Chapter 20
Scalable Multi-camera Tracking in a Metropolis

Yogesh Raja and Shaogang Gong

This work is dedicated to Colin Lewis, in memory of his lifelong passion for pushing the boundaries in making academic research relevant to meeting real-world challenges, and for his unequivocal support in making this work possible.

Abstract The majority of work in person re-identification is focused primarily on the matching process at an algorithmic level, from identifying reliable features to formulating effective classifiers and distance metrics in order to improve matching scores on established 'closed-world' benchmark datasets of limited scope and size. Very little work has explored the pragmatic and ultimately challenging question of how to engineer working systems that best leverage the strengths and tolerate the weaknesses of the current state of the art in re-identification techniques, and which are capable of scaling to 'open-world' operational requirements in a large urban environment. In this work, we present the design rationale, implementational considerations and quantitative evaluation of a retrospective forensic tool known as Multi-Camera Tracking (MCT). The MCT system was developed for re-identifying and back-tracking individuals within huge quantities of open-world CCTV video data sourced from a large distributed multi-camera network encompassing different public transport hubs in a metropolis. There are three key characteristics of MCT, *associativity*, *capacity* and *accessibility*, that underpin its scalability to spatially large, temporally diverse, highly crowded and topologically complex urban environments with transport links. We discuss a multitude of functional features that in combination address these characteristics. We consider computer vision techniques and machine learning algorithms, including relative feature ranking for inter-camera matching,

Y. Raja
Vision Semantics Ltd, London, UK
e-mail: yraja@visionsemantics.com

S. Gong (✉)
Queen Mary University of London, London, UK
e-mail: sgg@eecs.qmul.ac.uk

S. Gong et al. (eds.), *Person Re-Identification*,
Advances in Computer Vision and Pattern Recognition,
DOI: 10.1007/978-1-4471-6296-4_20, © Springer-Verlag London 2014

global (crowd-level) and local (person-specific) space–time profiling, attribute re-ranking and machine-guided data mining using a 'man-in-the-loop' interactive paradigm. We also discuss implementational considerations designed to facilitate linear scalability to an aribitrary number of cameras by employing a distributed computing architecture. We conduct quantitative trials to illustrate the potential of the MCT system and its performance characteristics in coping with very large-scale open-world multi-camera data covering crowded transport hubs in a metropolis.

20.1 Introduction

Human investigators tasked with the forensic analysis of video from multi-camera CCTV networks face many challenges, including (1) data overload from large numbers of cameras, (2) a short attention span leading to important events and targets being missed, (3) a lack of contextual knowledge indicating what to look for and (4) a lack of or inability to utilise complementary non-visual sources of knowledge to assist the search process. Consequently, there is a distinct need for technology to alleviate the burden placed on limited human resources and augment human capabilities.

As reflected in the published literature, much research effort has been expended in developing low-level methods for the automatic visual re-identification of people and other objects appearing in different places and at different time across multiple cameras. The ultimate goal is to build 'black-box' systems capable of unilaterally solving this problem. However, this is an inherently challenging task, especially if visual appearance is the only available cue for discrimination, as shown in Fig. 20.1. Much of the focus so far has been on finding the most reliable representative features to employ in constructing templates of individuals' visual appearance (e.g. major colours [12], combinations of colour and texture [14], complex structural layouts [4]), along with distance metrics (e.g. Bhattacharyya distance [14], L1-Norm [20]) or classifiers (e.g. K-Nearest Neighbour [5], Ranking SVM [2]) for matching. Such work is generally conditioned towards maximising ranking performance on small, carefully constructed *closed-world* benchmark datasets largely unrepresentative of the scale and complexity of *open-world* scenarios where the number of cameras, spatial size of the environment and numbers of people are all at a significantly larger scale, with a search space of unknown size and a potentially unlimited number of candidate matches for a target. Re-identification of targets in such open environments can potentially scale to arbitrary levels, covering huge spatial areas spanning not just different buildings but different cities, countries or even continents, leading to an overwhelming quantity of 'big data'.

To date, very little work has focused on addressing the practical question of how to best leverage the current state of the art in re-identification techniques, while tolerating their limitations in *engineering* practical systems that are *scalable* to typical real-world operational scenarios. In this work, we describe the design rationale and implementational considerations of building a prototype system known as *Multi-Camera Tracking* (*MCT*), a tool which human operators may employ for generating

Fig. 20.1 An illustration of the difficulties of visual matching across different camera views. Individuals may undergo significant variability in appearance due to changes in lighting, scale and viewpoint. Other difficulties are caused by partial or complete occlusion which results in a lack of complete visual information, and the tendency for variability between people (inter-variation) to be less than the variability for a single person at different times and camera views (intra-variation). All of these problems are compounded by spatially large environments with significant numbers of cameras and levels of crowding

a *global target trail* by retrospectively searching, 'back-tracking' and reconstructing the movements of targets of interest across multiple disjoint camera views in a large public space spanning a city. The system takes the basic approach of searching within multiple camera views for a specified target from a watchlist and producing *ranked lists* of candidate matches. Rather than attempting to solve the challenge of a fully automatic black-box winner-take-all solution for Rank-1 re-identification, the system takes the more practical approach of implementing mechanisms that: (a) quickly and effectively narrow the search space of candidates for human operators to perform target verification; and (b) incrementally increase the ranks of likely correct matches without making hard decisions that may inadvertently discard them at too early a stage during the re-identification process.

The overall design of the MCT system takes into account three key characteristics in order to systematically address the challenge of scalability: (1) *Associativity*, concerning the ability of the system to help users accurately extract targets of interest from an extremely large search space; (2) *Capacity*, relating to computational resources and the ability to process large numbers of camera inputs simultaneously; and (3) *Accessibility*, the speed with which users can conduct searches of targets and reconstruct their movements. In order to scale to arbitrarily large, busy and visually complex spaces, the MCT system requires various augmentations to satisfy these three requirements, in addition to the implementation of standard computer vision and machine learning techniques. This is addressed through a highly modular, flexible network-centric implementation able to incrementally leverage multiple hardware components in order to process arbitrary numbers of cameras, and a

carefully designed user interface combining several mechanisms that support a coherent iterative *piecewise search strategy* for efficiently retracing multiple target movements through large complex multi-camera environments.

In Sect. 20.2, we discuss six key mechanisms that enable the MCT system to address the requirement of *associativity*. In Sect. 20.3, we detail implementational considerations that permit the system to address the requirements of *capacity* and *accessibility*. In Sect. 20.4, we describe a highly challenging open-world dataset encompassing two transport hubs in a metropolis. This is used to conduct quantitative trials of the MCT system, results of which are provided in Sect. 20.5. Finally, in Sect. 20.6 we conclude with lessons learned and open questions for future work.

20.2 Key Mechanisms

The *associativity* of a scalable multi-camera tracking tool is related to the efficiency and reliability with which it can aid users in locating targets of interest amongst very large numbers of individuals. Consequently, the fundamental objectives for the system are to reduce user workload by: (a) appropriately narrowing the search space and producing a minimal set of candidates containing the target; and (b) ranking the target highly within the candidate set. There are six key mechanisms of the MCT system that combine to address these objectives.

20.2.1 Relative Feature Ranking

The MCT system employs a comprehensive set of 29 types of visual features encompassing the colour and texture appearance of individuals for matching across camera views. More specifically, the colour features incorporate different colour spaces including RGB, Hue-Saturation and YCrCb, with texture features derived from Gabor wavelet responses at eight different scales and orientations, as well as thirteen differently parameterised Schmid filters [18]. Details can be found in [15]. Image patches within bounding boxes corresponding to people automatically detected by a parts-based person detector [3] are resampled to 300 pixels wide for consistency of scale. They are then split into six equal horizontal segments, with separate normalised histograms generated for each segment before concatenation into a single *feature vector*. Given 16 bins for the histogram corresponding to each of the 29 feature types for each of the 6 horizontal strips, we thus have a 2784-dimensional feature vector per bounding box, which is used as an *appearance descriptor*.

