Efficient Second Order Multi-Target Tracking with Exclusion Constraints

Chris Russell †[1]
http://www.eecs.qmul.ac.uk/~chrisr/
Francesco Setti †[2]
settif@eecs.qmul.ac.uk
Lourdes Agapito[1]
http://www.eecs.qmul.ac.uk/~lourdes/

[1] School of Electronic Engineering and Computer Science
Queen Mary University of London
[2] University of Trento

Abstract

Current state of the art multi-target tracking (MTT) exists in an "either/or" situation. Either a greedy approach can be used, that can make use of second-order information which captures object dynamics, such as "objects tend to move in the same direction over adjacent frames", or one can use global approaches that make use of the information contained in the entire sequence to resolve ambiguous sub-sequences, but are unable to use such second order information. However, the accurate resolution of ambiguous sequences requires both a good model of object dynamics, and global inference.

In this work we present a novel approach to MTT that combines the best of both worlds. By formulating the problem of tracking as one of global MAP estimation over a directed acyclic hyper-graph, we are able to both capture long range interactions, and informative second order priors. In practice, our algorithm is extremely effective, with a run time linear in the number of objects to be tracked, possible locations of an object, and the number of frames. We demonstrate the effectiveness of our approach, both on standard MTT data-sets that contain few objects to be tracked, and on point tracking for non-rigid structure from motion, which, with hundreds of points to be tracked simultaneously, strongly benefits from the efficiency of our approach.

1 Introduction

The tracking of multiple points or objects is a prerequisite for many types of video analysis. For example: non-rigid structure from motion makes use of the long term tracks of interest points in performing 3D reconstruction [8, 25]; in surveillance the tracking of individuals is of interest in its own right. Traditional multi-target tracking has relied upon either a naive approach to tracking, that treated the location of each object track as being independent of one another [10], or by exhaustively mapping the set of valid combinations of object locations [19] – given *exclusion constraints* which say that only one object may occur in a particular location at a particular time. These approaches suffer from different drawbacks: the naive approaches may merge object tracks (see fig. 1, **far left**); while the computational cost of the combinatorial approaches grows exponentially with the number of objects considered, making them ill-suited for tracking problems containing a large number of points.

†CR and FS assert joint first authorship. This work was funded by the European Research Council under the ERC Starting Grant agreement 204871-HUMANIS. FS was supported by the European Commission and Provincia Autonoma di Trento under Marie Curie Action - Cofund project ABILE.

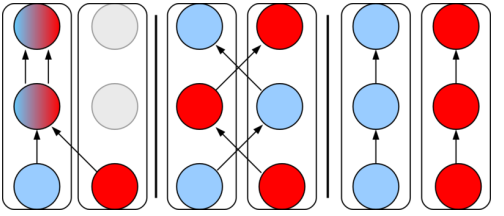


Diagram showing common failures of MTT, *as two tracks closely pass by each other. The boxes show the true tracks of objects, while the colour of the balls and the arrows indicate their object labelling.* **Left:** *Without the use of exclusion constraints (forcing only one object to occur in a particular location at any time) object tracks may merge.* **Centre:** *Even with the use of exclusion constraints, if the distance between two object tracks is smaller than the distance moved by a single object in two frames flickering may occur.* **Right** *The use of second order terms to penalise objects sharply changing direction eliminates flicker. See section 1 for details. Fig. 3 for real world examples.*

MTT Method	Persistent Appearance model	Complexity in number of frames	Global Optimisation	Exclusivity prior	Second Order priors
Naive Dynamic Prog. [1, 2]	✓	Poly †	✓	✗	✗
Greedy [26]	✓	Linear	✗	✓	✓
Kalman [3]	✓	Linear	✗	✓	✓
Particle [3]	✓	Linear	✗	✓	✓
L.P.[13]	✓	Poly	✓	✓	✗
Flow[4, 30]	✗	Poly †	✓	✓	✗
K-short Path[5]	✗	Poly †	✓	✓	✗
Our approach	✓	Linear	✓	✓	✓

† While we report worst time running times, the authors of [1, 4, 5] report that in practice a linear, rather than polynomial, growth in complexity may be obtained.

