CANDELA - STORAGE, ANALYSIS AND RETRIEVAL OF VIDEO CONTENT IN DISTRIBUTED SYSTEMS: REAL-TIME VIDEO SURVEILLANCE AND RETRIEVAL

E.G.T Jaspers[a] R.G.J. Wijnhoven[a] A.H.R. Albers[a] X. Desurmont[b] M. Barais[c] J. Hamaide[b] B. Lienard[b]

[a]Bosch Security Systems
Glaslaan 2
5616 LW, Eindhoven, The Netherlands

[b]Multitel
Parc Initialis - Avenue Copernic 1
7000, Mons, Belgium

[c]Vrije Universiteit Brussel
Pleinlaan 2
1050, Brussels, Belgium

Although many different types of technologies for information systems have evolved over the last decades (such as databases, video systems, the Internet and mobile telecommunication), the integration of these technologies is just in its infancy and has the potential to introduce "intelligent" systems. This paper describes the novelties of a video content analysis in a surveillance system, demonstrating the benefits for fast retrieval in huge video databases.

1. INTRODUCTION

The CANDELA project, which is part of the European ITEA program, focuses on the integration of video content analysis into a storage and retrieval system to unleash its full capabilities. When observing the advances in video processing, one can observe a trend in increasing content dependency. Although, techniques such a content-adaptive processing have become a commodity, the current focus of content analysis adds the notion of content awareness. Systems that start to "understand" the video signal are currently being introduced. This implies: computer vision algorithms that analyse the video and segment the content into objects; generation of metadata for large databases of video content and; the use of smart search devices. Processing is becoming more and more application-specific, making the technology less generically applicable. And even though standardization effort (MPEG-7) attempts the generalize the technology, it hardly addresses the application-specific requirements. For example, how can we detect and identify a shoplifter in a warehouse without manually observing hundreds of security cameras? Or, how can we retrieve information about a certain vehicle on a mobile device by applying abstract search queries on huge databases?

Even though the applications are very diverse, the CANDELA project has identified a general system architecture (see Figure 1). It comprises the integration of content analysis, storage and searching, providing a platform for a large range of applications. This paper explores a system implementation for the surveillance application to demonstrate the benefits for searching through large content databases. The corresponding demonstrator comprises one or more camera(s) to observe a secured area and applies real-time video content analysis. Moving objects are being tracked and classified. The analysis information, including properties such as size, speed and behavior are stored into a relational database. Simultaneously, a remote client demonstrates on-the-fly retrieval of the lively recorded content by higher-semantic search queries.

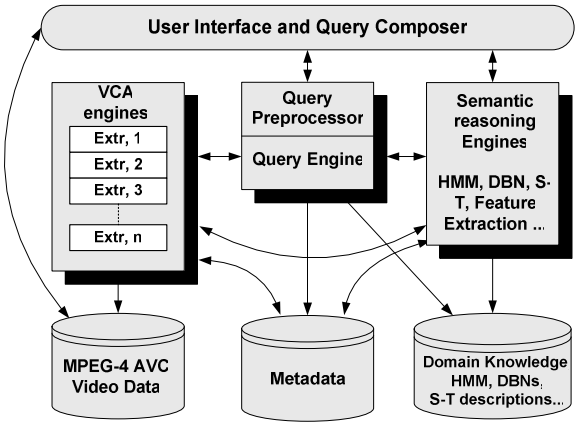

Fig. 1. General system architecture.

Section 2 discusses the novelties of the system in the field of content analysis. Subsequently, Section 3 explains how the analysis results are being exploited for search and retrieval. Section 4 finalizes with some conclusions.