Rather than considering each feature type equally in terms of relevance, we dynamically learn the importance of each of these feature types to more strongly weight those features most relevant for matching across different cameras [15, 21, 22]. Such a model is trained from a dataset of pairs of feature vectors derived from single detections of the same person taken from different cameras [17]. More pre-

cisely, given a training set of m samples $X = (\mathbf{x}_i, y_i)_{i=1}^{m}$ where $\mathbf{x}_i \in R^d$ is a feature vector for a specific individual and y_i a corresponding label, and given the feature vectors $\mathbf{x}_j^+$ from the same training set X corresponding to the same person from another view (called *relevant* feature vectors) along with $\mathbf{x}_j^-$ corresponding to different people (called *irrelevant* feature vectors), we learn a ranking function δ to rank vector pair similarity such that $\delta(\mathbf{x}_i, \mathbf{x}_j^+) > \delta(\mathbf{x}_i, \mathbf{x}_j^-)$. This takes the form of a support vector machine (SVM) known as RankSVM [2, 6]. The RankSVM model is characterised by a linear function for ranking matches between two feature vectors as $\delta(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{w}^\top |\mathbf{x}_i - \mathbf{x}_j|$. Given a feature vector $\mathbf{x}_i$, the required relationship between relevant and irrelevant feature vectors is $\mathbf{w}^\top (|\mathbf{x}_i - \mathbf{x}_j^+| - |\mathbf{x}_i - \mathbf{x}_j^-|) > 0$, i.e. the ranks for all correct matches are higher than the ranks for incorrect matches. Accordingly, given $\hat{\mathbf{x}}_s^+ = |\mathbf{x}_i - \mathbf{x}_j^+|$ and $\hat{\mathbf{x}}_s^- = |\mathbf{x}_i - \mathbf{x}_j^-|$ and the set $P = \{(\hat{\mathbf{x}}_s^+, \hat{\mathbf{x}}_s^-)\}$ of all pairwise relevant difference vectors required to satisfy the above relationship, a corresponding RankSVM model can be derived by minimising the objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C \sum_{s=1}^{|P|} \xi_s \tag{20.1}$$

with the constraints:

$$\mathbf{w}^\top (\hat{\mathbf{x}}_s^+ - \hat{\mathbf{x}}_s^-) \geq 1 - \xi_s \tag{20.2}$$

for each $s = 1, \ldots, |P|$ and restricting all $\xi_s \geq 0$. C is a parameter for trading margin size against training error.

20.2.2 Matching by Tracklets

For comparing individuals, the Munkres Assignment algorithm, also known as the Hungarian algorithm [7, 13], is employed as part of a multi-target tracking scheme to increase the number of samples for each individual by locally grouping detections in different frames as likely belonging to the same person. This process yields *tracklets* encompassing individual detections over multiple frames, representing short intra-camera trajectories. An individual D is accordingly represented as a tracklet $T_D = \{\alpha_{D,1}, \ldots, \alpha_{D,J}\}$ comprising a set of J individual detections with appearance descriptors $\alpha_{D,j}$. Two individuals are then matched by computing the median match score between each combination of detection pairs, one each from their respective tracklets. This approach mitigates the difficulties that might be faced by object tracking techniques in highly crowded environments, where irregular movement and regular occlusion causes tracking failure. Computing the median as a tracklet match score permits a degree of robustness against erroneous assignments, where tracklets may inadvertently comprise samples from multiple individuals.

More precisely, tracklets are built up incrementally over time, with an incomplete set updated after each frame by assigning individual detections from that frame to

a tracklet according to their appearance similarity and spatial proximity. That is, given: (1) a set $S = \{\alpha_{1,f}, \ldots, \alpha_{M,f}\}$ of M appearance descriptors for detections in frame f with corresponding pixel locations $\{\beta_{1,f}, \ldots, \beta_{M,f}\}$; (2) a current set of N incomplete tracklets $R = \{\hat{T}_1, \ldots, \hat{T}_N\}$ with their most recently added appearance descriptors $\{\hat{\alpha}_{n,f_n}\}$; and (3) corresponding *predicted* pixel locations $\{\hat{\beta}_{n,f}\}$, an $M \times N$ cost matrix $\mathbf{C}$ is generated where each entry $\mathbf{C}_{m,n}$ is computed as:

$$\mathbf{C}_{m,n} = \omega_1 |\hat{\alpha}_{n,f_n} - \alpha_{m,f}| + \omega_2 |\hat{\beta}_{n,f} - \beta_{m,f}| \tag{20.3}$$

In essence, this cost is computed as a weighted combination of appearance descriptor dissimilarity and physical pixel distance. Predicted pixel locations $\hat{\beta}_{n,f}$ for frame f are estimated by assuming constant linear velocity from the last known location and velocity. The Munkres Assignment algorithm maps rows to columns in $\mathbf{C}$ so as to minimise the cost, with each detection added accordingly to their mapped incomplete tracklets. Surplus detections are used to initiate new tracklets. In practice, an upper bound is placed on cost, with assignments exceeding the upper bound being retracted, and the detection concerned treated as surplus. Additionally, tracklets which have not been updated for a length of time are treated as complete.

For re-identification, completed tracklets are taken as a representation for an individual, though individuals may comprise several tracklets. When matching two individuals D_1 and D_2 with corresponding tracklets T_{D_1} and T_{D_2}, the score S_j for each pairing of appearance descriptors $\{(x, y) : x \in T_{D_1}, y \in T_{D_2}\}, j = 1, \ldots, J_1 J_2$ where $J_1 = |T_{D_1}|$ and $J_2 = |T_{D_2}|$ is computed using the RankSVM model as:

$$S = \mathbf{w}^\top (|x - y|) \tag{20.4}$$

where $\mathbf{w}$ is obtained by minimising Eq. (20.1). The match score S_{D_1,D_2} for the two tracklets as a whole is computed as the median of these scores over all pairs of their appearance descriptors:

$$S_{D_1,D_2} = \text{median}\left(\{S_1, S_2, \ldots, S_{J_1 J_2}\}\right); \tag{20.5}$$

A set of candidate matches is ranked by sorting their corresponding tracklet scores in descending order.

20.2.3 Global Space–Time Profiling

Given the inherent difficulties in visual matching when visual appearance lacks discriminability, not least in real-world scenarios where there are a very large number of possible candidates for matching, it becomes critical that higher level prior information is exploited to provide space–time context and significantly narrow the search space [10, 11, 17]. Our approach is to dynamically learn the typical movement

patterns of individuals throughout the environment to yield a probabilistic model of when and where people detected in one view are likely to appear in other views. This top-down knowledge is imposed during the query process to drastically reduce the search space and dramatically increase the chances of finding correct matches, having a profound effect on the efficacy of the system.

More specifically, we employ the method proposed in [10, 11]. Each camera view is decomposed automatically into regions, across which different spatio-temporal activity patterns are observed. Let $\mathbf{x}_i(t)$ and $\mathbf{x}_j(t)$ denote the two regional activity time series observed in the ith and jth regions, respectively. These time series comprise the 2,784-dimensional appearance descriptors of detected individuals (Sect. 20.2.1). Cross Canonical Correlation Analysis (xCCA) is employed to measure the correlation of two regional activities as a function of an unknown time lag τ applied to one of the two regional activity time series. Denoting $\mathbf{x}_j(t) = \mathbf{x}_i(t+\tau)$, we drop the parameters t and τ for brevity to denote $\mathbf{x}_j = \mathbf{x_i}$. Then, for each time delay index τ, xCCA finds two sets of optimal basis vectors $\mathbf{w}_{\mathbf{x}_i}$ and $\mathbf{w}_{\mathbf{x}_j}$ such that the projections of $\mathbf{x}_i$ and $\mathbf{x}_j$ onto these basis vectors are mutually maximally correlated.

That is, given $\mathbf{x}_i = \mathbf{w}_{\mathbf{x}_i}^T \mathbf{x}_i$ and $\mathbf{x}_j = \mathbf{w}_{\mathbf{x}_j}^T \mathbf{x}_j$, the canonical correlation $\rho_{\mathbf{x}_i,\mathbf{x}_j}(\tau)$ is computed as:

$$\rho_{\mathbf{x}_i,\mathbf{x}_j}(\tau) = \frac{E[\mathbf{w}_{\mathbf{x}_i}^\top \mathbf{C}_{\mathbf{x}_i\mathbf{x}_j} \mathbf{w}_{\mathbf{x}_j}]}{\sqrt{E[\mathbf{w}_{\mathbf{x}_i}^\top \mathbf{C}_{\mathbf{x}_i\mathbf{x}_i} \mathbf{w}_{\mathbf{x}_i}]}\sqrt{E[\mathbf{w}_{\mathbf{x}_j}^\top \mathbf{C}_{\mathbf{x}_j\mathbf{x}_j} \mathbf{w}_{\mathbf{x}_j}]}} \tag{20.6}$$

where $\mathbf{C}_{\mathbf{x}_i\mathbf{x}_i}$ and $\mathbf{C}_{\mathbf{x}_j\mathbf{x}_j}$ are the within set covariance matrices of $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively, and $\mathbf{C}_{\mathbf{x}_i\mathbf{x}_j}$ is the between-set covariance matrix.

