A Relaxation Algorithm for Real-time Multiple View 3D-Tracking

Yi Li, Adrian Hilton* and John Illingworth
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, GU2 7XH, UK
a.hilton@surrey.ac.uk
(* corresponding author)

Abstract

In this paper we address the problem of reliable real-time 3D-tracking of multiple objects which are observed in multiple wide-baseline camera views. Establishing the spatio-temporal correspondence is a problem with combinatorial complexity in the number of objects and views. In addition vision based tracking suffers from the ambiguities introduced by occlusion, clutter and irregular 3D motion. In this paper we present a discrete relaxation algorithm for reducing the intrinsic combinatorial complexity by pruning the decision tree based on unreliable prior information from independent 2D-tracking for each view. The algorithm improves the reliability of spatio-temporal correspondence by simultaneous optimisation over multiple views in the case where 2D-tracking in one or more views are ambiguous. Application to the 3D reconstruction of human movement, based on tracking of skin-coloured regions in three views, demonstrates considerable improvement in reliability and performance. Results demonstrate that the optimisation over multiple views gives correct 3D reconstruction and object labeling in the presence of incorrect 2D-tracking whilst maintaining real-time performance.

Key words: 3D-tracking, combinatorial optimisation, relaxation.

A Relaxation Algorithm for Real-time Multiview 3D-Tracking

Abstract

In this paper we address the problem of reliable real-time 3D-tracking of multiple objects which are observed in multiple wide-baseline camera views. Establishing the spatio-temporal correspondence is a problem with combinatorial complexity in the number of objects and views. In addition vision based tracking suffers from the ambiguities introduced by occlusion, clutter and irregular 3D motion. In this paper we present a discrete relaxation algorithm for reducing the intrinsic combinatorial complexity by pruning the decision tree based on unreliable prior information from independent 2D-tracking for each view. The algorithm improves the reliability of spatio-temporal correspondence by simultaneous optimisation over multiple views in the case where 2D-tracking in one or more views are ambiguous. Application to the 3D reconstruction of human movement, based on tracking of skin-coloured regions in three views, demonstrates considerable improvement in reliability and performance. Results demonstrate that the optimisation over multiple views gives correct 3D reconstruction and object labeling in the presence of incorrect 2D-tracking whilst maintaining real-time performance.

Key words: 3D-tracking, combinatorial optimisation, relaxation.

1. Introduction

Tracking of multiple objects in multiple view image sequences requires the solution of two labeling problems: spatial correspondence of observations between views and temporal correspondence of the observations in a single view with an object. Commonly these problems are treated independently leading to sub-optimal solutions in the presence of ambiguities such as incorrect correspondence due to occlusion, clutter, changes in appearance and complex motion. In this paper, instead, we present a novel approach to reliable 3D-tracking by simultaneous optimisation over multiple views which achieves computationally efficient integration of observations using prior knowledge from individual views.

Multiple view tracking of multiple objects has combinatorial complexity in the number of objects and observations, which is prohibitive for real-time applications. We introduce the uncertain prior knowledge from the independent 2D-tracking in each view into the optimisation algorithm to identify the most likely correspondence. Relaxation based on our uncertainty in the prior knowledge is used to efficiently identify the solution, which provides global optima across multiple views, at the same time avoids the uncertainty. This approach provides a computationally efficient solution to the spatio-temporal correspondence, which enables real-time multi-view 3D-tracking. It is also robust to errors in the prior knowledge, because the possibility of objects disappearing and reappearing due to occlusion with respect to a particular view and the presence of clutter due to identification of background objects which do not correspond to any of the objects being tracked, are efficiently taken into account.

1.1 Previous Work

The problem of 3D-tracking of multiple moving objects observed in either single or multiple view image sequences is common in computer vision. Typically image features such as edges, colour or texture are used to identify a sparse set of 2D features corresponding to observations of the moving objects. Feature or token-based tracking has been investigated to establish the temporal correspondence in the presence of scene clutter and occlusion [Rangarajan and Shah 1991, Zhang and Faugeras 1992]. Multiple object tracking in the presence of clutter has also been addressed in the context of general target based tracking [Bar-Shalom and Fortmann, 1988, Bar-Shalom1996, Blackman 1986].

Tracking from multiple view image sequences opens up the possibility of 3D reconstruction of the object trajectory. This requires the solution of both the spatial correspondence of observations between views and the temporal correspondence of observations with objects. Consistent labelling according to a set of a priori known constraints has combinatorial complexity [Haralick and Shapiro 1979, Faugeras and Maybank 1990]. The optimal labelling problem can be resolved by techniques such as relational graph matching, graph isomorphism, tree search and relaxation labelling [Faugeras and Bethod 1981, Hummel and Zucker 1983, Chen and Huang 1988]. Approaches to reducing the combinatorial complexity include knowledge-based clipping, heuristic search, divide-and-conquer and dynamic programming. In general these approaches reduce the complexity but may fail to identify the optimal solution for the ambiguous situations which occur in 3D-tracking, as discussed in section 1.2. Constraints on the 3D motion are commonly used to reduce the search space such as rigidity [Philip 1991], co-planarity [Tsai and Huang 1981, Weng et al. 1991], local coherence [Roy and Cox 1994], epi-polar geometry [Faugeras 1993] and tri-focal tensor [Hartley and Zisserman, 2000]. Symbolic optimisation methods have been employed such as best-first or greedy search [Sethi and Jain 1987, Hwang 1989, Salari and Sethi 1990, Jenkin and Tsotsos 1986], beam searching [Bar-Shalom and Fortmann 1988] and competitive linking [Chetverikov and Verestoy 1998]. These approaches address the search for the global optima but either still suffer from local optima, or do not reduce the computational complexity to a level where they can be readily employed for real-time vision applications, or both. The approach introduced in this paper addresses the issue of reducing the inherent computational complexity of multiple view 3D-tracking for real-time applications whilst maintaining the robustness of global optimisation.

Numerous approaches to object matching based on shape, appearance, motion and other a priori knowledge have been developed in computer vision [Martin and Aggarwal 1979, Lowe 1992, Thompson etl al. 1993, Jang et al. 1997]. Motion prediction in 2D or 3D has been widely used in vision based tracking systems. Typically isolated objects are tracked using a Kalman filter approach to predict and update object location estimates from observations [Zhang and Faugeras 1994]. Previous work, such as [Sethi and Jain 1987, Hwang 1989, Salari and Sethi 1990], focuses on the cost function definition according to motion smoothness or geometric constraints taking into account object occlusion and reappearance. Techniques for statistical data association have also been applied to motion correspondence [Cox 1993, Zhang and Faugeras 1994]. Stochastic optimisation and random sampling techniques using statistical priors have also been develop to achieve robust tracking in the presence of clutter [Cox 1993, Isard and Blake 1998]. These approaches to robust tracking in the presence of clutter and occlusion are not computationally efficient for real-time tracking across multiple views.

Simultaneous optimisation of object-observation and observation-observation correspondences across multiple views has combinatorial complexity in both the number of objects and number of views. Previous approaches, typically [Faugeras and Maybank 1990, Huang and Netravali 1994], have handled the problem of computational complexity using a divide-and-conquer strategy. The global optimisation problem is reduced to the sub-problems of spatial and temporal matching which are treated independently. The divide-and-conquer approaches for 3D-tracking from multiple views can be categorised into two distinct strategies:

Reconstruction-Tracking (RT): First identify the inter-view observation-observation spatial correspondence then resolve the 3D object-observation temporal correspondence.

Tracking-Reconstruction (TR): First perform 2D-tracking in each view independently to obtain the object-observation temporal correspondence and then reconstruct the 3D location from the resulting set of object observations.

These strategies reduce the computational complexity by decoupling the combinatorial optimisation of correspondence into separate problems with smaller combinatorics sometimes leading to real-time 3D-tracking solutions. However, they may lead to failure in the reconstruction due to the inherent ambiguities in both 2D-tracking of 3D objects is a single view or matching of observations between views. Ambiguities caused by self-occlusion and clutter are discussed in further detail in the section 1.2. To achieve reliable 3D-tracking for objects observed in multiple views it is necessary to simultaneously optimise over spatial and temporal correspondence.

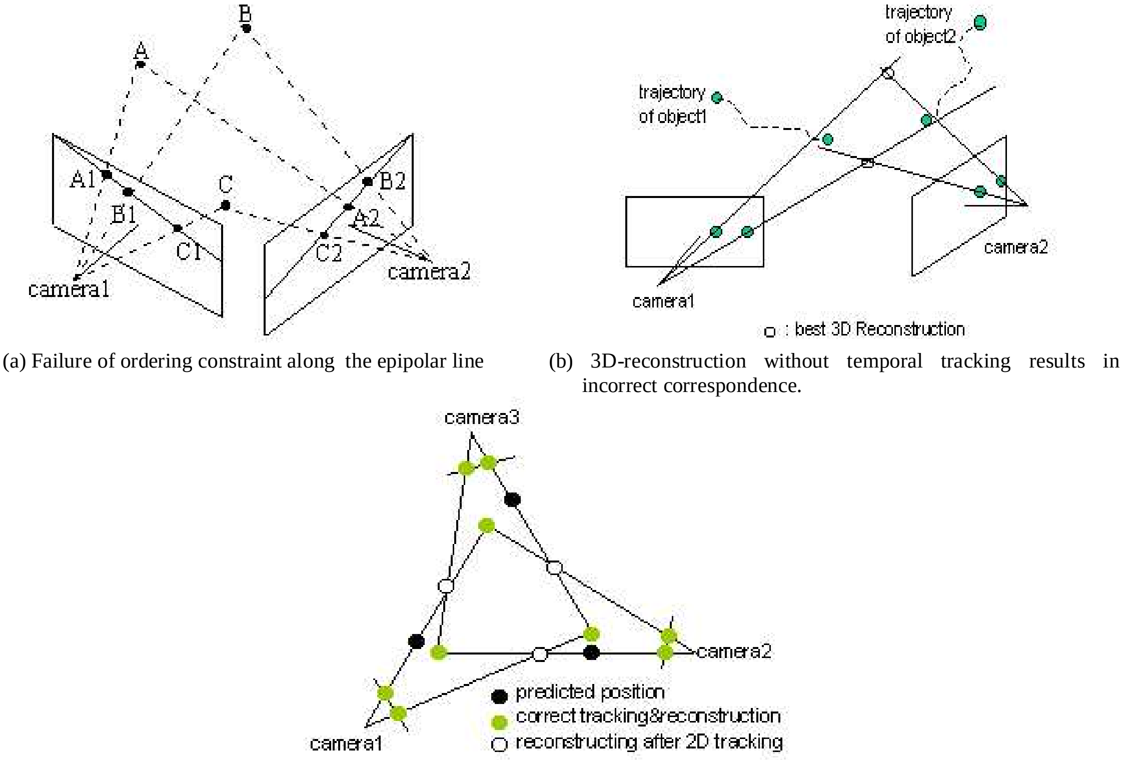
This problem of 3D-tracking from multiple views is of increasing interest in computer vision for applications such as video surveillance [Collins et al. 2000] and human motion capture [Hilton and Fua 2001]. Due to the inherent self-occlusion and clutter in human movement reliable 2D feature tracking in single view image sequences is problematic. Recent reviews of research addressing human motion capture identify this as a problem [Aggarwal et al.1999, Gavrilla et al.1999, Moeslund et al.2001]. Recent research [Song et al. 2001] investigates the problem of feature labelling and reconstruction in a probabilistic framework using the underlying kinematic model to resolve the labelling problem. Other researchers such as [Isard and Blake1998, Sidenbladh et al.1998, Gong et al. 2000] have developed model-based tracking frameworks which utilise knowledge of the prior distribution to sample the space of possible solutions. Currently such solutions enable reliable tracking over a range of movement from a single view but do not in general address the problem of real-time performance. In this paper we apply the multiple view 3D-tracking framework to address the issue of computational efficiency in the reliable capture of human movement by simultaneous tracking a person in multiple views.