2. CONTENT ANALYSIS

In literature, many algorithms for video content analysis (VCA) have been proposed [1]. These algorithm are for example able to detects specific object features, output the position of the objects on a frame-by-frame basis or classify objects using sophisticated object models. Typically, the descriptions from the analysis have a low abstraction level. For example, a moving object is described by a bounding box and the trajectory of the object. On a higher abstraction level, additional analysis of the metadata from the VCA is required to enable human understandable search queries. To obtain a higher semantic level of the metadata, *a priori* knowledge about the environment is required. Petkovic

0-7803-9332-5/05/$20.00 ©2005 IEEE

and Jonker [2] have already proposed a system that separates general VCA processing and additional knowledge about the application-specific environment. The following subsections will mainly discuss the analysis of metadata to raise the semantic level of search queries. Subsection 2.1 gives a general explanation of how the low-level metadata is being filtered. Subsequently, Subsection 2.2 discusses the transformation from pixel coordinates into real-world coordinates to enable intuitive search queries on size, speed and distance. Subsequently, Subsection 2.3 briefly explains how these real-world-size coordinates are being exploited to provide object classification. Subsection 2.4 will discuss the novelties of geographic querying for object trajectories.

2.1. Metadata analysis

Basically, the video content analysis (VCA) modules in the system analyze incoming video frames and segments the images into a static background and moving foreground objects. In addition, the objects are tracked over time, giving each object a unique object Id. Summarizing, for each frame and for each object the VCA module outputs the location, the bounding box and an Id. Figure 2a shows an example of this output over the lifetime of an object. However, this format is not is suitable for storage and retrieval. Therefore, the system contains separate Metadata Analysis modules (MDAs) that analyse the results from the VCA modules. The frame-based descriptions from the VCA modules are converted into object-based descriptions to remove redundancy and to provide a data format that is more suitable for retrieval. Notice that human reasoning is more object-oriented and hence search queries of this type are preferred.

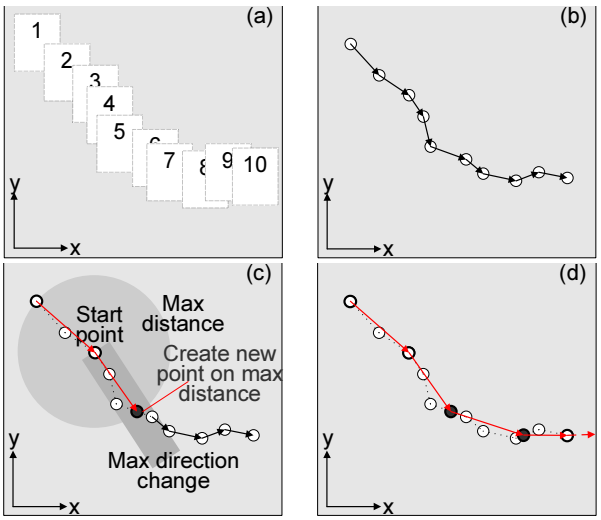

Fig. 2. Bounding box for each video frame (a), location for each video frame (b), filtering locations over time (c), original and filtered locations (d).

2.2. Perspective transformation

As mentioned before, each object is described by its location and the bounding box, denoted in the number of pixels. However, units of pixels are not desired due to the perspective distortion that is introduced by the 2-D image acquisition of the 3-D world. Note for example that the object size decreases when the object moves further away from the camera. Therefore, in order to use the location and bounding box information for intuitive search queries, the pixels coordinates are transformed to real-world size coordinates. This requires manually or automatic calibration of the camera [3] [4], i.e. the height of the camera and the distance to two points in the scene have to be determined (see Figure 3). Subsequently, perspective transformation can be applied to compute the real-world sizes of detected objects.

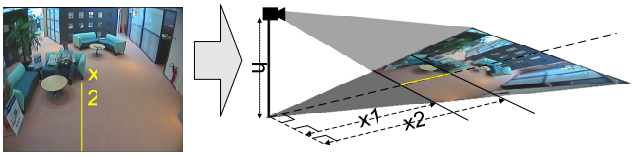

Fig. 3. Pixels to meters, using perspective transform.