The time delay that maximises the canonical correlation between $\mathbf{x}_i(t)$ and $\mathbf{x}_j(t)$ is then computed as:

$$\hat{\tau}_{\mathbf{x}_i,\mathbf{x}_j} = \text{argmax}_\tau \frac{\sum^{\Gamma} \rho_{\mathbf{x}_i,\mathbf{x}_j}(\tau)}{\Gamma} \tag{20.7}$$

where $\Gamma = \min\big(\text{rank}(\mathbf{x}_i), \text{rank}(\mathbf{x}_j)\big)$.

Given a target nominated in camera view j for searching in camera view k, the search space is narrowed by considering only tracklets from k with a corresponding time delay less than $\alpha\hat{\tau}_{\mathbf{x}_j,\mathbf{x}_k}$ (with α a constant factor) for matching. This candidate set is then ranked accordingly.

20.2.4 'Man-in-the-Loop' Machine-Guided Data Mining

The MCT system is an interactive 'man-in-the-loop' tool designed to enable human operators to retrospectively re-trace the movements of targets of interest through a spatially large, complex multi-camera environment by performing *queries* on generated metadata. A common-sense approach to doing so in such an environment is to employ an iterative *piecewise search strategy*, conducting multiple progressive

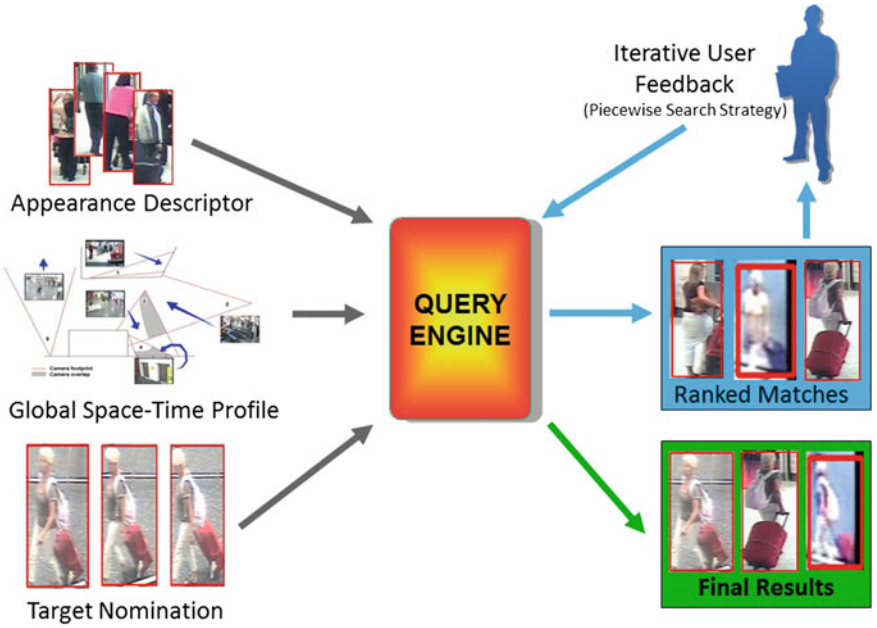

Fig. 20.2 Usage of the MCT system. Given automatically extracted appearance descriptors from across the multi-camera network along with a global space–time profile (Sect. 20.2.3), users nominate a target and then iteratively search through the network in a piecewise fashion, marking observed locations and times of the target in the process. The procedure stops when the target has been re-identified in a sufficient number of views for an automatically generated reconstruction

searches over several iterations to gradually build a picture of target movements, or *global target trail*.

More precisely, given the initial position of a nominated target, the first search is conducted in the place most likely to correspond to their next appearance, such as the adjacent camera view depending on direction of movement. Further detections of the target provide constraints upon the next most likely location, within which the next search iteration is conducted. The search thus proceeds in a manner gradually spanning out from the initial detected position, marking further detections along the way and building a picture of target movements, until the number of locations has been exhausted or the picture is sufficiently detailed for an automatically generated reconstruction of the target's movement through the environment. This approach ensures that the problem is tackled piecemeal, with the overall search task simplified and the workload on users minimised. Figure 20.2 illustrates the top-level paradigm for system usage.

Additionally, in the process of conducting a query, unexpected associations such as previously unknown accomplices may be discovered. These are not only highly relevant to the investigation at large, but may be exploited as part of the search process itself. Such associates may naturally and seamlessly be incorporated into the query,

forming a parallel branch of enquiry which proceeds in the same way. This allows: (a) accomplices to aid in the detection of the target of interest, for example if the latter is not visible for the system to detect but inferable by way of their proximity to the detectable accomplice; and (b) accomplices to be tracked independently at the same time as the original target should their trajectories through the multi-camera network diverge.

The basic MCT query procedure is as follows: A user initiates a query Q_0 which comprises a nominated target with tracklets $T_0 = \{t_{0,j_0}\}$, $j_0 = 1, \ldots, J_0$ from camera view ξ_0. The first search iteration is conducted in camera view ξ_1, resulting in a set of J_1 candidate matches $T_1 = \{t_{1,j_1}\}$, $j_1 = 1, \ldots, J_1$. Any number of these can be tagged by the user, whether they correspond to the initial target or a relevant association, yielding a set R_1 of K_1 indices for 'relevant' flags $R_1 = \{r_{k_1}\}$, $k_1 = 1, \ldots, K_1$. The set $C_1 = \{t_{1,r_{k_1}}\}$ is then used to initiate the next iteration of the query Q_1 in camera view ξ_2, yielding J_2 new candidate matches $T_2 = \{t_{2,j_2}\}$, $j_2 = 1, \ldots, J_2$. These are again marked accordingly by the user, yielding a set R_2 of K_2 indices for 'relevant' flags $R_1 = \{r_{k_2}\}, k_2 = 1, \ldots, K_2$. The new set $C_2 = \{t_{2,r_{k_2}}\}$ is combined with the set from the previous iteration C_1 as well as the initial nomination to produce an aggregate set $\hat{C}_2 = \{T_0 \cup C_1 \cup C_2\}$. The search proceeds for as many iterations as required, finding relevant matches in each camera view. After n iterations, we have the aggregate pool of matches tagged as relevant by the user over all previous search iterations, plus the original nomination:

$$\hat{C}_n = \{T_0 \cup C_1 \cup C_2 \cup \cdots \cup C_n\} = \{t_{0,j_0}\} \cup \{t_{1,r_{k_1}}\} \cup \cdots \cup \{t_{n,r_{k_n}}\} \tag{20.8}$$

This set constitutes the final associated evidence from which a video reconstruction of target movements is automatically created by the system and instantly viewable. Note that the search process is not generally linear. The interface provides the flexibility to search in multiple cameras at once and then analyse the results from each camera one-by-one. A user may also select matches from previous iterations to conduct searches in a future iteration. This enables multiple targets to be tracked as part of a single query as well as tracking movements both backwards and forwards in time.

20.2.5 Attribute-Based Re-ranking

The RankSVM model (Sect. 20.2.1) [2, 6] employs appearance descriptors comprising a multitude of low-level feature types which are weighted by the RankSVM model. However, such a representation is not always sufficiently invariant to changes in viewing conditions, leading to blunted discriminability. Furthermore, to a human observer, such feature descriptors are not amenable to descriptive interpretation. For example, depending on the tracking scenario, human operators may focus on unambiguous characteristics of a target, such as attire, colours or patterns. Consequently, we incorporate mid-level *semantic attributes* [8, 9] as an intuitive complementary method of ranking candidate matches. Users may select multiple attributes descrip-

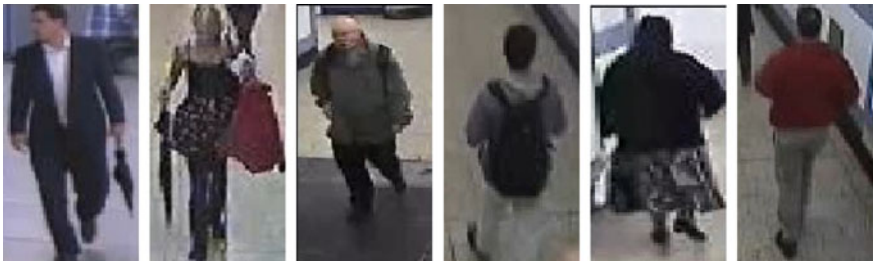

Fig. 20.3 Examples of images associated with semantic mid-level attributes. Some images can be associated with multiple attributes simultaneously—for example, the second example labelled 'Femal' can be also be labelled 'Skirt' (i.e. she is also wearing a skirt), and the third example labelled as 'Bald' can also be labelled 'Backpack'

tive of the target to re-rank candidates and encourage correct matches to rise, reducing the time taken for localisation.

