Argus: Visual Sensing for Large-Scale Tracking

Colin Low
Internet Systems and Storage Laboratory
HP Laboratories Bristol
HPL-2005-12(R.1)
February 6, 2006*

remote sensing, computer vision, tracking, scene analysis

An architecture and theory are provided for tracking many thousands of people with a large scale network of cameras. Cameras are not used to identify individuals; instead they generate a robust and characteristic visual signature, and probabilistic techniques derived from air and naval theatre tracking are used to construct maximum likelihood tracks. A novel feature of the system is that the topology of the camera network is deduced from very noisy data, and the system performs deductively without an explicit a priori representation of the camera network. Detailed simulation confirms the theoretical analysis and shows the algorithms can perform well. Applications are likely to include reconstructive analysis of criminal acts in public places, route analysis in commercial zones, and security in transport termini.

* Internal Accession Date Only
Approved for External Publication

Argus: Visual Sensing for Large-Scale Tracking

Colin Low

Hewlett Packard Laboratories,
Filton Road
Bristol, UK
colin_low@hp.com

Abstract

An architecture and theory are provided for tracking many thousands of people with a large scale network of cameras. Cameras are not used to identify individuals; instead they generate a robust and characteristic visual signature, and probabilistic techniques derived from air and naval theatre tracking are used to construct maximum likelihood tracks. A novel feature of the system is that the topology of the camera network is deduced from very noisy data, and the system performs deductively without an explicit *a priori* representation of the camera network. Detailed simulation confirms the theoretical analysis and shows the algorithms can perform well. Applications are likely to include reconstructive analysis of criminal acts in public places, route analysis in commercial zones, and security in transport termini.

Keywords: Remote Sensing, Computer Vision, Tracking, Scene Analysis

1. Introduction

This paper describes how a large-scale network of video cameras can be used to track the movements of pedestrians through a connected mesh of public spaces, such as might be found in a city centre, a mass transit system, or an air terminal. The ambition is to use hundreds of cameras to track thousands of people.

This kind of ambition causes feelings of unease, but the level of public surveillance in the UK has already reached a point where the average person can expect to appear on camera dozens of times each day. Video tapes from surveillance cameras are routinely used by police in the forensic analysis of serious crimes. What is missing is the means to carry out intelligent reconstruction, as the effort currently required to comb through potentially relevant video recordings is enormous.

The problem is to correlate observations of people across a large, spatially extended system of cameras. "Large" means the number of cameras could vary from hundreds to many thousands. "Spatially extended" means a city centre, or a mass transit system such as London Underground. There could be thousands of people within view of at least one camera. It is possible that the fields-of-view of some cameras will overlap, so that people could be tracked camera-to-camera using known techniques, but this is not assumed, and in general, fields-of-view will not overlap.

The scale of this problem is atypical for coordinated surveillance involving multiple cameras. A more typical scenario is where several cameras with overlapping fields-of-view are monitoring a limited area such as a car park [14]. The problem described more closely resembles multi-sensor tracking situations with military applications, such as ballistic missile tracking, air theatre tracking, and naval theatre tracking, in particular, submarine tracking. These applications are characterised by many targets, many sensor types, and the need to fuse data from multiple sensors into coherent observations, and so create accurate target tracks [18]. A key step in making large-scale camera tracking possible is not to think of a camera as an observing "eye", capable of high-level visual semantic processing, but as a sensor capable of originating simple observations in much the same way as radar or sonar.

2. Solution Outline

A track consists of a sequence of observations derived from the same person made by different cameras at different locations. One would expect the track to form a piecewise approximation to the actual physical path of a pedestrian within the spatial region monitored by the camera network.

If each person passing a camera could be identified using a unique visual signature, then the tracking problem becomes trivial. A batch of observations taken from many cameras would be partitioned according to signature. Each observation of a signature corresponds to an element of the track of a single person. When multiple observations of a signature are time-ordered, they constitute the track of a person.

This method is complicated by the fact that signatures will not be unique. This is outlined in section 8. In summary, current visual processing techniques are not adequate to provide a unique visual signature, and so it is necessary to live with the fact that the same person may produce a range of visual signatures according to how the person is presented to a camera, and multiple people may generate signatures that overlap, and cannot be disambiguated using purely visual means.

The first step is to generate a batch of observations. These are partitioned by clustering in signature space. This is described in more detail in Section 4. A subset of observations, comprising a cluster in signature space, is referred to as a *primary track*. A primary track consists of observations that could belong to a single individual, but will more likely consist of observations belonging to many individuals. These observations are disambiguated by using spatial and temporal information about the camera network in the form of a Transit Graph. The Transit Graph contains the likelihood that a person at location i will appear at location j after a time Δt. Bayesian techniques, and a probability maximisation algorithm, are used to disambiguate a primary track into a set of refined tracks. Refined tracks are a best-fit to the raw data, and constitute, with some level of confidence, a possible set of movements for a single person.

The Transit Graph is the key to disambiguating (or refining) primary tracks into refined tracks. A system of N cameras will have a Transit Graph with N^2 elements, and when $N = 100$ this requires *O(10000)* parameters to be acquired to a statistical level of significance. A unique and innovative feature of the approach described in this paper is that the Transit Graph is acquired in real time and is bootstrapped using observations in a process of mutual refinement - raw observations are used to construct a Transit Graph, and the Transit Graph is used to refine the raw observations, which in turn are used to construct a more accurate Transit Graph. After a number of iterations the bootstrapped Transit Graph is capable of refining observations to a high level of accuracy. A rationale for why this works is given is Section 6.

There are scaling limits to this bootstrapping approach. These are discussed in Section 9.

3. Solution Architecture

The functional components in the solution architecture are shown in Figure 1.

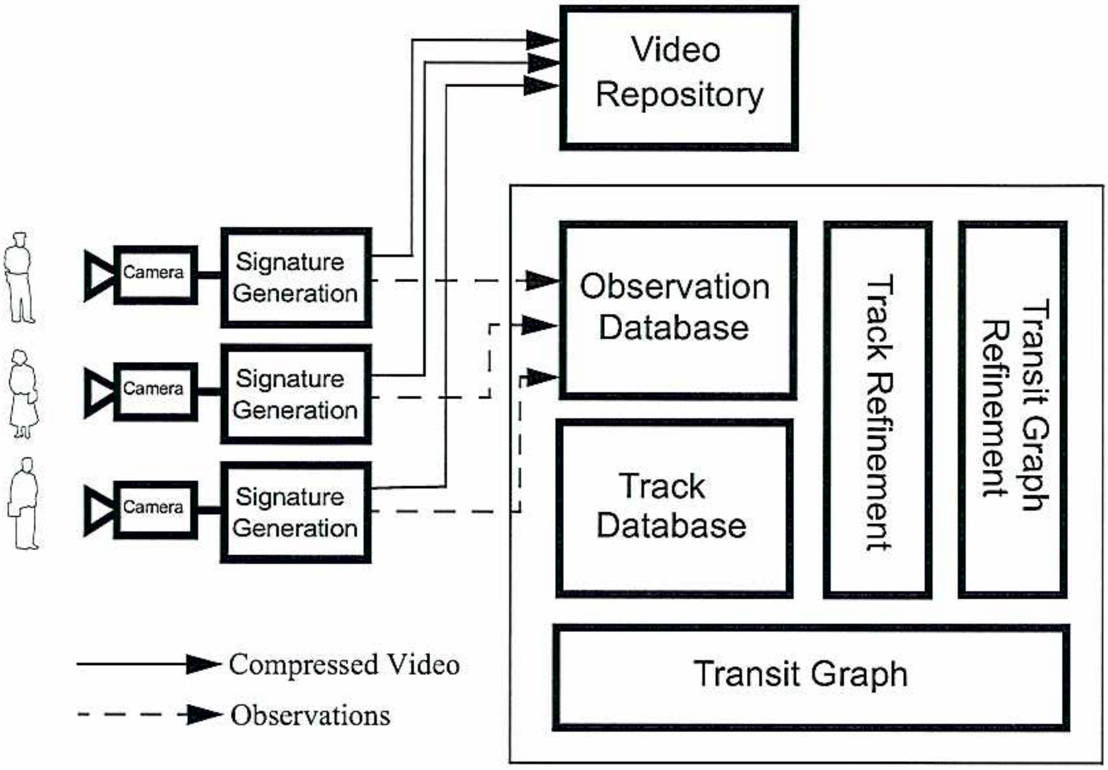

Figure 1: Solution Architecture

Cameras & Signature Generation.

