A Non-Iterative Greedy Algorithm for Multi-frame Point Correspondence

Khurram Shafique and Mubarak Shah
School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL-32816
khurram,shah@cs.ucf.edu

Abstract

This paper presents a framework for finding point correspondences in monocular image sequences over multiple frames. The general problem of multi-frame point correspondence is NP Hard for three or more frames. A polynomial time algorithm for a restriction of this problem is presented, and is used as the basis of proposed greedy algorithm for the general problem. The greedy nature of the proposed algorithm allows it to be used in real time systems for tracking and surveillance etc. In addition, the proposed algorithm deals with the problems of occlusion, missed detections, and false positives, by using a single non-iterative greedy optimization scheme, and hence, reduces the complexity of the overall algorithm as compared to most existing approaches, where multiple heuristics are used for the same purpose. While most greedy algorithms for point tracking do not allow for entry and exit of points from the scene, this is not a limitation for the proposed algorithm. Experiments with real and synthetic data show that the proposed algorithm outperforms the existing techniques and is applicable in more general settings.

1 Introduction

In motion correspondence, given an image sequence, the problem is to find the correspondences between the feature points in the images that occur due to the same object in the real world at different time instances. We assume that the only information available about the feature points, is their position in the image, and there is no other distinguishing feature among these points. This assumption is particularly useful in applications like particle tracking, or tracking of dense field of similar objects.

We formulate the problem as follows. The same notation will be used throughout the rest of the paper. Let a sequence of n frames F_i (each of dimensions $S_x \times S_y$), $1 \leq i \leq n$, and let $X_i = \{x_1^i, x_2^i, \ldots, x_r^i\}$ be the set of r points detected in frame F_i (The number of points detected in each frame need not be the same). We define a *track* T of length m, to be a sequence of m points $\langle x_{a_1}^{i_1}, x_{a_2}^{i_2}, \ldots, x_{a_m}^{i_m} \rangle$, such that $1 \leq i_1 < i_2 < \ldots \leq n$ and $1 \leq a_j \leq |X_{i_j}|$. The *backward correspondence* and *forward correspondence* of a point $x_{a_j}^{i_j}$ in track T are respectively defined by the points preceding and succeeding $x_{a_j}^{i_j}$, i.e. $x_{a_{j-1}}^{i_{j-1}}$ and $x_{a_{j+1}}^{i_{j+1}}$.

The point correspondence problem is to find a set of tracks $A = \{T_1, T_2, \ldots, T_m\}$, such that $\forall T_i \in A$, either one of the following is true:

- If $\exists x_j^k \in T_i$, such that x_j^k is a 2D projection of *only* point Z_i in the real world, then every point in T_i is a 2D projection of Z_i (or more points in occlusion with Z_i), and no other track T_r contains a 2D projection of *only* Z_i.
- $\forall x_j^k \in T_i$, x_j^k is not a 2D projection of any real world point Z_i.

The first condition requires each real world point to have exactly one track associated with it, and vice versa. The second condition disallows any overlap between tracks corresponding to a real world point and tracks composed of noise. The distinction between these two types of tracks is usually done by higher level processes and is not in the scope of this paper.

The major contribution of this paper is formulation of a framework for efficient and robust solution to the multi-frame correspondence problem as defined above. We propose a look-ahead technique to solve the correspondence problem by using a sliding window over multiple frames. Our framework deals with the problems of occlusion handling, missed detections and false positives, by using a single greedy optimization scheme as compared to most existing approaches, where different heuristics are used for the same purpose.

The organization of the paper is as follows. In the next section, we present a survey of the related work. In Section 3, we define the terminology and notation for this paper, and provide a graph theoretical formulation of the correspondence problem and its solution in Section 4. We refine the solution of Section 4 and present details of our algorithm in Section 5. In Section 6, we demonstrate the results of the proposed approach on a variety of synthesized and real sequences, and compare our results to the previous approaches. Section 7 concludes the paper.

