2, then they originate from two different objects (or the object is deformed), and splitting is the only way we can do. However, if they satisfy the rigidity constraints, we can resolve the ambiguity using the motion continuity. The displacement of a rigid object between two successive frames in a sequence with high sample frequency cannot be large due to physical law. A reasonable constraint, for example, is that the rotation angle between two successive frames must be less than some threshold, say 60 degrees. To explain how to exploit this constraint, we refer to Fig. 2 and consider the two-dimensional case. If we assign S_1 to T_1 and S_2 to T_2, the rotation angle is about 45 degrees. On the other hand, if we assign S_1 to T_2 and S_2 to T_1, the rotation angle will be about 135 degrees, which is of course not reasonable. We thus resolve the ambiguity.

The reader is referred to [17, 18] for a complete set of rigidity constraints for 3D line segments. As the data we have are always corrupted with noise, the equalities hardly ever hold true. We have formulated the rigidity constraints by explicitly taking into account the uncertainty of measurements. The reader is referred to [19, 20, 21, 22] for other formalisms of rigidity constraints.

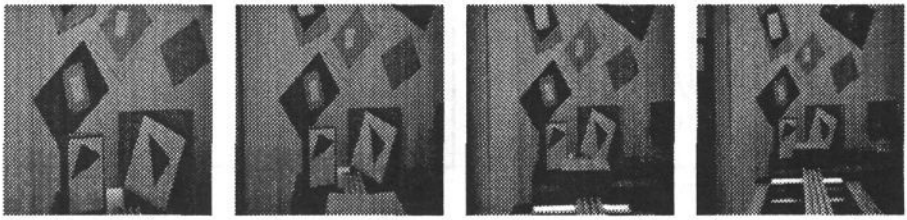
7 Experimental Results

We have incorporated the strategies described above into a tracking algorithm previously developed [11, 12]. The algorithm tracks 3D line segments in a sequence

Table 1. The numbers of scene tokens and active tokens in each frame

frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
scene tokens	37	42	42	44	43	43	48	50	51	58	66	73	102	101	98	103
active tokens	37	54	61	51	51	50	52	60	61	69	79	86	115	134	146	147

of 3D frames reconstructed by a trinocular stereo system. It computes at the same time the 3D kinematic parameters for *each* line segment, and can segment the scene into objects by grouping line segments based on motion similarity.



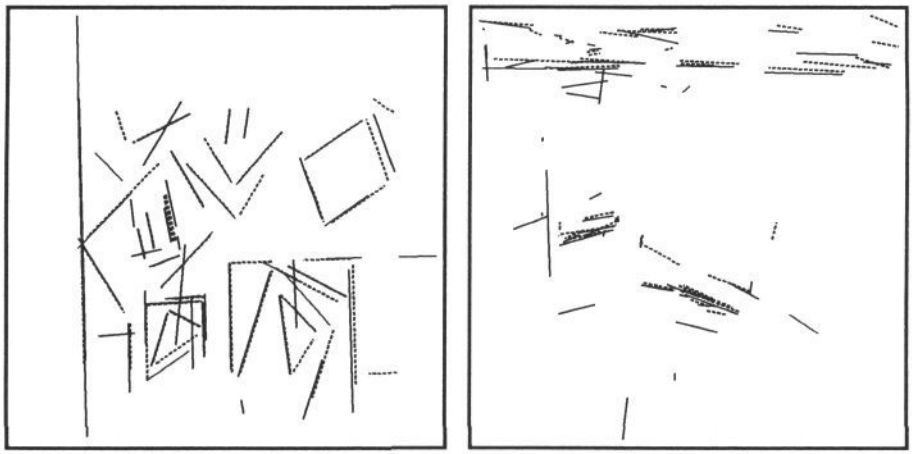
(the first) (the sixth) (the eleventh) (the sixteenth)

