A Framework for Distributed Human Tracking

Konstantinos Bitsakos*, Dimitrios Tsoumakos*, Yiannis Aloimonos* and Nick Roussopoulos*
kbits@cs.umd.edu dtsouma@cs.umd.edu yiannis@cfar.umd.edu and nick@cs.umd.edu
*Department of Computer Science
University of Maryland,College Park, MD 20742

***Abstract*— Today, more than ever, monitoring and surveillance systems play an important role in many aspects of our lives. Technology plays a vital role in our efforts to create, store and analyze vast amounts of data for both security and commercial purposes. In this paper, we propose an application which combines research performed in computer networks, multimedia databases and computer vision. We consider the problem where a number of networks are interconnected. Each of the individual nodes (networks) are collecting, processing and storing data from several sensors (cameras). Specifically, we emphasize on how the data (images) are processed by the individual nodes and how the information is transmitted, so that queries involving multiple nodes can be answered. During this process, we also identify several challenges related to sharing voluminous content provided by visual surveillance devices.**

Keywords: Peer-to-Peer, Human Tracking, Distributed Applications

I. Introduction

In our times, more than ever, technology is called upon to play a big role in providing crucial security and public safety information. Vast amounts of data and the immense rates at which they are collected make human-controlled systems increasingly error-prone. An ever-growing number of devices, ranging from photo and video cameras, microphones, etc to wireless sensors that measure pressure, temperature, weight, etc, collect data. In many public, military and commercial locations, these devices operate on an everyday basis monitoring human activity, environmental conditions, object positions, etc. In most cases, the answers to interesting and complex questions cannot be directly extracted from the collected data. Sophisticated and often time consuming operations need to be applied on them before some important information can be derived.

As an example, imagine a large area such as an airport, mall, business or government facility, etc. These locations comprise of large numbers of rooms occupied by many people, both visitors and workers. Such facilities are often monitored by a number of surveillance cameras. In practice, the output of the cameras is being monitored in real time by a small number of security personnel and/or stored in analog tapes. There are various disadvantages in this system: First, the number of cameras is usually larger than the observing staff, which makes real-time combination of visual information hard. Second, storing video in such devices makes off-line querying very slow. We believe that such a system could potentially operate faster, more accurately and in a more automated manner. In order to achieve that, three basic components must be added or improved:

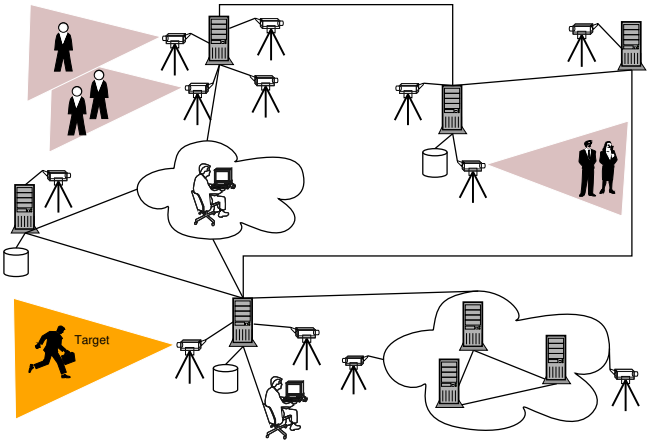

Fig. 1. Graphical description of our motivating application

- Operation on digital data
- Fast and accurate object tracking
- Cooperative network operation

Having the video/image data in digital format has many advantages: First, it can be stored and retrieved in large volumes by fast, off-the-shelf technology. Second, it can be directly analyzed using existing and new image processing tools. Third, it can be efficiently compressed. The extended use of digital cameras together with the increase in cheap available storage makes this extension easy to deploy.

The second important addition requires a tool that

enables the extraction, within multiple frames, of interesting objects. In other words, given a series of images, we require a process that can identify "interesting" properties in them, such as individuals, moving objects, etc. As we describe next, such tools exist although they work in isolation.

Third and equally important is the ability of such a system to identify and automatically monitor its target(s) across multiple observation sites controlled by different nodes. Specifically, we want to be able to identify the position or movements of a target as time progresses.

Figure 1 describes our motivating application pictorially. The whole facility is divided into a number of observing sites/rooms. Interconnected nodes are responsible for monitoring one or more of those sites through digital cameras. Each node can either be a single PC or a PC cluster. Security personnel monitors some of the cameras' outputs in real time to identify tracking targets. Visual data is stored at various nodes in the network in order to be queried on or off-line.