2 Related Work

A large number of correspondence methods have been proposed in recent years. Ullman [15] proposed a minimal

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

mapping approach, where the probabilistic cost function was based on the distance between the points in consecutive frames. A linear programming approach was used to minimize the cost function. The cost function was further improved by Jenkin [7], who introduced the smoothness constraint along with the nearest neighbor relationship and used a greedy approach for optimization. Barnard and Thompson [1] used relaxation based approach to solve this problem. Sethi and Jain [13] proposed an iterative greedy exchange algorithm using both nearest neighbor and smoothness constraints. The self initializing version of the algorithm repeats the optimization step in forward and backward directions until an equilibrium state is achieved. The algorithm, however, assumes that the points do not enter or leave the scene and that there is no occlusion and detection errors. The latter condition was relaxed by Salari and Sethi in [12].

Our work is closely related to the GOA Tracker of Veenman et al [16], who proposed the Hungarian search as an optimization tool for their GOA Tracker, along with the different motion models defined in [3, 10, 13]. However, we present a solution to the "Multi-frame" correspondence problem, as opposed to the 2-frame correspondence problem in [16]. The latter is a special case of the former and is inherently an easy problem, for which a polynomial time optimal solution exists. In addition, GOA assumes that the number of points in the scene remains constant, which is not a restriction for the proposed algorithm. Further, self initializing version of GOA is a two-pass algorithm compared to the proposed algorithm, which is a single pass algorithm and hence, is applicable in real time systems.

Apart from these methods, quite a few algorithms have been proposed in the statistical domain [4], of which the most well known is Multiple Hypothesis Tracking (MHT) [11], which though being optimal, suffers from very high computational complexity. More efficient approximations of MHT have been presented. Some of these techniques use Murti's Algorithm to find the k best hypotheses to reduce the search space [5], while others reduce the search space by using a limited temporal scope and a sliding window technique. However, the problem remains intractable and further approximations are used for efficient implementation [6, 9].

3 Definitions and Notations

Let $D = (V, E)$ be an edge weighted directed graph without self loops and multiple edges, where V and E are respectively, the set of vertices and edges of digraph D. A *vertex disjoint path cover* C of D is a set $\{P_1, P_2, \ldots, P_k\}$ of directed paths P_i (of length ≥ 0), if $V = \bigcup_{i=1}^{k} V(P_i)$ and $V(P_i) \cap V(P_j) = \emptyset$ whenever $i \neq j$, where $V(P_i)$ is the set of vertices of directed path P_i. For simplicity of notation, we will refer to vertex disjoint path cover as path cover. Let $W(C)$ denote the weight of path cover C, where $W(C)$ is defined by the sum of weights of all the edges in the cover C. A *maximum weight path cover* of D is a path cover C', such that $C' = \arg\max_{C_i} W(C_i)$, for all path covers C_i of D.

A *Split* of an edge weighted digraph D, is an edge weighted bipartite graph G, whose partite sets V^+, V^- are copies of $V(D)$. For each vertex $x \in V(D)$, there is one vertex $x^+ \in V^+$ and one vertex $x^- \in V^-$. For each edge e from u to v in D, there is a corresponding edge e' with endpoints u^+, v^- in G, such that $w(e') = w(e)$.

A *matching* in a graph G is a set of edges with no shared end-vertices. A *maximum matching* in a weighted graph is a matching with the maximum weight among all matchings in the graph.