1.2 Ambiguities in Multiple View 3D-Tracking

When dealing with the multiple view based real-time tracking of sparse and independent 3D motion, for the purpose of computational efficiency most approaches address the problems of correspondence and reconstruction separately following either the RT or TR strategies outlined in section 1.1. However, in the presence of occlusion and clutter it is often not possible to achieve reliable 3D-tracking of multiple objects in multiple views without considering correspondence and reconstruction simultaneously.

In the reconstruction-tracking RT approach we first match observations between multiple views. However, for wide angle views the observed shape and appearance of image features are generally substantially different. The order of observations along the epipolar line (the ordering constraint) [see Faugras 1993] provides a widely used constraint for matching observations between pairs of views. However, multiple noisy observations in one view can easily appear on or near the same epipolar line in a second view causing ambiguity in the correspondences. The order of observations along the epipolar line may change between views, as illustrated in Figure 1(a). If the motion trajectory of 3D objects is not taken into account, the best 3D-reconstruction may yield incorrect spatial correspondence. Figure 1(b) shows an example why the RT approach would fail when temporal information is omitted. In practice there are many instances where the RT approaches would fail due to incorrect correspondence.

In the case of the tracking-reconstruction TR strategy we independently track the temporal correspondence in each view based on the predicted position using techniques such as the Kalman filter. However, this may lead to incorrect correspondence due to rapid or irregular motion which can not be accurately predicted. Multiple objects can easily appear in close proximity in a single view resulting in correspondence ambiguities despite being spatially separated in 3-space. In the worst case objects may occlude each other with respect to a particular view and have similar projected motion trajectories. In this situation there is no possibility of resolving the ambiguity unless we have either observations from multiple views or strong priori knowledge of object shape, appearance and movement. For many real situations objects have similar shape and appearance which are changing over time. Together with independent irregular movement, the a priori knowledge is not sufficient to efficiently resolve ambiguities. An example of object-object occlusion resulting in ambiguous correspondence and failure of the TR strategy is shown in Figure 1(c). In this extreme case one of the three 3D-objects is occluded by another with less depth in each of the three views, so only two 2D-objects are observed in each view. The independent 2D-tracking in each view may easily result in incorrect correspondence. All three views must be considered simultaneously to resolve the ambiguity and correctly track the 3D-objects.



(a) Failure of ordering constraint along the epipolar line

(b) 3D-reconstruction without temporal tracking results in incorrect correspondence.

(c) Independent 2D-tracking without inter-view information results in incorrect spatial correspondence

Figure 1. Ambiguities in multiple view tracking

Due to the inherent ambiguities in visual tracking of multiple objects in 3D-space we must address the simultaneous optimisation of the temporal correspondences as well as the spatial correspondence between views. This leads to the combinatorial optimisation problem to consider the set of all possible spatio-temporal matches and occlusions for multiple objects in multiple views to obtain a global optima. Evaluating the set of all possible correspondence is prohibitively expensive even for a small number of objects and views. For example with five objects in three views the total number of combinations to be considered is in excess of 1.7x10^6. In practice, heuristics from application knowledge can be used to prune the search space and reduce the combinatorics. However, for real-time applications the complexity remains prohibitive, because tracking performance is dependent on the worst cases, whose complexity can not always be reduced by heuristics. As a result, reduction of the intrinsic complexity of the problem is highly advantageous to practical systems.

1.3 Overview of Approach to Reduction of Computational Complexity

Since estimating the 2D temporal correspondence for each view independently is computationally much cheaper than solving the multiple view 3D-tracking problem, it may be possible that we can use the results of the 2D-tracking as prior information for the resolution of multiple view correspondence. In many situations the 2D-tracking will achieve either all or partially correct correspondences. However, in general the single view 2D temporal correspondences are unreliable due to the ambiguities caused by occlusion and clutter, as noted in section 1.2 with the TR strategy. The key question. which this paper addresses, is how can we utilise the unreliable prior information from independent 2D-tracking to reduce the complexity of the combinatorial optimisation of estimating correspondences between observations of multiple objects in multiple views whist maintaining reliable 3D-tracking.

In this paper we first model the multiple view 3D-tracking problem as a multi-layer box-ball allocation problem, section 2.1 and 2.2. We then introduce a discrete relaxation algorithm which uses unreliable prior information from independent 2D-tracking to reduce the combinatorial complexity, section 2.3. The algorithm sorts all possible combinations according to their consistency with the prior information. Relaxation directs the search for optimal results among all combinations in the descending order of their consistency to the prior. The search terminates when all possible combinations, which are consistent with our uncertainty in the prior information, have been evaluated. This results in an optimisation, which is robust to errors in our prior knowledge from independent 2D-tracking and greatly reduces the computational complexity. This is a general algorithm for integration of unreliable prior information into a combinatorial optimisation problem. Furthermore, the algorithm could be applied in conjunction with symbolic optimisation algorithms such as heuristic search or genetic algorithms to maintain their high performance while reduce the risk of local optima.

This algorithm is applied to the problem of 3D-tracking of skin-colour objects in multiple views, for the purpose of human motion capture, section 3. Results demonstrate that simultaneous optimisation over multiple views resolves the inherent ambiguities due to occlusion and clutter which produce errors in the independent 2D-tracking. The relaxation algorithm resolves errors in the 2D-tracking and significantly reduces the computational complexity to achieving reliable 3D-tracking by integration of observations from multiple views. Introduction of the 2D-tracking information as a prior greatly reduces the computational complexity enabling real-time performance. In the case of unambiguous 2D information the computational cost reduces to that of direct 3D-reconstruction from the 2D-tracking.

2. Modelling the 3D-Tracking Problem

In this section we cast the problem of 3D-tracking in the presence of occlusion and clutter as a combinatorial optimisation problem. This is modelled as a 'box-ball allocation problem' where the object labels are the boxes and the set of observation labels are the balls. The box-ball model is a standard representation of the possible label assignments for combinatorial optimisation problems [Wood 1993]. Modelling the problem in this way allows integration of information across multiple views, which enables simultaneous optimisation over spatio-temporal correspondence for the 3D-tracking problem. Solution requires finding the global optima of a cost function across all possible combinations. First we introduce the box-ball model for the single view case, then extend the model to the multiple view case using a 'multi-layer' box-ball model. Finally we show how introducing unreliable prior information can be used to enable efficient search whilst maintaining reliable correspondence.

2.1 Single View Tracking

The problem of tracking multiple objects in a single view image sequence can be stated as:

> Given a set of object labels $L_{t,}=\{l_{t,i}|i=1,..,n_t\}$, at time t and the set of observation labels $L_{t+1}=\{l_{t+1,j}|j=1,..,n_{t+1}\}$, at time $t+1$, find the set R_{t+1} of object-observation correspondence pairs $r_k=<l_{t,i},l_{t+1,j}>$ such that:
>
> $$R_{t+1}=\{r_k=<l_{t,i},l_{t+1,j}>|\ l_{t,i}\in L^*_t\subseteq L_t,\ l_{t+1,j}\in L^*_{t+1}\subseteq L_{t+1}\ \}; \qquad (1)$$

where L^*_t and L^*_{t+1} are the sub-set of 'stable' objects during the interval $[t,t+1]$. *Stable* objects are those observed at both time t and time t+1. It should be noted that L_t can contain labels for objects not observed at time t, thus if we know we are tracking a set of N objects the labels for each of these objects will always be present in the label set even if they were not observed at time t. Other objects due to scene clutter will appear as additional labels.

Finding correspondence pairs must allow for the possibility that objects appear and disappear due to occlusion and clutter. In general we minimise a cost function of the form:

$$E(R_{t+1})= \sum_{r_k\in R_{t+1}}\varepsilon_c(r_k)+\sum_{l_{t,i}\notin L^*_t}\varepsilon_d(l_{t,i})+\sum_{l_{t+1,j}\notin L^*_{t+1}}\varepsilon_a(l_{t+1,j}) \qquad (2)$$

ε_c measures the cost of a correspondence pair $r_k=<l_{t,i},l_{t+1,j}>$. Objects not in L^*_t are objects that disappeared at time $t+1$, with ε_d measuring the cost of 'disappearing'. Observations not in L^*_{t+1} are objects appearing at time $t+1$, with ε_a measuring the cost of 'appearing', both of these cases are generally treated as outliers for single-view tracking. The form of individual terms is dependent on the particular application as discussed in section 3.

The tracking problem of equation (1) can be cast as a combinatorial box-ball assignment problem [Wood 1993]. In the generic box-ball assignment problem we have a set of boxes and a set of balls. Each possible assignment of balls into boxes will generate an associated cost. In the general case some boxes may be empty and some balls may not be assigned to boxes. Empty boxes and unassigned balls will also contribute an associated cost. The optimal solution of the 'box-ball assignment problem' is the assignment of balls to boxes which gives the minimum cost. This is a combinatorial optimisation problem for which the cost of all possible assignments must be evaluated to guarantee that the global minima is found. A problem tree is commonly used to represent all possible assignment configurations as leaf nodes. The optimal solution is then obtained by evaluating the leaf node with minimum cost in either depth-first or breadth-first order. Searching the problem tree to evaluate the cost of all possible configurations provides a general solution. It should be noted that the assignment is symmetric, exchanging the meaning of the boxes and balls results in a different problem tree structure but exactly the same optimal solution.

In the single view tracking problem, we let the object labels $L_{t,}$ be the boxes and the observation labels $L_{t+1,}$ be the balls. Equivalently the observations could be boxes and objects balls resulting in the same set of object-observation combinations. Due to occlusion and clutter some boxes may be empty if an object dissappeared and 'new' boxes are required to allow for objects which may appear. All possible object-observation combinations are represented in a problem tree with each level representing a unique box (object) and each node representing the assignment of an observation to that box (object). Each child node is a possible observation to be assigned in the next layer. New boxes must be included in the problem tree as possible new object labels.

Figure 2(a) illustrates the problem tree for a simple combination of two boxes (objects) and two balls (observations) with the observation labels as (1), (2) or empty ( ) meaning the object is not observed. New box 1 and 2 allow for the possibility of one or both of the two observations corresponding to new objects. Each leaf node of the tree is a candidate solution, and finding the global optima requires evaluation of the cost for each leaf node according to equation (2). In this simple case there are a total of seven combinations to be evaluated.

The complexity of the problem is the total number of possible combinations, for n boxes and m balls, this can be calculated by the following recursive formula:

$$\begin{cases} F(n,m) = nF(n-1,m-1) + F(n,m-1) = mF(n-1,m-1) + F(n-1,m) \\ F(n,0) = F(0,m) = 1 \end{cases} \tag{3}$$

which gives the close-formed solution:

$$\begin{cases} F(n,m) = \sum_{i=0}^{m} C_m^i P_n^i \qquad for \quad n \geq m \\ F(n,m) = \sum_{i=0}^{n} C_n^i P_m^i \qquad for \quad n \leq m \end{cases} \tag{4}$$

where $P_m^i = \frac{n!}{(n-i)!}$ is the number of purmutations of i things from a set of n and $C_m^i = \frac{P_m^i}{P_i^i} = \frac{n!}{i!(n-i)!}$ is the number of combinations of i things from a set of n.