We identify 19 semantic attributes, including but not limited to *bald*, *suit*, *female*, *backpack*, *headwear*, *blue* and *stripy*. Figure 20.3 shows some example images associated with these attributes. We then create a training set of 3,910 sample images of 45 different individuals across multiple camera views and for each sample j generate an appearance descriptor α_j of the form used for the RankSVM model (Sect. 20.2.1). These are manually annotated according to the 19 attributes. Given this data, a set of *attribute detectors* a_i, $i = 1, \ldots, 19$ in the form of support vector machines using intersection kernels are learned [8, 9] using the LIBSVM library [1]. Cross-validation is employed to select SVM slack parameter values.

The outputs of the detectors are in the form of posterior probabilities $p(a_i|\alpha)$, denoting the probability of attribute i given an appearance descriptor α. Given I user-selected attributes $\{a_1, \ldots, a_I\}$ and a set of K candidate matches $\{t_1, \ldots, t_K\}$ where candidate $t_k = \{\alpha_{k,1}, \ldots, \alpha_{k,J}\}$ is a set of J appearance descriptors, the score $S_{i,k}$ for each attribute a_i is computed for each candidate t_k as an average of the posterior probabilities for each of the J appearance descriptors:

$$S_{i,k} = \frac{1}{J} \sum_{j=1}^{J} p(a_i|\alpha_j) \tag{20.9}$$

Accordingly, each candidate t_k has an associated vector of scores $[S_{1,k}, \ldots, S_{I,k}]^\top$. The set of candidates is then ranked separately for each attribute, averaging the ranks for each candidate and finally sorting by the average rank.

20.2.6 Local Space–Time Profiling

Global space–time profiles (Sect. 2.1.2) significantly narrow the search space of match candidates by imposing constraints learned from the observed movements of *crowds in-between camera views*. To complement this, local space–time profiles further reduce the set of candidate matches by imposing constraints implied by observed movements of *specific individuals within each camera view*. Ultimately, this may incorporate knowledge of scene structure and likely trajectories of individuals within the view, for example depending on which exit they are likely to take in a multi-exit scene.

For the MCT system, we employed a simple method of filtering known as *Convergent Querying*. For each camera view i, $i = 1, \ldots, 6$, we selected a small set of example individuals (e.g. 20) at random and manually measured the *length of time* they were visible in that view, i.e. from the frame of their appearance to the frame of their disappearance. Temporal windows τ_i were then estimated for each camera view as:

$$\tau_i = E[X_i] + 3\sqrt{(\mathrm{Var}(X_i))} \tag{20.10}$$

where X_i denotes the random variable for the observed transition times in frames from Camera i.

Given a set $T = \{t_1, \ldots, t_J\}$ of J candidate matches (tracklets) from Camera i, the user may tag one of the matches t_j for local space–time profiling, resulting in the pruned set:

$$\hat{T} = \{\hat{t} \in T : |\phi(\hat{t}) - \phi(t_j)| \leq \tau_i\} \tag{20.11}$$

where $\phi(t)$ is a function returning the average of the first and last frame indices of the individual detections of tracklet t. Consequently, the filter removes all tracklets lying outside the temporal window, narrowing the results to those corresponding to the tighter time period within which the specific target is expected to appear.

20.3 Implementation Considerations

The *capacity* of a multi-camera tracking system relates to the ability to process, generate and store metadata for very large numbers of cameras simultaneously. A related characteristic is *accessibility*, the ability to query the generated metadata in a speedy fashion. Accordingly, the ability of the system to scale to typical open-world scenarios where the quantity of data can arbitrarily increase depends upon careful design and implementation of the processing architecture, the user interface and in particular, the metadata storage scheme.

In order to enable on-the-fly analysis of videos streams which may be pre-recorded and finite or live and perpetual, the general top-level approach we take towards implementing the MCT prototype is to produce two independently functioning subsystems.

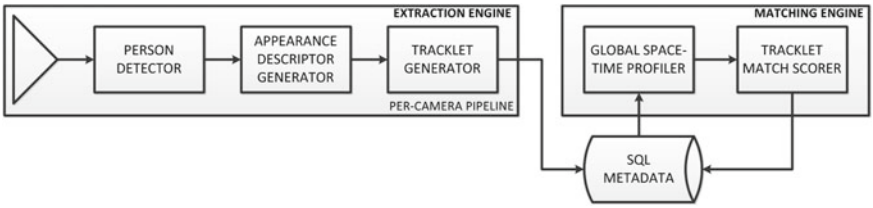

Fig. 20.4 MCT Core Engine, depicting the asynchronous Extraction and Matching engines. The Extraction Engine takes the form of a multi-threaded processing pipeline, enabling efficient processing of multiple inputs simultaneouslyon multi-core CPU platforms

First, the *Generator Subsystem* is responsible for processing video streams and generating *metadata*. This metadata includes *tracklets* of detected people in each camera view and the storage of this metadata in a backend database. Targets which may be nominated by individuals are restricted to those that can be *automatically detected* by the system rather then permitting users to arbitrarily select image regions that may correspond to objects of interest but which may not be visually detectable automatically. Second, the *Interrogator Subsystem* provides a platform for users to query the generated metadata through a secure, encrypted online browser-based interface. These two subsystems operate asynchronously, enabling users to query metadata via the Interrogator Subsystem as and when they become available by way of the Generator Subsystem functioning in parallel. The MCT system is designed to be flexible and for its components to inter-operate either locally or remotely across a network, in order to permit the incremental utilisation of off-the-shelf hardware. For example, the entire system may operate on a single server, or with each component on separate servers connected via the Internet.

Metadata is stored in an *SQL Metadata* backend database component. A *Video Streamer* provides video data from recorded or live input to an *MCT Core Engine* and multiple *User Interface (UI) Clients* that encapsulate the essential functionalities of the MCT system.

The MCT Core Engine comprises two asynchronous sub-components known as the *Extraction* and *Matching* Engines, which form the primary processing pipeline for generating metadata for the Generator Subsystem (Fig. 20.4). This MCT pipeline employs a multi-threaded approach, efficiently utilising multi-core CPUs to process multiple camera inputs simultaneously. This implementation enables additional processing resources to be added as available in a flexible manner. For example, multiple camera inputs may be processed on a single machine, or allocated as desired across several machines. Such flexibility also applies to the Extraction and Matching Engines, which can be allocated separately for each camera. This facilitates potentially unlimited incremental additions to hardware resources with ever-increasing numbers of cameras.