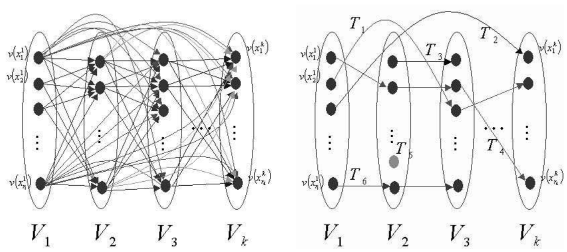
4 Graph Theoretical Formulation

There is an obvious graph theoretical formulation of 2-frame correspondence problem [15, 16]. The problem can be viewed as finding a maximum matching of a bipartite graph G. Where the partite sets V_1, V_2 correspond to the sets of points X_1 and X_2, detected in frames F_1 and F_2 respectively. An edge between two points correspond to a match hypothesis between those points, and the weight of the edge is the gain associated with this match. The total gain among all points is maximized by the maximum matching of graph G, which can be found in polynomial time [8]. Unfortunately, the extension of this approach to multiple frames ($k-$Dimensional matching problem) is NP-Hard for $k \geq 3$. The other drawback of using $k-$D matching is that, it requires the point to be visible in all k frames and hence, does not allow for occlusions, missed detections and tracks of length less than k. Researchers have worked their way around these problems by using Lagrangian approximations to reduce the complexity and by introducing multiple heuristics to incorporate occlusions and missed detections, e.g., [6, 9].

In our approach, instead of using a $k-$D hypergraph to model the $k-$frame problem, we construct a weighted digraph $D = (V, E)$, such that $\{V_1, V_2, \ldots, V_k\}$ partitions V, and each vertex $v(x) \in V_i$ corresponds to a point $x \in X_i$, detected in frame F_i. Further, $E = \left\{v(x_a^i)v(x_b^j) | v(x_a^i) \in V_i \wedge v(x_b^j) \in V_j, \ \forall i < j\right\}$, i.e., there is a directed edge from every vertex in set V_i to every vertex in set V_j, such that $i < j$. Once again, each edge $e = v(x_a^i)v(x_b^j)$ corresponds to a match hypothesis of point x_a^i in frame F_i to point x_b^j in frame F_j, where the edge weight $w(e)$ is the gain $g(x_a^i, x_b^j)$ associated with this match. A sample digraph formed this way is shown in Figure 1a.

By the definition of correspondence problem in section 1, the task is to find a set of vertex disjoint directed paths (Tracks) of length 0 or more, such that the total gain

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE



(a) (b)

Figure 1: (a) An instance of digraph D as defined in Section 4 and (b) a candidate solution

is maximum among all such paths, i.e., we want to find a maximum weight path cover of the directed graph D. A sample solution to the problem is shown in Fig 1b.

Once again, the problem of finding maximum path cover is NP-Hard, even in the case of unweighted graphs [2]. However, by the following theorem, a polynomial solution exists if the directed graph is acyclic:

Theorem 1 *The edges of maximum matching of the split graph G of an acyclic edge-weighted digraph D correspond to the edges of a maximum path cover of D.*

The proof of the above theorem is straight forward and is omitted for the sake of brevity. By the construction of digraph D, all the edges in D are in the direction of increasing time, thus D is acyclic. Hence, given the weighted directed graph D, an optimal set of tracks that maximizes the overall gain can be obtained in polynomial time.

Let $T = \langle x^{i_1}_{a_1}, x^{i_2}_{a_2}, \ldots, x^{i_m}_{a_m} \rangle$ be a track corresponding to some real world point Z_i, we require that $\forall p, q,\ 1 < p+1 < q \leq m$, the gain function $g(x^{i_p}_{a_p}, x^{i_q}_{a_q})$ satisfies the following inequality:

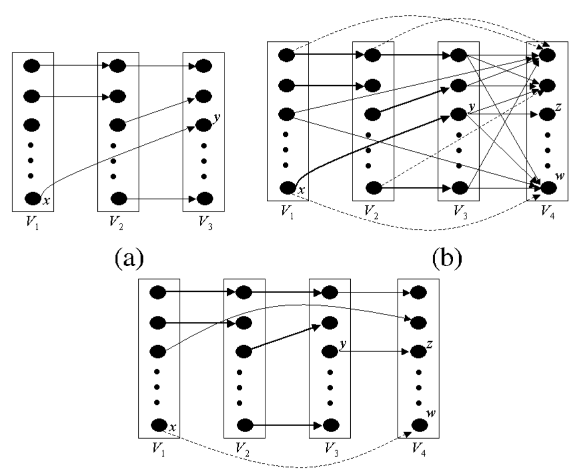
$$g(x^{i_p}_{a_p}, x^{i_q}_{a_q}) < g(x^{i_p}_{a_p}, x^{i_{p+1}}_{a_{p+1}}) + g(x^{i_{q-1}}_{a_{q-1}}, x^{i_q}_{a_q}) \qquad (1)$$

This condition guarantees that the total gain is maximized only if all the edges of T are in the path cover and penalizes the choice of a shorter track when a longer valid track is present.

5 Greedy Algorithm

The construction of digraph, as mentioned in section 4, assumes the gain $g(x^i_a, x^j_b)$ to be independent of backward correspondences of x^i_a. For simpler cases, such as gain function based on the nearest neighborhood criteria [15] or correlation, this condition is satisfiable. However, this condition is not satisfied if the gain function, $g(x^i_a, x^j_b)$, requires velocity or acceleration of point x^i_a (which is computable only if the backward correspondence of x^i_a is known). We present a solution to this problem by proposing a greedy algorithm based on the framework of section 4.

Assume first, that the correspondences of points $X_1, X_2, \ldots, X_{k-1}$, in $k-1$ frames, $F_1, F_2, \ldots, F_{k-1}$, $k > 2$, have been established, and let C_{k-1} be the set of these correspondences. These correspondences of $k-1$ frames were made by the information available till time instant t_{k-1}, and may be changed once more information is available. Also, let F_k be the current frame and construct a digraph $D = (V, E)$ as follows: $V = V_1 \cup V_2 \cup \ldots \cup V_k$ such that $V_1, V_2, \ldots, V_k$ are pairwise disjoint and each vertex $v(x) \in V_i$ corresponds to a point $x \in X_i$. For every vertex pair $v(x) \in V_i$, $v(y) \in V_j$, there is an edge from $v(x)$ to $v(y)$, if $i < j = k$ or there is a correspondence from x to y in C_{k-1}. Hence, apart from the edges defining the correspondences in C_{k-1}, say *old edges*, all the other edges have some vertex in V_k as their end-vertex . For a point, say x, which has a forward correspondence in C_{k-1}, the new edges from $v(x)$ to the vertices in V_k represent the possibility of forward correspondence in C_{k-1} to be false and that the point x was mis-detected or occluded till frame F_k. We refer to such edges as *correction edges*. For the other points, these edges represent the extension of correspondence in C_{k-1} to frame F_k, and are referred to as *extension edges*. The digraph obtained this way is called *extension digraph* (Figure 2). Since, all the backward correspondences except for the points in X_k have been established, all the edge weights in D can now be computed, regardless of the type of gain function used.



(a) (b) (c)

Figure 2: (a) An initial correspondence. (b)The extension digraph D. (c) A maximum path cover of extension digraph. The correction edges are shown as dotted lines, while the old edges are shown by bold lines. (Not all edges and vertices are shown)

Once the digraph, D, is constructed, we again seek a set of vertex disjoint paths (tracks) in D that maximizes the total gain. A candidate solution C_k may contain all three types of edges, i.e., old edges, correction edges and extension edges. While an extension edge does not change any correspondence in C_{k-1}, a correction edge $e' = v(x)v(w)$ always replace some old edge $e = v(x)v(y)$. Suppose now,

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

that the point y has a forward correspondence z in C_k. Since this correspondence was obtained by assuming the correspondence xy in C_{k-1}, and since the correspondence xy is voided in C_k, the correspondence yz and all such forward correspondences must be removed from C_k and if possible be replaced with new edges. We define an edge to be a *false hypothesis*, if it has a directed path from an edge that is replaced by a correction edge, e.g., the edge yz in Figure 2(c) is a false hypothesis. The replacement of false hypotheses with the new edges can be performed by the following recursive scheme:

Procedure FalseHypothesisReplacement()
for $i = 1$ *to* $k - 2$
begin
While there is a false hypothesis originating from some vertex in V_{i+1}
begin
Delete all the false hypotheses in graph D
Solve the $(k-i+1)-$ *frame correspondence problem for all the uncorresponded vertices in sets* $V_i, V_{i+1}, \ldots, V_k$
end
end

We have found through experiments that the following efficient greedy step also performs reasonably well in most cases, even for $m = 2$:

Procedure GreedyFalseHypothesisReplacement(m)
for $i = 1$ *to* $k - m + 1$
begin
Delete all the false hypotheses in graph D
Solve the $m-$ *frame correspondence problem for all the uncorresponded vertices in sets* $V_i, V_{i+1}, \ldots, V_{i+m-1}$
end

Once the greedy step is completed, we obtain a new correspondence set, C_k, of all points up to frame F_k. Similarly a correspondence can be extended for any number of frames by adding one frame at a time.

The initialization is done by first using the 2-frames algorithm to obtain the correspondence of first two frames F_1 and F_2. This correspondence is then extended for each new frame by using the algorithm described above till the k^{th} frame. At this stage, a backtracking is performed by applying the same algorithm in the reverse direction, i.e., on frames $F_k, F_{k-1}, \ldots, F_1$, using the established correspondences. This takes care of any wrong correspondence that was made when no or less motion information was available, and is done just once for the first k frames.

6 Results

In this section, we present the results of our algorithm on both synthetic and real sequences. Given a point $x^{ij}_{a_j}$, and its predicted position in frame b, x^b, we define the gain function $g(x^{ij}_{a_j}, x^b_a)$ to be the convex combination of two terms, referred to as *directional coherence* and *speed consistency*, as follows:

$$g(x^{ij}_{a_j}, x^b_a) = \alpha \left[\frac{1}{2} + \frac{x^{ij}_{a_j}x^b \cdot x^{ij}_{a_j}x^b_a}{2\|x^{ij}_{a_j}x^b\|\|x^{ij}_{a_j}x^b_a\|}\right] + (1-\alpha)\left[1 - \frac{\|x^{ij}_{a_j}x^b - x^{ij}_{a_j}x^b_a\|_p}{\sqrt{S_x^2+S_y^2}}\right],\ \alpha \in [0,1]$$

To satisfy the constraint of equation 1, we add a small constant penalty ϵ to the gain function $g(x^{ij}_{a_j}, x^b_a)$ whenever $b > i_j + 1$. For all of our experiments, we used constant acceleration motion model, $\alpha = 0.1$, $\epsilon = -10^{-3}$, and a sliding window of size 5. The procedure *GreedyFalseHypothesisReplacement*($m = 2$) is used instead of its recursive counterpart. The results are compared with the self initializing version of GOA tracker with the smooth motion model, as defined in [16]. Since Veenman et al have shown experimentally [16] that, in most situations, the GOA Tracker outperforms the algorithms in [3, 10, 11, 13], a comparison against GOA Tracker implies a comparison against all these algorithms.

The synthetic sequences in this section are generated by a data set generator called *Point Set Motion Generator (PSMG)* [17]. The generator provides control over the size of image space, number of points, number of frames, mean and variance of initial velocity, mean and variance of the change in velocity, probability of occlusion etc. For every experiment, we consider the following three scenarios separately; i) Points are not allowed to enter or leave the scene, though they may be occluded or miss-detected. ii) Points are allowed to leave the scene but new points may not enter. iii) Points are allowed to leave the scene and for every point that leaves the scene, a new point enters the scene. To analyze the performance of tracking and to compare the results, we use *track-based error*, E_T [17], defined as $E_T = 1 - \frac{T_c}{T_t}$, where T_t is the total number of true tracks, and T_c is the number of completely correct tracks generated by the tracker. The error is calculated by averaging the error of 100 sequences generated by using the same parameters.

