A GENERIC VIDEO ANALYSIS AND SEGMENTATION SYSTEM

Ebroul Izquierdo$^{(*)}$, Jianhui Xia$^{(*)}$ and Roland Mech$^{(**)}$

$^{(*)}$ Department of Electronic Engineering
Queen Mary, University of London
London E1 4NS, United Kingdom

$^{(**)}$Communication Technology and Information
Processing Institute,
University Hannover, Hanover, Germany

ABSTRACT

A generic video analysis system for supervised and unsupervised segmentation is described. The idea behind the presented concept is to integrate different advanced segmentation techniques to obtain a robust, efficient and modular segmentation system for natural video and still images. The system entails several independent modules. Each one of these modules encapsulates a complete video processing technique The intermediate results obtained from each single module are merged and further processed by a set of intelligent rules to achieve a highly accurate final segmentation. The modular structure of the system allows it to be extended continuously and with ease by adding new independent modules. The intermediate segmentation results of newly added modules are linked to the other system results via the rule processor. A user friendly graphical interface (GUI) is also provided. The functionality of the GUI is twofold: it serves as input interface to pass processing parameters to the system and as semi-automatic segmentation tool for user interaction and manually refinement of automatically generated segmentation masks. Selected results obtained with the current version of the video analysis system are reported.

1. INTRODUCTION

The European-Algorithmic Group COST 211 [1] is a forum and research network on video analysis. During the 5th framework, this forum has focused on video segmentation based on a test model, called the COST 211 Analysis Model (AM). The adopted approach has been to investigate, compare and optimize algorithms for image and video analysis in an experimental framework. In contrast to research conducted on an individual basis, this approach enables the comparison of competing technology based on a well-defined testbed and under agreed experimental conditions and performance measures. The purpose of a test model in COST - very similar to that in MPEG - is to describe completely defined Common Core algorithms, such that collaborative experiments performed by multiple independent parties can produce identical results and will allow the conduction of “Core Experiments'” under controlled conditions in a common environment. A test model specifies the formats for the input and the output. It fully specifies the algorithm for image analysis and the criteria to judge the results.

The COST 211 meeting in Ankara, Turkey, in October 1996 witnessed the definition of the 1st AM, which consists of a full description of tools and algorithms for automatic and semi-automatic image sequence segmentation, object detection, extraction and tracking. The AM was then refined by partners co-operating in this forum and by the year 2000 it progressed to its 5th Version. Concurrently with the full description of the AM framework, a software implementation was developed to enable COST partners to have a convenient platform to perform experimentation and to integrate provisions for improvement.

Underpinning the AM 5.0 at the technical level different video processing modules were implemented and integrated. Each one of these modules encapsulates a complete video processing technique The intermediate results obtained from each single module are merged and further processed by a set of intelligent rules to achieve a highly accurate final results. The modular structure of the system allows it to be extended continuously and with ease by adding new independent modules. The intermediate results of newly added modules are linked to the other system results via the rule processor. The implemented techniques, performance and evaluation results of the AM 5.0 has been reported in several papers [1], [5], [6], [7].

In this paper several extensions of the AM leading to the already released version 5.2 as well as very recent improvements which will lead to version 5.3 are reported. The main goal is to introduce this last version of the software in which a highly robust and stable segmentation system comprising supervised and unsupervised functionality as well as a user friendly graphical interface (GUI) have been realized. The purpose of the GUI is to allow non experts to use and experiment with the system and to provide the research community with a new highly efficient segmentation tool. To achieve this goal the GUI has been implemented in Java, while the video analysis software is implemented mainly in ANSI C. The GUI interacts with the ANSI C modules via the JAVA Native Interface. Further extensions will include a low complexity GUI as Java applet in order to allow anyone to use the system from everywhere over the Net.

2. BASIC STRUCTURE OF THE ANALYSIS MODEL

For the sake of completeness, the main modules of the AM 5.0 will be briefly described in this section. The result of applying the AM version 5.0 is a binary mask of moving foreground objects in front of a static or moving background. To deal with global changes in the scene camera motion estimation, compensation and scene cut detection methods have been implemented. Temporal coherency of the segmentation results is achieved by object tracking and compensation. To avoid distortions introduced by moving cast shadows, these are detected and used to compensate scene changes in the shot detection and segmentation modules. The output of the AM are the segmentation and the shadow masks, as well as, parameters describing global, local motion and other events. In the remaining of this section the most important modules of the AM 5.0 are described briefly. For a more detailed description of this release the reader is referred to [1], [5], [6], [7].

Global Motion Estimation and Compensation Given two successive frames in a video sequence, the apparent camera motion is estimated using an affine motion model. The eight parameters to be estimated encoded usual global displacements including zoom and pan. Camera motion compensation is achieved by bilinear interpolating.