2.3. Classification

After deriving the real-world sizes, the metadata processing applies straightforward classification by using *a priori* knowledge on the typical sizes of persons, cars, trucks, and smaller objects like rabbits or suitcases. Moreover, the trajectory coordinates are used to determine the speed of the objects. This enable a more accurate classification, since it is unlikely that for example a person walks at a speed of 40 km/h. Hence, the combination of these properties enables some mean of higher level reasoning.

2.4. Trajectory processing

From a user point of view, it is desirable to search through the video database without any additional expert knowledge. For the surveillance application, we have amongst others looked at search queries that are related to the trajectories (motion paths) of the objects [5][6]. Typically, such trajectory data is conveyed by the VCA algorithm on a frame-basis. However, this format is not suitable for efficient storage in a database (DB) nor for matching the trajectory with a query request. Let us now consider an objects bounding box at a certain time stamp. We can model these bounding boxes over time to retrieve an object trajectory in the spatio-temporal plane (X,Y,t) (see Figure 4). At the Graphical User Interface (GUI) side, a user is able to sketch a trajectory in the plane of the camera-view (X,Y) to search for all objects with a similar trajectory pattern. However, a fundamental problem is that all (X,Y,t) points of all objects in the database have to be examined to find the

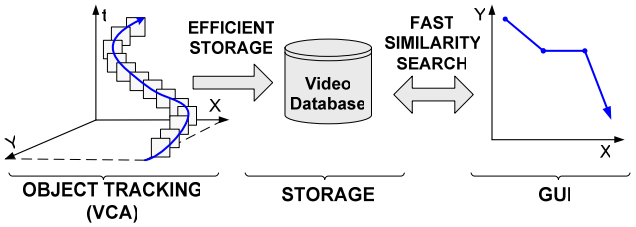

Fig. 4. Trajectory storage, search and retrieval overview.

correct matches. Therefore, a fast but still accurate method is required that reduces this computational burden. Three different challenges can be distinguished. Firstly, the definition of the *data representation* to model trajectory data for efficient indexing in the database. Secondly, which *database indexing structure* is used that provides fast searching, without scanning the whole database? Thirdly, we need to define the *similarity model*: Which metric is going to be used as a distance (quality) measure between trajectories? As a requirement, the chosen data structure should support different types of queries: retrieval of parts (sub-trajectories) of the trajectory data that match with the sketched query; retrieval of objects that crosses a user drawable line and; retrieval of objects in a selected interesting area. The following will describe the three challenges separately.

Data representation - The filtering step filters the spatio-temporal trajectory data into a spatial representation. It uses a combination of the Piecewise Linear Approximation (PLA) [7] and the Piecewise Aggregate Approximation (PAA) [8]. This results in a more compact description of the trajectory and enables fast search-queries. The filtering is applied for each new object that is detected in the scene. The location of the object is defined by the center of the bounding box (see the example in Figure 2a and 2b). When the filter decides that the location of the object at the current frame is relevant, it is stored into the database. The relevance is determined by two criteria. Firstly, the maximum distance between two filtered location points and secondly, the maximum deviation of the direction in which the object is moving. Both criteria are visualized in Figure 2c. When the filtering engine decides that any of these two criteria are exceeded, a new trajectory point is generated and stored. To compute the location of this new point, interpolating between the current and previous frame is applied. This filtering algorithm is continued until the object has disappeared. The results from the filtering are shown in Figure 2d.