The User Interface (UI) Clients are Java web-based applets which interface remotely with a *Query Engine Server* to enable the search of metadata stored in the SQL Metadata component. Usage of the system only requires access to a basic

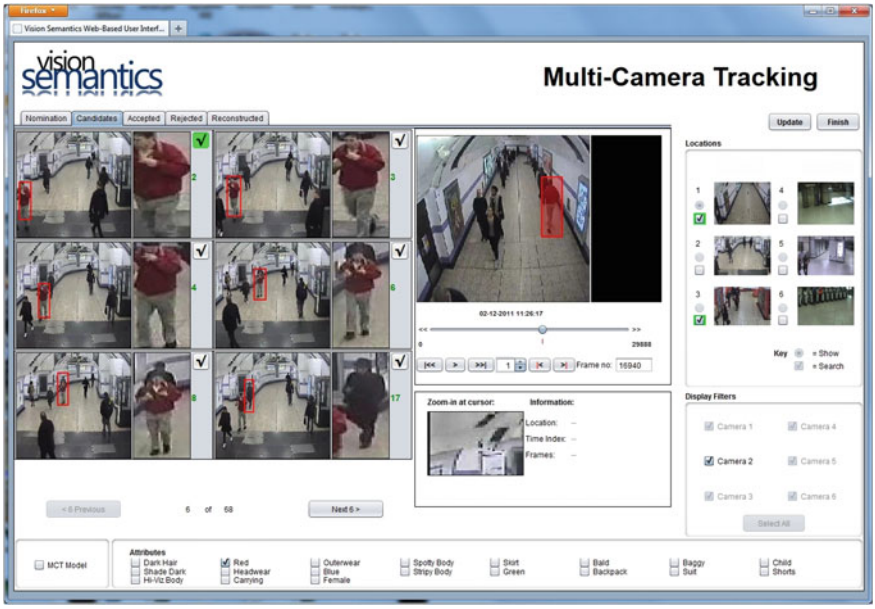

Fig. 20.5 MCT User Interface Client example screenshot. Here, users examine a paginated set of match candidates from a search iteration, locating and tagging those relevant including target associates. The local space–time profile Convergent Querying filter is employed here, using tagged candidates to immediately narrow the set being displayed to the appropriate temporal window. Attributes appropriate to the target may also be selected, which instantly re-rank the candidate list accordingly

terminal equipped with a standard web browser with a Java plugin. Security features include password protected user logins, per-user usage logging, automatic time-outs and fully encrypted video and metadata transfer to and from the Query Engine Server. The interface includes functions to support the piecewise search strategy (Sect. 20.2.4), as well as for viewing dynamically generated chronologically-ordered video reconstructions of target movements.

Figure 20.5 depicts an example screenshot of the MCT User Interface Client. This screen lists all candidate matches returned from a search iteration in paginated form. Here, users may browse through candidate matches from a search iteration, locating and tagging those which are relevant to the query. Two key features available are: (1) the *Convergent Querying filter* which is applied when tagging a candidate, instantly imposing local space–time profiling on the currently displayed set (Sect. 20.2.6); and (2) *Attribute selection* checkboxes for instant re-ranking of candidates by user-selected semantic attributes (Sect. 20.2.5).

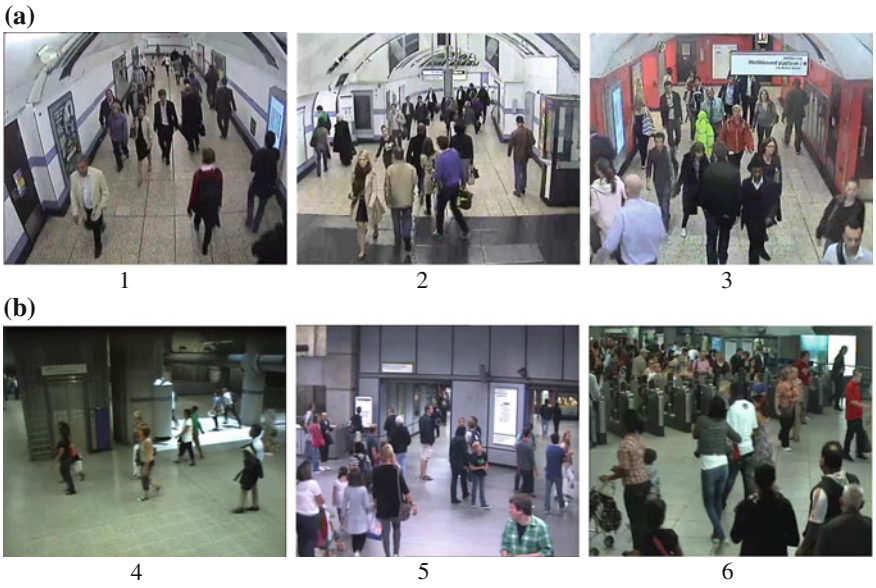

Fig. 20.6 MCT trial dataset example video images. **a** Cameras 1, 2 and 3 (from Station A). *Left* Corridor leading from entrance to Station A; *Centre* Escalator to train platforms; *Right* Entrances to platforms. **b** Cameras 4, 5 and 6 (from Station B). *Left* Platforms for trains to Station A; *Centre* Platforms for those arriving from Station A; *Right* Ticket barriers at entrance to Station B

20.4 MCT Trial Dataset

As defined in Sect. 20.1, there are three key characteristics that influence scalability: *associativity*, *capacity* and *accessibility*. The scale of the environment concerned profoundly impinges upon all of these factors since it is correlated with the quantity of data to process, as well as the number of individuals to search through and for whom metadata must be generated and stored. We conducted an in-depth evaluation of the MCT system in order to determine its scalability in terms of these three factors.

The MCT system has previously been tested [17] using the i-LIDS multi-camera dataset [19]. The i-LIDS data comprised five cameras located at various key points in an open environment at an airport. A key limitation of this dataset is that the five cameras covered a relatively small area within a single building, where passengers moved on foot in a single direction with transition across the entire network taking at most 3 min. As such, the scale of the i-LIDS environment is limited for testing typical open-world operational scenarios. Trialling the MCT system requires an open-world test environment unlike all existing closed-world benchmark datasets.

To address this problem, we captured a new trial dataset during a number of sessions in an operational public transport environment [16]. This dataset comprises six cameras selected from existing camera infrastructure covering two different transport hubs at different locations on an urban train network, reflecting an open-world

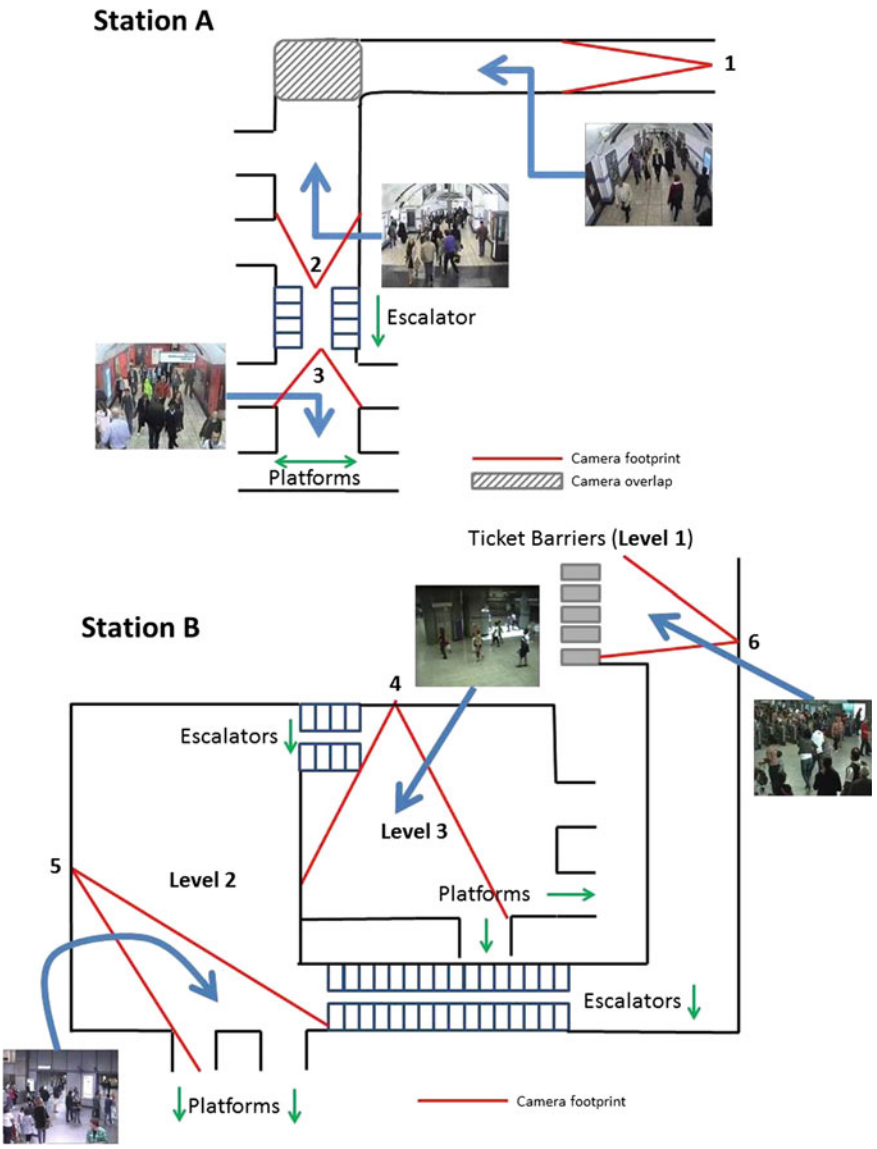

Fig. 20.7 Topological layout of Stations A and B. Three cameras (sample frames shown) were selected from each and used for data collection and MCT system testing

operational scenario. Camera locations are connected by walkways within each hub and a transport link connecting the two hubs. Lighting changes and viewpoints exhibit greater variability, placing more stress on the matching model employed by a re-identification system. Furthermore, passenger movements are multi-directional

and less constrained, increasing uncertainty in transition times between camera views. The average journey time between the two stations across the train network takes approximately 15 min. Example video images are shown in Fig. 20.6, and the approximate topological layout of the two hubs and the relative positions of the selected camera views are shown in Fig. 20.7.

