Author manuscript, published in IEEE International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 117-124, London, UK, June, 2008

BEHAVIOR AND EVENT DETECTION FOR ANNOTATION AND SURVEILLANCE

Zoltán Szlávik[1], Levente Kovács[1], László Havasi[1], Csaba Benedek[1], István Petrás[1], Ákos Utasi[2], Attila Licsár[2], László Czúni[2], Tamás Szirányi[1]

[1]{szlavik; levente.kovacs; havasi; bcsaba; petras; sziranyi}@sztaki.hu
MTA-SZTAKI, H-1111 Budapest Kende u 13-17. +36 1 2797300
[2]University of Pannonia, Dept. of Image Processing and Neurocomputing
H-8200 Veszprém, Egyetem u. 10, Hungary

ABSTRACT

Visual surveillance and activity analysis is an active research field of computer vision. As a result, there are several different algorithms produced for this purpose. To obtain more robust systems it is desirable to integrate the different algorithms. To achieve this goal, the paper presents results in automatic event detection in surveillance videos, and a distributed application framework for supporting these methods. Results in motion analysis for static and moving cameras, automatic fight detection, shadow segmentation, discovery of unusual motion patterns, indexing and retrieval will be presented. These applications perform real time, and are suitable for real life applications.

1. INTRODUCTION

Visual surveillance and activity analysis has attained great interest in the field of computer vision research [1][2]. Several algorithm libraries are available on-line (open-source or proprietary), however their integration into a complex system is hindered by the inhomogeneity of the implementation language, format, processing speed, etc. The result of this development is a flexible system for activity analysis. We provide a transparent and distributed architecture for easy integration of third party modules into a common framework to facilitate easier research collaboration and evaluation. The setup is hierarchical thus helping the scalability of the whole framework. The actual implementation integrates diverse algorithms forming a test-bed for unusual activity detection. Various complex surveillance related algorithms, such as analysis algorithm for static and moving cameras, automatic fight detection, shadow segmentation, discovery of unusual l motion patterns are integrated into this framework. In the case of detecting unusual motion occurrences, we refer to the term *unusual* in statistical sense.

In the paper a framework system is demonstrated for visual surveillance and activity analysis. The demonstrate how to complete a complex surveillance system by integrating video-processing algorithms with adaptive and self-learning filters for discovery unusual motion patterns and retrieval interfaces.

2. SYSTEM ARCHITECTURE

The architecture according to the current trend and software tools is as flexible as possible. The modules can be distributed over the network (either LAN or WAN); they are organized into a hierarchical structure. The structure can be separated into four main entities: a) the clients, b) the server (optionally including the web server) c) the communication interface embedded into the user module (see Fig. 2). Each component operates autonomously communicating through RPC requests over TCP/IP.

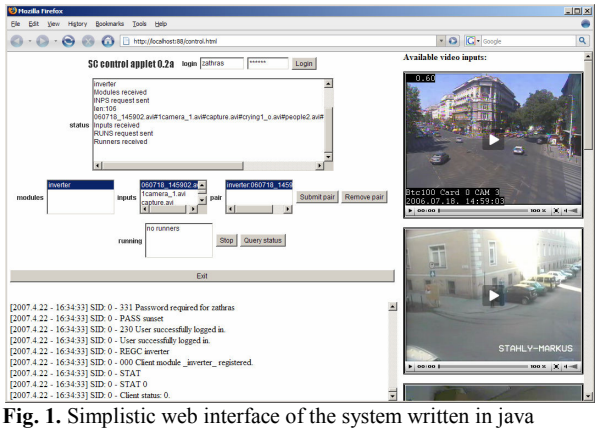

Fig. 1. Simplistic web interface of the system written in java

a) The web-based control interface of the server (Figure 1) gives the user transparent access to the control of the modules and can display their current status and output. It is written in Java and XHTML/ActionScript.

b) The server is the central element of the system; it sits on the top of the hierarchy. It delegates the tasks to the modules and coordinates the execution of the different modules: starts and stops modules and sets their parameters.

c) The communication interface seamlessly integrates into the user module. It is implemented as a C++ class. It requires as little changes to the user modules as possible and it is designed to be easily integrated into third parties' existing applications. Through the interface the modules can transparently exchange images frames and other data.
One advantage of the architecture is that there is no need to publicly provide any proprietary source code, yet it is still possible to integrate heterogeneous modules through TCP/IP. Third parties' IP rights can be easily protected as they can run their own modules on their own servers; they only need to incorporate the communication interface classes we provide.