In this paper we describe an interesting application that utilizes P2P technology in order to store, query and identify interesting patterns collected by a group of cameras. Specifically, we propose a system that tracks humans across multiple cameras, with an ability to do on or off-line processing. During this process, we identify several challenges related to sharing voluminous content provided by visual surveillance devices.

II. RELATED WORK

A. Multimedia Databases

The area of Multimedia Databases has produced a lot of work on storing, organizing and querying image and video objects. We describe some representative examples in this part of the section. A more detailed presentation can be found in [1].

Many commercial RDBMSs (e.g., Oracle 10g, Informix, DB2, etc) have been extended to manage multimedia content. In most of these systems, images are stored as binary large objects (*BLOB*) and querying can be as complex as similarity between images on their color distribution or texture.

In Mirror [2], images are indexed by manually provided annotations and their respective URLs. The system also supports feature extraction daemons which operate on images (e.g., color histograms, texture algorithms) for additional querying abilities.

DISIMA [3] is an image database system that enables content-based querying, such as identifying all images where a person is present as well as define a visual querying language. Modeling of the images is based on semi-automatic building of hierarchies of themes for images and objects inside them.

B. Tracking Systems

The area of vision-based tracking is very active. In this section, we present work on multi-person whole-body real-time tracking that is related to our approach.

In the W^4 person-tracker [4], a single stationary camera is used to detect humans in an outdoor environment using gray-scale video. It was extended in W^4S [5] by using a pair of cameras in order to incorporate stereo information to the existing algorithm. In *Hydra* [6], a more sophisticated algorithm where the silhouettes are also taken into account is described.

EasyLiving [7] is an intelligent indoor environment that performs multi-person tracking. The system uses two color *stereo cameras*[1] to perform background subtraction and eventually locate humans, using a rough 3D model of the human body.

Q. Cai et al. in [8]–[10] and [11] describe a system for tracking moving humans in an indoor environment using an array of synchronized fixed cameras. They emphasize on the tracking portion of the system and they provide an algorithm for automatic camera switching when the tracked subject is moving out of the viewing boundaries of the current camera.

In [12], the authors present a framework for human tracking using a pair of cameras. They integrate a face detection module, performing human identification based on facial features such as hair and skin color, besides body shape and color or clothes. Furthermore, they introduce three different time-scales for identifying humans, short-range (frame to frame), medium-range (seconds, minutes) and long-range (hours, days). Depending on the time-scale, different cues are used to identify a human, for instance in the long-range scale only body shape and facial cues are used.

The work in [13] presents a system for video surveillance in an outdoor environment using a distributed network of cameras. It relies on a central authority receiving video and data from the sensors and combining them together. The final result is a higher level description of the events happening in the area. A number of extensions such as object tracking/classification and human activity recognition are also described.

[1]Stereo cameras provide disparity maps (i.e. depth information) along with the color images.

The wide-baseline, multiple-cameras tracker in [14] adopts a unified approach to image segmentation, person detection and tracking. The model and the ground plane position of a person are estimated iteratively. The advantage is that the system is robust against occlusions. On the other hand it processes frames at a smaller rate.

III. DESIRED CHARACTERISTICS

We would like our system to have the following characteristics:

1) It should be easily deployed and configurable. Thus, the network should be able to automatically handle the case of node arrivals, departures and failures. In addition, the tracking algorithm should be able to create a new representation for each new person entering the scene and identify previously seen ones.
2) The system should be robust, namely no single point of failure should exist. This requirement, combined with the rate at which data is produced, makes centralized storage prohibitive.
3) It should operate in "real-time", thus efficiency is required by all system modules. This affects the number of cameras connected to a node, the percentage of frames per camera processed, the complexity of the tracking algorithm, the selection of the network topology, the replication of data on other nodes, the lookup algorithm, the indexing scheme and the querying capabilities of our distributed database.
4) The system should be able to identify each person in a "consistent" way. Differences in camera characteristics and in the lighting of a single scene make this task very difficult. Moreover, changes in a person's appearance, posture or occlusions further complicate this process. Thus, the identification of a person should be based on consistent measures, such as height and facial features.
5) It should be able to locate the position in the world of each person with good accuracy. In order to perform this task, at least two cameras looking at the same scene should be present. Because of occlusions and point correspondence errors more than two cameras are usually required to get a good estimate of a person's 3D location [14].

Some of the aforementioned requirements are conflicting, e.g., the efficiency requirement dictates to keep each camera's frames in a single node, while in order to have a robust architecture we would like to have replicas of the frames in multiple nodes.