Equations (3) and (4) are symmetrical with respect to n and m ($F(n,m)$=F(m,n)), because the box and ball can alternate their roles without ambiguity. For single view tracking (2D tracking), this symmetry indicates that either the objects at time t or the observations at time t+1 can be used as the boxes for the box-ball assignment problem. The total complexity will rapidly explode with the increase of *n* and *m*, for example, *F(3,3)=34, F(4,4)=209, F(5,5) =1546.*

This is an NP-complete problem [Wood 1993], with no dynamic programming solution, because finding an optimal sub-problem does not guarantee the global optima. Branch clipping to prune invalid branches of the problem tree can in general reduce the complexity of such problems. In practice due to the inherent combinatorial complexity the labelling problem remains prohibitively expensive for multiple objects and observations [Haralick and Shapiro 1979]. Heuristic search using knowledge of the specific problem or stochastic techniques such as simulated annealing and genetic algorithms can also be applied but suffer from local optima. For 3D-tracking, possible heuristic clipping rules include the maximum number of objects, the maximum distance between objects and observations and their similarity in appearance (shape, colour etc.) [Zhang and Faugeras 1994]. The cost function is based on the distance between observations and predicted object positions together with fixed penalty cost for appearing and disappearing objects.

2.2 Multiple View Tracking

In this section we extend the standard box-ball model to represent all possible object-observation correspondences from multiple camera views in a single problem tree. A 'multi-layer' box-ball model is introduced to represent all possible combinations from multiple views. The different layers or levels within each box correspond to each view. Observations (balls) from each view are only assigned to the appropriate layer within a box. This leads to a problem tree hierarchy with multiple layers within each box, one for each camera view. For multiview tracking, stable objects at time *t* have to be the boxes and observations from all views at time *t*+1 will be the balls, therefore the symmetry of the single view assignment problem is lost.

Figure 2(b) illustrates the multi-layer box-ball model for the simple problem of one object (box) and two views (layers) with one observation (ball) for each view. Two new boxes are required to represent the possibility of one or both of the observations corresponding to a previously unobserved object due to clutter and the possibility that the observations correspond to different objects. The leaf nodes of the problem tree represent all possible combinations irrespective of the ordering of the layers within each box. It should be noted that even in this very simple case of a single object from two views there are five possible combinations to be evaluated. As in the single view case equation (2) defines the total cost of each solution where the cost function is now summed across all label correspondences in all views.

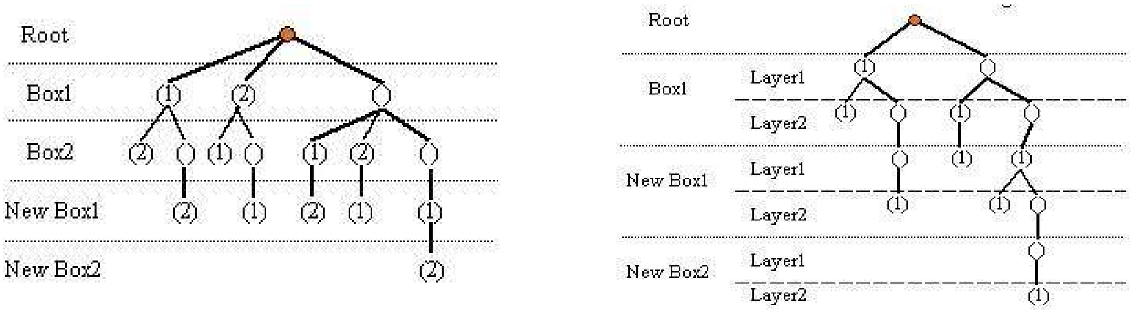
The complexity of the multi-layer box-ball problem is the total number of possible combinations across all views. For N boxes (objects) and K layers (views) with M_1, …, M_K balls (observations in each view) , the complexity is given by:

$$F(M_1,...,M_K,N) = \sum_{i_1=0}^{\min(M_1,N)} C_{M_1}^{i_1} P_N^{i_1} \quad (\sum_{i_2=0}^{\min(M_2+M_1-i_1,N)} C_{M_2}^{i_2} P_N^{i_2} \quad (\ ... \ (\sum_{i_K=0}^{\min(M_K+M_{K-1}-i_{K-1},N)} C_{M_K}^{i_K} P_N^{i_K}) \ ... \ )) \qquad (5)$$

In the special case where there are no outliers equation (5) simplifies to:

$$F(M_1,...,M_K,N)=\prod_{i=1}^{K}P_N^{M_i} \cdot \tag{6}$$

Equation (6) gives the complexity for the simplest case, without occlusion or clutter, which gives an indication of the minimum complexity for multiple view tracking. In practice for real tracking applications due to occlusion and clutter the complexity, from equation (5), is likely to be significantly greater than the simple case of equation (6). To achieve reliable real-time tracking we have to address the problem of finding the global minima whilst reducing the computational cost by efficient tree search.



(a) Single layer (b) Two layers

Figure 2. Problem trees for single and multi-layer box-ball assignment

2.3 Relaxation using Prior Knowledge

In this section we address the central issue of how to utilise unreliable prior knowledge to reduce the computational complexity in combinatorial problems such as multiview 3D-tracking. Independent 2D-tracking in each view provides a computationally efficient mechanism to obtain a set of prior 'unreliable' object-observation correspondences. Results of this tracking provide useful prior information to direct the 3D tracking for simultaneous optimisation of correspondence and reconstruction across multiple views. Here the term "prior" is used to denote the set of estimated correspondences from independent 2D tracking which provide prior knowledge for solving the correspondence problem at time t+1. In this case the prior is a discrete set of correspondence estimates rather than a continuous probability distribution on the likelihood of matches.

The use of 2D tracking as a prior for 3D tracking assumes consistency between the 2D temporal observation-observation correspondences and 3D spatial object-observation correspondence, for example if 2D observation label $l_{j,k,t}$ of layer (view) j is found to correspond to 3D object label l_i by the 3D-tracking at time t, and to correspond to $l_{j,m,t+1}$ by 2D tracking at time t+1 with the correspondence pair < $l_{j,m,t+1}$, $l_{j,k,t}$>, then if the prior 2D tracking is correct $l_{j,m,t+1}$ corresponds to l_i. . As a result we get the correspondence pair <l_i, $l_{j,m,t+1}$> as an uncertain prior for the 3D-tracking at time t+1. If the prior is correct then this object-observation correspondence does not have to be considered in the optimisation. In every layer, each 2D label observed at both time t and t+1 will contribute such an object-observation pair, and the final prior is a table of label correspondences. If the 2D tracking is correct, then all correspondences in the prior are correct too, and 3D-tracking only needs to deal with 2D objects that have disappeared and appeared resulting in lower complexity.

Table 1 shows an example of how the prior for a two-layer problem is derived. After 3D tracking at time *t*, we know that there are three 3D objects “A”, “D”, “E”. Also at time t in the first view (layer1) we observed two 2D objects “a” and “b”, with “a” corresponding to “A” and “b” to “E”; in the second view (layer2) we observed three 2D objects “f”, “g”, and “h”, with “g” corresponding to “A”, “f” to “D” and “h” to “E”.At time *t*+1, in the first view (layer1) we observed two objects “c” and “d” and in the second view (layer 2) we observed three objects “i”, “j” and “k”. From the 2D-tracking in the first view between time *t* and time *t*+1, we estimate that object “c” corresponds to “b”, and object “d” does not correspond to a stable object from time t. Since observation “b” corresponds to object “E” at time , we get the prior correspondence pair <c, E>. Following the same process, we obtain the set of prior correspondence for both views at time t+1 {<i, A>, <c, E>, <j, E>}. For imperfect 2D-tracking, the objects that appear such as label “d” in layer 1 might actually correspond to previous observations. The estimated prior correspondences between observations at *t* and *t*+1, such as label “i” and “j” in layer 2, might also be erroneous. The problem is then given these prior correspondence estimates with possible errors to establish the object-observation correspondence at time t+1.

	3D-tracking at time *t*			2D-tracking at time *t*+1 (<$l_{j,m,t+1}$, $l_{j,k,t}$ >)			Prior table for 3D-tracking at time *t*+1		
3D labels	A	D	E				A	D	E
2D labels for Layer1	a		b	<c, b>	<d, *null*>				c
2D labels for Layer2	g	f	h	<i, g>	<j, h>	<k, *null*>	i		j
Note	3 3D-objects are observed (A, D, E) with 2 and 3 2D-objects observations at layer 1 and 2, respectively.			In layer 1, old observation a has disappeared and a new observation d appeared; in layer 2, old observation f disappeared and new observation k appeared.			There are 1 and 2 prior information for layers 1 and 2, respectively. The result of 2D-tracking shows that the original 3D-object D has disappeared.		

Table 1. Prior Generation for 3D-tracking from 2D-tracking

In the presence of ambiguities such as occlusion, clutter and irregular movement there will be errors in the prior information which must be taken into account in searching for the optimal spatio-temporal correspondence. The problem is how to utilise the unreliable prior to improve the efficiency whilst ensuring that the global optima is reached. Relaxation labelling processes are an established mechanism for dealing with ambiguities and noise in vision systems [Hummel and Zucker 1983]. We introduce a ‘discrete relaxation’ algorithm which orders object-observation correspondences according to their consistency with the prior. Relaxing the consistency with the prior to allow for the maximum number of errors, according to a measure of the minimum reliability, ensures that the global optima will be included in the set of object-observation correspondences. Ordering enables efficient search for the optimal solution taking into account errors in the prior. Relaxation labelling processes are an established mechanism for dealing with ambiguities and noise. This is a general methodology for combinatorial optimisation problems for introducing uncertain prior knowledge with known reliability to reduce the computational complexity.

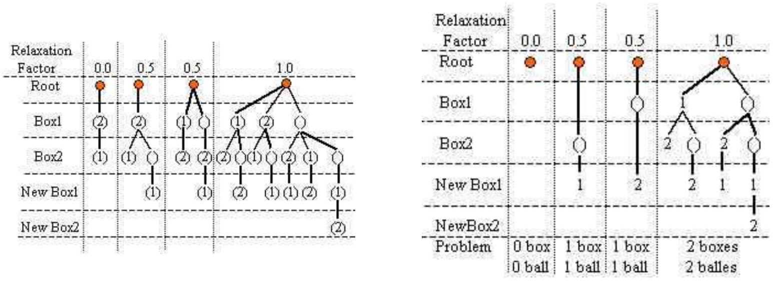
Given a set of prior object-observation correspondences, $R^{P}_{t+1}=\{r^{P}_{k} =<l_{t,i},l_{t+1,j}>|\ l_{t,i}\in L^{*}_{t}\subseteq L_{t}\ ,\ l_{t+1,j} \in L^{*}_{t+1}\subseteq L_{t+1}$ $k=0...N_{P}\}$, the prior is composed of two sub-sets the correct correspondences, R^{PC}_{t+1}, and the erroneous correspondences, $R^{PE}{}_{t+1}$, such that $R^{P}{}_{t+1} = R^{PC}{}_{t+1} \cup R^{PE}{}_{t+1}$. Taking an arbitrary sub-set of the priors, $R^{PS}{}_{t+1} = \{r^{P}{}_{j} \mid r^{P}{}_{j} \in R^{P}{}_{t+1} \;\; j=0...N_{PU}\}$, we can prune the problem tree such that for N objects and M observations the total number of combinations to be evaluated for a single-layer problem is given by $F(N\text{-}N_{PU}, M\text{-}N_{PU})$ in equation (4). A similar result will hold for the multi-layer problem in equation (5) where priors correspond to observations in specific views. As we do not know which subset of the priors are correct we must consider problem trees for multiple subsets, $R^{PS}{}_{t+1}$, of the prior to establish the set of correct correspondences $R^{PS}{}_{t+1}=R^{PC}{}_{t+1}$.