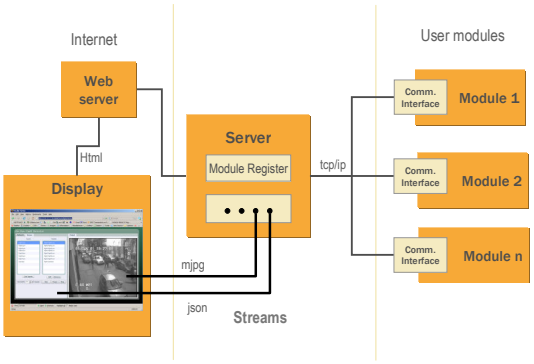

Fig. 2. The block diagram of the distributed system.

3. SYSTEM MODULES

As the number of installed CCTV cameras dramatically increases, the number of personnel watching over them ought to be also increased. As a result, the existing personnel will be in charge of the excess number of cameras. This eventually leads to performance loss. Thus, one of the most important goals of visual surveillance systems is to analyze the activity of the observed objects in order to detect anomalies, predict future behaviors, or predict potential unusual events before they occur.
Event detection modules implemented are mainly based on the analysis of the observed motion and can be used to focus operator's attention to unusual events and by means of this system'
There have been a lot of approaches to model the activity of dynamic scenes. Analysis of motion patterns is an effective approach for learning the observed activity. For the most of the time, objects in the scene do not move randomly. They usually follow well-defined motion patterns. Knowledge of usual motion patterns can be used to detect anomalous motion patterns of objects. Current systems mainly base their analysis of motion patterns on a predefined classification of tracked data [3] or of optical flow patterns [4].

3.1. Motion and shadow detection - Index parameters through background detection

Background subtraction is a key issue in several computer vision applications like video surveillance, aerial exploitation, traffic monitoring and control, or disaster protection. Although in practical systems, foreground extraction is usually only a preliminary filter used before applying high level event detector or analysis methods, it can directly provide useful information for content based video indexing. These descriptors can be considered as "middle level" event features. They are at a higher level than e.g. mpeg descriptors, but they provide just indicators and not an exact interpretation of the video content in contrast to e.g. fight detection or human identification.
For foreground detection in videos recorded by static cameras, we have used our previously introduced method [5]. This approach is an extension of the widely used Gaussian mixtures model [7] with and adaptive shadow detector, and an improved spatial foreground model supported by a color-texture fusion process.
Thereafter, the following features are automatically extracted are stored in the metadatabase.
Ratio of the area of dynamic and static scene parts: with this feature one can distinguish e.g. videos containing dense streets scenarios from surveillance shots of closed objects.
Another interesting issue is related to detecting changes in the background. In a usual surveillance application, only the moving objects are focused.

Fig.3 Examples of shadow and motion detection in real outdoor videos.

So, the background model should be adaptive only to produce less false foreground alarms. On the other hand, in video indexing the background changes give us useful content information as well. Two main cases can be distinguished here: *global illumination changes*, which results in a smooth or abrupt modification of the mean background values of all pixels and *background object changes* (e.g. an abandoned bag or a new parking car),

Fig. 4. Example of fight detection in a real video.

which can be detected as local changes in the indexes of the dominant Gaussian background components [7].

In the above way, periodicity effects in the background can be easily detected by minor modifications of the proposed model. They may provide efficient indicators of some events, like presence of traffic lamps with periodically changing red-yellow-green lights, opened and closed doors and windows or multi-state advertisements in the background.

In tracking applications, shadows are usually mentioned as very harmful effects, because they make difficult the extract silhouettes of the moving objects. However, detected shadows include much information about the illumination conditions in the scene. Based on [5], we automatically detected the shadow regions and examined the following factors. The darkness of shadow informs us if an outdoor shot was taken in sunlit, or in overcast weather. If multiple shadows are observable with different darkness then more light sources must be present in the scene. We can also consider the area ratios of the corresponding shadowed and foreground regions. That factor may help to detect the part of the day: shadows are smaller at noon and longer in the morning or in the late afternoon. As well, if we have more videos recorded by the same camera, the relative offset vector between shadows and foreground objects describes, when the shot was taken.