Since GOA-tracker does not allow the points to enter or leave the scene, the output of GOA is only shown for the first scenario. To analyze the noise handling capability of the algorithms, we consider the scenario when the new points are generated in the middle of the sequence and use a modified track based error E^c_T, for both GOA and the proposed tracker. E^c_T is defined as $E^c_T = 1 - \frac{T^c_c}{T^c_t}$, where T^c_t is the total number of true tracks of points that were visible in both first and last frame, and T^c_c is the number of completely correct such tracks generated by the tracker. The points that enter or leave the image in these sequences are then considered as noise, while only the points that are visible in both first and last frames are considered as valid tracks.

Our first experiment demonstrates the effectiveness of the proposed initialization scheme (i.e., backtracking after

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

first k frames). The experiments were performed for varying number of points with three different modes, i) Manual Initialization, ii) Self Initialization by backtracking and iii) No Initialization. The Track errors are shown in Figure 3. The results show that the proposed initialization scheme is almost as good as the manual initialization and improves the results significantly as compared to no initialization.

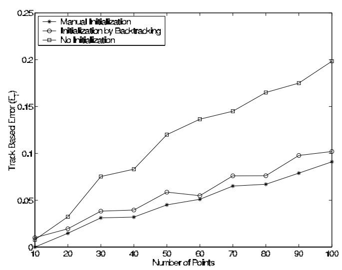
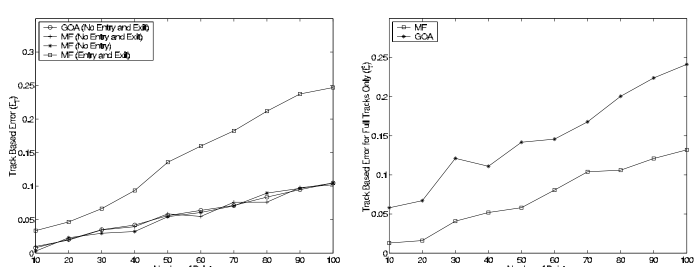


Figure 3: Track errors with different modes of initialization. The upper curve is obtained by no initialization, while the middle and lower curves are the errors of backtracking and manual initialization respectively.

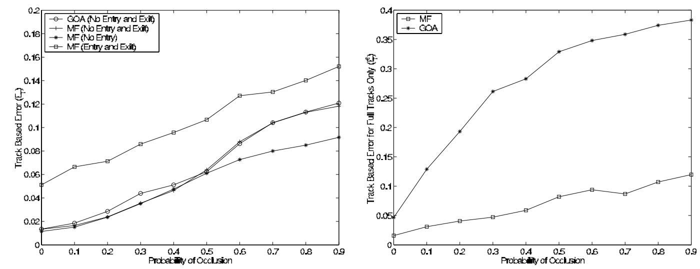
In the next experiment, we analyze the performance of the proposed multiframe algorithm (MF) with respect to point density. The experiments were performed by increasing the number of point tracks in a fixed image space. In Figure 4(a), the track based errors E_T, are shown for applying MF on three different types of sequences, as described above. In addition, the track based errors of GOA tracker are also shown on the sequences, where points are not allowed to leave or enter the scene. In Figure 4(b), we show the effect of noise on both trackers by using the modified track based error E_T^c, and allowing the points to enter and exit the scene.