Figure 1: **Left:** *A diagram demonstrating common failures of* MTT **Right:** *A comparison of existing methods against our approach. See section 1 for more details.*

Recently, there has been substantial progress in modelling these *exclusion constraints* explicitly [4, 5, 13, 30]. This has allowed them to be enforced without enumerating all possible permutations of object tracks. These recent works can again be loosely categorised into two families: Greedy methods such as the Kalman [19] or particle filters [3] are able to make use of second order cues such as the fact that tracks tend to continue moving smoothly in the same direction, but become lost in ambiguous sub-sequences, where it is unclear how objects have moved; and global methods [4, 5, 13] which do not capture these second order relationships but make use of strong cues in other frames to resolve local ambiguities.

In this work, we present a novel global formulation for MTT which, alongside compactly representing exclusion constraints, is able to make use of second-order prior information about motion, eliminating the common failure case in which object tracks flicker back and forth between their correct locations (as shown in fig. 1 **centre left**).

Second order approaches The use of second order information, in particular the prior that objects being tracked are likely to have a smoothly varying velocity, is common to greedy approaches such as the Kalman and particle filters [3, 19]. However, while the complexity of such methods grows linearly with the number of frames, in some cases, they may grow exponentially in the number of objects [19], or in the size of their search window [3]. Such methods are easily confused by ambiguous sub-sequences, as they are unable to use information regarding the position of objects in future frames to resolve the confusion.

Global approaches Outside of computer vision, global methods, including the Viterbi algorithm and shortest path algorithms, have been frequently combined with the use of a large state space which enumerates all possible permutations of objects. While such approaches are valid for domains such as radar tracking, where few objects are tracked, and there are relatively few locations an object might be in at a particular point in time, the most efficient algorithms require $\mathcal{O}(L^O \cdot F)$ run time, and many require $\mathcal{O}(L^{2N} \cdot F)$ operations, where L is the number of possible locations an object may occur, N the number of objects, and F the number of frames the object is tracked for. See [23] for an overview. Such approaches are

inappropriate for vision problems — even when there are relatively few objects in a scene, the intrinsic ambiguities mean that there are often many locations where each object may be.

The work [13] first combined the explicit modelling of exclusion constraints with a linear program that could find the global solution to the problem in polynomial time. Our work can be seen as improving on theirs — we make two important modifications of their formulation: we replace their higher-order constraints that multiple objects may not exist in the same location, with a set of pairwise constraints that say "No pair of objects may exist in the same location", allowing for efficient inference with existing vision algorithms; furthermore, we augment their pairwise soft constraint on spatial consistency[1] with an additional soft tertiary constraint that encourages objects to have similar velocities in adjacent frames.

Several flow [2, 30], and path based [5] global approaches have been proposed. These approaches are often efficient in practice, while they have a worst case polynomial run times, the authors of these methods report a linear growth in complexity on real data. Compared to our approach, which makes use of persistent appearance models, and second-order cues that encourage objects to change velocity smoothly, such methods are only able to match appearance between two adjacent frames. This carries significant disadvantages: firstly, although such methods can make use of similarities in appearance over adjacent frames [5], they are unable to use the complementary information in a persistent model of object appearance. Such persistent models can be highly informative [1, 7], and the absence of them makes it harder to recover from temporary lighting changes, and prevents the recognition of a previously seen object reentering a sequence. These methods are also unable to capture the second order prior that the velocity of a tracked object changes smoothly, leading to errors such as those shown in fig. 1 **centre left**.

A summary of the advantages of our approach can be seen in fig. 1 **right**. In brief: Ours is the only global approach to MTT that incorporates exclusion constraints and has a guaranteed linear running time; the only efficient global approach [2, 30] that can make use of persistent appearance models; and the only global approach to make use of highly informative second order priors which describe the motion of a tracked object.