Images from a distributed array of cameras are processed to extract relevant targets[1] (i.e. people) and each *target observation* is associated with a *target signature*. An observation is added to a central *Observation Database*. Each observation contains a target signature, a camera identifier, a location, a time/timecode, and a Video Repository reference.

In a practical implementation, observations are communicated to the observation database in the form of *observation event messages*, using some form of networking such as TCP/IP.

Portions of the raw video from each camera are compressed and forwarded to a *Video Repository*. The Video Repository reference and observation timecode provides a way to access the raw video (and possibly some processed video) associated with each observation.

Observation Database

All observations events generated by the distributed camera system are held in an Observation Database.

[1] An unfortunate term used in tracking literature.

Track Database

Observations are organised into tracks. Tracks can be *primary tracks* or *refined tracks*. Observations are clustered into primary tracks using signature information. Primary tracks are refined using information contained in the Transit Graph.

Transit Graph

Each camera is identified using a unique integer id. If there are *N* cameras the transit graph is an *N x N* matrix with each cell containing accumulated information about the likelihood that a target which appears at camera *i* will appear some time later at camera *j*. Although the Transit Graph can be initialised using off-line measurements, it can also be constructed in real time using observations, as described in more detail later in this document.

Track & Transit Graph Refinement

Primary tracks are refined using bootstrapped or *a priori* information about movements between cameras. The Transit Graph is refined using the observations in refined tracks. The iterative bootstrapping method used to generate the Transit Graph is one of the features of the algorithm described later in this document.

Video Repository

The Video Repository contains raw video from each camera indexed by camera and timecode, and the portion corresponding to an observation event is capable of being accessed remotely using a reference.

4. Signature Classification

The tracking system described in Section 5 requires visual signatures capable of classification. This means that when the signatures generated by the same individual are plotted in an appropriate high-dimensional space, then all the signatures should be 'close' according to some measure. It would be impossible to carry out meaningful classification if signatures generated by appearances that were perceptually similar appeared in very 'distant' parts of classification space.

It is assumed that a visual signature is a finite bitstring of some kind. It is also assumed that this bitstring can be mapped into a high-dimensional metric space with the property that visually similar appearances map into signatures that are close in metric space. Signatures can then be grouped into clusters using well-known techniques. A considerable literature on clustering/classification is reviewed in [7].

The signature clustering algorithm is not described, as it depends on the specific method of signature generation. For the remainder of this paper it is assumed that signatures can be generated and classified. There are many technical difficulties involved in achieving this and these are discussed later in Section 8.

5. Tracking

A track consists of an ordered sequence of observations related to a single target. Given a set of observations involving several targets, there are many possible ways to sequence the observations into tracks. The number of ways a set of *N* observations can be partitioned into tracks is the same as the number of partitions of a set of size *N* into non-empty subsets, and is given by *Bell(N)*, where *Bell(N)* is the Bell number for integer *N*. This is the sequence (1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597, ...) for N = (1, 2, 3 ...).

The 52 partitions of a set of 5 elements are shown in Figure 2 below[1]. The visual representation employed in the diagram is highly appropriate and shows how a set of observations can be interpreted as a set of tracks. Note that tracks must begin and end, so a single point in a set above would correspond to a track with a single observation event.

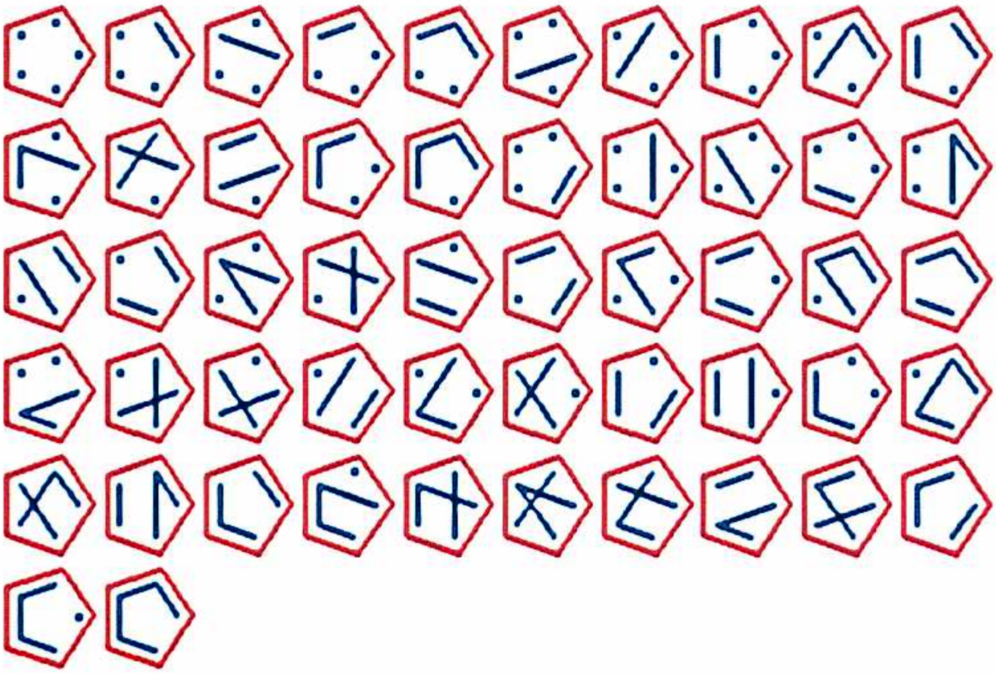

Figure 2:Partitions of a set of cardinality 5

One way to group and order observations into individual tracks is to enumerate all possible partitions of a set of observations, interpret each partition as a track hypothesis, and grade each set of tracks using *a priori* probabilities to determine the likelihood of each hypothesis. The partition with the greatest likelihood would then be used as the best track hypothesis for that set of observations. Unfortunately the combinatorics work against this approach - after as few as 12 observations there are 4,213,597 competing hypotheses to consider. This approach ("track splitting" or "multiple hypothesis tracking" is one of the standard approaches described in the literature [18][1], but it is not well suited to situations where large numbers of observations are involved. Simulation of the problem (see section 7.) routinely deals with set sizes of several hundred observations, and even when low probability hypotheses are culled, the combinatorial explosion is too severe for a practical implementation using this technique.

[1] Robert M. Dickau, http://mathforum.org/advanced/robertd/bell.html, used with permission.

The approach used in this document is influenced by tracking techniques developed for air traffic control, missile tracking and submarine tracking, with a clear bias towards military applications using radar and sonar [18][1]. The data association techniques used have substantial generality, and the specific approach employed uses the assignment problem formulation described by Poore [15], and explicitly adapted to visual tracking by Kettnaker and Zabih [9][8].