(a) (b)

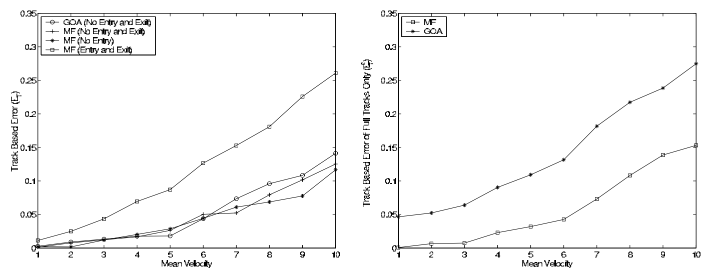
Figure 4: Variable point density performance: (a) E_T of the proposed Multiframe algorithm (MF) and GOA algorithm. The lower three curves are the errors of GOA (with no entry and exit) and MF (with no entry and exit, and no exit), while the upper curve is the error of MF (with both entry and exit). (b) Effect of noise: E_T^c, when points are allowed to leave and enter the scene. The upper curve is the error of GOA, and lower curve is the error of MF.

Similar experiments were performed on occlusion handling (Figure 5) and variable velocity performance (Figure 6), where the probability of occlusion was varied in former and the mean velocity was increased in the latter.



(a) (b)

Figure 5: Occlusion Handling: (a) E_T, (b) E_T^c.



(a) (b)

Figure 6: Variable Velocity Performance: (a) E_T, (b) E_T^c.

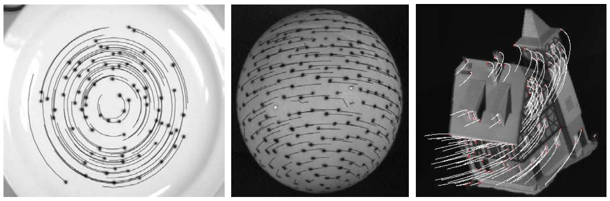
The results show that the proposed algorithm performs equally well as GOA tracker when the points are not allowed to enter or leave the scene. However, the performance of the proposed tracker is unaltered when the points are allowed to leave the scene, or when additional noise is introduced. The proposed tracker also performs reasonably well on sequences where points are allowed to leave and enter the scene simultaneously, given the higher degree of ambiguity in such sequences. In addition, the results clearly show that the proposed algorithm outperforms GOA tracker in the presence of noise.

Next, we show the results of our algorithm on real data. Our first set of experiments is based on the standard sequences in dense point correspondence literature. In the first experiment, we use a sequence from [16], where 80 black seeds are placed on a rotating dish. Figure 7 (a) shows that all 80 seeds were correctly tracked over the sequence (This claim is also verified by the ground truth).

In the next two standard real sequences, we used KLT method [14] to only detect the feature points, then used our algorithm to establish correspondences. The visual analysis of both outputs (Figure 7 (b) and (c)) show that most of the tracks were perfectly tracked through out the sequence.

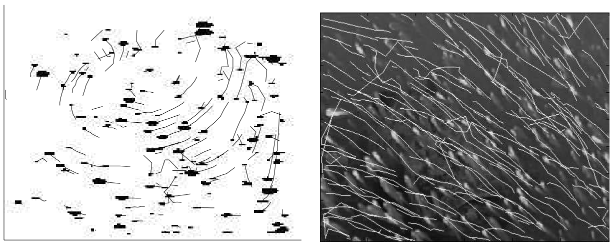
Our second set of experiments is based on natural sequences with very dense feature points and high occlusion scenarios. The moving objects are detected by background

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE



(a) (b) (c)

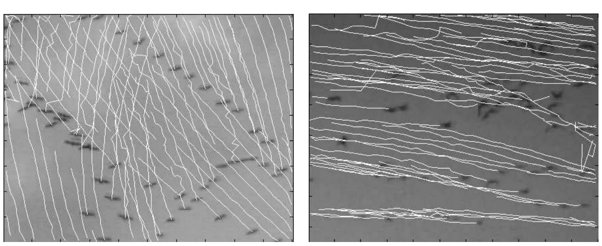
Figure 7: Tracks generated for (a) rotating dish sequence. (b) rotating ball sequence. (c) house sequence.