As for moving cameras, [6] model can be applied, before similar investigations to the static case.

3.2. Fight detection

Our new algorithm helps to ease the burden of focusing the valuable attention of security personnel. The algorithm detects fights and sends a signal when disorderly motion patterns are detected in the video stream. The tuning of this robust and effective algorithm is easy and mostly invariant to the characteristics of video (spatial resolution, refresh rate, view parameters etc.)

Automatic detection of events is a must in the leading video surveillance and video retrieval systems. Our algorithm provides a valuable index key for the database engine when searching for suspicious activities.

The algorithm was tested on several fight videos from publicly available surveillance databases [11][12] and on real videos also. Example shots of detection are shown in Figures 4 and 5.

Fig. 5. Example detections of "simulated" fights.

3.3. Unusual global motion detection

This method is based on the construction of average global motion histograms on video segments, learning the average motion directions, and creating alarm events in the case when unusual – in direction, in length, in location - motions occur. Such unusual events can be e.g. someone goes against the traffic in a one-way street, someone crosses a high traffic street where no crossing usually occurs, a traffic jam occurs on a street where the traffic's flow is usually constant, and so on.

The method is based on length- and time-based segmentation of local optical flow patterns, which is called a sample, then learning those samples. During the training period the samples are classified by a K-means algorithm into k classes. If the centers of these classes are closer than a threshold, they are combined. Also, an outlier-detection step is used to drop extreme samples/classes, which might be the result of bad detections. During the recognition/alarm phase the same sample extraction is performed on the incoming frames, and if they do not fit into one of the learned classes, an alarm is risen. Figures 6 and 7 show some examples.

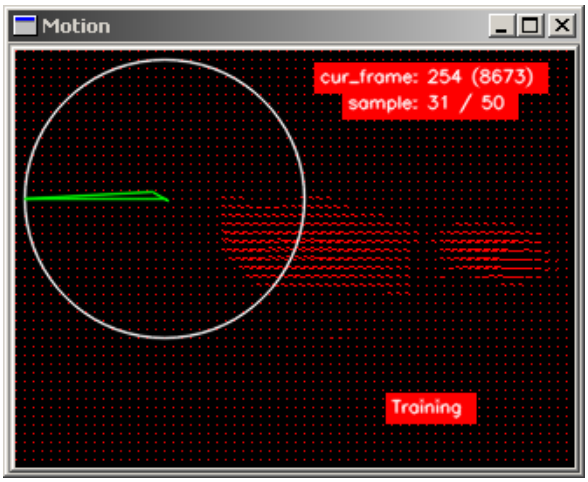

Fig. 6. A typical screen of a learning phase. In the circle the actual sample's directional histogram is shown, the background is the actual extracted optical flow field.

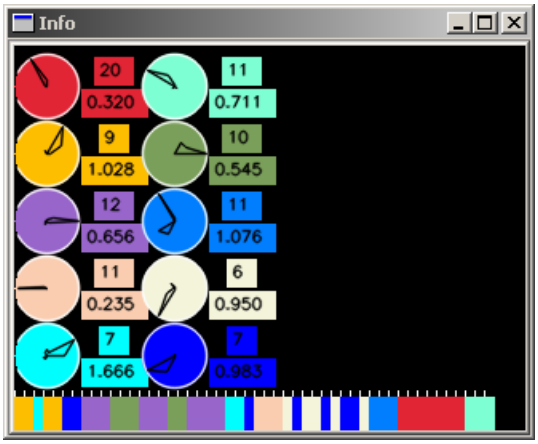

Fig. 7. A typical screen of an alarm phase, showing the learned directional histogram classes in the circles (color coded), and the time histogram in the bottom showing in which category the actual frames have been classified into.