IV. OUR PROPOSAL

Our proposed solution comprises of three basic parts: Tracking algorithm, storage/indexing and querying/retrieval.

A. Tracking Algorithm

1) Camera's Setting and Self-Calibration: In the calibration process there are two different sets of parameters, internal and external, that need to be estimated. Various methods and toolkits (e.g., [15]–[17]) have been developed in order to estimate the *internal characteristics* of each camera (focal length, principal point, skew coefficient, radial and tangential distortion). In the case of multiple cameras working together, the *external parameters* of each camera should also be estimated. Those parameters describe the location and orientation of the camera in the 3D world. The manual estimation of the extrinsic parameters for an array of cameras is an arduous task. Elaborate techniques have been developed in the recent years that automate this process [18]. We should note that when the cameras are stationary, calibration need be performed only once at the initialization stage.

2) Image Segmentation and Background Extraction: The first part of every tracking system amounts to segmenting an image taken from a single camera into foreground and background regions. Many different approaches have been proposed, varying in complexity and execution time. Here we will describe a computationally simple and fast approach that takes into advantage the immobility of the cameras in order to perform background removal.

For each pixel i three values are kept, the minimum and maximum intensity (I^i_{min}, I^i_{max}) and the maximum intensity difference (D^i). Initially, the pixel is considered to be part of the foreground if $\|I^i_{max} - I^i\| > D^i$ or $\|I^i_{min} - I^i\| > D^i$, namely the current pixel intensity is much greater than the maximum intensity or much smaller than the minimum intensity. Due to illumination changes, this step alone is not sufficient, hence a number of morphological operations (erosion and dilation) are applied in order to improve the result. In the final step of the segmentation procedure, a connected components algorithm assigns labels to each foreground region.

3) Person Detection: Using a *cardboard* [19] or other model for the human body, we can find the probability of a particular region (or clusters of regions) to represent a human. This probabilistic framework can easily be extended using prior knowledge, such as the prior position of a human and its velocity. The final step of this process

is to apply a classifier, thus separating humans from non-human objects.

4) Human Identification: The simpler approach for human identification is to split each person into horizontal stripes and record the color characteristics of each one of them. In addition, we can record easily-obtainable body characteristics such as the width of each stripe, the global approximate height of a person and its skin hue. A more complex approach is to assume that at some point, we capture an image of the face of each person, for instance by zooming with a moving camera or by placing a camera at a strategic place (on a corridor, near the door of a room, etc). Using this image, we can add the person's facial features to the person's characteristics that we store for long-term identification.

In conclusion, the output of our tracking scheme is a representation of the identified humans along with their 3D positions in each processed frame. Both the human representations and their positions have the form of probability distributions over multiple attributes. For instance, we assume that each measured attribute and position is described by a gaussian probability function with mean value and standard deviation.

B. Why Peer-to-Peer?

Peer-to-Peer (hence P2P) computing represents the notion of sharing resources available at the edges of the Internet [20]. Its success can still be largely attributed to file-sharing applications (e.g., [21]–[23]), which enable users worldwide to exchange popular content. Nonetheless, the number of applications utilizing this technology is constantly increasing (e.g., web caching [24], instant messaging [25], email [26], etc).

The P2P paradigm is gaining an increasing amount of attention from both the academic and the large Internet community. It represents many plausible characteristics, which are particularly desirable for our application. P2P systems present a decentralized system model, without single points of failure or need of supervision. They allow for easy sharing of resources (content, indices, space, etc) as well as data availability inside the network. Equally important, such systems provide with increased robustness in the face of node arrivals, departures or unexpected failures.

Today, many popular P2P applications operate on *unstructured* networks. In these systems, peers connect in an ad-hoc fashion, the location of the documents is not controlled by the system and no guarantees for the success or the complexity of a search are offered. Distributed Hash Tables (*DHTs*), on the other hand, provide a more structured design, where each peer is responsible for a part of the objects. In DHTs, routing is performed in a more deterministic manner and requires $O(log(n))$ steps, n being the total number of nodes, to identify all relevant objects. Unstructured systems require no costly operations during peer insertions/deletions, whereas DHTs perform very efficient searches with a slight performance deterioration during dynamic operations.

The role that P2P can play in our system can be summarized in two parts: Data storage (i.e., who keeps what) and network reconfiguration. Organizing our nodes into a structured P2P system where each one is responsible for a set of human fingerprints has many advantages, especially if most queries are human-related. Because we assume that camera or node failures exist at a fairly low rate, we argue for a DHT-based overlay. Objects in this system are camera frames captured throughout the facility. Video data are locally stored, while the DHT is responsible for distributing the indices using a globally defined hash function. With this scheme, each node will store the data from the local cameras, as well as the locations of frames that contain certain humans.