The following presentation is based on that given by Poore, which should be consulted for background.

Given a set Z of N observations, let Γ^* be the collection of all partitions of Z and let Γ be a discrete random element defined on Γ^*. Then a partition $\gamma \in \Gamma^*$ represents a specific hypothesis about how the observations can be grouped into tracks, and the problem of organising the set of observations into tracks can be stated as

$$\text{maximise}\{P((\Gamma = \gamma) \mid Z) \mid \gamma \in \Gamma^*\}$$

That is, given the set of observations Z, find the track hypothesis with the greatest probability. This formula can be expanded using Bayes Theorem to give

$$\text{maximise}\left\{\frac{P(Z \mid (\Gamma = \gamma)) \cdot P(\Gamma = \gamma)}{P(Z)} \mid \gamma \in \Gamma^*\right\}$$

where the first term in the numerator is the *likelihood* of the observations given a specific track hypothesis, and the second term in the numerator is the *prior*, the *a priori* probability for the specific track hypothesis being true. The term in the denominator is constant across all track hypotheses and can be ignored in the maximisation procedure.

It is normal to continue the analysis by decomposing the likelihood in terms of the specifics of the actual problem, and both Poore [15] and Kettnaker [8] provide examples of this. Kettnaker provides a highly elaborate Bayesian analysis down to the frame level of individual cameras, but the simplifying assumptions used means that the likelihood information factors out, and only the prior terms remain. The key architectural separation made in this document between signature generation and tracking exposes the intuition behind this result more clearly.

Suppose that we have a set of observations that are consistent with many possible track hypotheses. The likelihood term encodes the fact that while all hypotheses are *possible*, some are more *likely* than others. If one were to exhibit a specific track hypothesis, on could say whether the observations were more or less likely given that hypothesis.

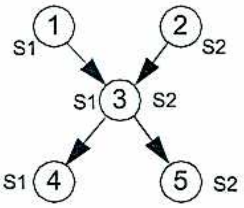

Figure 3: Signature confusion & Likelihood

Consider the situation in Figure 3. Assume signature *S1* is very different from signature *S2*, and the probability that *S1* could be confused with *S2* very small. Then all other things being equal, one would prefer to believe that the signatures were generated by the track hypotheses (i.e. underlying object movements) 1,3,4 and 2,3,5 rather than 1,3,5 and 2,3,4. If S1 was similar to S2 (signature confusion) then both hypotheses would be viable, and we would have to depend on additional (e.g. prior) information to choose one hypothesis over the other.

Let t_i be a track in the partition $\gamma \in \Gamma^*$, and let t_i consist of the observations $o_1, o_2 .. o_n$ which in turn contain the signature values $s_1, s_2, .. s_n$. Then the contribution of t_i to the overall likelihood of the partition $\gamma \in \Gamma^*$ is the probability that $s_1, s_2, .. s_n$ are generated by the successive appearances of the same individual. If $s_1, s_2, .. s_n$ are identical or similar, we would be inclined to assign a high likelihood to t_i, but if $s_1, s_2, .. s_n$ are very different then we would assign a low likelihood.

It is unclear that identifying a quantitative measure for the likelihood of $s_1, s_2, .. s_n$ being the successive appearances of same individual is a meaningful procedure. The difficulty is that the system in question, a large-scale visual tracking system for use in public spaces, has a non-stationary population of observed subjects. The visual appearance of subjects may be moderately constant over short periods of time (e.g. hours), but certainly not days. Each collection of observations/signatures needs to be considered as unique, and this is consistent with the underlying situation, a changing population of individuals with constantly changing appearances.

An alternative procedure is to use a signature classifier to partition a set of observations Z into subsets. Any signature that is “sufficiently like” another signature is collected into the same subset. Track hypotheses are only considered within each subset. In this way the collection of partitions considered in the maximisation procedure is drastically culled; it is equivalent to assigning many partitions a likelihood of zero, and the remainder a likelihood of one.

Let $Z = Z_1 \cup Z_2 .. \cup Z_N$ where $Z_1..Z_N$ are non-empty subsets of Z generated by a signature classifier such that $Z_i \cap Z_j = \varnothing$ for all $i, j,\ i \neq j$. Let Γ^i be the collection of all partitions of Z_i and let

$$\Gamma^+ = \Gamma^1 \cup \Gamma^2 .. \cup \Gamma^N$$

Then $\Gamma^+ \subset \Gamma^*$ and the likelihood can be defined as

$$P(Z \mid (\Gamma = \gamma)) = 1 \text{ if } (\gamma \in \Gamma^+)$$

$$P(Z \mid (\Gamma = \gamma)) = 0 \text{ if } (\gamma \in \Gamma^* - \Gamma^+)$$

An advantage of this approach is that the design of a suitable signature classifier is orthogonal to the design of the tracking system. There is also a considerable computational advantage. A set of 12 observations has 4213597 partitions as potential hypothesis, but two sets of 6 observations have 406 partitions. The advantage in pre-partitioning the set Z increases as the size of the set increases.

This is the approach used in this work. Rather than try to estimate a function that yields the likelihood that a collection of signatures comes from the same observed subject, a function that will almost certainly vary for every subject, the assumption is made that signatures can be projected into a high dimensional signature space, and that the camera/signature generation system is well-behaved in the sense that successive appearances of the same subject produces signatures that occupy a "well-defined region" in this space. It is also assumed that it is possible to devise a classifier which can divide the signature space into regions such that each region contains signatures corresponding to all the observations of a single subject. It is not the case that each region contains signatures for *only* one subject - if that was possible, each subject would be uniquely identified, and the tracking algorithm described below would be redundant.

These are strong assumptions, perhaps unrealistic, as they imply that the output from each camera can be normalised with respect to positioning, scaling, lighting, environment, internal settings and manufacturing tolerances. However, in a network consisting of hundreds or thousands of cameras, this is an important architectural separation - cameras do camera things, and tracking does tracking. We do not know in advance what people will look like, what they will wear, how they will move. The best we can do is demand that the camera subsystem produce signatures that are localised in a metric space such that it is possible for a classifier to find well-defined clusters. This assumption is revisited in more detail in Section 8.

The prior probability $P(\Gamma = \gamma)$ can be expressed using the following definitions. Let

$P_N(x,t)$: the prior probability density that a new object appears (becomes visible) in location x at time t.

$P_D(x,t)$: the prior probability density that an object disappears at location x and time t.

$P_{Select}(x, x')$: prior probability that an object at location x will select location x' as a next destination.

$P_{Delay}(x, t, x', \Delta t)$: prior probability density that an object moving at time t from x to x' will reappear after a delay of Δt.

$P_M(x, t, x', t')$: the prior probability density that an object last visible at location x and time t moves to location x' and becomes visible at time t' after an interval $t' - t = \Delta t$

The situation is shown in Figure 4. The hypothesis is that a new observed subject appears in location 1, moves through locations 2, 3 and 4, and disappears at location 4. The appearance of a new subject (from some notional location N "offstage") marks the initiation of a track, and the disappearance of the subject (to some notional location D "offstage") marks the termination of the track. An alternative hypothesis - 4 singleton tracks - consistent with the same observations is shown in Figure 5..