Fig. 3. Sample images of the stereo sequence studied

We have tested the modified algorithm on the sequences described in [11, 12] (but it has not been reported). In this paper, we provide the results on a new sequence, consisting of 16 triplets of images. The 1st, 6th, 11th and 16th images taken by the first camera of the stereo rig are shown in Fig. 3. The sequence was acquired by manually moving the stereo rig away from a wall on which we have put several posters to increase the number of line segments. The interframe displacement was supposed a pure translation of 10 centimeters. It is in fact almost true except for the thirteenth frame, as can be seen later. This sequence is interesting in that more and more tokens are visible when time goes on, i.e., the appearance is remarkable. (Several line segments are not observable in the 3D frames due to the *absence problem* described in the introduction section.) If we process the sequence in the reverse direction, more and more tokens would disappear. However, the appearance problem is more difficult to tackle than the disappearance in tracking. The number of line segments reconstructed by the stereo system in each frame is shown in the first row of Table 1.

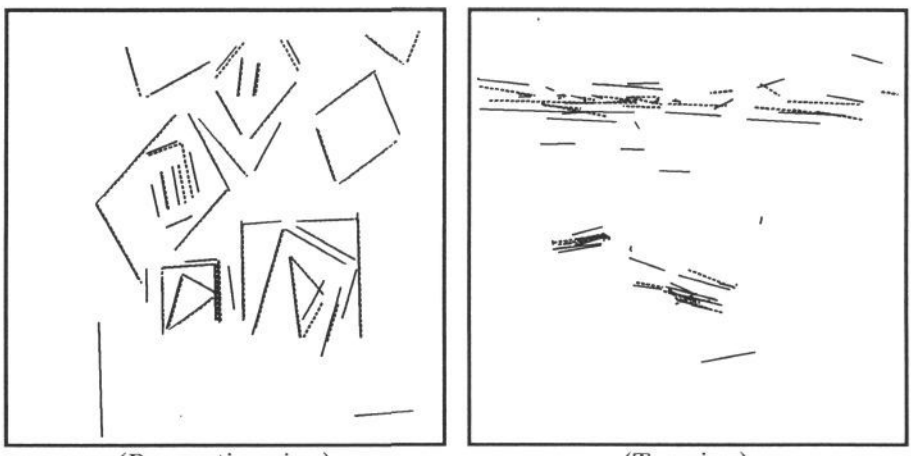
Each segment in the first frame is initialized as a token to be tracked. Since the motion tracking algorithm is recursive, some a priori information on the kinematics is required. A reasonable assumption may be that objects do not move, as the inter-frame motion is expected to be small. The kinematic parameters are thus all initialized to zero, but with fairly large uncertainty: the standard deviation for each angular velocity component is 0.0873 radians/unit-time, and that for each translational velocity component is 150 millimeters/unit-time.

Those tokens are then predicted for the next instant t_2 and the predicted tokens are compared with those in the new frame. Of course, since we have assumed no motion, the predicted position and orientation of each token remains unchanged, but its uncertainty changes and becomes very large. As expected, multiple matches occur for most of tokens. Techniques based only on the best match usually fail at this stage, since the nearest segment is not always the correct match. We retain the two best matches if a token has multiple matches. Furthermore, the strategies described in this paper are exploited to reduce the matching ambiguities. The token updates its kinematic parameters using its best match. A new token is initialized by combining the token and its second best match which is used to estimate its kinematic parameters. We continue the tracking in the same manner. Usually the tokens originated from false matching in the preceding instants are losing their support for existence as more frames are processed, and are eventually



(Perspective view) (Top view)

Fig. 4. The superposition of the predicted (in solid lines) and the observed (in dashed lines) segments at time t_3