2 Notation and Cost Function

We formulate tracking as a discrete problem, in which the aim is to assign a set of objects to a set of fixed locations. These locations are either given by interest point or object detectors, or are an *occluded* state, that indicates that the object has not been detected in a frame, but that it is believed to be in a particular region of the image. We use N to refer to the number of objects being tracked, O the set of occluded states, F for the number of frames, and L the number of possible locations. We write $x_{o,t}$ for the location of object o at time t, and $\mathbf{x}$ for the set of complete tracks of all objects over all frames. To find a set of good tracks, we seek the complete tracks $\mathbf{x}$ which leads to a minimal cost solution of some cost function $C(\cdot)$. This cost function can be decomposed into 4 components as follows:

- Unary Potentials: $U_{o,t}(x_{o,t})$ The cost of placing an object o in location $x_{o,t}$ at time t. This typically takes into account how closely the appearance of the object matches the pixels in the location. It may also take advantage of prior knowledge of a particular object's expected location at a given time.

- Pairwise Potentials: $P_{o,t}(x_{o,t}, x_{o,t+1})$ These potentials describe the cost of an object o transitioning from location $x_{o,t}$ at time t to location $x_{o,t+1}$ at time $t+1$. This cost takes

[1] This constraint encourages objects to appear in similar locations in adjacent frames.

into account the distance between the locations $x_{o,t}$ and $x_{o,t}$, and is typically of the form

$$P_{o,t}(x_{o,t}, x_{o,t+1}) = k_t ||x_{o,t} - x_{o,t+1}||_2^2, \tag{1}$$

where k_t is an arbitrary constant used to weight the pairwise cost. The pairwise costs may also be used to penalise large changes in appearance from location $x_{o,t}$ to $x_{o,t+1}$ [5].

- Tertiary Potentials: $T_{o,t}(x_{o,t}, x_{o,t+1}, x_{o,t+2})$ While our inference approach supports any form of potentials over the 3 locations, $x_{o,t}, x_{o,t+1}$ and $x_{o,t+2}$, we will use them to penalise an object suddenly changing direction. In this work, these potentials typically take the form

$$\begin{aligned} T_{o,t}(x_{o,t}, x_{o,t+1}, x_{o,t+2}) &= k'_t ||(x_{o,t} - x_{o,t+1}) - (x_{o,t+1} - x_{o,t+2})||_2^2 \\ &= k'_t ||2x_{o,t+1} - x_{o,t} - x_{o,t+2}||_2^2, \end{aligned} \tag{2}$$

where again k'_t is an arbitrary weighting constant. We will also use these tertiary potentials to force an object to reappear next to the location it disappeared from if it is occluded, or undetected, for a single frame. These tertiary costs which transition through an occluded state take a similar form to the pairwise costs (1)

$$T_{o,t}(x_{o,t+1}, y, x_{o,t+2}) = k''_t ||x_{o,t} - x_{o,t+2}||_2^2, \tag{3}$$

where $y \in O$ is an occluded state.

- Exclusion constraints: These are hard constraints which enforce that only one point may occupy a single location at any time. We relax this constraint for occluded locations, and allow multiple objects to be present in them, as they are essentially used as pigeon holes to store the location of objects which are currently undetected.

Putting these constraints together, we arrive at the following objective: we seek a labelling $\mathbf{x}$ that minimises the cost function $C(\cdot)$ defined as

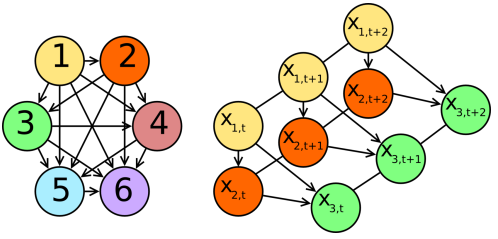
$$\min_{\mathbf{x} \in L^{N \cdot F}} C(\mathbf{x}) = \sum_{o \leq N} \Big[\sum_{t \leq F} U_{o,t}(x_{o,t}) + \sum_{t \leq F-1} P_{o,t}(x_{o,t}, x_{o,t+1}) + \sum_{t \leq F-2} T_{o,t}(x_{o,t}, x_{o,t+1}, x_{o,t+2}) \Big] \tag{4}$$

$$\text{such that } \sum_{o \leq N} \Delta(x_{o,t} = l) \leq 1 \; \forall t, \;\; \forall l \in L, \;\; l \notin O. \tag{5}$$

This final constraint is the exclusion constraint, which says that at most one object may appear in any unoccluded location at one time.