Database indexing structure - After studying several database indexing structures that can store spatial data, R-tree variants [9] seem to fit our requirements best. Many geographical information systems (GIS), already extensively use R-trees for similar applications [10]. Spatial indexing is done by grouping the approximate trajectory representations into sub-trails and representing each of them with a minimum bounding rectangle (MBR). For our application,

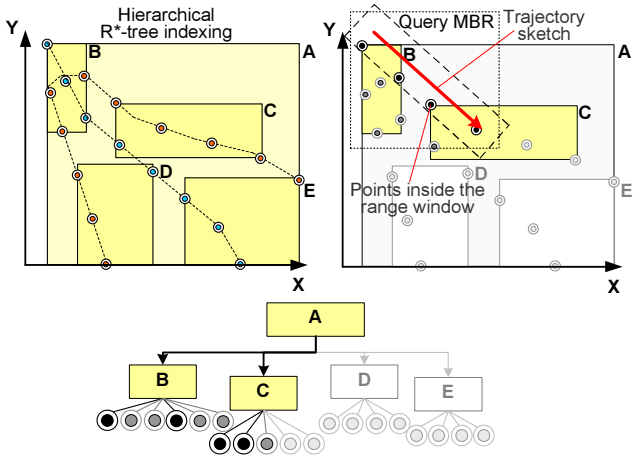

Fig. 5. Hierarchical storage of trajectory data.

a special variant of R-trees (R*-tree) is used to store the MBRs in an hierarchical way (see the left part of Figure 5). After the user sketches a trajectory to search for, a window is placed over the drawn line segments to define a distance range. This range defines the area of the trajectory points to search for. After this process, the hierarchical R*-tree filled with trajectory data is traversed for each query MBR (see Figure 5 for an example where two trajectories are present in the R-tree).

Similarity model - For matching the sketched line(s) with the trajectories in the database, two different metrics are adopted: the Euclidean point-to-line distance and the directional difference between the queried line and the stored line segments. If the sketch trajectory query contains more than one MBR, the matching is first applied to each MBR. To enable a Google-like ranking of the retrieved objects and to provide preliminary results for fast feedback to the user, the ranked results from each MBR query are combined into one global result set. Therefore, for each two MBR query result sets, a rank-join algorithm is executed that joins the trajectory points from the two sets. Finally, one large result set, ranked in the order of similarity, is left that contains all trajectories that match with the user sketch. The ranking phase in the rank-join algorithm is adaptive to the size of the MBR and its number of processed points [11].

3. RETRIEVAL

To search through the recorded video, a platform-independent client application is implemented. The graphical user interface (GUI) is divided into a query tab and a video-results tab. Figure 6 shows a screenshot of the GUI. Several types of queries are supported to retrieve potentially suspicious events in the video database. The different query types can be combined by the user to define a more sophisticated search query. The most important queries are: object type to search for e.g. persons; events to search for e.g.

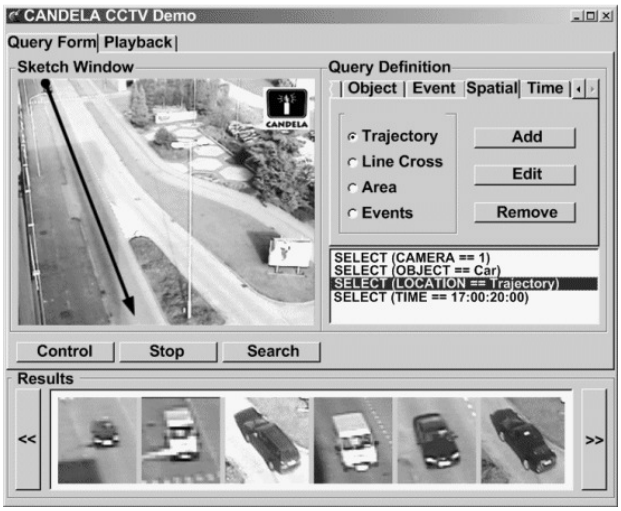

Fig. 6. Screenshot of the graphical user interface.

cars that stopped or persons that abandoned some luggage; time to search in as specific time interval; and specific positions and trajectories of the objects in the scene. Combining these query types, one can search for a car that drove from the parking lot to the gate, where it stopped between 15.00 and 16.00 hours. At the bottom of the figure, thumbnails of matching video parts are displayed. Object-trajectory results are displayed in the video window of the query tab. Subsequently, the selection of a thumbnail or trajectory invokes the video tab and starts playback of the corresponding video. Besides normal playback, also trick-play modes (e.g. fast/slow forward and backward, pause) are implemented in the demonstrator.