Scene Cut Detection In the AM 5.0 a mean square error (MSE) based scene cut detector is used. For two consecutive frames the MSE is computed for the background. If the MSE is greater than a threshold a scene cut or pan is identified. In this case, relevant segmentation parameters are reset to their initial values and the system started from that point.

Change Detection The change detection mask (CDM) encodes luminance changes due to object displacements. The algorithm for CDM estimation is subdivided into four processing steps: computation of the initial CDM, relaxation of the initial CDM for spatial homogeneity, detection and elimination of moving shadows and temporal coherency of the object shapes.

Shadow Detection This module detects image regions changed by moving cast shadows in the background. To detect luminance changes due to moving cast shadows all pixels belonging to the CDM are evaluated according to the results obtained in the three processing sub-modules: detection of static background edges, detection of uniform illumination changes and penumbra detection.

Shadow Integration Image regions changed by moving cast shadows are classified and temporally integrated. This classification is conducted pixel-wise for any occluded or uncovered background. The shadow binary mask is initialized with the binary value 0. If the luminance value changes in a given pixel after processing, the binary value 1 is assigned to the corresponding sampling position. The final binary mask of moving cast shadows is then obtained by temporal smoothing and integration.

Color Segmentation In this module segmentation is carried out using only color information. A recursive-shortest-spanning-tree (RSST) based segmentation method is used. This technique allows simple control over the number of regions and therefore over the degree of detail in the segmentation mask. The RSST initially maps the input image into a weighted graph. The nodes of the graph form the regions and links between two nodes. These links represent the “distance” between the two neighboring regions. Initially each pixel is a node, i.e. a region. In the first processing step the RSST method checks all the links between regions, and merges the two regions separated by the link which minimizes the used distance measure. The merging continues until a desired number of regions is reached.

Local Motion Analysis A dense displacement vector field is estimated by hierarchical block matching. The estimation is performed in three levels. In each level global displacements are estimated and used as potential displacements in the next level. Finally, the estimated block motion vectors are interpolated in order to obtain a dense motion field [2], [3].

The Rule Processor The Rule Processor evaluates the results from the change detection, color segmentation and local motion analysis in order to distinguish between moving objects and background. In a first step, the uncovered background areas are eliminated from the estimated change detection mask. Only the estimated motion information for pixels within the changed regions is used. A pixel is set to “foreground” if both, the starting and ending points of the corresponding displacement vector, are in the “changed” area of the change detection mask. Otherwise the pixel is set to “background”. Since the color segmentation has accurate boundary information, the final segmentation mask, is copied from the color segmentation mask.

3. THE EXTENDED ANALYSIS MODEL

Recently, several modules of the AM 5.0 have been improved and totally new ones have been also added to that release. The most significant changes carried out in the already released AM 5.1 are in the color segmentation module. Here a more complex segmentation was implemented. The AM 5.2 introduced a new Rule Processor which gives more weight to the results of the enhanced colour segmentation approach. This new Rule Processor improved the accuracy of the object mask and allows the detection of uniform colored moving objects, where temporal changes usually only appear at the object border. In the following the new modules and functionality of the AM, leading to version 5.3 are described.

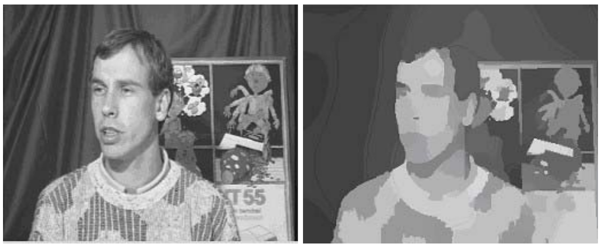


Fig. 1: Original image (top) and simplified image after 2000 iterations (bottom).

Enhanced Rule Processor According to the colour segmentation the next set of rules were added to the system: for a given region, if at lease 80% of its pixels are in foreground of the preliminary object mask, then the pixels that have been detected as foreground are set to foreground in final object mask. After that, this mask is processed by a morphological dilation. If the remaining pixels are detected as foreground in the dilated image, then they will be labeled as foreground in the final object mask, otherwise they are kept as background.

If less then 80% of pixels have been detected as foreground, then the pixels that have been detected as background will be set to background in the final object mask. After that the object mask is eroded. If the remaining pixels in the eroded mask are labeled as background, then they will be also labeled as background in the final object mask. Otherwise they will be set to foreground in the final object mask.

Video Segmentation For two consecutive video frames (previous frame and current frame), the whole segmentation processing is conducted in the following order: colour segmentation of current frame, global estimation between current and previous frame, scene cut detection, shadow detection, change detection and local motion analysis. Finally, The rule processor is used to analyze the results of the remaining modules and produce a final object mask.