3.4. Unusual local motion masking

This method's purpose is to learn the locally typical motions, and raise an alarm when an unusual local motion occurs (based on [9]). The goal is to identify the unusually moving areas and patterns and give a local mask identifying those regions. This way alarms can be risen based on some types of motions occurring over some selected areas, or in a global case, an unusually moving region can automatically be segmented and masked, and an alarm can be shown.

The base of this approach is a Stauffer-Grimson [7] background-foreground separation step, which uses the output of a local optical flow extraction module to generate the different layers (up to six). Then, a mean shift [8] segmentation step is performed to group similar, and closely (in space) occurring motions into the same classes. During the alarm phase, on every region – or a hand-selected region – motion patterns which do not fit into the such generated classes are signaled as being unusual.

Figure 8 shows a synthetic example for this approach, where multiple horizontally moving objects were generated, their motion patterns have been learnt, then an object moving in a crossing direction has been clearly masked.

Fig. 8. A synthetic example: top-left: input frame, top-right:typical motion patterns, color coded (horizontal two-directional motions, thus the two colors), bottom-left: a screen from the learning phase where masks show the moving objects, intensity corresponds to motion speed, bottom-right: a screen from the alarm phase where a diagonally moving object is marked as unusual and an alarm is raised.

3.5 Storage and search of events

All the modules of the unusual event detection framework produce alarm and module-output data in XML markup which is imported into a relational database with native markup language support. This provides us with the event data and details that can be later looked through and searches can be performed upon.

The main data that gets stored into the database are the id of the alarm module, the time of the occurrence, and description of the event, associated filter data for later processing, the id of the camera feed on which the event occurred, optional annotations that can manually be assigned to events, and a frame taken at the time of the event from the live camera feed. Figure 9 shows the search dialog where past events can be browsed, and searches can be performed based on the type of the event, on time constraints and on annotation data.

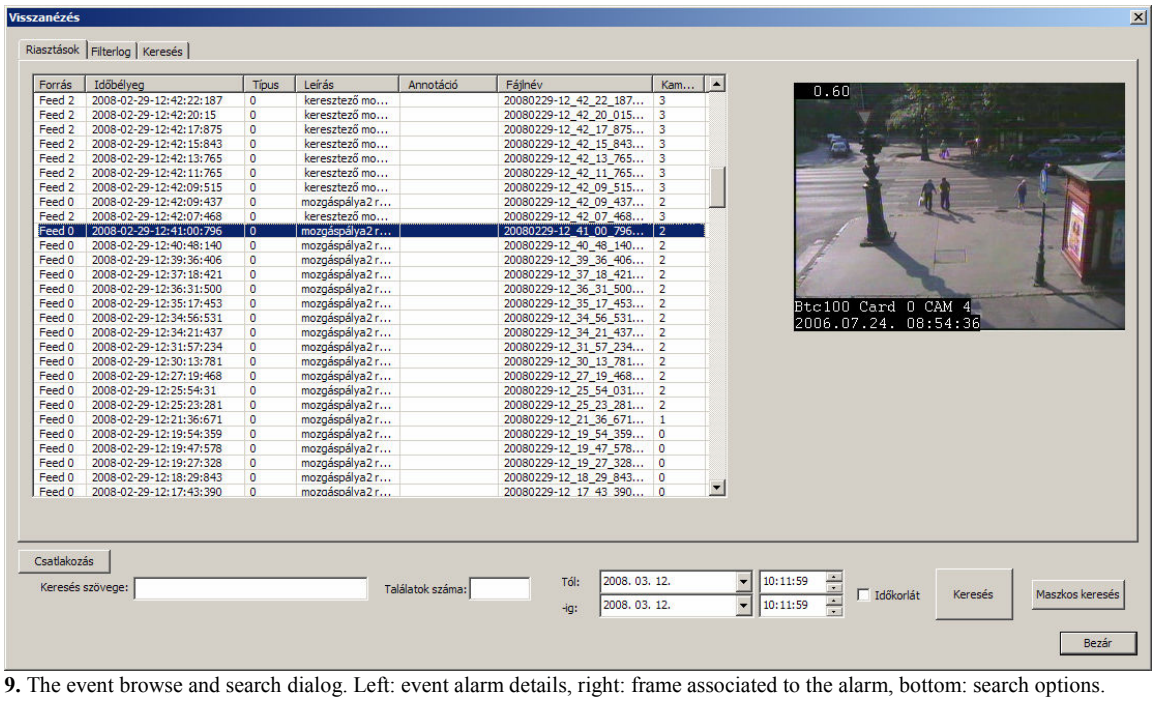

Fig. 9. The event browse and search dialog. Left: event alarm details, right: frame associated to the alarm, bottom: search options.