3 Graph structure and Inference

With the exception of the exclusion constraints, the optimal solution to the above cost function can be exactly solved in polynomial time, using the second order Viterbi algorithm [17]. Note that, in general, the tertiary potentials we have introduced are incompatible with standard graph based MTT approaches [5, 30]. While third order submodular costs can be embedded in a graph using one additional node [14], embedding arbitrary tertiary costs requires the use of negative edges [29], which prohibits the use of efficient flow algorithms. Higher-order Viterbi algorithms and belief propagation do not suffer from such issues as they exploit the acyclic structure of the tracks. Given this, it is natural to ask if we can also



Graphical models demonstrating our approach. **Left:** *A slice of the graph showing the fully connected* DAG *between 6 object tracks at a fixed time.* **Right:** *The full* DAG *showing the interdependence between nodes of the graph over several frames. Different colours indicate different object tracks. For the sake of clarity, the tertiary hyperedges are omitted from the graph. Directed edges are indicated by arrows, while undirected edges are indicated by straight lines connecting nodes of the graph.*

Inference Scheme

for $k = \{1,2,\dots N\}$ **do**
  Forward (k)
  Backward(k)
  Message down (k)
end for
for $k = \{N, N-1,\dots 1\}$ **do**
  Undo sent message(k)
  Backward(k)
  Viterbi(k)
  Constraint up (k)
end for

High-level pseudo code for inference. See supplementary materials for complete pseudo code.

Figure 2: MAP *estimation of the* DAG

incorporate exclusion constraints into our graph structure, without introducing dense cycles which make belief propagation based inference perform poorly [15, 24].

The answer to this question is yes; these constraints can be formulated as a *Directed Acyclic Graph* (DAG) between object tracks. Given a set of object tracks $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, for all t, we add an extra set of directed edges from $x_{i,t}$ to all $x_{j,t}$ such that $i > j$ (see fig. 2 **left** for an illustration of the resulting graph structure). We associate the pairwise cost

$$E(x_{i,t}, x_{j,t}) = \begin{cases} 0 & \text{if } x_{i,t} \neq x_{j,t} \\ -\infty & \text{otherwise} \end{cases} \tag{6}$$

with each such edge. The full graph is still a DAG (see fig. 2 **centre**), and can be solved using standard belief propagation techniques [20] (see fig. 2 **right** for an overview of the algorithm and Supplementary Materials for the full pseudo code). By formulating this problem as inference of a DAG rather than an undirected graph, we are making assumptions about the causal structure of the problem that may be unwarranted [21]. However, in practice, such assumptions are relatively common in computer vision [12, 16, 18], and except for our use of second-order belief propagation, inference is identical to that of [16].

3.1 Efficient Inference

Using the approach of [17], the complexity of performing a single iteration of the second order Viterbi algorithm or the forward backward algorithm for a single object, is $\mathcal{O}(L^3 \cdot F)$. Similarly, the time taken to pass messages from one particular object track to all others has an average run time of $\mathcal{O}(L^2 \cdot N)$ using the techniques of [20]. We will now show how these computations can be efficiently performed in $\mathcal{O}(k^2 \cdot L \cdot F)$[2] and in $\mathcal{O}(L)$ time respectively, using techniques similar to those of [9].