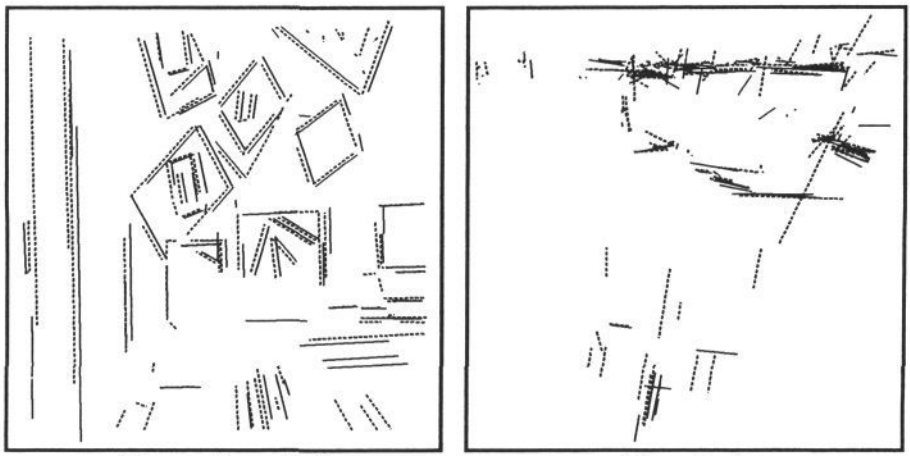
(Perspective view) (Top view)

Fig. 5. The superposition of the predicted (in solid lines) and the observed (in dashed lines) segments at time t_5

deactivated. The number of active tokens after processing each frame is shown in the second row of Table 1. The number does not become overwhelming, even though there is a significant increase in the number of scene tokens.

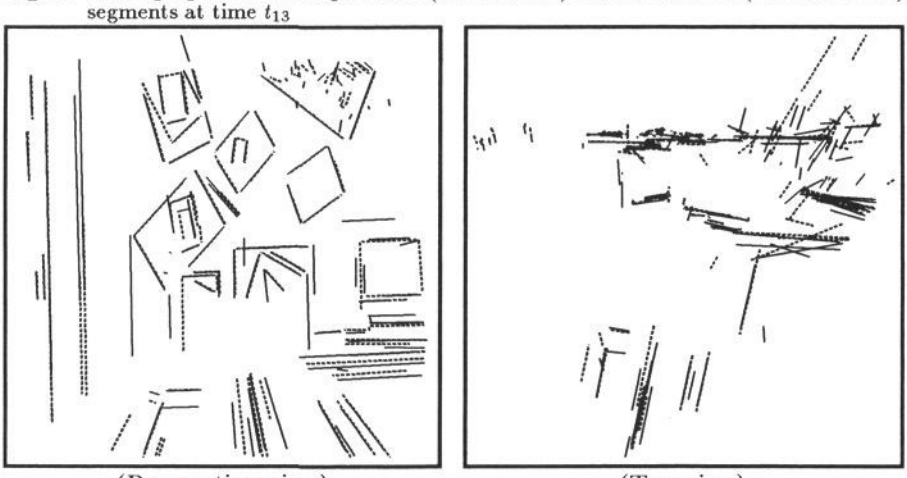
In Fig. 4, we show the superposition of the predicted (in solid lines) and the observed (in dashed lines) segments at time t_3. As can be observed, more active tokens (in solid lines) exist at this moment: some have been activated due to multiple matches at time t_2 and some just entered the field of view. We observe that the tokens originated from good matching coincide well with the scene tokens. After having processed the fourth frame, a number of false tokens disappear, as shown in Fig. 5, where the predictions for t_5 are overlayed on the observations at t_5.

As said earlier, the thirteenth frame was taken in a shifted position. Figure 6 shows the superposition of the predicted (in solid lines) and the observed (in dashed lines) segments at time t_{13}. Compared with the results shown in Fig. 4 and Fig. 5, we can observe a relatively big difference between the prediction and the observation. After several frames, such occasional incoherent motion will be



(Perspective view) (Top view)

Fig. 6. The superposition of the predicted (in solid lines) and the observed (in dashed lines) segments at time t_{13}



(Perspective view) (Top view)

Fig. 7. The superposition of the predicted (in solid lines) and the observed (in dashed lines) segments at time t_{16}

compensated for by the algorithm. Figure 7 shows the superposition of the predicted (in solid lines) and observed (in dashed lines) segments at t_{16}. Quite a good fitting between the prediction and observation can be observed.

As described in[11, 12], we can group the individual tokens into objects based on the motion coherence. Here there is only one object. The final estimate of the interframe rotation is 1.1 milliradians, or 0.063 degrees. The final estimate of the interframe translation is 99.52 millimeters. Recall that the supposed displacement is a pure translation of 100 millimeters.