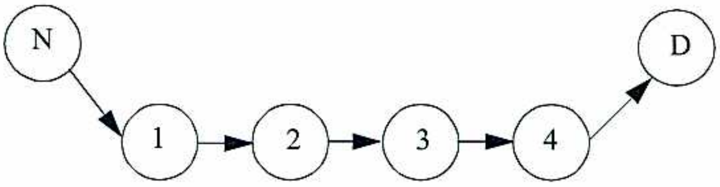

Figure 4: Track Hypothesis

Thus we have that$P_M(x, t, x', t') = P_{Select}(x, x') \cdot P_{Delay}(x, t, x', t' - t)$

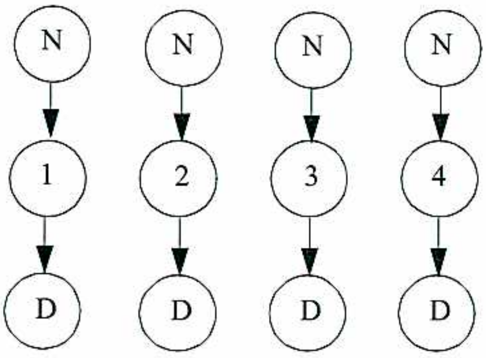

Figure 5: Alternative Track Hypothesis - Singleton Tracks

Note that the priors P_{Select} and P_{Delay} can be continuously estimated from high-confidence tracks using standard estimation techniques [1].

For any hypothesised track consisting of K consecutive locations in the location graph, we can order the locations using integer numbering from $1 .. K$, where an object appears in location 1 and disappears in location K.

The probability of a given track hypothesis τ is

$$P(\tau) = P_N(1, t_1)dt_1 \cdot \left(\prod_{x = 1 .. K-1} P_{Select}(x, x+1) \cdot P_{Delay}(x, t_x, x+1, \Delta t_x)dt_{x+1} \right) \cdot P_D(N, t_K)dt_N$$

The probability of a partition is the product of the probabilities of each track in the partition:

$$\begin{aligned} P(\Gamma = \gamma) &= \prod_{\tau} P(\tau) \mid \tau \in \gamma \\ &= \prod_{\tau} P_D(\tau) \cdot dt_1 \cdot dt_2 \cdot dt_3 .. dt_N \mid \tau \in \gamma \end{aligned}$$

where$P_D(\tau)$ is a probability density

$$P_D(\tau) = P_N(1, t_1) \cdot \left(\prod_{x = 1 .. N-1} P_{Select}(x, x+1) \cdot P_{Delay}(x, t_x, x+1, \Delta t_x) \right) \cdot P_D(N, t_N)$$

It is possible to factor out the differentials dt_i because for any partition, each observation appears exactly once in exactly one track, and so the product will always contain $dt_1 dt_2 .. dt_N$ exactly once. Absolute probabilities are not required for the maximisation procedure, and we are at liberty to divide by an arbitrary scaling factor $G \cdot dt_1 \cdot dt_2 \cdot dt_3 .. dt_N$, so that the *weight* $W(\gamma)$ assigned to a partition is

$$W(\gamma) = \frac{1}{G} \cdot \prod_{\tau} P_D(\tau) \mid \tau \in \gamma$$

Given the preceding discussion, the solution to the tracking problem can now be given as

$$\text{maximise}\{P(\Gamma = \gamma) \mid \gamma \in \Gamma^{+}\}$$

$$\equiv \text{maximise}\{W(\gamma) \mid \gamma \in \Gamma^{+}\}$$

$$\equiv \text{maximise}\left\{\sum_{\tau} \log P_D(\tau) \mid \tau \in \gamma, \gamma \in \Gamma^{+}\right\}$$

The final step can be justified by the fact that the partition for which the product is maximised is the same partition that maximises the sum over all tracks of the logarithm of $P_D(\tau)$. Reframing the problem in terms of maximising a sum of terms is convenient because there are powerful algorithms for solving problems of this kind.

The maximisation procedure over all partitions and tracks is the heart of the tracking algorithm. Poore [15] shows how data association problems of this type can be solved as multi-dimensional assignment problems, and the solution adopted here is inspired by Kettnaker & Zabih [9], who approach the problem as a two-dimensional assignment using a solution to the weighted matching (assignment) problem in bipartite graphs.

A bipartite graph is a graph which can be partitioned into two sets X and Y of vertices such that there are no edges between vertices within each set. A matching is a subset of the edges between X and Y such that each vertex is attached to only one edge. A perfect matching between two sets of equal size is a matching that connects every vertex in X to a vertex in Y and forms a one-to-one and onto mapping between the two sets. If each edge in the graph has an associated weight, then a maximum weighted matching is a perfect matching that maximises the edge weights.

An advantage of formulating the tracking problem in terms of weighted matching on a bipartite graph is that a solution can be found in polynomial time: if the cardinality of X is $|X|$, then solutions exist that execute in $O(|X|^3)$ time. This is a considerable improvement on Bell($|X|$).

Let Z be a sequence of observations $z_1, z_2, z_3, .. z_N$ sorted in time-ascending order. Let $X = Y = Z$, so that the bipartite graph is a mapping from Z to itself. A vertex $z_i \in X$ can only have an edge connecting to a vertex $z_k \in Y$ if $i < k$, because observations are temporally ordered - that is, early observations must precede later observations. Let the weight associated with edge e_{ik} be

$$W(e_{ik}) = P_{Select}(i, k) \cdot P_{Delay}(i, t_i, k, \Delta t_i)$$

A complete matching of maximum weight on this graph will result in every $z_i \in X$ being linked to a $z_k \in Y$ such that $i < k$ (temporal order preserved) in such a way that the overall likelihood is maximised - this is exactly what is needed to find the partition with the greatest likelihood.

Unfortunately this is insufficient. The sequence of observations Z contains the interleaved tracks of several observed subjects, and each track must be initiated at some point in the sequence, and must terminate at some point in the sequence. The scheme just described provides no way for tracks to initiate or terminate; indeed, the condition $i < k$ guarantees that no perfect matching is even possible, as there is always an observation z_1 which precedes all and cannot be matched, and an observation z_N which has no observation following it, and it also cannot be matched.

To represent track beginnings and ends it is necessary to add another N vertices $\{N_i\}$ to X to represent track initiation ("new"), and another N vertices $\{D_i\}$ to Y to represent track termination ("disappeared"). The new vertices follow the z_i in the bipartite graph as shown in Figure 6 .

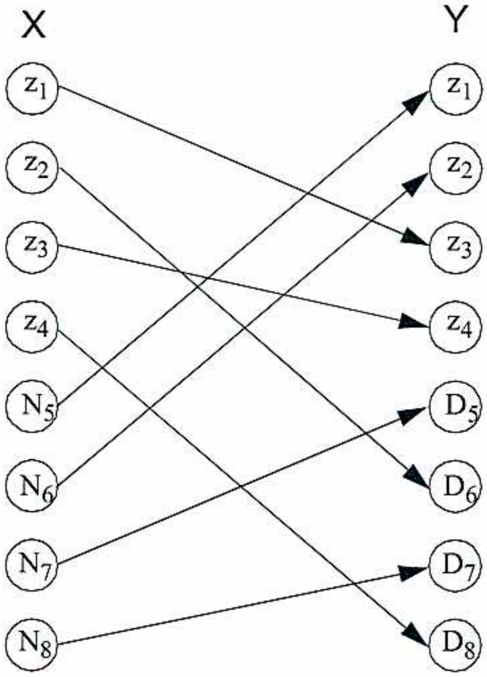

Figure 6: Bipartite Graph perfect matching

Figure 6 represents the following situation:

- a track *(N_5, z_1, z_3, z_4, D_8)* that is initiated in location 1, transits locations 3 and 4, then terminates.
- a singleton track *(N_6, z_2, D_6)* that is initiated in location 2, and terminates in location 2.