Efficient Second Order Viterbi for Tracking *Gating* is a standard technique for improving the efficiency of global tracking methods [23]. It is based on the observation that if the soft costs of transitioning between two locations very far away is sufficiently high, such a transition never occurs in practice. Consequently, such soft costs penalising these transitions can be replaced with hard constraints that prohibit such transitions without altering the

[2] k is a constant substantially smaller than L

global optima. In graph based algorithms, such as k-shortest paths [5], *gating* is equivalent to deleting edges with very high cost, before running the shortest path algorithm. In both the forward-backward and Viterbi algorithms the same approach can be taken where we make an a priori decision to only update the marginals of locations that are sufficiently close to one another. See [22] for more details. If we only allow a transition from any location to at most k other locations, this reduces the run time of the standard Viterbi algorithm from $\mathcal{O}(L^2 \cdot F)$ to $\mathcal{O}(k \cdot L \cdot F)$. We take exactly the same approach with the second order forward-backward and Viterbi algorithms; by restricting the space of valid transitions in the same manner, there are at most k^2L valid transitions over three adjacent frames, and this leads to second order algorithms with a run time of $\mathcal{O}(k^2 \cdot L \cdot F)$. See Supplementary Materials for full code.

Second order Viterbi and Occluded States The above speed-up can not be used if there is only a single occluded state, as this would imply that, over three frames, an object in any location can transition to any other location via the occluded state. This would lead to an overall run-time complexity of $\mathcal{O}(k \cdot L^2 \cdot F)$. Fortunately, if we have multiple occluded states associated with different locations, this is not the case. By blocking a transition to an occluded state at a particular location from a location far from it, we can restrict ourselves to at most k transitions to/from any single occluded state, and perform the Viterbi algorithm in $\mathcal{O}(k^2 \cdot L \cdot F)$ with occlusions. In practice, the use of multiple occluded states leads to much better final results, as it prevents points from moving to an occluded state in order to jump across the image. It is one of the few cases in vision where a more discriminative model is also more computationally efficient.

Efficient Updates between Object Tracks Efficiently passing messages between object tracks requires a different approach. A naive approach to this problem would involve passing a message between every pair of objects, at every frame. The cost of computing these messages is $\mathcal{O}(L^2)$ leading to an overall run time of $\mathcal{O}(N^2 \cdot L^2 \cdot F)$. However, the inherent symmetry of the messages can be exploited to reduce the run time of this process to $\mathcal{O}(N \cdot L \cdot F)$. Ordering the tracks as before, we define $M_{o,t}$ as the marginals of a single track, as computed by the forward-backward algorithm (see lines 1-2 of **Message down** Procedure Supplementary Materials for definition). We first note that the same messages are passed from any object track i to every track $j > i$. Letting $\hat{l} = \arg\max_{l_2} M_{o_1,t}(l_2)$, we can compute these messages passed down, which we call $\downarrow M_{i,j,t}$ from object i to all objects $j > i$ as

$$\downarrow M_{i,j,t}(l) = \max_{l_2} \left(M_{i,t}(l_2) + E(l,l_2)\right) = \begin{cases} M_{i,t}(\hat{l}) & \text{if } l \neq \hat{l} \\ \max_{l_2 \neq \hat{l}} M_{i,t}(l_2) & \text{otherwise} \end{cases} \tag{7}$$

in $\mathcal{O}(L)$ time. Under constant reparameterisation this is equivalent to

$$\downarrow M_{i,j,t}(l) = \begin{cases} \max_{l_2 \neq \hat{l}} M_{i,t}(l_2) - M_{j,t}(\hat{l}) & \text{if } l = \hat{l} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

and can be computed in $\mathcal{O}(L)$ time. To avoid sending messages of this form $\mathcal{O}(N^2)$ times, we exploit the fact that the same message is sent by object track i to all objects $j > i$, and amend a single message as it is passed in sequence through all object tracks. In this case, object i updates this message $\downarrow M_t(l)$ as follows

$$\downarrow M_t(\hat{l}) = \downarrow M_t(\hat{l}) + \max_{l_2 \neq \hat{l}} M_{o_1,t}(l_2) - M_{o_1,t}(\hat{l}). \tag{9}$$

Using this technique, the downward max-margins can be iteratively computed for objects tracks 1 through to n, at which point, following [20] we can perform assignment, to find

the optimal solution. This is done by first removing the message previously passed down (see **Undo message** Supplementary materials), and then using the second order Viterbi algorithm to perform assignment. Any constraint due to this assignment is then appended to the messages (see **Constraint Up**).