A discrete relaxation labelling algorithm is applied to relax the consistency with the prior information allowing for erroneous priors. This produces ordered sets of object-observation correspondences according to their consistency with the prior. We identify the set of all possible solutions which are consistent with the prior up to a 'relaxation factor'. The relaxation factor, λ, is defined from the total number of priors, N_P, and number of priors used, N_{PU}, as: $\lambda=(N_P - N_{PU})/N_P$ where $\lambda \in [0,1]$. A relaxation factor $\lambda=0$ corresponds to complete consistency with the prior information (no errors) and $\lambda=1$ corresponds to no prior information such that the complete problem tree must be evaluated. For a given value of λ there are a number of possible sub-sets of the prior, $\{R^{PS}{}_{t+1}\}$, such that the number of priors in each set is N_{PU}. Each sub-set of priors produces a pruned problem tree which must be evaluated.

Let us assume a lower bound, c_{min}, for the reliability of the prior information, $R^{P}{}_{t+1}$, such that the number of correct correspondences $N_{PC} \geq c_{min}N_P$. Then we can define the maximum relaxation factor, $\lambda_{max} = (1\text{-}c_{min})$, such that the number of prior correspondences used $N_{PU} \geq (1\text{-}\lambda_{max})N_P$. Gradual increase of the relaxation factor λ from 0 to λ_{max} in discrete steps corresponding to $N_{PU}=[N_P, (N_P - 1),...,(c_{min}N_P)]$ produces all possible subsets, $R^{PS}{}_{t+1}$, in the order of their consistency with the prior. By using each of these subsets, the original problem tree is simplified to a set of sub-trees. The optimisation is performed with reduced complexity by evaluation of combinations for each of the sub-trees. Given a strict lower bound for the reliability, c_{min}, the set of sub-trees is guaranteed to contain the optimal solution. In practice an appropriate lower bound for the reliability can be estimated for a particular application from the worst-case failure of the prior estimator across a wide range of examples, such as 2D tracking as discussed in section 3.

Figure 3(a) illustrates the problem sub-trees which are generated in the simple single layer (view) problem of two boxes (objects) and two balls (observations) and two priors with relaxation factors 0, 0.5 and 1. The prior correspondences are <1,2> and <2,1> where the correspondence $<l_{t,I}, l_{t+1,j}>$ is between the object label, $l_{t,\mathrm{i}}$, and the observation $l_{t+1,j}$. Relaxing this prior with a relaxation factor $\lambda=0.5$, gives two possible sub-trees with either $<1,2>$ or $<2,1>$ as the prior, Figure 3(c). With λ_{max}=0.5, the original problem tree is decomposed into four sub-trees three of which must be evaluated as shown in Figure 3(a). This reduces the number of combinations to be evaluated from seven to five. Many intermediate nodes are repeated in the sub-trees, so the computational complexity of finding the global optima is not significantly reduced. This repetition increases with problem size resulting in increased complexity, elimination of this inherent redundancy is addressed in the next section.

In practice when evaluating correspondences according to a cost function as in equation (2) we may terminate the search after the cost falls below a threshold based on the expected noise level in the data. Likewise if given a maximum relaxation factor the optimal matching cost does not fall below the expected noise level then we may have a greater number of errors in the prior and further relaxation has to be performed on the priors. For the application of 3D tracking presented in section 3 having set the prior reliability from the worst-case 2D tracking failure for a set of example sequences all solutions were found within the maximum relaxation factor.



Step	*f*	Prior	Tabu

1	0.0	2(1)
		1(2)	

2a	0.5	2(1)	1(2)
2b	0.5	1(2)	2(1)

3	1.0		1(2)
			2(1)

(a). Directly using prior rules without tabu.

(b). Tree decomposition by uncertain prior and tabu

(c). Relaxation steps, prior, and tabu

Figure 3. Problem Tree Decomposition using Prior Knowledge

2.4 Eliminating Redundancy in Relaxation

In the previous section we introduced a gradual relaxation algorithm which produces a set of problem sub-trees in the order of their consistency with the prior. Direct evaluation of all combinations for the problem sub-trees results in redundancy due to the repetition of nodes, as illustrated in Figure 3(a). Redundancy results from prior correspondences being included as non-prior correspondences in the relaxed problem sub-trees. For example, in the second sub-tree of Figure 3(a) with $\lambda=0.5$ and prior <2,1>, although <1,2> is no longer a prior, the corresponding nodes are still generated in the problem sub-tree. Appendix A gives a mathematical analysis of the complexity resulting from this redundancy.

In order to avoid this redundancy, we introduce the idea of "negative" prior or tabu. Unlike the "positive" prior, negative prior forbids a certain ball (observation) to be allocated into a certain box (object). In a specific relaxation step which uses the prior sub-set R^{PS}_{t+1} with N_{PU} prior rules, the set of all unused priors, $R^{T}_{t+1}=\{r^{P}_{k}\notin R^{PS}_{t+1}\}$, serves as the tabu-set of $N_T=N_P-N_{PU}$ negative prior rules. The tabu-set are used as clipping rules for generation of the problem sub-tree in each step of the discrete relaxation algorithm. All nodes corresponding to negative priors are not included in the problem sub-tree. As a consequence, the resulting sub-trees are *not* topological sub-trees of the original problem tree. No nodes are repeated for all sub-trees generated by increasing the relaxation factor from 0 to 1. This eliminates the inherent redundancy in the relaxed problem sub-trees, which minimises the computational complexity in evaluating the global optima for a given maximum relaxation factor.

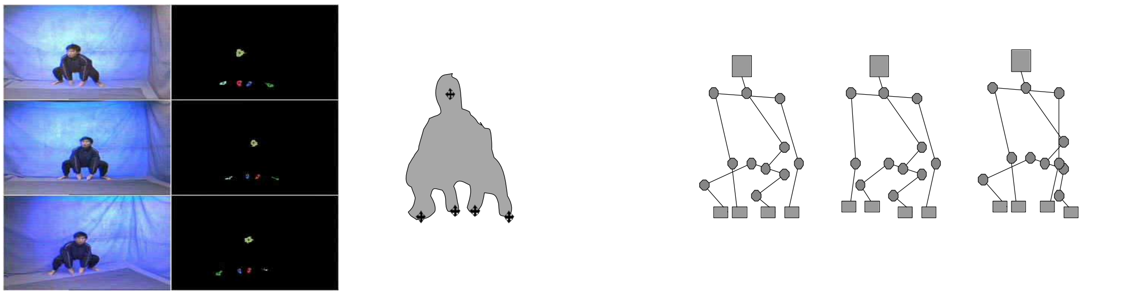
Figure 3(b) shows the decomposed problem sub-trees when using the tabu set on the same example as Figure 3(a). For the same maximum relaxation factor λ_{max}=0.5, the complexity is further reduced from five to three and no node is repeated. An analysis of the computational complexity of this approach is given in Appendix A, which shows that under ideal condition this relaxation algorithm reduces the intrinsic combinatorial complexity with respect to the number of layers (views), K, number of priors, N_P, and the maximum relaxation factor, λ_{max}, by a factor of at least $(1/(\lambda_{max} \cdot N_P)!)^K$. For example with five objects, three views and five priors with a relaxation factor of λ_{max}=0.6, even under ideal conditions the resulting complexity is 1/216 of the original size. If empty boxes and new boxes are allowed, the reduction rate will be even greater. For example, in the simple case of Figure 3, if empty and new boxes are not allowed, the complexity will be reduced by a factor 1/((0.5·2)!)=1.0, i.e., no reduction. However, allowing for objects appearing and disappearing the complexity is reduced by a factor 3/7, as shown in Figure 3(b). The reason is that when outliers exist, there will be more boxes for lower layers of the problem tree because of the new generated boxes in upper layer, and the reduction in complexity will increase with the increase in the number of boxes. Appendix A presents a detailed analysis of the algorithm complexity and Appendix B presents details of the algorithm implementation.

3. Application: Real-time 3D-Tracking of Human Movement Using Multiple Cameras

In this section we apply the relaxation algorithm to the problem of image based human motion capture from multiple calibrated cameras. This is an inherently difficult 3D-tracking problem due to the high number of degrees-of-freedom, occurrence of self-occlusion and clutter from both the subject and background, and difficulty of stereo correspondence between views with a large baseline.A large baseline between camera views is necessary to ensure that salient features are not self-occluded in all views for movements such as turning around. In this section we investigate the application of the framework developed in section 2 for a three camera system. The intention here is to demonstrate the utility of the framework to obtain correct spatio-temporal correspondence. To capture the full range of human movement a larger number of cameras are required and the framework could equally be applied to such a system.

Feature based tracking using cues such as skin-colour has been shown to provide a useful source of information for efficient tracking of human movement [Schiele et al. 1995, McKenna et al. 1998, Wren et al.1996, Yonemoto et al. 2000]. Resulting feature correspondence can be used to provide strong constraints on a kinematic model and hence reduce the degrees-of-freedom in posture estimation. However, appearance based correspondence methods will fail for ambigous situations such as multiple observations close to the same epi-polar line, as discussed in section 1. This situation is common for skin tracking during human movement. For the specific case of three cameras use of the trifocal tensor [Hartley and Zisserman, 2000] provides an addtional constraint. However, in general this does not help resolve multiview ambiguities. Crowley et al. [Crowley et al. 1992] presented the underlying theory of the TR approach, due to inherent ambiguities of self-occlusion and clutter independent tracking and reconstruction fails for human movement.Figure 4 shows the captured images and 2D skin feature tracking for a typical ambiguous human posture which may result in tracking failure. The camera configuration for the three views is illustrated in Figure 6(a) , the large rectangular regions are the three image planes and the small regions the reconstructed hand, head and feet locations. Detailed calibration parameters are listed in Appendix C.

In this section, we address the problem of tracking and labelling multiple parts of a person performing complex movements where 2D tracking from a single view or direct spatial correspondence between views fails. The purpose of this application is to demonstrate that the framework introduced in section 2 enables us to reliably and efficient integrate observations from multiple views in the presence of ambiguities. In this work we do not introduce a kinematic model and treat object motion independently. A kinematic model could be used for motion prediction if desired but may result in unreliable prediction for irregular movement. Results of the 3D tracking could be used to constrain the degrees-of-freedom of a kinematic model. Independent 2D tracking of skin-colour features is performed in each view with low computational complexity. Results of the 2D tracking serve as a prior for the discrete relaxation algorithm. Simultaneous 3D tracking over multiple views is used to resolve ambiguities and errors which result for tracking in any single view.



(a) Multi-view images/regions (b) Silhouette & Features (c) Three ambiguous postures with same feature/silhouettes

Figure 4. Example of posture ambiguity for the silhouette and skin-regions without 3D-tracking

3.1 2D-Tracking

In this application the head, hands and feet of a person are tracked in individual 2D views using pixel classification based on skin-colour and region segmentation. An overview of the system used for 2D-tracking is illustrated in Figure 5. Motion prediction is performed assuming a constant acceleration model. Matching between regions identified in subsequent time frames is performed by comparing the distance to the predicted location together with the similarity of the colour distribution and shape for candidate regions. A single layer box-ball model, as presented in section 2.1, is used to find the optimal match between 2D regions. In this work results of 2D-tracking serve as prior information for the 3D-tracking. The performance of the algorithm and conditions under which 2D-tracking fails due to self-occlusion and clutter are expected to be similar to previous approaches using colour regions or blobs, such as [Wren et al. 1996, McKenna et al.1998]. A blue screen studio is used to reduce background clutter although some erroneous background regions occur due to shadowing. Three synchronised calibrated cameras (30Hz) are used with a baseline of approximately 1.5m between them so that all parts of the body are visible in at least one camera view throughout the captured sequences.