As a comparison between the MCT trial dataset and the i-LIDS multi-camera dataset, each i-LIDS video ranges from between 4,524 to 8,433 frames, yielding on average 39,000 candidate person detections and around 4,000 computed tracklets. In contrast, each 20 min segment of the MCT trial dataset contains typically 30,000 video frames with around 120,000 candidate person detections and 20,000 tracklets. Consequently, the complexity and volume of the data to be searched and matched in order to re-identify a target demonstrates an increase by one order of magnitude over the i-LIDS dataset [19], making it *significantly* more challenging.

The MCT trial dataset was collected over multiple sessions for prolonged periods during operational hours spanning more than 4 months. Each session produced over 3 h of testing data. To form ground-truth and facilitate evaluation, in each session a set of 21 volunteers containing a mixture of attire, ages and genders travelled repeatedly between Stations A and B. These volunteers formed a *watchlist* such that they could all be selected as probe targets for re-identification. Since reappearance of the majority of the travelling public is not guaranteed due to the open-world characteristics of the testing environment, this ensured that the MCT trial dataset contained a subgroup of the travelling public known to reappear between the two stations, facilitating suitable testing of the MCT system.

20.5 Performance Evaluation

We conducted an extensive evaluation of the MCT system against the three key scalability requirements: *associativity* (tracking performance), *capacity* (processing speed) and *accessibility* (user querying speed). The results are as follows:

20.5.1 Associativity

The performance of the MCT system in aiding cross-camera tracking, i.e. re-identification, was evaluated by conducting queries for each of the 21 volunteers on our watchlist making the test journey between Stations A and B. The total number of search iterations (see Sect. 20.2.4) conducted over all 21 examples was 95. We were primarily interested in measuring the effectiveness of the three key ranking mechanisms: *relative feature ranking* [15]; *attribute-based re-ranking* [8, 9]; and *local space–time profiling*, in increasing the ranks of correct matches, as well as gauging the more holistic effectiveness of all six mechanisms (the above in

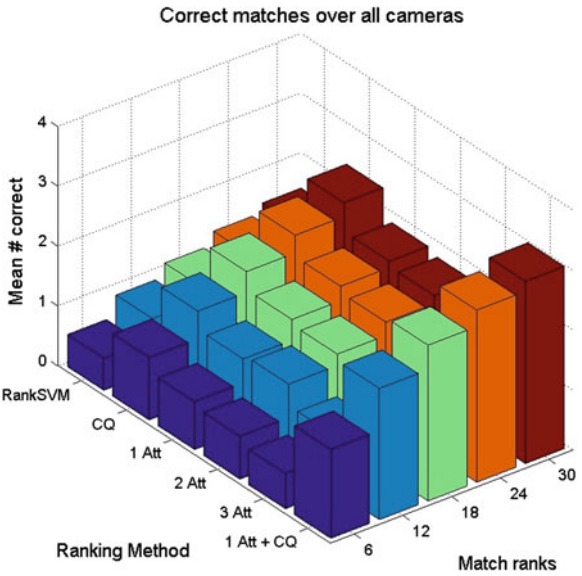

Fig. 20.8 The cumulative number of correct matches appearing in the top 6, 12, 18, 24 and 30 ranks, averaged over all search iterations and all camera views. The Convergent Querying (CQ) filter doubled the average number of correct matches in the first 6 ranks over the RankSVM model alone, from around 0.5 to 1. Selecting a single attribute was more beneficial than two or three, improving on the RankSVM model. Overall, a single attribute combined with CQ demonstrated the greatest improvement of around 200 % over the RankSVM model

addition to *matching by tracklets*; *global space–time profiling*; and *machine-guided data mining*) in tracking the targets across the multi-camera network.

We measured two criteria: (1) the *number of correct matches* in the first 6, 12, 18, 24 and 30 ranks after any given search iteration, averaged over all 95 search iterations, indicating how quickly a user will likely find the target amongst the candidates; and (2) *overall re-identification rates* in terms of the average percentage of cameras targets were successfully re-identified in, indicating tracking success through the environment overall. The exact querying procedure adhered to the iterative piecewise search strategy described in Sect. 20.2.4.

Number of Correct Matches

Figure 20.8 shows the cumulative number of correct matches that appeared in the top 6, 12, 18, 24 and 30 ranks viewed by a user, averaged overall search iterations and all camera views.

Using the RankSVM model alone [2, 6], the average number of correct matches in the first six ranks was around 0.5. Using the Convergent Querying (CQ) filter significantly improved upon the RankSVM model at all ranks, and approximately

Table 20.1 Effect of convergent querying filter

Stage of query	Mean candidate set size
Before CQ	392.9
After CQ	79.6

The mean reduction in candidate set size when employing the CQ filter, averaged over all query iterations and camera views. The effect was significant, resulting in an average 72.1 % reduction by more acutely focusing on the time period containing the target and removing the bulk of irrelevant candidates

doubled the number of correct matches in the first six ranks from around 0.5 to 1. The primary reason for this was its ability to remove the vast majority of incorrect matches by focusing on the appropriate time period. This is demonstrated by Table 20.1, showing that the reduction in the number of candidates invoked by the CQ filter was over 72 %, averaged over all query iterations. A single attribute model also showed around 50 % improvement, whereas adding a second and third attribute was less effective. However, the combination of a single attribute with the CQ filter provided the most significant improvement, with a 200 % increase over the RankSVM model. Consequently, local space–time profiling was critical for narrowing the search space more acutely and finding the right target more quickly amongst very large numbers of distractors. Combining this with a single attribute model provided an extra 50 % performance boost on average by providing an additional context for narrowing the search further.

Overall Re-identification Rates

Table 20.2 shows the percentage of watchlist targets that were explicitly detected by the system in each camera view. Apart from Cameras 4 and 5, detection rates were above 80 %. For Camera 5, the slightly larger distances to individuals resulted in slightly lower performance for the MCT person detector [3]. The profile views common in Camera 4 were responsible for lower person detection performance.

It is important to note that detection failure does *not* imply tracking failure, due to the facility for tagging visible associates of targets (refer to the piecewise search strategy in Sect. 20.2.4). Consequently, targets may still be tracked through camera views in which they may not be detected.

Table 20.3 shows the percentage of all six cameras that watchlist targets were tracked within on average; more specifically, the percentage of cameras within which users could tag matches that contained the target and which could be incorporated into a reconstruction, regardless of whether that target was *explicitly detected* by the system.

It can be seen that tracking coverage, i.e. re-identification, was very high, approaching 90 % over the entire network on average for both directions of movement. The result for the Station B to Station A journey was lessened due to the rele-

Table 20.2 Detection rates per camera

Camera	Target detection rate (%)
1	85.7
2	85.7
3	81
4	72.7
5	76.2
6	85.7

The percentage of watchlist targets explicitly detected in each camera view. Apart from Cameras 4 and 5, detection rates were above 80 %. For Camera 5, the slightly larger distances to individuals resulted in slightly lower performance for the MCT person detector [3]. The profile views common in Camera 4 were responsible for lower detection performance

Table 20.3 Overall tracking coverage

Direction of journey	Mean tracking coverage (%)
Station A to B	88
Station B to A	84.6

The average percentage of all six cameras within which a watchlist target could be found and incorporated into a reconstruction, whether or not explicitly detected by the system. Often targets were found for all cameras, with the few failures occurring due to: (a) unpredictable train times operating outside the global temporal profile, resulting in a loss of the target between stations; and (b) target occlusion due to crowding or moving outside the video frame

vance of Camera 4 for this journey (Sect. 20.4) and its corresponding lower detection reliability.