Nonlinear Diffusion Filtering Module A nonlinear filter module based on anisotropic diffusion [4] has been added to the extended AM 5.3. In this module local averaging is inhibited in image edges and the diffusion velocity is controlled by the magnitude of the gradient intensity. In this context a set of images $I(x,y,t)$ is generated, with $I(x,y,0)$ as the original image and t as scale parameter, by applying the parabolic diffusion equation

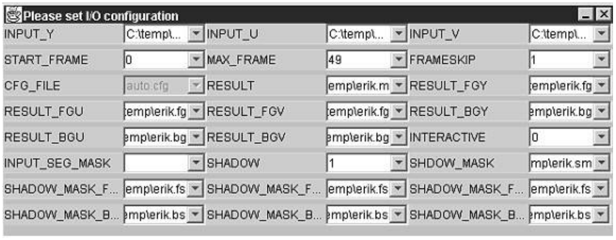
$$I_t = div(c(x,y,t)\nabla I) = \nabla \cdot [c(x,y,t)\nabla I]\,. \qquad (1)$$

If c is chosen as a suitable function of the image edges, the diffusion process tends to a piece-wise constant solution representing a simplified image with sharp boundaries. Using the diffusion equation (1) with $c(x,y,t) = f(\|\nabla I(x,y,t)\|^2)$, a set of simplified images is generated. The Perona-Malik diffusion function

$$f(w) = A \Big/ (1 + \tfrac{w}{B})$$

is used in this work. Fig. 1. shows how relevant edges are kept and even enhanced while texture of objects and noise is smoothed.

Region Growing Based Segmentation The purpose of this module is to segment simplified images obtained after nonlinear diffusion filtering. This module uses of recursive shortest spanning tree algorithm described in section 2. The main differences are: only intensity information (rather than the three colour channels) is used; the simplified images supplied by the non linear diffusion module are taken as initial segmentation and the calculation of each link cost or distance between regions is based on intensity information only. Furthermore, the fact that the diffusion-filtered image shows more accurate and sharp object boundaries, is used by the rule processor to find true edges of semantic objects in the scene.



Please set I/O configuration

INPUT_Y	C:\temp\...	INPUT_U	C:\temp\...	INPUT_V	C:\temp\...
START_FRAME	0	MAX_FRAME	49	FRAMESKIP	1
CFG_FILE	auto.cfg	RESULT	emp\erik.m	RESULT_FGY	emp\erik.fg
RESULT_FGU	emp\erik.fg	RESULT_FGV	emp\erik.fg	RESULT_BGY	emp\erik.bg
RESULT_BGU	emp\erik.bg	RESULT_BGV	emp\erik.bg	INTERACTIVE	0
INPUT_SEG_MASK		SHADOW	1	SHDOW_MASK	mp\erik.sm
SHADOW_MASK_F...	emp\erik.fs	SHADOW_MASK_F...	emp\erik.fs	SHADOW_MASK_F...	emp\erik.fs
SHADOW_MASK_B...	mp\erik.bs	SHADOW_MASK_B...	mp\erik.bs	SHADOW_MASK_B...	mp\erik.bs

Fig. 2: IO parameter window.

4. GUI VIDEO SEGMENTATION ENVIRONMENT

A GUI based Video Segmentation (GUI-VSE) Environment based on enhanced AM was implemented in JAVA 2. Because the AM is coded in ANSI C, the JAVA native interface is used to call C functions from the JAVA interface. The GUI-VSE comprises automatic and semi-automatic video segmentation functionality and other useful tools. Multithread is also used in sequence and mask viewers to read sequence and mask files.

Initially the GUI presents two default parameter windows to the user. The first window serves as input/output of video and segmentation masks. In the second window processing parameters can be input, changed and adjusted. A hash table is used to read and write configuration files. In Fig 2 a screen shot of the first IO window is shown. After the processing parameters have been configured, the automatic video segmentation program is called via the JNI and a new background process is crated to carry out the video segmentation. Thus, the user does not need to wait until the segmentation is finished, he can start a parallel job in the same GUI-VSE or anything else.