3.6. Indexing and retrieval of video data

For indexing and retrieval of frame and video data we created a searchable video database application, on which textual and content-based searches can be performed, and also an annotation application which can be used to assign annotations to archived video footage. After a video – with or without annotations – is imported into the database, several content based feature extractors are automatically run (i.e. average color, color structure, texture, motion, edge histogram, day/night, grass/sky, etc.) that will be indexed (by a modified BK-tree based approach [10]) and used when searches based on a model will be performed. Textual data, which comes from the annotations, is also stored into the database, thus textual and content based queries can both be performed.

Figures 10-13 contain some examples. Figure 10 shows an output of a textual search (in this case "football" in Hungarian), where each displayed result frame is in fact a representative frame of a respective video shot, whose annotation data contained the query text. Figure 11 shows the detail view of the database for a selected video shot, displaying the video information, the annotation, the shot itself. Figure 12 and 13 show examples for content based query results, in the first case we used an edge content based feature for searching for similar images, and in the second case we used a texture-based feature to search for similar images (the query and the result show representative frames from shots of a running contest video).

3.7 Adding new modules, monitoring

The system is architected and built to be highly modular and distributed. That is, by following our interface guidelines, third parties can build compatible modules from their algorithms, which can later be easily integrated into the system. The distributed architecture makes it possible to run the different modules on physically different networked machines, even from the developer's own machine. The current detector modules all have real-time performance, moreover, a state-of-the-art PC can run multiple modules on multiple camera feed sources concurrently.

Monitoring the currently running modules' output can be done by various ways. Currently the status of the modules can be viewed by a web interface (java or flash based, e.g. Figure 1.). But there are no real constraints for the implementation of a monitoring application, it just has to respect the internal communication protocols the modules and the controller server use.

4. CONCLUSIONS

A framework system is presented for the integration of advanced video-processing algorithms for motion and shadow detection and discovery of unusual behavior, indexing and retrieval interfaces into a next generation distributed video surveillance system.

Future work includes developing further event detection and feature extraction modules, publishing the protocol details for the inter-module and inter-server communication and the module interfaces, creating a user friendly web interface for the video database and making it public for web users to upload, index and search video contents. The final goal is to create a versatile and robust distributed system for video surveillance, event detection, automatic content-based indexing and retrieval.

5. ACKNOWLEDGEMENTS

The authors kindly acknowledge the financial support received from MUSCLE FP6 Network of Excellence and from the Economic Competitiveness Operative Programme of Hungary, grant Nr. GVOP-3.1.1.- 2004-05-0388/3.0.

6. REFERENCES

[1] W. Hu et al., "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 34, no. 3, 2004, pp. 334–352.

[2] Hilary Buxton: Learning and understanding dynamic scene activity: a review. Image Vision Comput., vol. 21, no. 1: pp: 125-136, 2003.

[3] C. Stauffer, W. Eric L. Grimson: Learning patterns of activity using real-time tracking, IEEE Trans. PAMI, vol. 22(8), pp. 747-757, 2000.

[4] E. L. Andrade, R. B. Fisher, S. Blunsden: Detection of emergency events in crowded scenes, Proc. of IEE Int. Symp. on Imaging for Crime Detection and Prevention, pp. 528-533, 2006.

[5] Cs. Benedek and T. Szirányi: "Bayesian Foreground and Shadow Detection in Uncertain Frame Rate Surveillance Videos", IEEE Trans. on Image Processing, vol. 17, no. 4, pp. 608-621, 2008.

[6] Cs. Benedek, T. Szirányi, Z. Kato and J. Zerubia: "A Multi-Layer MRF Model for Object-Motion Detection in Unregistered Airborne Image-Pairs," IEEE International Conference on Image Processing (ICIP), vol. 6, pp. 141-144, 2007.

[7] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real time tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 22–29, 1999.