Failures were due to two main reasons. First, abnormal train waiting or transition times resulted in two watchlist targets being lost in between stations. These times fell outside the range of the learned global space–time profile, resulting in a faulty narrowing of the search space. In very large-scale multi-camera networks such as those spanning cities where different parts of the environment are connected by transportation links, this danger can be compounded by multiple unpredictable delays. This suggests that integrating live non-visual information, such as real-time train updates, should be exploited to override or dynamically update global space–time profiles in order to ensure correct focusing of the search as circumstances change. Second, lone targets could occasionally become occluded and thus remain undetected or untrackable by association, due to excessive crowding or moving outside the view of the camera. This highlights the value of careful camera placement and orientation. Nevertheless, occasional detection failure in some camera views was not a barrier to successful tracking since searches could be iteratively widened when required and the target successfully reacquired further along their trajectory.

Table 20.4 Module processing time per frame

Module	Processing time (%)
Person detector	39.8
Appearance descriptor generator	57.7
Other	2.5

The relative computational expense of key processing modules of the Extraction Engine

20.5.2 Capacity

A critical area of system scalability is the *speed* of the system in processing video data depending on the size of the multi-camera environment. Consequently, a major area of focus is the effective use of acceleration technologies such as GPU acceleration and multi-threading. Table 20.4 shows the relative time taken by two key processing modules of the Extraction Engine in the Generator Subsystem (Sect. 20.3) to process a single video frame. The most computationally expensive processing module, namely the Appearance Descriptor Generator, was re-implemented to employ GPU acceleration in order to conduct an initial exploration of this area. Additionally, multi-threading was employed to specifically exploit the computational capacity of multi-core processors.

In exploring the characteristics of processing capacity, four GPU and multi-threading configurations were evaluated in order to highlight the importance and effectiveness of applying acceleration technologies in working to achieve acceptable processing speeds: (1) single thread, no GPU acceleration; (2) single thread, GPU acceleration of the Appearance Descriptor Generator; (3) multi-threading of pipelines to parallelise the processing of individual camera inputs, no GPU acceleration; and (4) both GPU acceleration of the Appearance Descriptor Generator and multi-threading together. The hardware platform employed contained an Intel Core-i7 quad-core processor operating at 3.5 GHz, running Microsoft Windows 7 Professional with 16 GB of RAM and two Nvidia GTX-580 GPU devices.

Figure 20.9 shows the average time in seconds taken for each of the four acceleration configurations to process a frame for 2, 3, 4, 5 and 6 cameras simultaneously. It can be clearly seen that GPU accelerating the Appearance Descriptor Generator alone (requiring more than 50 % of the computational resources when unaccelerated) resulted in halving the processing time for a video frame. This amounted to the vast majority of the processing time for that component being eliminated. Significantly, it can also be seen that the use of multi-threading enabled six cameras to be processed on the same machine with negligible overhead, demonstrating that multi-threading, in addition to the distributed architecture design, facilitates scalability of the system to arbitrary numbers of cameras (i.e. the ability to process multiple video frames from different cameras simultaneously) by exploiting the multi-core architecture of off-the-shelf CPUs. A quad-core processor with hyper-threading technology is capable of processing eight cameras simultaneously with little slow-down; more cameras

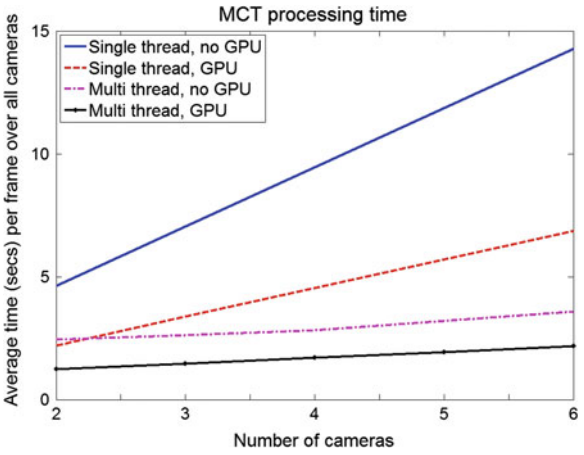

Fig. 20.9 Time taken in seconds for the MCT system to process a single frame across all camera streams for 2, 3, 4, 5 and 6 cameras in 4 different acceleration configurations. This demonstrated both the efficacy of employing multi-threading and GPU acceleration as well as the scalability of the system to arbitrary numbers of cameras. It can be seen that employing GPU acceleration dramatically improved the *time to process a single video frame*, and multi-threading facilitated the ability to *process frames from multiple cameras simultaneously*, demonstrating *linear scalability* of the system to larger camera networks

may be processed by simply adding another quad-core machine to provide another eight camera capability. Future processors with greater core numbers promise to efficiently increase scalability yet further.

20.5.3 Accessibility

The quantity of metadata generated by the system is strongly correlated with the size of the multi-camera environment, influencing the speed and responsiveness of the user interface in the course of a query being conducted. As such, this is a critical factor where scalability to typical real-world scenarios is concerned. Here, we investigate two key areas determining accessibility: (1) *query time versus database size*, relating system usability with the quantity of data processed; and (2) *local versus remote access*, comparing the speed of querying when running the User Interface Client locally and remotely in three different network configurations.

Query Time Versus Database Size

Open-world scenarios will typically present arbitrarily large numbers of individuals forming the search space of candidates during a query. The key factor in querying

Table 20.5 Length of video versus number of metadata match entries

Video length (min)	Number of tracklets	Number of match entries (millions)
10	8000	8.7
20	16000	26.9
40	31000	83.1

Relationship between the length of processed videos from six cameras, the number of extracted tracklets from those videos and the number of match entries in the corresponding metadata. A few minutes of video from all cameras yielded thousands of tracklets and millions of match entries in the database. Here we see that a 40 min segment of the six-camera data produced more than 30,000 tracklets and more than 83 million match entries

Table 20.6 Video length versus query time

Video length (min)	Mean query time (s)
10	82
20	124
40	284

The average time for the same query conducted three times for databases generated from 10, 20 and 40 min segments of the six-camera MCT trial dataset. While the 10 and 20 min segments resulted in acceptable times of around 1.5–2 min, the 40 min segment more than doubled the query time for the 20 min segment. The significant increase in query time with the quantity of video data processed highlighted a key bottleneck of the current system

time is the number of tracklets which have been generated for those individuals, and the size of a corresponding *match table* in the metadata which contains the matching results for appropriate global space–time filtered sets of tracklets between camera views. Table 20.5 shows the relationship between the number of tracklets and the corresponding number of match entries in the metadata match table for three different processed video segment lengths. It can be seen that 20 min of processed video from six cameras produced on the order of tens of thousands of tracklets and tens of millions of match entries in the database.

The question arises as to what effect this increase in the size of the database has on querying times. Table 20.6 shows the average time for the same query conducted three times over the same LAN connection for each of these three database sizes. While the 10 and 20 min segments resulted in acceptable times of around 1.5–2 min, the 40 min segment more than doubled the time over the 20 min segment. The significant increase in query time with the quantity of video data processed highlights a key bottleneck of the current system and a major focus on improving the scheme for metadata storage and access in working towards a deployable system.

Local Versus Remote Access

Table 20.7 shows the difference in query time for the same query conducted on the same metadata database accessed: (a) locally on the same machine as the Query Engine Server and SQL Metadata database; (b) remotely on a 1 Gbps local area

Table 20.7 Network access environment versus query time.

Query environment	Mean query time (s)
Local	97
LAN	125
Internet	181

Comparison of a typical query involving two feedback iterations for local, remote LAN (1 Gbps) and remote Internet (1 Mbps upload) access to the web server. Using the system over the internet with a very modest upload bandwidth resulted in almost doubling the query time over local access. A dedicated server with sufficient bandwidth would alleviate this drawback

network connected to the machine hosting the Query Engine Server and SQL Metadata database; and (c) remotely from the Internet with a server-side upload speed of approximately 1 Mbps. The query was conducted on metadata generated from a 20 min segment, and involved two search iterations examining and tagging appropriate candidates.