Routing will be efficiently performed by the overlay, while we can take advantage of the resilience that DHTs exhibit when a node leaves the system or in the face of large workloads. Many proposed DHTs allow for automated index replication, so that more than a single node can identify the location of an object, providing with increased availability and load-balancing. The necessary procedures also exist that involve index relocation and redistribution when nodes wish to depart or enter the network.

Another issue concerns the relative positions of the cameras. This is a step that is added to the calibration process and is necessary at network initialization. It can be done either manually or automatically using the tracking mechanism. We assign to each camera a unique identifier as well as define its logical *neighbors*. A plausible camera identification scheme is to represent each one using the tuple $\{$CameraId,NodeId$\}$. Nodes are responsible to assign a unique identifier to each camera they control, while network addresses can be used to identify the nodes. We define the *neighbors* of camera C to be those cameras whose observation areas are the closest to C's. In effect, the neighbors of C are those that will capture a moving object that escapes C's view the soonest. Obviously, this can either be done manually (by simple observation), semi-automatically (by using the mounted tracking mechanism with a sample moving

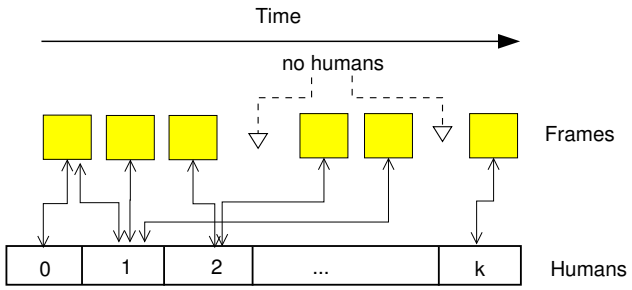

Fig. 2. Stored video from a single camera and its indexing. Frames are identified either by their capture time or the representations of the humans that were tracked inside them

human) or fully-automatically (by using a 3D description of the whole area). In any event, we require that the system is aware of at least 2-3 neighbors per camera to cope with possible malfunctions. As the cameras are memoryless devices, neighboring information should be stored at the node responsible for each camera.

C. Indexing

Another interesting question relates to the form of the stored data and its indexing, since this will directly affect our system's efficiency. Our data can be viewed as a 3-dimensional data set along the Time, Position, Human axes. Each frame is timestamped and we can assume without loss of generality, a synchronization between different cameras, so that no past frames have smaller timestamps than present/future ones. Each frame can also be described using the individuals it contains as well as their relative positions, as these are output by the tracking algorithm. In our design, we choose a 2-layer indexing over the captured frames, over time and human identifiers[2]. This is pictorially described in Figure 2, which shows what is stored at a local node serving a camera: Frames that do not contain any humans can be ignored. The rest of them are stored by time, while each identified person's representation is also stored. Each processed frame is then linked to the id(s) of the person(s) that were identified in it.

D. Querying

Given this representation, we now have to describe how the system would work in order to answer the following query: We identify person X in camera C_1 and want to track his movements till he leaves the mall. The node responsible for C_1 sends to the node(s) responsible for C_1's neighbors X's identifier. This process assumes that each camera can match X's global identifier to its local fingerprint. As soon as he leaves C_1's view, nearby nodes only will already have its identifier and will mark all frames that match it. This process continues till X leaves the premises. Our output will be all the relevant frames and their timestamps. More complex questions such as which individuals were closest to X can be answered with little extra processing. Obviously, given our indexing scheme, the off-line case is also easy to answer. By contacting the node responsible for a human's identifier, we can locate all nodes holding relevant frames. Querying on 3D positions would require a 2-step process. This is deliberately made, since we assume that the cameras provide with an approximate position.

[2]Identifiers are unique strings/numbers assigned to each different human representation registered in our network.

V. Discussion

In this position paper we presented an interesting application of sharing and querying digital video content using a Peer-to-Peer system. We consider this to be a particularly attractive and suitable application, based on a completely decentralized system model. We also believe that applications spanning computer networks, databases and computer vision will increase in number in the next few years, because of the evolution of both hardware and software. Furthermore, we anticipate that the areas will benefit a great deal by the introduction of novel applications requiring faster and more accurate algorithms. There exist a lot of interesting questions which we intend to pursue, both research and implementation oriented.