4. CONCLUSION

Many VCA solutions have already been proposed in the literature. However, for integration into a search and retrieval system the results for the analysis should be formatted into a suitable database structure. To raise the semantic level of the search queries, perspective transformation is applied to determine the real-world sizes. This enables low-complex classification and reasoning of object behavior and only requires relative low-complex VCA techniques.

Apart from raising the semantic level, filtering of the metadata is applied to provide efficient storage and fast retrieval. Frame-by-frame based video descriptions are transformed to a memory-efficient object-based description, which fits to the human way of reasoning.

The structure of the descriptions should enable efficient indexing. Therefore, the trajectory data is reduced and stored in a hierarchical R-tree structure, thereby decreasing the amount of data to search through.

All concepts are implemented on a demonstrator platform. A graphical user interface enable fast search and retrieval and shows the benefits of the metadata processing by applying intuitive semantic search queries.

This paper clearly demonstrates the benefits of video content analysis in a surveillance application. However, the performance of the retrieval strongly depends on the quality of the video descriptions. Testing and validation of the VCA results remains an open topic for research [12] and will be addressed in the successor of the CANDELA project.

5. REFERENCES

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *SMC-C*, vol. 34, no. 3, pp. 334–352, August 2004.

[2] M. Petkovic and W. Jonker, *Content-Based Video Retrieval, A Database Perspective*, Multimedia Systems and Applications, Vol. 25. Springer, 2003.

[3] S. Deng, Y. Yang, and X. Wang, "A calibration method using only one plane for 3d machine vision," in *16th International Symposium on Electronic Imaging, Storage and Retrieval Methods and Applications for Multimedia, SPIE 2004*, Jan. 2004.

[4] J.R. Renno, J. Orwell, and G.A. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," in *The 13th British Machine Vision Conference*, Sept. 2002, pp. 607 – 616.

[5] Y. Yanagisawa, J.I. Akahani, and T. Satoh, "Shape-based similarity query for trajectory of mobile objects," in *MDM '03: Proceedings of the 4th International Conference on Mobile Data Management*, 2003, vol. 2574, pp. 63–77.

[6] S. Satoh K. Aizawa, Y. Nakamura, "Sketchit: Basketball video retrieval using ball motion similarity," in *Advances in Multimedia Information Processing - PCM 2004: Proceedings of the 5th Pacific Rim Conference on Multimedia*, Tokyo, Japan, October 2004, vol. 3332, p. 256, Springer-Verlag GmbH.

[7] C.B. Shim and J.W. Chang, "Spatiotemporal compression techniques for moving point objects," in *Advances in Database Technology - EDBT 2004: Proceedings of the 9th International Conference on Extending Database Technology*. 2004, pp. 765–782, Springer-Verlag.

[8] E.J. Keogh and M.J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*. 2000, pp. 122–133, Springer-Verlag.

[9] Y. Manolopoulos *et al.*, "R-trees have grown everywhere," 2003.

[10] R.K.V. Kothuri, S. Ravada, and D. Abugov, "Quadtree and r-tree indexes in oracle spatial: a comparison using gis data," in *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2002, pp. 546–557, ACM Press.

[11] I.F. Ilyas, W.G. Aref, and A.K. Elmagarmid, "Joining ranked inputs in practice," in *VLDB*, 2002, pp. 950–961.

[12] X. Desurmont and R.G.J Wijnhoven *et al.*, "Performance evaluation of real-time video content analysis systems in the CANDELA project," in *Proc. of the SPIE - Real-Time Imaging IX*, Jan. 2005.