A typical set of images captured from the three camera views is shown in Figure 4(a) together with the extracted colour objects. Due to the wide-baseline it is not possible to directly match image regions between views based on their appearance. Use of the epipolar constraint to match observations of body parts between views fails for postures such as that shown in Figure 4. For our experimental conditions the average error rate for 2D body-part tracking in each view for complex human movement is around 5% with a worst-case error of 40%. In this work the worst-case error was evaluated by taking the worst 2D-tracking failure (total number of incorrectly labelled objects) for any image frame across all three views for ten test sequences of complex movements comprising a total of approximately 13K frames. The 2D-tracking error is composed of three factors: false positive errors (background clutter); true negative errors (occlusion of body part); and 2D-tracking errors (incorrect correspondence) with average error rates of 1.8%, 3%, 0.2% and worst case 40%, 33%, 40% respectively. The 2D skin-region tracking is performed in approximately 10ms per frame on a 450MHz PentiumIII CPU.

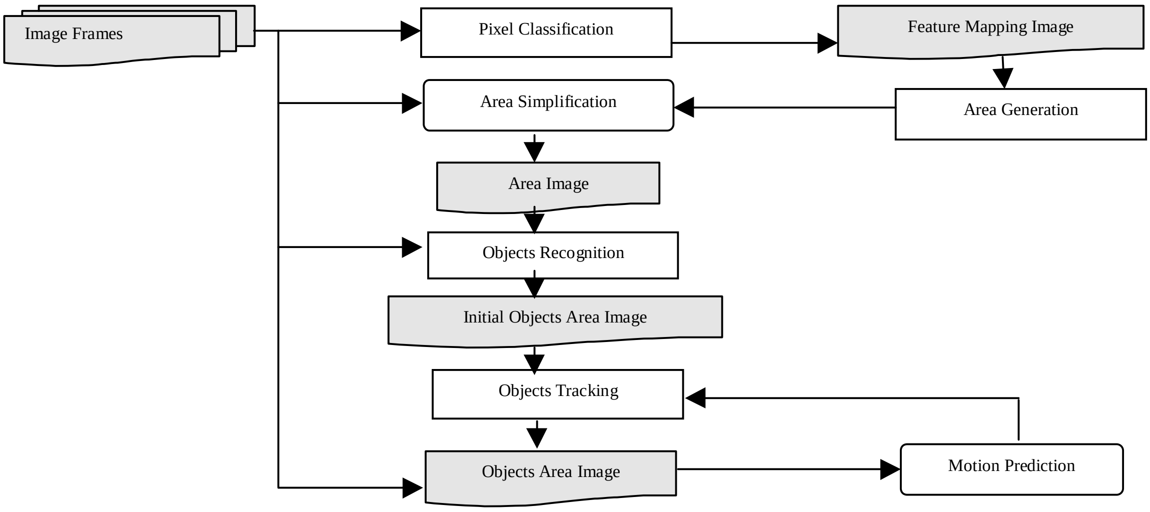


Figure 5: Outline of 2D-tracking Framework

Prior information is derived from “stable” 2D and 3D labels in the previous 3D-tracking and current 2D-tracking results, as described in section 2.3. Constant acceleration motion prediction is used for both 2D and 3D-tracking, with velocity *x'* and acceleration *x"* at time *t*+1 evaluated from previous positions:

$$\begin{aligned}
x''_{t+1} &= x''_t = x'_t - x'_{t-1} = [x_t - x_{t-1}] - [x_{t-1} - x_{t-2}] = x_t - 2x_{t-1} + x_{t-2} \\
x'_{t+1} &= x'_t + x''_{t+1} = 2x_t - 3x_{t-1} + x_{t-2} \\
x_{t+1} &= x_t + x'_{t+1} = 3x_t - 3x_{t-1} + x_{t-2}
\end{aligned}$$

A more sophisticated approach could be used to predict the object position using a similar dynamic model with the covariance of previous estimates such as the Kalman filter [Bar-Shalom 1988]. For the purposes of this work, the simple dynamic model has been found to be sufficient and practical. In the case of human motion, which can be highly irregular, a more sophisticated model may not significantly increase performance.

3.2 3D-Tracking

Applying the multi-layer box-ball model introduced in section 2.2 to 3D-tracking, the boxes are the 3D objects to be tracked, and the balls are actually 3D rays defined from 2D image feature points as well as camera parameters. Figure 6(a) shows an example of tracking five skin-coloured objects from three views. The three grey regions are the image planes of three cameras; the lines are the 3D rays defined by 2D feature points, which are the centroids of skin-coloured objects from 2D area-based tracking; and the dark boxes are the 3D objects reconstructed and tracked.

For a multi-camera network, as illustrated in Figure 6(b), the position and orientation of a specific camera is given by the external rotation and translation parameters (***R***,***T***) which define the relation between the world co-ordinate *p* and the local co-ordinate *p*' as:

$$p' = \boldsymbol{R} \cdot p + \boldsymbol{T}$$

Using a pin-hole camera model the intrinsic parameters for the centre of projection (u,v) and focal lengths (f_x, f_y) give the 3D ray for image point (x,y), as illustrate in Figure 6(b):

$$s = -\boldsymbol{R}^{-1} \cdot \boldsymbol{T}$$

$$d = \boldsymbol{R}^{-1} \cdot ((x-u)/f_x,\ (y-v)/\ f_y,\ 1)^{\mathrm{T}}$$

where s is the start point, i.e., the focal centre of the camera, and d the direction vector.

If a set of observation rays, $\boldsymbol{S}=\{R_i(s_i,d_i),\ i=1,\ldots,\mathrm{N}\}$ where s_i is the start point and d_i the normalised directional vector, are allocated into one box, then the 3D point reconstructed from these rays $p_r(\boldsymbol{S})$ is the current estimated position of the 3D object corresponding to this box. Here we perform a standard least-square reconstruction method, which minimises the sum of square distances from p_r to all rays in $\boldsymbol{S}$**:**,

$$\arg\min_{p_r} \sum_{i=1}^{N} dis^2(p_r, R_i) \tag{7}$$

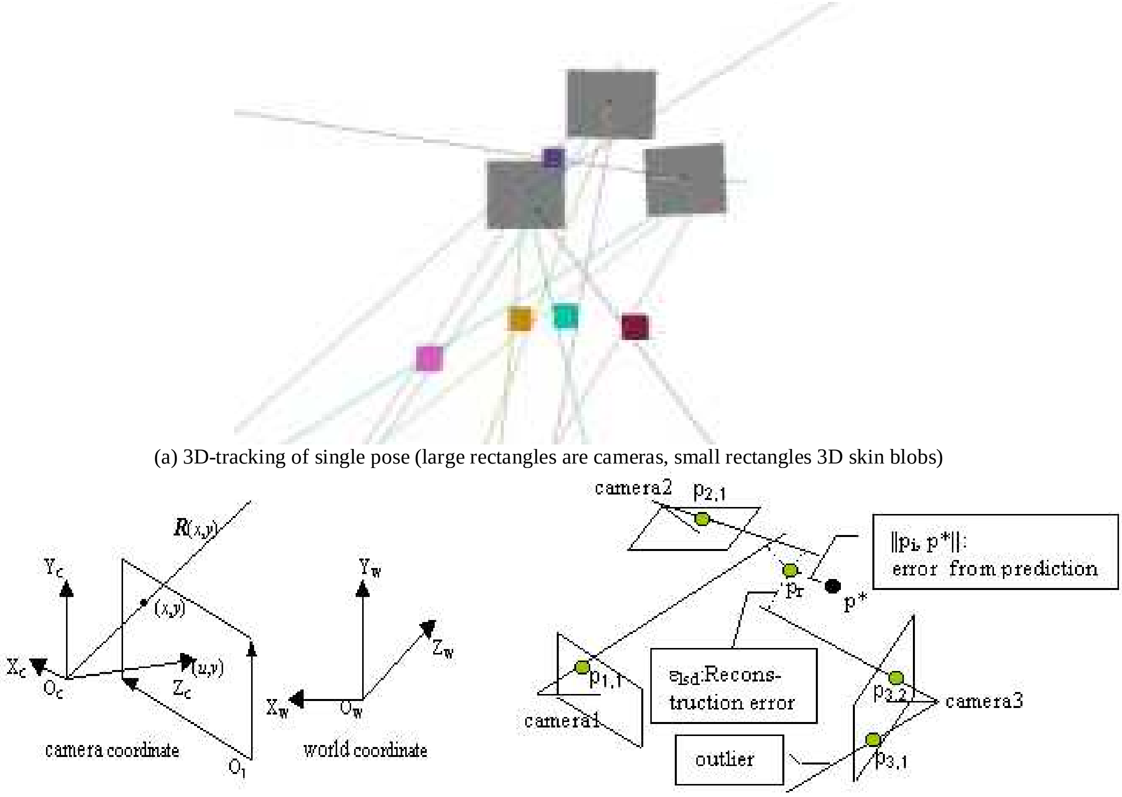
where $dis^2(p_r, R_i)$ is the squared Euclidean distance between the 3D point p_r and 3D ray $R_{\cdot\mathrm{i}}$. The reader is referred to [Faugeras 1993] for more details about pinhole camera geometry and least-square 3D reconstruction from multiple views.

For a specific 3D object with label $l_{t,i}$ and predicted position $p_p(l_{t,i})$, if a set of 2D objects from different views (layers) with labels $<l_{t+1,1,\ldots\ldots}, l_{t+1,n}>$ is allocated to it, the cost of this correspondence $r_j=<l_{t,i}\ , l_{t+1,1,\ldots\ldots}, l_{t+1,n}>$ is then evaluated as a truncated sum of the reconstruction error and the prediction error:

$$\varepsilon_c(r_j) = \alpha \bullet \varepsilon_{lsq}(l_{t+1,1\ldots\ldots}, l_{t+1,n}) + (1-\alpha) \| p_p\ (l_{t,i}) - p_r(l_{t+1,1\ldots\ldots}, l_{t+1,n}) \| \tag{8}$$

where $p_r(l_{t+1,1\ldots\ldots}, l_{t+1,n})$ is the 3D point reconstructed from the rays define $<l_{t+1,1,\ldots\ldots}, l_{t+1,n}>$, and ε_{lsq} is the least-square reconstruction error for the 3D point p_r defined by (7). The scalar weight, α, is a normalisation factor proportional to the number of 2D labels allocated to this 3D object. Outliers corresponding to objects that are appearing and disappearing between subsequent time frames are assigned a constant penalty.

Figure 6(c) illustrates how the 3D reconstruction from 2D points, $p_{1,1}$ on camera 1, $p_{2,1}$ on camera 2 and $p_{3,2}$ on camera 3, is performed, and how the cost is evaluated from this reconstruction and the predicted position p^* of the 3D object to be tracked.



(a) 3D-tracking of single pose (large rectangles are cameras, small rectangles 3D skin blobs)

(b) camera framework

(c) 3D-reconstruction and error measuring

Figure 6: 3D-tracking, camera framework, 3D reconstruction and error measurement

3.3 Results

The relaxation algorithm presented in section 2 has been applied to the problem of 3D-tracking of body parts for sequences of human movement totalling 13K frames. Results are presented for five sequences of movements captured in three views and three sequences captured in only two views. Many of the movements are of sufficient complexity that the 2D-tracking fails. In each camera view the location of five body parts (head, hands and feet) are tracked in 2D using the approach outlined in section 3.1. The 3D-tracking is then performed across all three camera views using prior information collected from previous 3D-tracking and current 2D-tracking results for the discrete relaxation algorithm of sections 2.3 and 3.2.