One question relates to the placement of the cameras. We believe that different placement strategies have a deep impact on the performance of the tracking algorithm. Another issue concerns the problem of corresponding human representations created by different cameras to a unique human identifier. We plan to experiment on the effect that different mapping functions as well as distance metrics between representations will have on our system.

In the network part, we intend to perform a thorough analysis on the existing DHTs and which one would be most suitable and adaptive to host our spatial data. A very interesting question that arises in this context is how to efficiently adapt the operation of a DHT (namely object insertion and routing), in the case that the data used for hashing is not an identifier but rather the probability distribution. This will allow for more expressive types of queries. Moreover, we intend to experiment with the effect that frame replication can have given intensive workloads.

In the database part, our indexing scheme purposefully left the 3D-location dimension untackled. We plan on investigating what will the effect of this choice be in the event of spatial queries. Finally, an interesting research question is how to create, index and query our data using a higher level description of video on our database. A sample query of this kind could ask for all children that were in the electronics store during the afternoon.

VI. Acknowledgments

This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number DAAD19-01-1-0494

References

[1] H. Kosch and M. Döller, "Multimedia database systems: Where are we now?," Special Session Talk to be given at the IASTED DBA-Conference in Innsbruck, February 2005.

[2] H. Blanken A. de Vries, M. van Doorn and Peter M. Apers, "The MIRROR MMDBMS architecture," in *VLDB*, 1999.

[3] B. Yao, "Building an Interoperable Distributed Image Database Management System," Tech. Rep. TR00-07, Department of Computing Science, University of Alberta, 2000.

[4] I. Haritaoglu, D. Harwood, and L. S. Davis, "w^4: Who? when? where? what? a real time system for detecting and tracking people," in *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*. 1998, p. 222, IEEE Computer Society.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis, "w^4s: A real-time system for detecting and tracking people in 2 1/2 d," in *Proceedings of the Fifth European Conference on Computer Vision*, 1998.

[6] I. Haritaoglu, D. Harwood, and L. S. Davis, "Hydra: Multiple people detection and tracking using silhouettes," in *International Conference on Image Analysis and Processing*, 1999, pp. 280–285.

[7] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," *Third IEEE International Workshop on Visual Surveillance*, July 2000.

[8] Q. Cai, A. Mitiche, and J. K. Aggarwal, "Tracking human motion in an indoor environment," in *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol. 1)-Volume 1*. 1995, p. 215, IEEE Computer Society.

[9] Q. Cai and J. K. Aggarwal, "Tracking human motion using multiple cameras," in *ICPR '96: Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276*. 1996, p. 68, IEEE Computer Society.

[10] Q. Cai and J. K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized Video Streams," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*. 1998, p. 356, IEEE Computer Society.

[11] Q. Cai and J. K. Aggarwal, "Tracking Human Motion in Structured Environments Using a Distributed-Camera System," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1241–1247, 1999.

[12] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 175–185, 2000.

[13] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456 – 1477, October 2001.

[14] A. Mittal and L. S. Davis, "M2 tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," in *International Journal of Computer Vision*. 2003, vol. 51(3), pp. 189–203, Kluwer Academic Publishers.

[15] Z. Zhang, "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," in *International Conference on Computer Vision*. 1999, IEEE Computer Society.

[16] J. Heikkil and O. Silvn, "A Four-step Camera Calibration Procedure with Implicit Image Correction," in *Computer Vision and Pattern Recognition*. 1997, pp. 1106–1112, IEEE Computer Society.

[17] "Caltech camera calibration toolbox for matlab," www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html.

[18] P. Baker and Y. Aloimonos, "Complete calibration of a multicamera network," in *In IEEE Workshop on Omnidirectional Vision*, 2000, pp. 134–144.

[19] S. Ju, M. Black, and Y. Yacoob, "Cardboard People: A Parameterized Model of Articulated Image Motion," in *International Conference on Face and Gesture Analysis*, 1996.

[20] Clay Shirky, "What Is P2P ... And What Isn't," *OpenP2P.com*, 2000.

[21] "http://www.napster.com," Napster website.

[22] "http://www.gnutella.com," Gnutella website.

[23] "http://www.kazaa.com," Kazaa website.

[24] S. Iyer, A. Rowstron, and P. Druschel, "Squirrel: A decentralized peer-to-peer web cache," in *PODC*, 2002.

[25] "http://web.icq.com/," ICQ web site.

[26] J. Kangasharju, K. Ross, and D. Turner, "Secure and Resilient Peer-to-Peer E-Mail: Design and Implementation," in *IEEE Intl Conf. on P2P Computing*, 2003.