8 Conclusion

In this paper we have presented our recent work on token tracking in a cluttered scene in the statistical data association framework. The main steps have been summarized. We have focused in this paper on several strategies including beam search for resolving multiple matches, support of existence for discarding false matches, locking tokens and maximizing local rigidity for handling combinatorial explosion. We have implemented those strategies in a 3D line segment tracking algorithm and found them very useful. Some new results have been provided.

References

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Academic, New York, 1988.

[2] S. S. Blackman, *Multiple-Target Tracking with Radar Application*. Artech House, Norwood, MA, 1986.

[3] S. Ullman, *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, 1979.

[4] R. Tsai and T. Huang, "Estimating 3-D motion parameters of a rigid planar patch, i," *IEEE Trans. ASSP*, vol. 29, pp. 1147–1152, December 1981.

[5] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.

[6] A. van Doorn and J. Koenderink, "Spatiotemporal integration in the detection of coherent motion," *Vision Research*, vol. 24, no. 1, pp. 47–53, 1984.

[7] M. Jenkin and J. Tsotsos, "Applying temporal constraints to the dynamic stereo problem," *Comput. Vision, Graphics Image Process.*, vol. 24, pp. 16–32, 1986.

[8] R. Deriche and O. Faugeras, "Tracking line segments," in *Proc. First European Conf. Comput. Vision*, (O. Faugeras, ed.), (Antibes, France), pp. 259–268, Springer, Berlin, Heidelberg, April 1990.

[9] Z. Zhang and O. Faugeras, "Tracking and motion estimation in a sequence of stereo frames," in *Proc. 9th European Conf. Artif. Intell.*, (L. Aiello, ed.), (Stockholm, Sweden), pp. 747–752, August 1990.

[10] J. Uhlmann, "Algorithms for multiple-target tracking," *American Scientist*, vol. 80, pp. 128–141, March-April 1992.

[11] Z. Zhang and O. Faugeras, "Tracking and grouping 3D line segments," in *Proc. Third Int'l Conf. Comput. Vision*, (Osaka, Japan), pp. 577–580, IEEE, December 1990.

[12] Z. Zhang and O. Faugeras, "Three-dimensional motion computation and object segmentation in a long sequence of stereo frames," *Int'l J. Comput. Vision*, vol. 7, pp. 211–241, March 1992.

[13] Z. Zhang, *Motion Analysis from a Sequence of Stereo Frames and its Applications*. PhD thesis, University of Paris XI, Orsay, Paris, France, 1990. in English.

[14] P. Maybeck, *Stochastic Models, Estimation and Control*. Vol. 1, Academic, New York, 1979.

[15] P. Maybeck, *Stochastic Models, Estimation and Control*. Vol. 2, Academic, New York, 1982.

[16] M. Orr, J. Hallam, and R. Fisher, "Fusion through interpretation," in *Proc. Second European Conf. Comput. Vision*, (Santa Margherita Ligure, Italy), pp. 801–805, May 1992.

[17] Z. Zhang, O. Faugeras, and N. Ayache, "Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints," in *Proc. Second Int'l Conf. Comput. Vision*, (Tampa, FL), pp. 177–186, December 1988.

[18] Z. Zhang and O. Faugeras, "Estimation of displacements from two 3D frames obtained from stereo," *IEEE Trans. PAMI*, vol. 14, pp. 1141–1156, 1992.

[19] W. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int'l J. Robotics Res.*, vol. 5, pp. 3–34, Fall 1984.

[20] S. Pollard, J. Porrill, J. Mayhew, and J. Frisby, "Matching geometrical descriptions in three-space," *Image and Vision Computing*, vol. 5, pp. 73–78, may 1987.

[21] H. Chen and T. Huang, "Maximal matching of 3-D points for multiple-object motion estimation," *Pattern Recog.*, vol. 21, no. 2, pp. 75–90, 1988.

[22] D. Murray and D. Cook, "Using the orientation of fragmentary 3D edge segments for polyhedral object recognition," *Int'l J. Comput. Vision*, no. 2, pp. 153–169, 1988.