(a) (b)

Figure 8: Tracks generated for (a) cylindrical reservoir sequence. (b) flock of fish.

subtraction and their centroids are used as the feature points for tracking. Our first example in this set is from particle tracking (Figure 8(a)). It is a ten frame sequence showing particles in a cylindrical reservoir containing liquid, and a tubular heater, which drives counter-clockwise rotating convection cells. There are more than 100 particles in each frame (some of them are almost stationary, while others appear for one or two frames only). In Figure 8(b), we show the tracking results of a flock of more than 150 fish in the sea. The next two examples (Figure 9) show the tracking results for bird flocks, where birds are at different altitudes and are having frequent occlusions.



(a) (b)

Figure 9: Tracks generated for two sequences of bird flocks

7 Conclusion

We have presented a framework for efficient and robust solution of multi-frame point correspondence problem. The proposed framework provides an optimization algorithm that optimizes the gain function over multiple frames and may be used for a large variety of motion models and cost functions that satisfy the constraints as posed by it. The presented algorithm is applicable in more general settings and is shown to perform well by extensive experimentation using synthetic data. Results on real data also support the experimental evaluation.

References

[1] S. T. Barnard and W. B. Thompson. "Disparity analysis of images". *IEEE Trans. on PAMI*, 2(4):333–340, July 1980.

[2] F. T. Boesch and J. F. Gimpel. "Covering the points of a digraph with point disjoint paths and its application to code optimization". *JACM*, 24(2):192–198, 1977.

[3] D. Chetverikov and J. Verestóy. "Feature point tracking for incomplete trajectories". *Computing*, 62(4):321–338, 1999.

[4] I. J. Cox. "A review of statistical data association techniques for motion correspondence". *IJCV*, 10(1):53–66, Feb 1993.

[5] I. J. Cox and S. L. Hingorani. "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking". *IEEE Trans. on PAMI*, 18(2):138–150, Feb 1996.

[6] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. "A generalized S-D assignment algorithm for multisensor-multitarget state estimation". *IEEE Trans. on Aero. and Elect. Sys.*, 33(2):523–538, 1997.

[7] M. R. M. Jenkin. "Tracking three dimensional moving light displays". In *Proceedings, Workshop Motion: Representation Contr*, pages 66–70, 1983.

[8] H. W. Kuhn. "The Hungarian method for solving the assignment problem". *Naval Reseach Logistics Quarterly*, 2:83–97, 1955.

[9] A. B. Poore and X. Yan. Data association in multi-frame processing. In *Proceedings of the Second International Conference on Information Fusion*, pages 1037–1044, 1999.

[10] K. Rangarajan and M. Shah. "Establishing motion correspondence". *CVGIP*, 54(1):56–73, july 1991.

[11] D. B. Reid. "An algorithm for tracking multiple targets". *IEEE Trans. on Auto. Contl.*, 24(6):843–854, Dec 1979.

[12] V. Salari and I. K. Sethi. "Feature point correspondence in the presence of occlusion". *IEEE Trans. on PAMI.*, 12(1):87–91, Jan 1990.

[13] I. K. Sethi and R. Jain. "Finding trajectories of feature points in a monocular image sequence". *IEEE Trans. on PAMI*, 9(1):56–73, Jan 1987.

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

[15] S. Ullman. *The Interpretation of Visual Motion*. M.I.T. Press, Cambridge, MA.

[16] C. J. Veenman, M. J. T. Reinders, and E. Backer. "Resolving motion correspondence for densely moving points". *IEEE Trans. on PAMI*, 23(1):54–72, Jan 2001.

[17] J. Verestóy and D. Chetverikov. "Experimental comparative evaluation of feature point tracking algorithms". In *Proc. Workshop on Eval. and Validation of Computer Vision Algos*, pages 183–194, 2000.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**