As the worst-case frame-to-frame error rate for the 2D-tracking is known to be 40% a maximum relaxation factor of, $\lambda_{max} = 0.6$, is used throughout to ensure that the global minima of the cost function is included in the sub-tree decomposition of the problem tree. All labels are initialised according to their label in the 1st frame. No further search is performed if the reconstructed object position does not fall below the maximum distance threshold, $d_{max} = 0.5m$, between the reconstructed and predicted position. This threshold is the maximum expected reconstruction error due to errors in the centroid calculation and camera calibration between views. Therefore, the results presented give a direct evaluation of the algorithm for relaxation using unreliable prior knowledge presented in section 2. For all sequences tested the 3D-tracking only fails to reconstruct the correct 3D position when the object is occluded in two or more of the three views. Costs for appearance and disappearance of objects in equation (2) are set to constant values $\varepsilon_a=\varepsilon_d=(d_{max}/3)^2$.

Tables 2 and 3 summarise the results of the tracking process in terms of the failure rate for each of the 2D views and the corresponding 3D-failure rate for incorrect labelling of the reconstructed body part locations in three and two view sequences of increasing complexity. These tables also give the average computation time per frame for 3D-tracking on a single 450MHz PentiumIII CPU with 128Mb RAM.

These results demonstrate that there is considerable improvement in the reliability of the tracking process when the correspondence is integrated over multiple views. The total tracking error is considerably reduced for all image sequences. For the spinning and waving hands sequences there are instances where one or more of the objects is occluded in all views resulting in possible failure of the 3D-tracking. In the running sequence rapid and irregular movement of the feet, which violate the constant acceleration assumption causes two 3D tracking errors. In all other cases the simultaneous optimisation over multiple views correctly tracks all 3D objects all the time, despite the large number of errors in the 2D-tracking.

Results for computation times given in Tables 2 and 3 demonstrate that the use of uncertain prior knowledge together with the relaxation algorithm presented in section 2.3 results in a considerable reduction in the computational cost whilst maintaining reliability. The 3D-tracking results are the same with and without relaxation. The average computational cost is reduced by at least an order of magnitude over direct computation without the prior-based relaxation algorithm. This enables 3D-tracking to be performed in real-time even when the correspondence of all objects is ambiguous. The graphs in Figure 7 show the number of cost function evaluations (iteration number) vs. frame number for four of the movement sequences with and without relaxation. Peaks in the computation time correspond to situations where there are errors in the 2D-tracking due to ambiguity in one or more views. In all cases these ambiguities are correctly resolved in considerably less time than evaluation without relaxation which is performed over all object-observation combinations where the maximum distance between the predicted position and observed image ray is less than d_{max}.

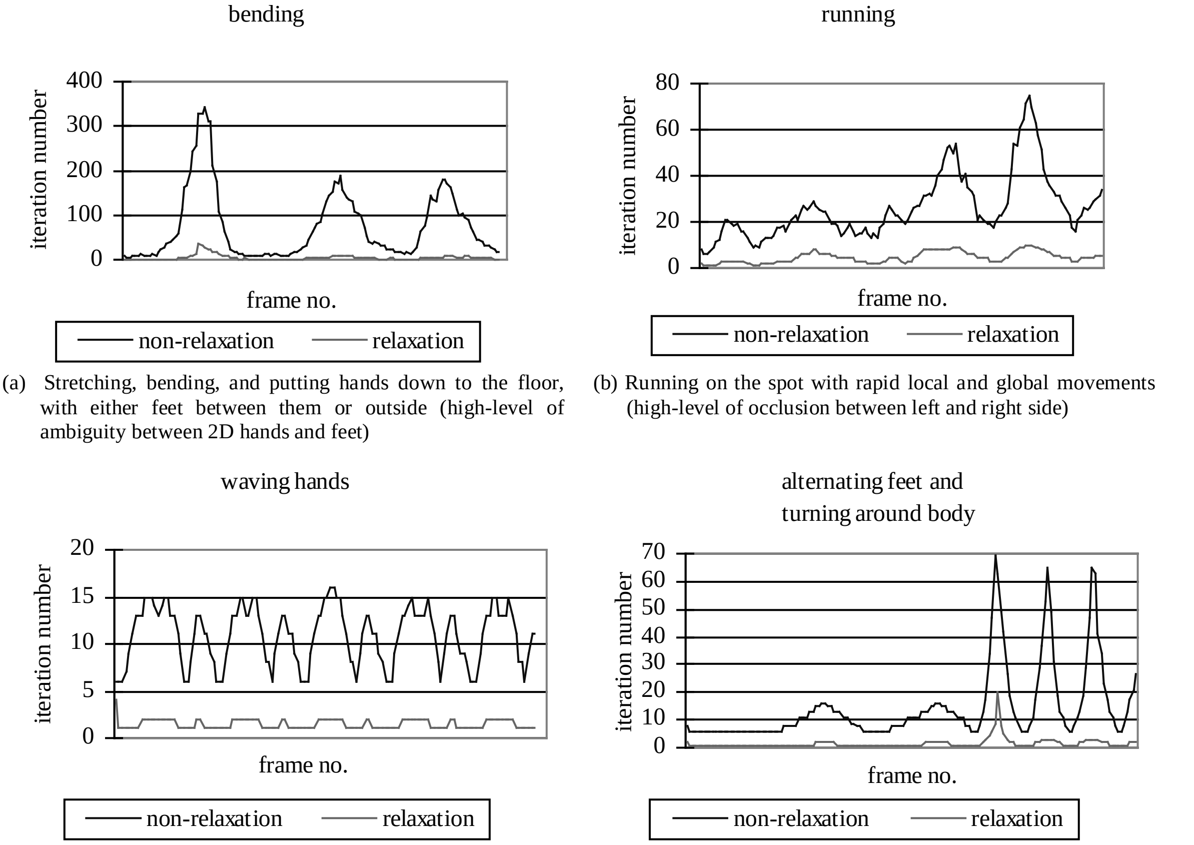
Sequence		Walking	Bending	Jumping	Running	Spin Hands
Frame number		3x450	3x300	3x300	3x300	3x200
2D-tracking	False Positive	90	145	110	110	40
	True Negative	20	190	120	240	120
	2D-tracking	0	5	4	12	17
3D-tracking	Total Tracking Error	0	0	0	2	4
Avg. Frame Rate (Frame/Second)	No Relaxation	2	0.3	0.5	0.5	0.3
	Relaxation	20	13	17	11	14

Table 2. Experiment results for five 3-view sequences of increasing complexity

Sequence		Waving Hands	Alternating Feet	Bend and Twist
Frame number		2x154	2x150	2x136
Approximate 2D-tracking error number	False Positive	0	0	0
	True Negative	38	22	0
	Error Tracking	22	12	0

3D-tracking	Error Tracking	1	0	0
Avg. Frame Rate (Frame/Second)	No Relaxation	5.5	8.5	6.2
	Relaxation	>20	>20	>20

Table 3. Experiment results for three 2-view sequences



(a) Stretching, bending, and putting hands down to the floor, with either feet between them or outside (high-level of ambiguity between 2D hands and feet)

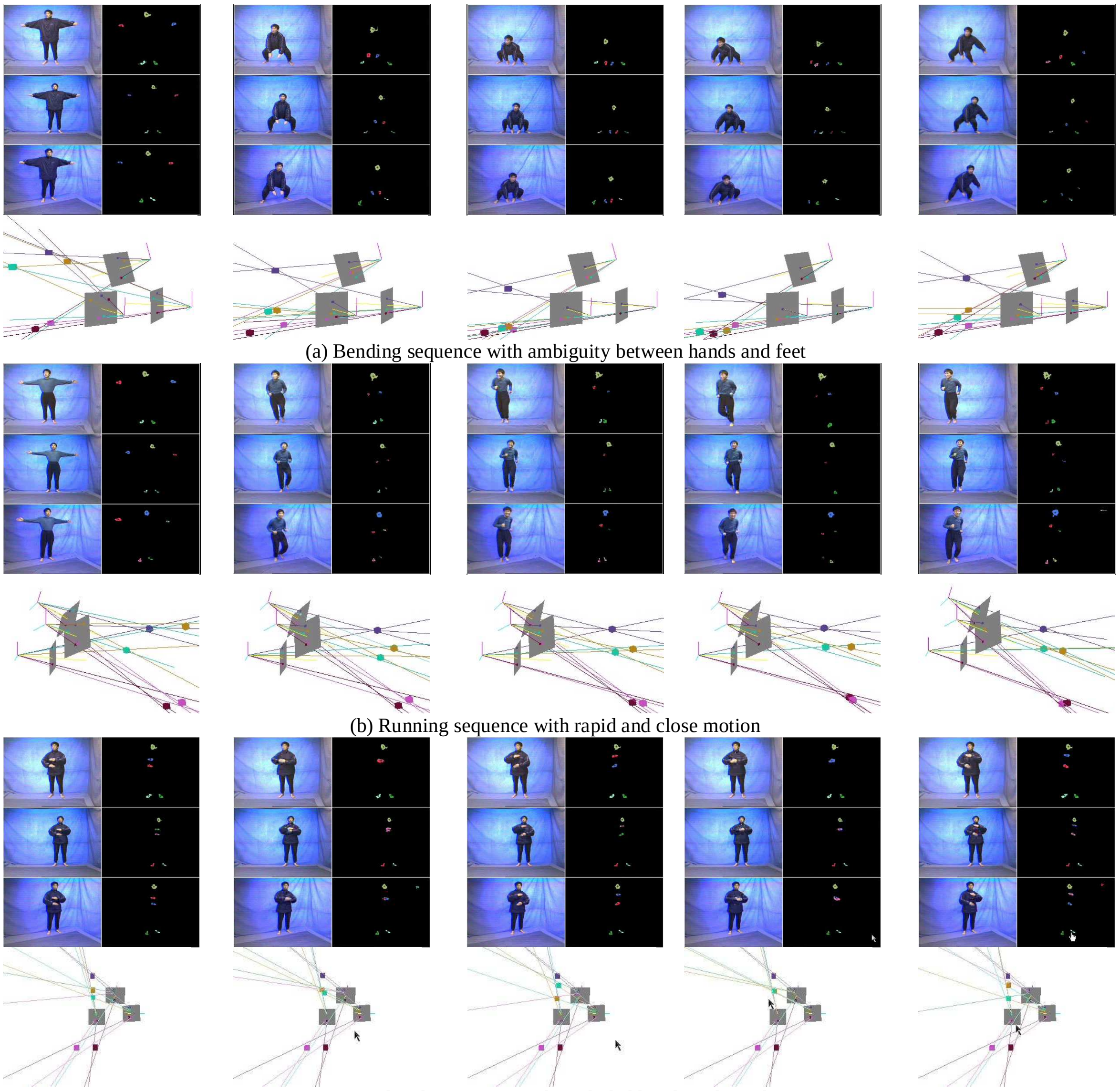
(b) Running on the spot with rapid local and global movements (high-level of occlusion between left and right side)

(c) waving hands in front of the body (rapid movement with occlusion)

(d) alternating left and right feet, then turning body around (feet disappearing and reappearing)

Figure 7: Performance comparison of 3D-tracking with and without the prior-based relaxation

Figure 8 shows the results of 3D-tracking for three of the sequences that contain highly ambiguous postures. The upper portion of each sequence shows the captured multiple view images together with the 2D skin-colour regions identified by tracking labels, the lower diagram shows the 3D-reconstruction with the camera views, 3D rays for region centroids and the labelled reconstructed 3D points (coloured according to their label). For all three sequences the camera configuration is the same as Figure 6(a), and the parameters are listed in Appendix C. To allow visualisation of the depth information, they are presented in different viewpoints and directions. Figure 8(a) is a sequence of crouching postures which results in incorrect 2D correspondence between hands and feet. Figure 8(b) is a running sequence in which the hands and feet performed rapid and close motion resulting in 2D-tracking failure. Figure 8(c) is a spinning hands sequence with self-occlusion and rapid motion. In all cases the 3D-tracking resolves the ambiguity by integrating information from multiple views. Movies of these experiments on our web page give more intuitive illustration (http://www.ee.surrey.ac.uk/CVSSP/3DVision/3DTracking).