4 Experimental Evaluation

We evaluate our approach on challenging data from several domains, including non-rigid structure from motion NRS*f*M, player tracking in a basketball match, and the tracking of people in crowds. In the evaluation, we make use of the standard MOTA and MOTP scores [6] to measure our efficiency[3].

For the tracking of basketball players, we made use of thresholded detection sites from [27], on a sequence of 500 frames from the Apidis project. The detection sites of [27] were projected onto an overhead camera view, and we made use of both the appearance and motion of players in this top view to form our potentials (see fig. 3). The distance between players from frame to frame, was also passed to [5] for evaluation. To evaluate the effect of our different potentials, we repeatedly ran our algorithm turning each of them off (see fig. 3). All of our potentials make a substantial contribution to the effectiveness of our approach.

In the basketball sequence, all approaches are limited by the quality of the detection sites. Assuming that every detection site was correctly classified the best possible MOTA score is .85, and given perfect detection sites our approach has a MOTA score of .99. Nonetheless, we feel that these experiments are valuable, as they show the suitability of our approach on real world detections. The only approach with a higher MOTA than ours is [2], which achieves this by making many false detections leading to a MOTP score 15 times worse than ours.

Various authors [2, 5] have claimed that approaches such as ours or [13], which can make use of persistent appearance models, cannot be used to track an unknown number of objects. To show this is not the case, we evaluated our approach against [5], on two sequences taken from their paper: the first is a monocular sequence from PETS 2009 (S2/L1) and the second involves the tracking of multiple ping-pong balls. In both sequences, detection cues were provided by [10] and we track an arbitrary number of objects. To do this, we estimate the greatest number of objects present in a single frame and perform inference with this many tracks. Tracks leaving the scene move to an occluded state, and are then allowed to re-enter from any side as a new object.

In this difficult scenario, where we are unable to make use of persistent appearance models or the knowledge of how many objects are present in the scene, we perform slightly worse than [5]. This can partially be attributed to incompatibilities between our approach and [10]. In practice, local maxima of [10] often occur relatively far from the true location of people, and if we do not perform non-maximal suppression, without knowledge of appearance or the number of objects in the image, we often get false positives.

For NRS*f*M, we took a synthetically rendered sequence of a flag moving in the wind allowing the comparison with ground truth [11]. As the number of points which could be tracked is arbitrarily large, the conventional measures of MOTA and MOTP are meaningless, and instead we report the root mean squared error of the tracks, as a measure of track drift. We were unable to evaluate the effectiveness of [5] on this data set, as we had insufficient RAM on our server. In practice, our algorithm took around 4 minutes to track 400 points over 200 frames.

[3] For MOTA higher is better, and 1 is optimal. For MOTP lower is better, and 0 is optimal.

Figure 3: The benefits of tertiary potentials on the basketball sequence.

Basketball	MOTP	MOTA
Pairwise only	0.6879	-1.064
Pairwise and unary	0.6021	-1.070
Pairwise and occlusion	0.8227	0.718
Pairwise, unary, and occlusion	0.8214	0.725
Complete method	0.7198	0.735

POM detections	PETS 2009		Balls sequence	
–	MOTP	MOTA	MOTP	MOTA
K-Shortest Path	1.2018	0.6783	0.3015	0.90
Our method	1.2742	0.5016	0.7687	0.8204

Basketball	memory usage	total time	optimisation time	MOTP	MOTA
Track-before[ ]	–	–	–	7.1	0.614
Trajectory assoc[ ]	–	–	–	12.7	0.781
K-Shortest Path[ ]	16 GB	2m 25s	1m 45s	1.09	0.586
Our method	550 MB †	1m 30s	1.1 s	0.72	0.735

NRSfM	RMS
Kanade-Lucas-Tomasi [ ]	142.59
Our method	32.66

Figure 4: **Top Right:**Comparison of MOTP and MOTA on PETS 2009 S2/L1 monocular and balls sequence.**Bottom Left:** Computational resources (RAM allocated), computation time, MOTP and MOTA for [ ] and our approach on the basketball sequence.**Bottom Right:** Feature tracking on the motion capture of a flag. Root Mean Square Error is shown for 400 feature points tracked. †Includes matlab instance.