[8] D. Comaniciu, P. Meer: "Mean Shift: A Robust Approach Toward Feature Space Analysis", IEEE Tr. on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002.

[9] Á. Utasi, L. Czúni, "Reducing the foreground aperture problem in mixture of Gaussians based motion detection," In Proc. of 6th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, EC-SIPMCS 2007.

[10] W. Burkhard, R. Keller: "Some approaches to best-match file searching", Communications of the ACM, vol. 16, no. 4, pp. 230-236, 1973.

[11] EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[12] Y. Dedeoğlu, B. U. Töreyin, U. Güdükbay, A., Enis Çetin, Silhouette-based Method for Object Classification and Human Action Recognition in Video, Int. Workshop on Human-Computer Interaction, (in conjunction with ECCV 2006). LNCS, vol. 3979, pp. 64-77, 2006.

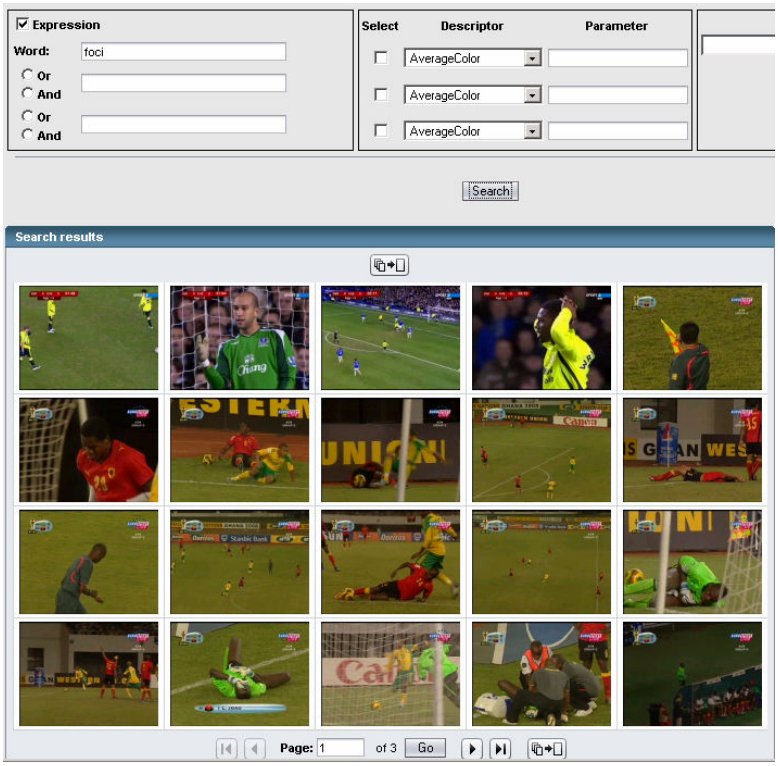

Fig. 10. Search by text: results of textual search for football sequences.

Fig. 11. Details of a shot from the database.