(a) Bending sequence with ambiguity between hands and feet

(b) Running sequence with rapid and close motion

(c) Spinning hands sequence with occluded hands and rapid motion

Figure 8. Results of 3D-tracking for 3 sequences with a high-level of 2D-tracking errors

4. Conclusion

In this paper we have introduced a discrete relaxation algorithm which uses unreliable prior knowledge to reduce the inherent computational complexity of combinatorial optimisation problems. This algorithm has been applied to the problem of 3D-tracking from multiple views to address two key issues:

- Simultaneous optimisation of multiple view correspondences and 3D reconstruction
- Computation of efficient solutions for real-time performance

Reliable 3D-tracking in the presence of occlusion and clutter requires the simultaneous correspondence and 3D reconstruction across multiple views. However, many approaches in computer vision address the problems of correspondence and reconstruction independently due to the inherent combinatorial complexity that prohibits real-time performance.

Gradual relaxation to allow for errors in the prior knowledge gives a set of sub-trees which are ordered according to their consistency with the prior. Sub-tree decomposition using tabu-sets eliminates redundancy in evaluating the optimal solution. Relaxation up to a maximum reliability in the prior greatly reduces the computational complexity and is guaranteed to find the optimal solution provided the reliability of the prior is correct. This is a general approach to using unreliable prior knowledge to reduce the complexity of combinatorial optimisation problems.

The discrete relaxation algorithm has been applied to the problem of tracking parts of the human body (head, hands and feet) from three camera views. Independent 2D-tracking in each camera view results in a high-level of tracking errors due the inherent ambiguity from occlusion and close proximity between observations which results in incorrect correspondences. Application of the relaxation algorithm using independent 2D-tracking as uncertain prior knowledge greatly reduces the failure rate whilst maintaining real-time performance. Results on multiple sequences containing ambiguous movements demonstrate that simultaneous optimisation over multiple views gives reliable 3D-tracking. The use of uncertain prior knowledge reduces the computational cost of 3D-tracking by an order of magnitude.

Further work will investigate the 3D-tracking of body parts for the full range of human movement with an increased number of camera views. Extension of this approach will incorporate kinematic and dynamic models of human motion to constrain the optimisation and improve 3D-motion prediction.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) and the UK Department of Trade and Industry (DTI) on Broadcast LINK project PROMETHEUS GR/M88075 and EPSRC Advanced Fellowship AF/95/2531. The authors would like to thank the project collaborators BBC and BT for facilitating this research.

Reference

[1] Aggarwal,J.K. and Cai,Q. Human Motion Analysis: A Review, Computer Vision and Image Understanding, 73(3):428-440, Academic Press, 1999

[2] Ayala, K., Orton, D.A., Larson, J.B., Elliott, D.F., Moving Target Tracking Using Symbolic Registration, PAMI 4(5), 515-520, 1982.

[3] Bar-Shalom,Y. Multitarget-multisensor tracking: Application and Advances, Storrs, CT, 1996.

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

[5] Blackman S. S., Multiple Targets Tracking with Radar Application, Artech House, Norwood, MA 1986.

[6] Borri, A., Bucci, G., Nesi, P., A Robust Tracking of 3D Motion, Proc. ECCV94, A:181-188.

[7] Burl, J.B., A Reduced Order Extended Kalman Filter for Sequential Images Containing a Moving Object, IEEE Trans. Image Processing 2(3), 285-295, 1993.

[8] Chen H., Huang T. S., Maximum Matching of 3D Points for Multiple-Object Motion Estimation, Pattern Recognition 21(7), 75-90, 1988.

[9] Chetverikov D., Verestóy J., Tracking Feature Points: A New Algorithm, Proc. International Conf. on Pattern Recognition, 1436-1438, 1998.

[10] Collins,R.T., Lipton,A.J. and Kanade,T. Introduction to the Special Issue on Video Sureveillance, PAMI 22(8):745-746, 2000.

[11] Cox, I. J., A Review of Statistical Data Association Techniques for Motion Correspondence, IJCV 10(1), 53-66, 1993.

[12] Cox I. J., A Maximum Likelihood N-camera Stereo Algorithm, Proc. CVPR(94), 733-739.

[13] Crowley J. L., Stelmaszyk P., Skordas T. and Puget P., Measurement and Integration od 3-D Structures by Tracking Edge Lines, IJCV 8(2), 1992.

[14] Faugeras O., Berthod M., Improving Consistency and Reducing Ambiguity in Stochastic Labelling: An Optimization Approach, PAMI 3, 412-423, 1983.

[15] Faugeras O. Maybank S., Motion from Point Matches, Multiplicity of Solutions, IJCV 4(3), 225-246, 1990.

[16] Faugeras O., Three-Dimension Computer Vision, MIT Press, Cambridge, 1993.

[17] Gavrila,D.M. and Davis,L.S. The Visual Analysis of Human Movement: A Survey, Computer Vision and Image Understanding, 73(1):82-98, Academic Press, 1999.

[18] Gong, S., McKenna, S.J. and Psarrou, A. Dynamic Vision: From Images to Face Recognition, Imperial College Press, London, 2000.

[19] Haralick R., Shapiro L., The Consistent Labelling Problem, PAMI 1, 129-139, 1979.

[20] Hartley D. and Zisserman A., Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[21] Hilton,A. and Fua,P. Forward to the Special Issue on Human Modelling, 81(2):143-144, 2001.

[22] Huang T. S., Netreravali A. N., Motion and Structure from Feature Correspondences: A Review, Proc. IEEE, 82(2), 252-268, 1994.

[23] Hummel R., Zuker S., On the Foundation of Relaxation Labelling Process, PAMI 5, 267-286, 1983.

[24] Hwang V., Tracking Feature Points in Time-varying Images Using an Opportunistic Selection Approach, Pattern Recognition, 22:247-256, 1989.

[25] Isard, M., Blake, A., A Mixed-State Condensation Tracker with Automatic Model-Switching, Proc. International Conf. on Computer Vision, 107-112, 1998.

[26] Jang D. S., Kim G. Y., Choi H. I., Model-Based Tracking of Moving Object, Pattern Recognition, 30(6), 999-1008, 1997.

[27] Jenkin M., Tsotsos J., Applying Temporal Constraints to the Dynamic Stereo Problem, CVGIP(24), 16-32, 1986.

[28] Kanade T., Yoshida A., Oda K., Kano H., Tanaka M., A Stereo Matching for Video-rate Dense Depth Mapping and its New Applications, Proc. CVPR(96).

[29] Lowe D.G., Robust Model-Based Motion Tracking Through the Integration of Search and Estimation, IJCV 8(2), 113-122, 1992.

[30] Martin W.N., Aggarwal, J.K., Computer Analysis of Dynamic Scenes Containing Curvilinear Figures, Pattern Recognition 11(3), 169-178, 1979.

[31] McKenna,S.J., Raja,Y. and Gong,S. Object Tracking using Adaptive Colour Mixture Models, IEEE Asian Conference on Computer Vision, pp.615-622, 1998

[32] Moeslund, T.B. and Granum, E. A Survey of Computer Vision-Based Human Motion Capture, , Computer Vision and Image Understanding, 81(2), Academic Press, 2001

[33] Philip J., Estimation of Three-Dimension Motion of Rigid Objects from Noisy Observations, PAMI 13(1), 61-66, 1991.

[34] Radig B., Image Sequence Analysis Using Relational Structures, Pattern Recognition 17(1), 161-167, 1984.

[35] Rangarajan K., Shah M., Establishing motion correspondence, CVGIP: Image Understanding, 54:56-73, 1991.

[36] Reid D., An Algorithm for Tracking Multiple Targets, IEEE Trans. AC., 24, 843-854, 1979.

[37] Roy S., Cox I. J., A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem, Proc. International Conference on Computer Vision pp. 492-499, 1988.

[38] Salari V., Sethi I. K., Feature Point Correspondence in the Presence of Occlusion, IEEE Trans. Pattern Analysis and Machine Intelligence, 12:87-91, 1990.

[39] Sethi I. K., Jain R., Finding Trajectories of Feature Points in a Monocular Image Sequence, IEEE Trans. Pattern Analysis and Machine Intelligence, 9:56-73, 1987.

[40] Schiele B., Waibel A., Gaze Tracking Based on Face Colour, International Workshop on Face and Gesture Recognition, Zurich, 1995.

[41] Sidenbladh,H., Black,M.J. and Fleet,D.J., Stochastic Tracking of 3D Human Figures Using 2D Image Motion, European Conference on Computer Vision, pp. , Spinger-Verlag, 2000

[42] Song,Y. Goncalves,L., Bernardo,E.D. and Perona,P. Monocular Perception of Biological Motion in Johansson Displays, Computer Vision and Image Understanding, 81(2), Academic Press, 2001

[43] Soon K. J., Kwang Y. W., A Model-based 3D-Tracking of Rigid Objects from a Sequence of Multiple Perspective Views, Pattern Recognition Letters, 19, 499-512, 1998.

[44] Thompson W.B., Lechleider, P., Stuck, E.R., Detecting Moving Objects Using the Rigidity Constraint, IEEE Trans. Pattern Analysis and Mchine Intelligence, 15(2), 162-166, 1993

[45] Tsai R. Y., Huang T. S., Estimation of Three-Dimension Motion Parameters of a Rigid Planar Patch, IEEE Trans. Acoust. Speech Sign. Proc. 29(6), 1147-1152, 1981.

[46] Weng J., Ahuja N., Huang T. S., Motion and Structure from Points Correspondence with Error Estimation: Planar Surfaces, IEEE Trans. Sign. Proc., 39(12), 2691-2717, 1991.

[47] Wood D., Data Structure, Algorithms and Performance, Reading, MA: Addison-Wesley, 1993.

[48] Wren,C., Azarbayejani,A., Darrell,T. and Pentland,A. Pfinder: Real-Time Tracking of the Human Body, IEEE Conference on Face and Gesture Recognition, pp.51-56, 1996

[49] Yonemoto S., Arita D., Taniguchi R., Real-time Human Motion Analysis and IK-based Human Figure Control, Proc. IEEE Workshop on Human Motion, 149-154, Dec. 2000.

[50] Zhang Z., Faugeras O., 3D Dynamic Scene Analysis, Springer, Berlin, 1992.

[51] Zhang Z., Faugeras, O., Token Tracking in a Cluttered Scene, Image and Vision Computing, 12 (3), 110-120, 1994.

Appendix A: Complexity Analysis of Relaxed Multi-Layer Box-Ball Allocation Problem

As shown in section 2, there is no close-form solution about the complexity of generalised multi-layer box-ball allocation problem, and possible clipping rules for specific problems will definitely simplify the solution space, therefore the precise analysis varies from application to application. In this section we give some analysis to a simplified problem by some assumptions, and compare the complexity of our relaxation algorithm to the original problem without the use of prior, so that the performance improvement of our algorithm can be understood in the order of magnitude. The assumptions we make are as follows:

1. No Clutter: Every observation (ball) corresponds to an object (box).
2. No Occlusion: All objects (boxes) are observed.
3. Equal Confidence: All priors have equal confidence.
4. Independence: Allocation of observations (balls) in each layer is independent.

These assumptions, which describe an ideal tracking situation, imply that the number of boxes in each layer is less than boxes. The problem, followed the definition in section 2, comes out to be:

> Given M boxes with L layers, in each of which $N_{l,\ l=1,\ldots,L}$ ($N_l \le M$), balls will be allocated under the direction of P_l , $P_l \le N_l$, priors with confidence $\lambda, 0 \le \lambda \le 1.0$, what is the number of all validate allocation ways.