The edges N_{i+N} -> z_i represent tracks which begin in location *i*, and the edges z_i -> D_{i+N} represent tracks which terminate in location *i*. The edges N_i ->D_k represent nothing. A complete matching is always possible because:

- the start of a track leaves an unmatched D_i in Y.
- the end of a track leaves an unmatched N_k in X.

- there is exactly one unmatched N_k for each unmatched D_i.

Edges between all N_k and D_i are given low (negative) weights, so that they are never considered in favour of a legitimate (positively weighted) edge, and are only used to complete the maximum weight matching. The edge from N_{i+N} to z_i is given a weight proportionate to $P_N(i, t_i)$, the likelihood that a new subject will appear at location i and time t_i. The edge from z_i to D_{i+N} is given a weight proportionate to $P_D(i, t_i)$, the likelihood that an observed subject will disappear from location i at time t_i.

The fact that this method can find a maximum weighted partition of N observations in $O(N^3)$ steps, and can represent explicitly all the relevant *a priori* likelihoods, including the initiation and termination of tracks, is remarkable given *Bell(N)*, the number of partitions of N observations. The method used differs from that described in Kettnaker [8] in that track termination is represented explicitly, using a location-dependent likelihood, and so the assignment problem formulation is similar in spirit but different in detail.

In the implementation of this method a maximum weighted matching was found using the Kuhn-Munkres algorithm [11][13]. In this algorithm the vertices of a bipartite graph are given a feasible labelling, an equality subgraph is found, and the equality subgraph is extended using the augmenting path method until it forms a complete matching.

6. Transit Graph Refinement

If a system contains V cameras, then movements between cameras can be characterised by a Transit Graph with V vertices and at most V^2 edges. This in turn can be represented by a $V \times V$ matrix, where each entry represents the *a priori* knowledge about the relative frequency and duration of transitions from camera m to camera n, with $1 \leq m, n \leq V$.

The Transit Graph can be created using an off-line method. For example, a small group of individuals with a distinctive appearance and an independent method of recording location (cellphone, GPS recorder etc.) could be used to prime the Transit Graph with transit data. The difficulty with this method is that it is not responsive to changing traffic flows and transition durations. It would be more satisfying to have a method which did not depend on off-line knowledge, and which updated the Transit Graph continuously using observed traffic.

The difficulty is that in order to produce a Transit Graph, accurate track data is required. In order to refine accurate track data from observations we need the *a priori* likelihood information contained in the Transition Graph. Starting with an uninitialised Transition Graph, and some observation events, it is not obvious that anything can be deduced, because there are many possible track hypotheses, and there is no *a priori* knowledge that would prefer one hypothesis over another. One of the contributions of the work described in this paper is an iterative refinement algorithm that enables a Transit Graph to be initialised and deduced purely from observation events, while simultaneously refining observations into tracks.

The reason that such an algorithm should exist can be deduced intuitively by considering how observation events are used to update the Transit Graph. This is presented as an informal argument.

Consider a network with V vertices and assume that the Transit Graph is regular, and that each vertex has valency C. The Transit Graph, considered as a matrix, will have V^2 entries, but only $E= V*C$ of these entries will represent genuine edges in the transit graph, the remainder (ideally) being null. Assume also a population of N signature-identical individuals where each individual causes observation events at a rate of m events/sec, and that the traffic flow is uniformly distributed among all E edges (ignoring any global flow problems that might arise).

Consider an event *e(i)* that occurs when an individual is observed at location *i*. If this event is followed in the event stream by an event *e(j)* we might deduce that there is an edge between locations *i* and *j*, and update the Transit Graph entry *TG(i,j)* with information about this transition. In most cases this deduction will be incorrect, because the event *e(j)* could be caused by any individual being observed at *j*, and not just the individual previously observed at *i* - this follows from the fact that individuals present identical signatures and cannot be distinguished from information in the observation event. In this case the Transit Graph will contain some spurious information. On the other hand, if there is a genuine path from *i* to *j*, and the transition at *j* is caused by the same individual that caused the transition at *i*, then correct information will have been added to the Transit Graph.

It is useful to compare the quantity of correct information added to the Transit Graph compared to the quantity of incorrect information. Although it is not entirely justifiable, one can assume that *en-masse*, observation events are not correlated and so one can can use the Poisson distribution to calculate probabilities.When an event *e(i)* occurs, on average it will take *1/m* seconds for the individual at *i* to cause another event *e(j)* at location *j*. During that time, $(N \cdot m)/m$ events could occur at any of the other vertices. The probability *P(n = 0)* that no other event occurs, and the event *e(j)* will correctly follow *e(i)* is

$$P(n = 0) = e^{-N}$$

and the probability that some other unconnected event *e(k)* $1 \le k \le V$ occurs is

$$P(n > 0) = 1 - e^{-N}$$

During some time interval a total of M events occur, and each time this happens an entry in the Transit Graph is incremented. A total of

$$M \cdot 1 - e^{-N}$$

events will be incorrectly assigned to any one of the V^2 entries in the transition graph matrix, so that each entry will receive

$$\frac{M \cdot 1 - e^{-N}}{V^2}$$

counts. A total of

$$M \cdot e^{-N}$$

events will be correctly assigned to one of the E entries in the Transit Graph matrix corresponding to an edge, so that each edge entry will receive a total of

$$\frac{M \cdot e^{-N}}{E} + \frac{M \cdot 1 - e^{-N}}{V^2}$$

counts (which includes its share of the incorrect counts).

A measure of how successfully the Transit Graph is being updated is the ratio of incorrect updates to correct updates for each real edge in the graph. Ideally this ratio would be << 1. If this ratio is >> 1 then it would be difficult to bootstrap the Transit Graph using nothing more than primary observation events.

The ratio of incorrect to correct counts in an edge entry is

$$\frac{E \cdot (1 - e^{-N})}{V^2 \cdot e^{-N}}$$

For a fixed value of N, this ratio will be < 1 for sufficiently large graphs of small valence. For example, if there are 100 vertices, and the valence is 4, then $\frac{E}{V^2}$ is 0.04. Set against this is the exponential sensitivity on the number of signature-identical individuals in the network.

A measure of N can be found by ignoring the exponential term in the numerator and setting

$$\frac{E}{V^2 \cdot e^{-N}} = 1$$

$$N = \ln \frac{V^2}{E}$$

If $V = 100$ and each vertex has 4 edges, then $N = 3.2$. If V = 1000 then $N = 5.5$. This extreme sensitivity to the number of identical individuals means the goal of bootstrapping a transit graph in a single system that observes a population of many tens of thousands is likely to be unachievable - even with a signature uniqueness of 99%, there would be too many ambiguous observation events. It should be noted that four heuristics are described later that greatly improve on this worst-case estimate - in particular, updating the transit graph using a subset of the observed population by ranking signature clusters in terms of quality.

The technique used to bootstrap the Transit Graph is to weight each entry in the Transit Graph matrix *TG(i,j)* according to the number of times an observation event *e(j)* follows an event *e(i)*, where each event falls into the same signature cluster. These weights are used for refining tracks using the weighted assignment algorithm. Even when the quality of the data in the matrix is poor, the weighted assignment algorithm is a powerful method for extracting whatever information is there, and the refined tracks are then used to rebuild the Transit Graph. This procedure is repeated for many iterations. The quality of the Transit Graph information improves with each refinement, which in turn leads to better refined tracks, which in turn yields a more accurate Transit Graph, as illustrated in Figure 7.