It can be seen that the same query took nearly twice as long over the Internet as compared to locally. The main slow-downs occurred in two places: (a) when retrieving either initial or updated candidate match lists, requiring the transmission of image data and bounding box metadata; and (b) when browsing the candidate tabs, again requiring the transmission of both image thumbnails and bounding boxes. This is a function of server-side upload bandwidth which in this case was very modest; a dedicated server offering higher bandwidth would result in lower delays and faster response times, important for open-world scenarios where highly crowded environments will typically result in larger numbers of candidates being returned after each query iteration.

20.6 Findings and Analysis

In this work, we presented a case study on how to engineer a multi-camera tracking system capable of coping with re-identification requirements in large scale, diverse open-world environments. In such environments where the number of cameras and the level of crowding are large, a key objective is to achieve *scalability* in terms of *associativity*, *capacity* and *accessibility*. Accordingly, we presented a prototype Multi-Camera Tracking (MCT) system comprising six key features: (1) *relative feature ranking* [2, 6, 15], which learns the best visual features for cross-camera matching; (2) *matching by tracklets*, for grouping individual detections of individuals into short intra-camera trajectories; (3) *global space–time profiling* [10], which models camera topologies and the physical motion constraints of crowds to significantly narrow the search space of candidates across camera views; (4) *machine-guided data mining*, for utilising human feedback as part of a *piecewise search strategy*; (5) *attribute-based re-ranking* [8, 9], for modelling high-level visual attributes such as colours and attire; and (6) *local space–time profiling*, to model the physical motion

constraints of individuals to narrow the search space of candidates within each camera view.

Our extensive evaluation shows that the MCT system is able to effectively aid users in quickly locating targets of interest, scaling well despite the highly crowded nature of the testing environment. It required 3 min on average to track each target through all cameras using the remote Web-based interface and exploiting the key features as part of a piecewise search strategy. This is in contrast to the significantly greater time it would require human observers to manually analyse video recordings.

It was observed that attribute-based re-ranking was on average effective in increasing the ranks of correct matches over the RankSVM model alone. However, employing more than one attribute at a time was generally not beneficial and often detrimental. Local space–time profiling was *extremely* effective under all circumstances and combining it with a single attribute always led to a significant increase in the ranks of relevant targets, with a tripling of the average number of correct matches in the first six ranks alone. These features are critical in enabling the MCT system to cope with the large search space induced by the data by focusing on the right subset of candidates, at the right place and at the right time.

Overall, out of the 21 watchlist individuals, all but two were trackable across both stations in the MCT trial dataset. The two exceptions were lost on a single train journey. This was due to the train time falling outside the learned global space–time profile. This emphasises the utility of employing non-visual external information sources such as real-time train updates to modify global space–time profiles on-the-fly. This would permit such profiles to be tighter and more relevant over time, making them more consistently effective in narrowing the search space. This can be a critical factor in very large open-world scenarios where different parts of the multi-camera network may be connected by unpredictable and highly variable transport links.

Our testing of system speed shows that employing GPU acceleration for the most computationally intensive component resulted in a 50 % reduction in computation time per frame. Furthermore, employing multi-threading on a quad-core CPU with multi-threading enabled all six cameras of the MCT trial dataset to be processed simultaneously with negligible slow-down. This suggests that, in conjunction with the modular distributed nature of the system architecture design, the processing capacity of the system is *linearly scalable* to an arbitrary number of cameras by adding more CPUs to the system architecture (e.g. another machine on the network). Furthermore, by focusing effort on optimising each processing component of the Extraction Engine, a real-time frame rate per camera is likely achievable.

The most significant bottleneck of the entire MCT system was found to be metadata storage. Using an off-the-shelf SQL installation and basic tables, stored metadata was found to become prohibitively large over time. Querying metadata from video data longer than 20 min would result in long waiting times for the Query Engine Server to return the relevant results. Given the typical number of cameras in a highly crowded open-world scenario, this highlights the criticality of designing an appropriate storage scheme to store data more efficiently, reduce waiting times during a query and improve accessibility to metadata covering longer periods of time.

It is clear that there is great promise for the realisation of a scalable, highly effective and deployable computer vision-based multi-camera tracking and re-identification tool for assisting human operators in analysing large quantities of multi-camera video data from arbitrarily large camera networks spanning large spaces across cities. In building the MCT system, we have identified three areas worthy of further investigation. Firstl, integration with non-visual intelligence such as real-time transportation timetables (e.g. flights, trains and buses) is critical for dynamically managing global space–time profiles and ensuring that the search space is always narrowed in a contextually appropriate manner. Second, careful optimisation of individual processing components is required, which also involves a proper mediation between multithreading and GPU resources to best harness availability in each machine comprising the distributed MCT system network. Finally, an optimised method for metadata storage is required for quick and easy accessibility regardless of the quantity being produced.

Acknowledgments We thank Lukasz Zalewski, Tao Xiang, Robert Koger, Tim Hospedales, Ryan Layne, Chen Change Loy and Richard Howarth of Vision Semantics and Queen Mary University of London who contributed to this work; Colin Lewis, Gari Owen and Andrew Powell of the UK MOD SA(SD) who made this work possible; Zsolt Husz, Antony Waldock, Edward Campbell and Paul Zanelli of BAE Systems who collaborated on this work; and Toby Nortcliffe of the UK Home Office CAST who assisted in setting up the trial environment and data capture.

References

1. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011)
2. Chapelle, O., Keerthi, S.: Efficient algorithms for ranking with SVMs. Inf. Retrieval **13**(3): 201–215 (2010)
3. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
4. Gheissari, N., Sebastian, T., Hartley, R.: Person reidentification using spatiotemporal appearance. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1528–1535 (2006)
5. Hahnel, M., Klunder, D., Kraiss, K.F.: Color and texture features for person recognition. In: IEEE International Joint Conference on Neural Networks, vol. 1, pp. 647–652 (2004)
6. Joachims, T.: Optimizing search engines using clickthrough data. In: Knowledge Discovery and Data Mining, pp. 133–142 (2010)
7. Kuhn, H.: The hungarian method for the assignment problem. Naval Res. Logist. Quarterly **2**, 83–97 (1955)
8. Layne, R., Hospedales, T., Gong, S.: Person re-identification by attributes. In: British Machine Vision Conference, Guildford, UK (2012)
9. Layne, R., Hospedales, T., Gong, S.: Towards person identification and re-identification with attributes. In: European Conference on Computer Vision, First International Workshop on Re-Identification. Firenze, Italy (2012)
10. Loy, C.C., Xiang, T., Gong, S.: Multi-camera activity correlation analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1988–1995 (2009)

11. Loy, C.C., Xiang, T., Gong, S.: Time-delayed correlation analysis for multi-camera activity understanding. Int. J. Comput. Vis. **90**(1), 106–129 (2010)
12. Madden, C., Cheng, E., Piccardi, M.: Tracking people across disjoint camera views by an illumination-tolerant appearance representation. Mach. Vis. Appl. **18**(3), 233–247 (2007)
13. Munkres, J.: Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**(1), 32–38 (1957)
14. Prosser, B., Gong, S., Xiang, T.: Multi-camera matching under illumination change over time. In: European Conference on Computer Vision, Workshop on Multi-camera and Multi-model Sensor Fusion (2008)
15. Prosser, B., Zheng, W., Gong, S., Xiang, T.: Person re-identification by support vector ranking. In: British Machine Vision Conference, Aberystwyth, UK (2010)
16. Raja, Y., Gong, S.: Scaling up multi-camera tracking for real-world deployment. In: Proceedings of the SPIE Conference on Optics and Photonics for Counterterrorism, Crime Fighting and Defence, Edinburgh, UK (2012)
17. Raja, Y., Gong, S., Xiang, T.: Multi-source data inference for object association. In: IMA Conference on Mathematics in Defence, Shrivenham, UK (2011)
18. Schmid, C.: Constructing models for content-based image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 30–45 (2001)
19. UK Home Office: i-LIDS dataset: Multiple camera tracking scenario. http://scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technology/i-lids/ (2010)
20. Wang, H., Suter, D., Schindler, K.: Effective appearance model and similarity measure for particle filtering and visual tracking. In: European Conference on Computer Vision, pp. 606–618, Graz, Austria (2006)
21. Zheng, W., Gong, S., Xiang, T.: Person re-identification by probabilistic relative distance comparison. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 649–656, Colorado Springs, USA (2011)
22. Zheng, W., Gong, S., Xiang, T.: Re-identification by relative distance comparison. IEEE Trans. Pattern Anal. Mach. Intell. **35**(3), 653–668 (2013)