This number, which we denote as $\Gamma(M, L, N_1, P_1 \ldots, N_L, P_L, \lambda)$, gives the intrinsic complexity of the problem that describes the searching steps of the algorithm to find the *absolute-global-optimal* allocation way that minimise the cost function.

From assumption 4 we understand that the total complexity is the multiplication of the complexity in all layers. We then denote this as

$$\Gamma(M, L, N_1, P_1 \ldots, N_L, P_L, \lambda) = \prod_{l=1,\ldots,L} \Gamma_1(M, N_1, P_l, \lambda) \qquad \text{(A.1)}$$

Without the use of prior, we have the complexity of each layer is the permutation number of M and N_1, i.e.,

$$\Gamma_1(M, N_1) = \mathrm{P}_M^{N_l} = \frac{M!}{(M - N_l)!} \qquad \text{(A.2)}$$

If k, $k <= N_l$ priors are used, the complexity is reduced to

$$\Gamma_l(M, N_l, k)=\Gamma_l(M\text{-}k, N_l\text{-}k)=\frac{(M-k)!}{M!}\cdot\Gamma_l(M, N_l) \qquad \text{(A.3)}$$

For P_l priors with confidenceλ, we understand that at least $P^*_l= P_l\cdot\lambda$priors are correct. If gradually relaxing the priors to P^*_l and decomposing the problem tree without tabu to eliminate repetition, we obtain the complexity of single layer

$$\Gamma_l(M, N_l,P_l,\lambda)= \sum_{k=P_l*\lambda}^{P_l}\Gamma_l\,(M, N_l, k)\cdot C_{P_l}^{k} = \frac{\sum_{k=P_l\cdot\lambda}^{P_l} C_{P_l}^{k}\cdot(M-k)!}{M!}\cdot\Gamma_l(M, N_l) \qquad \text{(A.4)}$$

This yields the complexity reduction rate for single layer

$$\Psi_l(M, N_l,P_l,\lambda)= \frac{\sum_{k=P_l\cdot\lambda}^{P_l} C_{P_l}^{k}\cdot(M-k)!}{M!} \qquad \text{(A.5)}$$

and for total problem

$$\Psi(M, L, N_l,P_l\ldots,N_L,P_L,\lambda)=\prod_{l=1,\ldots,L}\frac{\sum_{k=P_l\cdot\lambda}^{P_l} C_{P_l}^{k}\cdot(M-k)!}{M!} \qquad \text{(A.6)}$$

With relative smaller P_l and (or) λ, (A.5) may even yield a rate larger than 1.0, which means the degradation of the performance, subject to the following condition

$$\sum_{k=P_l*\lambda}^{P_l} C_{P_l}^{k}\cdot(M\text{-}k)!)> M!$$

which is illustrated in Figure 9.

As our relaxation algorithm does in section 2.3, priors which are used in less relaxed decomposition step but not in current more relaxed step work as the tabu to eliminate redundancy, therefore all leaf nodes that have been visited previously will not be generated again. As a result, the total complexity is just the same as the sum of complexity when using P^*_l among all P_l priors for layer l. This time we have

$$\Gamma^*_l(M, N_l,P_l,\lambda)=\Gamma_l(M, N_l, P_l*\lambda)\cdot C_{P_l}^{P_l\cdot\lambda} = C_{P_l}^{P_l\cdot\lambda}\cdot\frac{(M-P_l\cdot\lambda)!}{M!}\cdot\Gamma_l(M, N_l) \qquad \text{(A.4')}$$

The complexity reduction for single layer changes to

$$\Psi^*_l(M, N_l,P_l,\lambda)= C_{P_l}^{P_l\cdot\lambda}\cdot\frac{(M-P_l\cdot\lambda)!}{M!} \qquad \text{(A.5')}$$

and for total problem

$$\Psi^*(M, L, N_l,P_l\ldots,N_L,P_L,\lambda)=\prod_{l=1,\ldots,L} C_{P_l}^{P_l\cdot\lambda}\cdot\frac{(M-P_l\cdot\lambda)!}{M!} \qquad \text{(A.6')}$$

which is roughly to the exponent of L, M, and P_l. As the simplest situation, i.e., P_l=N_l=M, (A.6') reduces to

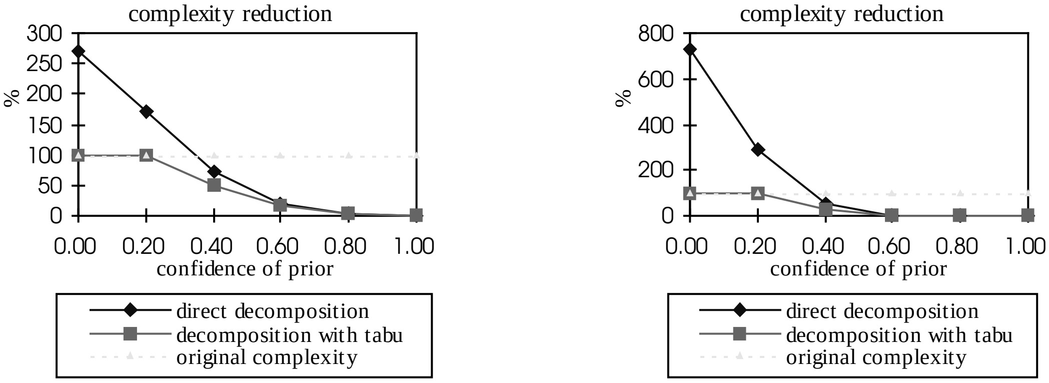
$$\Psi^{*}(M, L, M,M\ldots,M,M,\lambda)=\frac{1}{((M\cdot\lambda)!)^{L}}$$

Since $P_l \leq N_l \leq M$ and $0\leq\lambda\leq1.0$, it is easy to verify that from (A.5$'$) that

$$\Psi^{*}_{l}(M, N_1, P_l, \lambda)\leq1.0$$

while equal holds if and only if $\lambda=0$ or $P_l=0$.

Figure 9 illustrates the performance improvement of our algorithm for the case $M=N_l=P_l=5$, $L=1$ and $L=2$. As has been mentioned, practically the improvement can't be so large, as the solution space can be simplified by thresholding and knowledge based clipping. Anyway, our relaxation algorithm has made it more practical to resolve this complicated problem in real time.



$M=N_l=P_l=5$ and $L=1$ $\qquad$ $M=N_l=P_l=5$ and $L=2$

Fig. 9. Comparison of Complexity Reduction.

Appendix B: Algorithm Implementation

In this appendix we present details of the implementation of the relaxation algorithm presented in section 2 for the problem of 3D-tracking. The four data types are:

Box: a 3D point with a specific label, together with the prediction of current position.

Layer: a calibrated pin-hole camera system with a projection matrix of the intrinsic and external parameters.

Ball: a 2D-feature point in a specific layer (camera view) with a unique label, which represents a 3D ray.

Prior: a label table containing the label correspondences between the labels of boxes and all labels of balls in each view, as illustrated in Table 1. The raw number of the table is 1 (the box layer) plus the number of layers; the column number is the number of boxes.

The algorithm generates all possible arrangements of all balls in all boxes for all given a set of sub-set of the priors up to the maximum reliability in the prior and outputs the arrangements with the lowest cost. Implementation of the algorithm applies a standard tree-problem-solving (TPS) model for all sub-problems. TPS searches the nodes of a problem tree in the depth-first order, and sequentially outputs the branches from the root. The searching state is stored in a stack, with the standard last-in-first-out (LIFO) functionality. The TPS model can be described as the following C-style pseudo-code:

```
stack.push(root)
step_state=FORWARD
while (stack.not_empty()) do{
    if (branch_ok(stack)){
        if (step_state==FORWARD AND solution_ok(stack))
            output(stack)
            if (has_child(stack.top())){
                stack.push(next_child())
                step_state=FORWARD
            }
            else{
                stack.pop()
                step_state=BACK
            }
        }
        else{
            stack.pop()
            step_state=BACK
        }
}
```

Function *has_child* and *next_child* check and retrieve the next child for the current node at the top of the stack, in which heuristics can be embedded in to direct the search. *branch_ok* checks if the current branch is a valid one, in which knowledge-based clipping rules can be embedded. *solution_ok* checks if the current branch is a valid solution of the problem.

Two basic sub-problem solvers derived from TPS are: the permutation generator (PG), which generates the full set of permutation of M balls into M boxes; and the combination generator (CG), who selects M elements from all N (N>=M) elements.

For a previous layer some new boxes may have been generated. The problem tree for the current layer is the permutation of all possible arrangements of balls in this layer into all the original and new boxes. When a certain arrangement in the final layer has been generated, a possible solution is reached. Each solution is an arrangement table, with the boxes as the columns containing the labels of all balls allocated in this box. The algorithm then evaluates the cost of this solution as described in Section 2.3. When all possible solutions have been evaluated, we get the global optimal one. The implementation of the arrangement generator (AG) for a specific layer with M boxes and N balls is:

```
for (I=min(M,N) down_to 0) do{
    use CG to select I boxes from M boxes and for every I boxes{
        use CG to select I balls from N balls and for every M balls{
            use PG to generate every permutation of I balls into M boxes
            generate N-I new boxes and put the left N-I balls into without order
            output this arrangement
        }
    }
}
```

With a certain prior assigning P balls into P boxes for a specific layer with M boxes and N balls, and a tabu forbidding T balls to be put into T boxes (N>=P+T, M>=P+T), the problem is reduced to the arrangement of M-P boxes and N-P balls with T tabu rules. This gives the prior-based arrangement generator with tabu (PAG) from AG. The tabu rules are then applied in the *branch_ok* checking in the basic TPS model. If we know that at least R among all P assignments (P<=R) of the prior are correct with the reliability R/P, the prior-based relaxation arrangement generator (PRAG) is:

```
for (I=P down_to R) do{
    use CG to select I assignments from all P assignments and for very R assignments{
        use these R assignments as the prior
        use the left P-R assignments as the tabu
        use PAG to generate every arrangement
        output every arrangement
    }
}
```

The leaf-node-generator (LNG) for layer *l* (LNG$_l$) is a PRAG with M$_l$ boxes (subject to the current arrangement of all upper layers), N$_l$ balls, P$_l$ assignments as prior and λ_l as the confidence. The multi-layer box-ball problem is also solved by using the TPS model with the LNG as the child generator and the output of PRAG as the nodes. The clipping rules is applied in *branch_ok* vertically checking the arrangement table to see if a set of balls from different layers are allowed to be put into a specific box. After a solution is output, the reconstruction error and prediction error in each box are summed to measure the cost of this arrangement.

To ensure the reliability of the solution, we don’t apply any heuristic rules and do the full search in the multiview based 3D-tracking of skin objects for human motion capturing. Since the proposed prior-based relaxation algorithm dramatically reduced the inherent complexity real-time performance is still reached.

Appendix C: Camera Configuration

In this appendix the camera configuration for Figure 4, Figure 6(a) and Figure 8 are listed. There are three views involved, and the intrinsic parameters, the translation vector (measured in meter) and rotation matrix are listed respectively.

	Camera 1	Camera 2	Camera 3
$<f_x, f_y, c_x, c_y>$	341.9, 313.1, 151.5, 136.2	340.8, 311.8, 160.7, 125.8	343.9, 313.5, 149.7, 125.6
Translation Vector	0,0,0	0.9816, -0.0421, 0.1863	0.8752, 0.0528, 0.4809
Rotation Matrix	1,0,0 0,1,0 0,0,1	-0.0087, 0.9645, 0.2638 -0.1908, -0.2605, 0.9464 -0.5738, -0.7181, -0.0389	-0.0435, 0.9986, -0.0305 -0.4818, 0.0058, 0.8763 -0.9912, -0.0178, 0.1310