A useful measure of refinement progress is the track purity. Because signatures are not unique, any track one assembles may contain observations from several individuals sequenced together. Although the assembled track may not be an accurate representation of the movements of any one individual, it may contain subsequences of observations which are subsequences from ideal

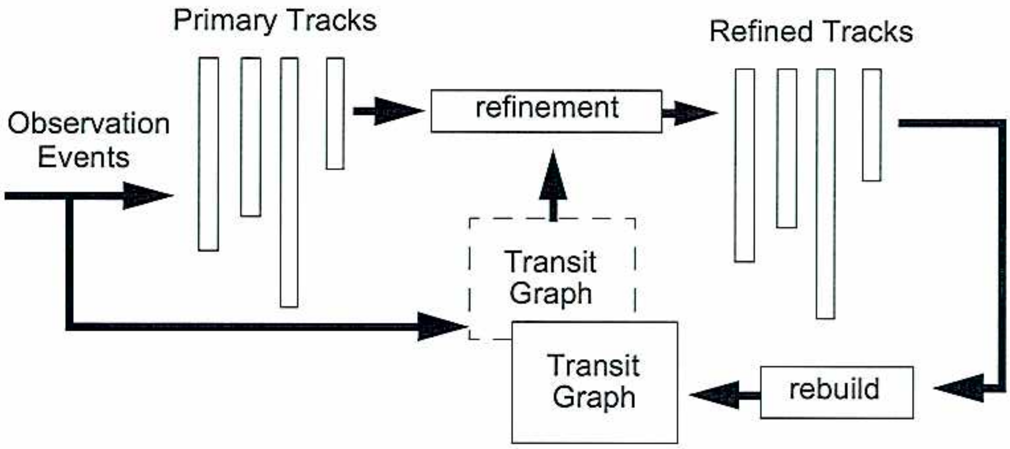

Figure 7: Refinement process

correct tracks. The more correct subsequences the track contains, the more "pure" it is. The Track Purity is the percentage of observations in a track which correctly follow a preceding observation. A track which is 100% pure is a subsequence of a correct track. A track which is 0% pure contains no subsequences from a correct track.

The result of applying the refinement procedure displayed in Figure 7 to a sequence of observations generated by a city simulation (Section 7.) is shown below in Figure 8 . The iteration process is repeated 30 times for a population of 1000 individuals with a signature uniqueness of 98% and 99%. When the signature uniqueness is 98%, the primary/unrefined track purity is about 6%, and the refined track purity begins at 36% and rises to 88%. When the signature uniqueness is 99%, the primary track purity is a higher 22%, and the refined track purity begins at 67% and rises to 95%.

This simulation demonstrates the extremely sensitive dependence on signature uniqueness, and also the relative ease with which it is possible to deduce a useful Transit Graph capable of refining tracks to high levels of purity from nothing more than very noisy input observations. The refinement process operates with no knowledge whatsoever of the camera graph topology.

7. Implementation

An implementation of a tracking system was built that conforms closely to the solution architecture described in Section 3. A key aspect was the architectural separation between the camera/signature generation sub-system that generates observation event messages, and the subsystem that organises the observations into tracks, constructs a Transit Graph, and uses the Transit Graph to process primary tracks into refined tracks.

A key motivation behind the implementation was to verify that it is feasible to construct a system capable of processing the outputs from hundreds of cameras observing thousands of people, and discover what level of success one might expect. A second motivation was to test the idea that it is possible to refine tracks by bootstrapping a Transit Graph without knowing anything about camera locations, transit times, and preferred routes - that is, to deduce the knowledge

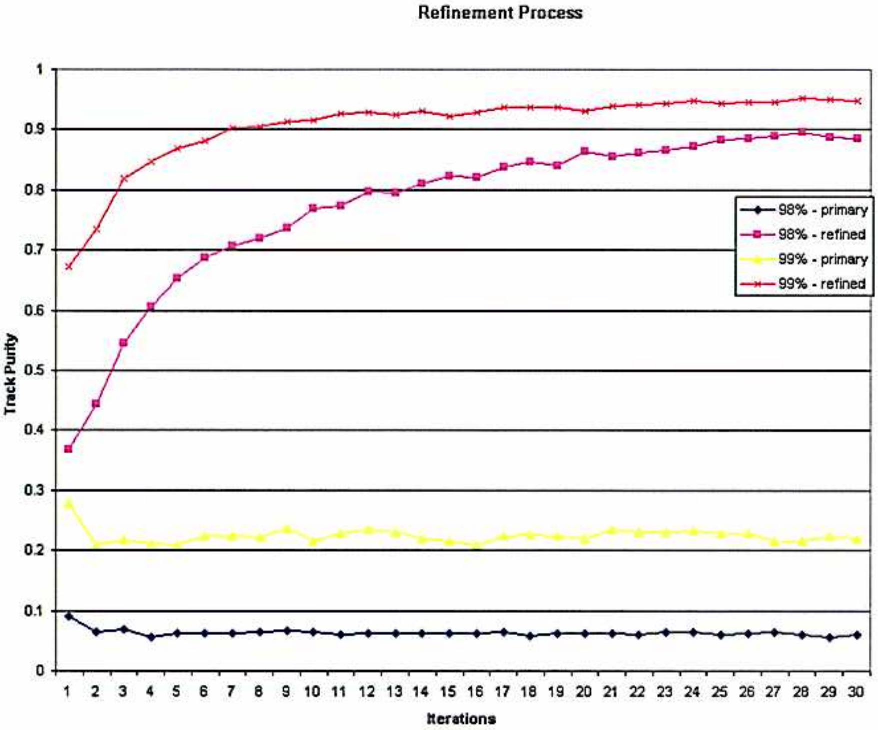

Figure 8: Iterative refinement. The Transit graph is refined for 30 iterations, to yield a track purity of 95%.

needed to refine tracks. A third motivation was to search for heuristics that might improve the tracking process.

The camera/signature generation system was provided using a discrete-event (Monte Carlo) simulation of a population of observed individuals making journeys through a complex network of places. Three types of places were provided: domicile places, transit places and public places.

Each simulated individual would begin in a domicile place, select a random public place as a destination, and negotiate a mesh of transit places to the public place, then make a return journey to the same domicile place. This was repeated for the duration of the simulation. A percentage of the transit places would be designated observation points, and would generate an observation event message each time an individual entered that place. A typical simulation run would consist of 500 - 1000 individuals, 100 - 400 transit places, 1000 - 2000 domicile places and 10 - 50 public places

Typically each home place would be connected to one transit place, each transit place would be randomly connected to 4-5 other transit places, and each public place would be connected to 2 transit places. The mesh of places would be generated randomly for each simulation run. Larger populations could be simulated, but the exponential dependency on the numbers of individuals in transit sharing the same signature classification (see Section 7) meant that an accurate Transit Graph could not be computed for larger simulations. Simulation confirms the approximate reasoning in Section 7, that there is a scale of a 100-1000 individuals and 100-200 transit nodes for which a Transit Graph capable of refining tracks can be computed from raw data. The only situations simulated were ones where the Transit Graph was empirically deduced from the observational data; there was no method for initialising the tracking system with externally generated information about the mesh of transit places.

The only significant simplification made in the population simulation was that individuals were pre-sorted into signature classes. If the signature uniqueness was set to 98%, then each individual would be randomly allocated to one of 50 signature classes, and all observations of that individual would be tagged with that signature. Each observation event message also contained the unique identity of each individual, so that the tracking subsystem could maintain a private set of "true tracks" which were used to assess the purity of refined tracks by computing the refined track purity with respect to the true track.

The tracking system uses knowledge about the average behaviour of a population to discriminate between what is a probable track and what is not. The journey time between two locations is particularly important; if the journey time between two locations is well defined, then transit time becomes a useful measure in deciding between track hypotheses. The more variation there is in the transit times, the less useful transit time becomes as a discriminator, an instance of the general rule that the noisier the data is, the harder it is to extract anything useful from it.