5 Conclusion

We have presented a state of the art approach to tracking and demonstrated its importance in interest point tracking in non-rigid structure from motion and multiple person tracking under highly challenging scenarios. Our approach is the first global method to make use of second order cues which describe the acceleration of points and our experimental results convincingly demonstrate their importance. Compared to other efficient algorithms [5] our approach is over a hundred times faster, and exhibits better worse case performance. This improved efficiency allows us to track hundreds of points easily, and to estimate the location of basketball players at almost 500 frames per second. Consequently, we expect this work to be of strong interest to both the non-rigid structure from motion community, and those working in surveillance, or real time multi-target tracking. Our code is available for download at http://www.eecs.qmul.ac.uk/~chrisr/tracking.tar.gz.

References

[1] Brian Amberg and Thomas Vetter. Graphtrack: Fast and globally optimal tracking in videos. In *CVPR*, 2011.

[2] Nadeem Anjum and Andrea Cavallaro. Trajectory association and fusion across partially overlapping cameras. In Stefano Tubaro and Jean-Luc Dugelay, editors, *AVSS*, pages 201–206. IEEE Computer Society, 2009.

[3] L. Bazzani, D. Bloisi, and V. Murino. A comparison of multi-hypothesis Kalman filter and particle filter for multi-target tracking. In *Performance Evaluation of Tracking and Surveillance (PETS) workshop at CVPR*, pages 47–54, Miami, Florida, 2009.

[4] Jerome Berclaz, Francois Fleuret, and Pascal Fua. Multiple Object Tracking using Flow Linear Programming. In *12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS 2009)*, 2009.

[5] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[6] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *J. Image Video Process.*, 2008:1:1–1:10, January 2008.

[7] A. Buchanan and A. Fitzgibbon. Interactive Feature Tracking using K-D Trees and Dynamic Programming. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 626–633, 2006.

[8] Joao Fayad, Lourdes Agapito, and Alessio Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In *ECCV*, 2010.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Belief Propagation for Early Vision. In *Proc. CVPR*, volume 1, pages 261–268, 2004.

[10] Francois Fleuret, J Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:267–282, 2008.

[11] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Robust trajectory-space tv-l1 optical flow for non-rigid sequences. In *EMMCVPR*, 2011. URL http://www.eecs.qmul.ac.uk/~lourdes/subspace_flow/.

[12] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 2006.

[13] Hao Jiang, Sidney Fels, and James J. Little. A Linear Programming Approach for Multiple Object Tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.

[14] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[15] Vladimir Kolmogorov and Carsten Rother. C.: Comparison of energy minimization algorithms for highly connected graphs. in: Eccv. In *In Proc. ECCV*, pages 1–15, 2006.

[16] Sanjiv Kumar and Martial Hebert. A hierarchical field framework for unified context-based classification. In *International Conference on Computer Vision*, 2005.

[17] Amlan Kundu, Yang He, and Paramvir Bahl. Recognition of handwritten word: first and second order hidden markov model based approach. *Pattern Recogn.*, 22:283–297, February 1989.

[18] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the Twenth-Sixth International Conference on Machine Learning*, 2009.

[19] John MacCormick and Andrew Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1998.

[21] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[22] G. W. Pulford and B. F. La Scala. Multihypothesis Viterbi Data Association: Algorithm Development and Assessment. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):583–609, April 2010.

[23] G.W. Pulford. Taxonomy of multiple target tracking methods. *Radar, Sonar and Navigation, IEE Proceedings* -, 152(5):291 – 304, october 2005.

[24] Chris Russell, Lubor Ladicky, Pushmeet Kohli, and Philip Torr. Exact and approximate inference in associative hierarchical networks using graph cuts. *UAI*, 2010.

[25] Mathieu Salzmann and Raquel Urtasun. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In *CVPR*, 2010.

[26] Khurram Shafique and Mubarak Shah. A non-iterative greedy algorithm for multi-frame point correspondence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 51–65, 2005.

[27] M. Taj and A. Cavallaro. Multi-camera track-before-detect. In *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*, 2009.

[28] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[29] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[30] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.