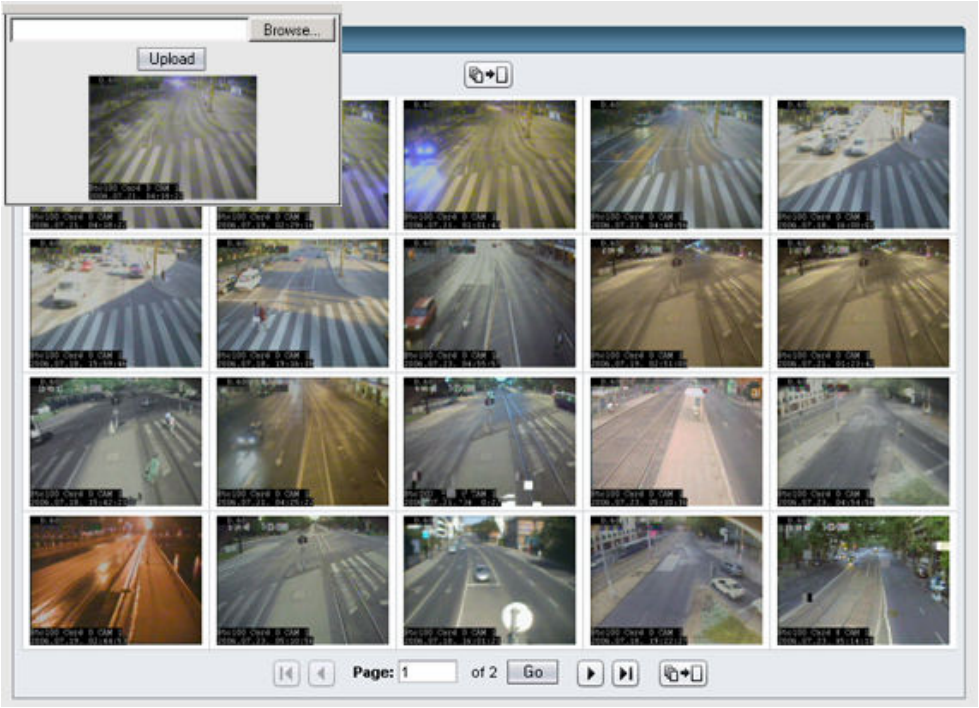

Fig. 12. Content based search: top-left: model query image, and search results based on edge histogram similarity.

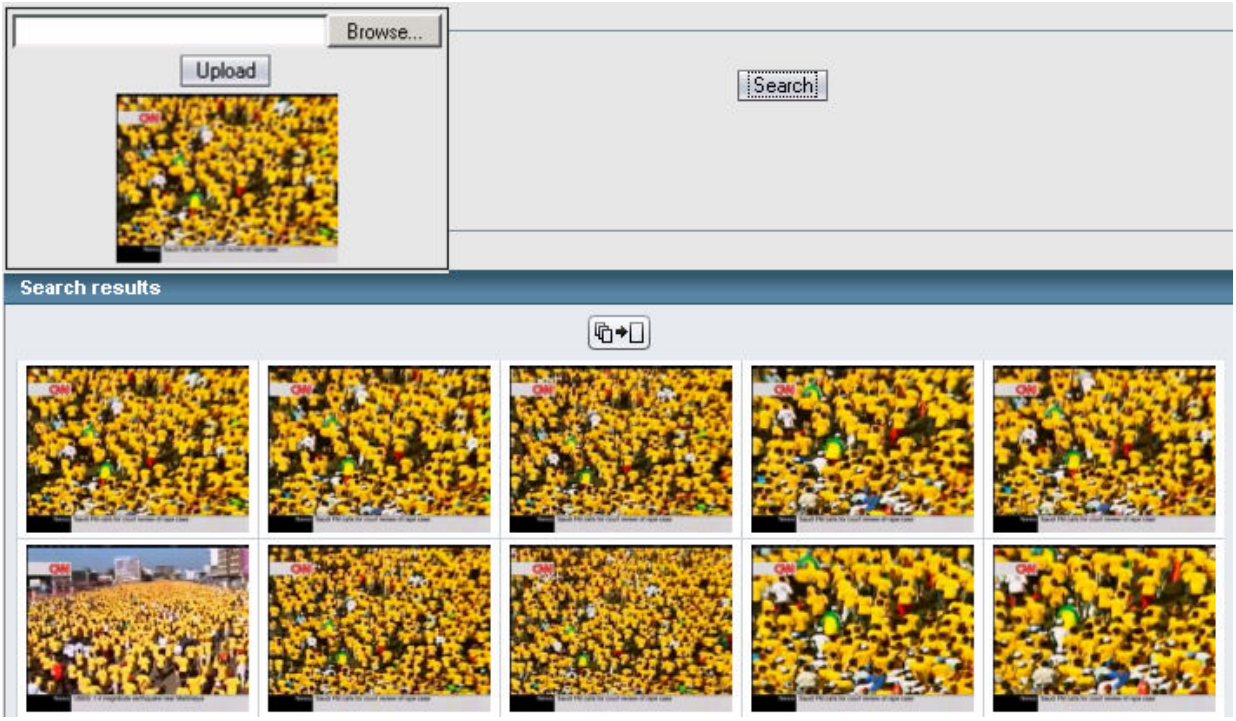

Fig. 13. Content based search: top-left: query image, and the retrieved results (each being a representative-frame of a video shot).