Each transit place in the mesh of places has a mean residency time defined, and the residency time for each individual is selected at each simulation step from a Gaussian distribution using the mean for the place, and a standard deviation that is a fixed proportion of the mean. Simulation confirms what is intuitively obvious: the larger the standard deviation, the less successful tracking is. Figure 9 shows the refined track purity plotted against the number of refinement interactions for the standard deviation equal to 0.2, 0.4, 0.6 and 0.8 of the mean transit time. The population is 1000 individuals, signature uniqueness is 98%, and there are 100 transit nodes. The refined track purity asymptotes to 95-96% when the standard deviation is 0.2 of the mean, and asymptotes to 80-83% when the standard deviation is 0.8 of the mean.

Transit times between locations are estimated from refined tracks. If location *j* follows location *i* in a refined track, then the time difference of the observation events is an estimate of the transit time. The times between any two locations in the Transit Graph are stored in the form of a rolling mean of size *K*; that is, the last *K* transit times are stored in the form of a first-in-last-out queue and used to compute the mean and standard deviation. In a non-steady state, the rolling mean is a method used to estimate statistical parameters, but also to slowly forget old data. For most of the simulations $K = 50$. The transition probabilities between locations are progressively estimated in a similar manner.

The tracking system was written in C++ for speed. The core of the tracker is an implementation of the Kuhn-Munkres bipartite graph weighted assignment algorithm written by the author. Because of the $O(N^3)$ behaviour of this algorithm, primary tracks were batch refined when the average primary track length consisted of about 50-70 observations. When the signature uniqueness was 98%, there would be 50 primary tracks, making about 2500-3500 observations in each batch. These would be refined and a 10,000 vertex Transit Graph rebuilt in about 4 seconds of real time on an 800 MHz Pentium running Windows 2000. From this one can deduce that even in its current prototype state an observation event rate of 900 events/second can be processed by the tracker.

The system is compute-bound and highly parallelisable. The speed is dominated by the weighted assignment algorithm, and each primary track can be refined in parallel, so a linear speed-up proportional to the number of primary tracks is trivial to achieve. In the example above, a speed

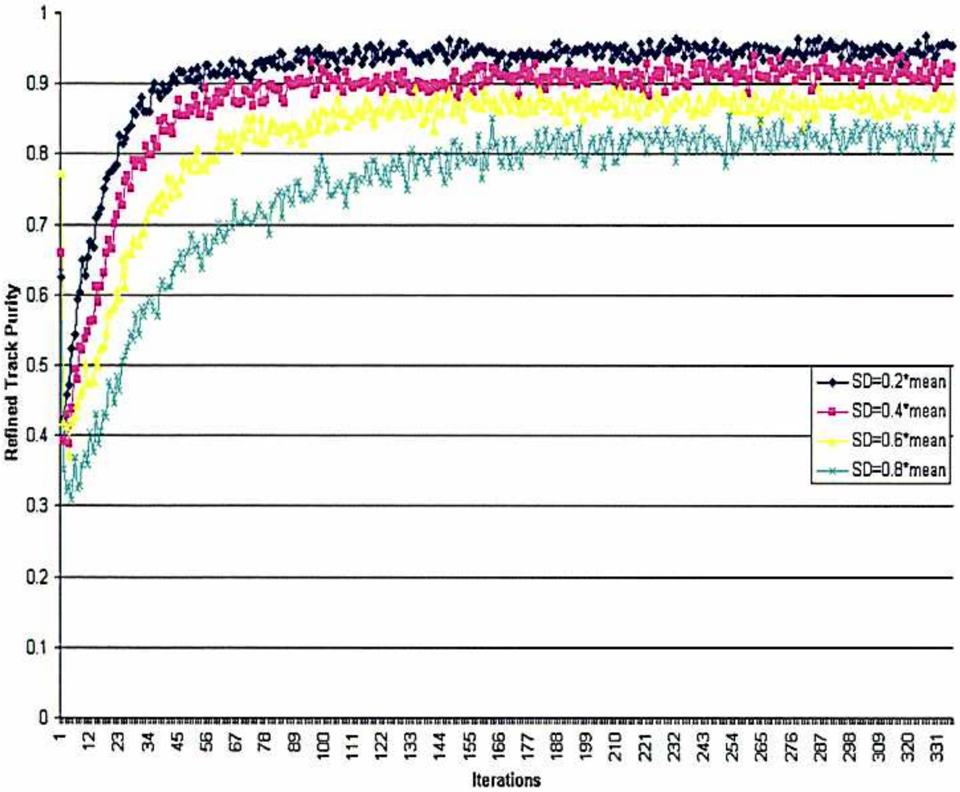

Figure 9: Track Purity dependence on transit time variance

of 45,000 events/second could be achieved using a compute array of 50 processors. The assignment algorithm itself can also be parallelised [2].

The most useful heuristic discovered is to use primary tracks of high purity to refine the Transit Graph. The most obvious way is to only update the graph during periods of light traffic, as primary track purity is a direct function of the number of individuals under observation sharing the same signature classification.

A second method is to calculate the mean track length over all primary tracks, and only update the Transit Graph with tracks whose length is less than the mean. The reasoning behind this is that short primary tracks are more likely to have been generated by a smaller number of individuals in the same signature classification, and by implication the track purity for those tracks is likely to be higher.

A third method is to tag primary tracks with information from the signature classifier, indicating whether the track corresponds to a signature in an infrequently populated part of signature space. Again, this track is more likely to be of higher purity.

A fourth method is to use the number of refined track hypotheses for a primary track as a guide to its quality. A primary track that refines into many tracks is likely to have a lower purity than one that refines to a single track.

Clearly these four methods can be combined in an heuristic to suggest which refined tracks are of high quality, and so suitable for updating the Transit Graph.

8. Signature Generation

The tracking system described depends on being able to generate a visual signature for *each person* within the field of view of a camera. This must take place reliably in the face of

- many individuals being within the field of view.
- occlusion of some individuals by others.
- occlusion of individuals by fixed foreground objects such as pillars, waste bins etc.
- occlusion of individuals by moving foreground objects such as shopping trolleys, vehicles, pushchairs etc.
- changes in light intensity.
- shadows.
- variation in cameras and light sources.
- distance, orientation, stance, gait and speed.

This is a formidable task. Tracking people using a single camera, or a small number of cameras with overlapping fields-of-view, is still an active area of research. It does not appear to be the case that there currently exists a system capable of dealing with all these requirements. On a positive note, it is unlikely that the problem will prove intractable, as various researchers claim to have solved significant parts of the problem.

Rather than attempt to be prescriptive about how to generate visual signatures, when the field is still open for developments, I will outline what appears to be a promising approach at this time, based on the techniques of colour image segmentation and colour indexing [3].

Colour indexing is one of the most important techniques in Content Based Image Retrieval. Its importance derives from the fact that objects can be distinguished/retrieved by using colour information alone, and colour feature extraction is relatively invariant to scale and orientation in a way that geometric features are not. It can also be carried out at real-time video frame rates.

A basic and popular technique is colour histogramming [17]. The colour space (e.g. RGB, or hue/saturation) is partitioned into buckets, and the number of pixels falling into each bucket is recorded and used as the basis for a colour signature. There is an issue of how to normalise the histogram for scale invariance. Another popular method is to fit the colour of the object to a parametric model such as a mixture of Gaussians; the parameters can then be used as a compact signature.

In both cases *colour constancy* must be considered. The light reflected from an object is dependent on the spectrum of the illuminating light, which in turn is composed of direct light and reflected light from the environment. The apparent colour of a given material can be different depending on the colour temperature of the main light source. For example, despite the apparent differences due to racial type, the colour of human skin is well defined in 2D hue/saturation space, but varies markedly according to the colour temperature of the light source [16].

There is also the effect of uneven illumination, so that a single colour may appear differently in different parts of an image. There are techniques for normalising the colour information in an image [5][6], but in general one would expect that a given camera would have to be calibrated with respect to available light sources.

Real-time tracking of people using colour image segmentation has been reported by many authors - the techniques used in the EasyLiving system [10] and Pfinder [19] are representative. The EasyLiving system uses colour histograms for identity maintenance, and so comes close to the signature idea presented here. However, no tracking system capable of generating a signature based on colour information has been identified that meets all the requirements described in this section, and there is an opportunity for work in this area.

The kinds of cameras typically used in today's surveillance applications are inadequate for the application described in this paper. The Smart Camera approach [20] would be ideal, which combines visual imaging with embedded processing for feature extraction in a compact, and (ultimately) low cost format. It is highly probably that binocular imaging would be used, as it is very effective in separating multiple foreground objects from a background, and provides distance information for image scaling.

9. Scaling Limitations

The tracking method described in this paper will work if the statistical parameters embodied in the Transit Graph are estimated using an offline method. This would be feasible for small camera networks. For a large network this may not be practical, and the bootstrapping method becomes attractive. The informal argument in section 6. shows that bootstrapping the matrix representation of the graph depends on three values: the number of vertices V in the camera network, the number of edges E between cameras, and the level of signature confusion in the observed population used to update the graph. It should be noted that only a subset of the total population need be used for updating, and four heuristics were described in section 7. that may be used to identify a subset of the population - literally, individuals who stand out and are easily tracked.

Without examining a real population and a practical signature generation scheme it is not possible to say what the maximum population is likely to be, but the simulations carried out suggest $O(100)$ cameras and $O(1000)$ people is likely to be an upper limit.

A solution to this limitation is to group cameras into cells, so that each cell covers a connected spatial extent, and the spatial union of the cells covers the total observed region. Cameras on the boundary of a cell would have multiple membership, belonging to two or more cells. Individuals would be tracked within a cell, and if a track terminated on the boundary of a cell it could be associated with a track or tracks initiated within the adjacent cell.

Work on this is proceeding. Preliminary indications are that this approach is feasible, and would permit scaling to very large networks.

10. Related Work

There is a large literature on using cameras to track human body parts and the human body [12], normally in a spatially limited environment. Although there is some literature on tracking individuals across cameras, this is normally with overlapping fields of view [14], or a small number of observed subjects [10]. There is a considerable literature on tracking large numbers of targets using a variety of sensor types with various military and air-traffic control applications [1][18], but these have not considered cameras in an urban setting.

The most relevant work which combines these approaches is described in Kettnaker's thesis and accompanying paper with Zabih [8][9]. Although many useful ideas have been adopted here from this work, it is limited in considering only a small number of cameras (4), a limited spatial environment (a building), and there is no architectural extension to large scale systems as proposed here. Transit likelihoods are measured by hand, and there is no process for generating and maintaining a Transit Graph.

11. Conclusions

A system has been described that would make it possible to track the movements of a large population of individuals over a spatially extended region, such as a town centre, an airport terminal, or a metro system. It uses cameras as visual sensors to identify key features of an individual (a visual signature) and uses tracking algorithms derived from radar and sonar applications to find the most probable tracks using knowledge of underlying traffic flows. One part of the algorithm is unusual in that it bootstraps knowledge of the background traffic flows from poor quality observational data, increasingly refining the quality of the observational data in parallel with the traffic flow patterns.

The tracking system has proved itself to be extremely robust and convergent; given observational data that is 1%-10% correct, it can refine that data to 90-95% correctness in the absence of any information about the network of visual sensors.

The system can easily handle real-time workloads, and work-in-progress indicates that the system can scale to practical levels.

The most immediate application for such a system would be retrospective analysis of crime scenes in public places, and it would be straightforward to add legal safeguards to such a system to prevent misuse and satisfy privacy concerns.

12. Acknowledgements

Thanks to Professor David Hogg for motivating words of encouragement, and to Dr. Dave Cliff for help with references.

13. References

[1] Yaakov Bar-Shalom, Thomas E. Fortmann, *Tracking and Data Association*, Academic Press, London 1988

[2] Dimitri P. Bertsekas, David A. Castanon, *Parallel Asynchronous Hungarian Methods for the Assignment Problem*, ORSA J. on Computing, Vol 5, pp. 261-274, 1993

[3] H.D. Cheng, X.H. Jiang, Jingli Wang, *Color image segmentation: advances and prospects*, Pattern Recognition 34, 2259-2281, 2001.

[4] Roger Clarke, *While You Were Sleeping ... Surveillance Technologies Arrived*, 2001, http://www.anu.edu.au/people/Roger.Clarke/DV/AQ2001.html

[5] Graham D. Finlayson, Bernt Schiele, James L. Crowley, *Using Colour for Image Indexing*, in *The Challenge of Image Retrieval*

[6] Graham D. Finlayson, Bernt Schiele, James L. Crowley, *Comprehensive Colour Image Normalisation*, ECCV'98, European Conference on Computer Vision, June 1998

[7] A.K. Jain, M.N. Murty, P.J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys, 31(3), 1999.

[8] Vera Maria Kettnaker, *Stochastic Models for the Analysis of Traffic Video*, PhD Dissertation, Cornell University.

[9] Vera Kettnaker, Ramin Zabih, *Bayesian Multi-camera Surveillance*, IEEE Conference on Computer Vision and Pattern Recognition, June 1999

[10] John Krumm et al, *Multi-Camera Multi-Person Tracking for EasyLiving*, Third IEEE International Workshop on Visual Surveillance, July 1 2000, Dublin, Ireland.

[11] H. W. Kuhn, *Variants of the Hungarian Method for assignment methods*, Naval Res. Logist. Quart. 3, 253-8, (1956)

[12] Thomas B. Moeslund, Erik Granum, *A Survey of Computer Vision-Based Human Motion Capture*, Computer Vision and Image Understanding, 81(3) 231-268

[13] J. Munkres, *Algorithms for the assignment and and transportation problems*, J. SIAM 5, 32-38 (1957)

[14] Ioannis Pavlidis, Vissilios Morellas, Panagiotis Tsiamyrtzis, Steve Harp, *Urban Surveillance Systems: from Laboratory to the Commerical World*, Proc. of the IEEE, Vol 89, 10, October 2001

[15] Aubrey Poore, *Multidimensional Assignment Formulation of Data Association Problems arising from Multitarget and Multisensor Tracking*, Computational Optimisation and Applications, 3, 27-57, 1994

[16] Moritz Storring, Hans J. Anderson, Erik Granum, *Skin colour detection under changing lighting conditions*, 7th. Symposium on Intelligent Robotics Systems, Coimbra, Portugal 1999

[17] M.J. Swain & D.H. Ballard, *Color Indexing*, International Journal of Computer Vision, 7(11):11-32, 1991

[18] Edward Waltz, James Lllinas, *MultiSensor Data Fusion*, Artech House, London 1990

[19] Christopher Wren, Ali Azerbayejani, Trevor Darrell, Alex Pentland, *Pfinder: Real-Time Tracking of the Human Body*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 780-785, 1997.

[20] Wayne Wolf, Burak Ozer, Tiehan Lv, *Smart Cameras as Embedded Systems*, IEEE Computer, September 2002.