Semiautomatic Segmentation Tool Although the main goal of the AM is to carry out fully unsupervised video segmentation, in many situations human interaction is still needed. For instance in studio applications it is often necessary to extract objects from images in which intensity variation along any portion of the object contour is imperceptible. For this reason we have designed another segmentation scheme to extract exact object contours in accordance with rough position specifications by the human operator. The scheme consists of an automatic tool for the refinement of rough object mask and a graphic interface for user interaction. For automatic refinement the same techniques described previously are used. The GUI contains diverse functions to add or remove control points and to delineate fragments of object contours that cannot be extracted automatically due to the absence of intensity information and only the experiences accumulated in the human brain can outline. If motion fields are available, the user interface can also be used to interact with the object information extracted automatically in order to add, remove or shift control points and motion vectors to more accurate positions

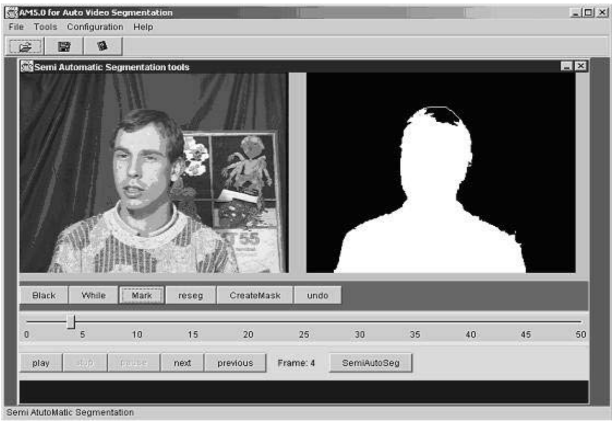


Fig. 3: Semi-automatic video segmentation tool

Using the GUI, the user can also view the video and mask sequences simultaneously. If a given object mask is not accurate enough, the user can refine the mask interactively. Using the refined object mask as an input, better segmentation result can be obtained with the unsupervised segmentation sysytem. In Fig 3 a screenshot of this user interface is shown.

Additional functionalities supported by the GUI are:

- **Video Sequence Viewer and Mask Sequence Viewer** can be used to stream the video sequence and the segmentation results.
- **Sequence Merger and Mask Merger** To append a sequences to another. This functionality can be used over video sequences or segmentation masks.

- **Sequence Partition and Mask Partition** To create or split sequences into different subsequences.

5. SELECTED RESULTS

In order to validate the system performance several experiments have been conducted using internationally known test sequences. A thoroughly validation of the AM version 5.0 has been performed and reported previously [5], [6]. In this section only few selected results of the AM extensions leading to version 5.3 are described. The images at the top of Fig. 4 show the extracted segmentation masks for the first and 20th frame of sequence ERIC. The images at the left-middle and left-bottom of Fig. 4 present original frames from COASTGUARD and TABLE-TENNIS. The images at the right show the extracted segmentation masks using the unsupervised mode in the AM 5.3. The subjective quality of the segmented objects is quite good. Although examples showing the temporal coherency of the results cannot be shown in this paper, it can be stated that the overall performance of the automatic segmentation system is very satisfactory.

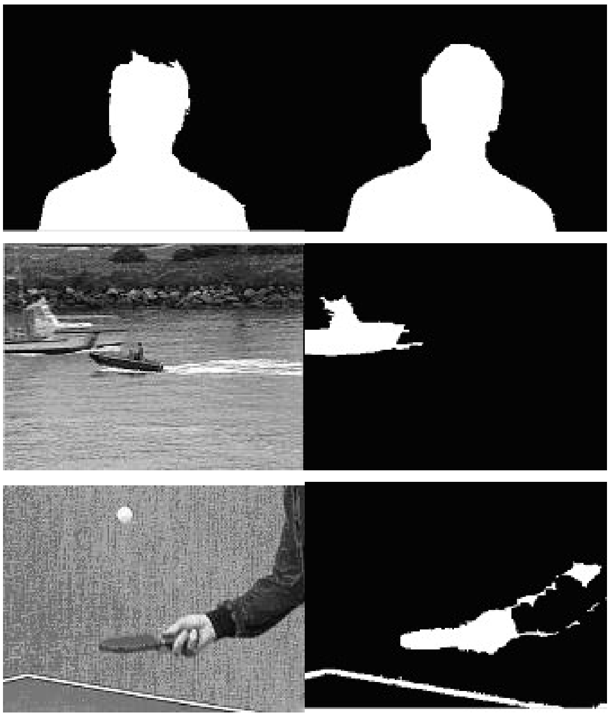


Fig. 4: Segmentation masks obtained with the unsupervised AM 5.3 system.

6. SUMMARY AND CONCLUSIONS

In this paper an advanced video analysis system is described. The backbone of the introduced system is an integration concept in which different advanced segmentation approaches merges in a cooperative framework. The objective is to obtain a robust, efficient and modular analysis and segmentation tool for natural video and still images. The system comprises several independent modules, each one of them encapsulating a complete video processing technique. The intermediate results obtained from each single module are merged and further processed by a set of intelligent rules to achieve a highly accurate final segmentation. The modular structure of the system allows it to be extended continuously and with ease by adding new independent modules. The intermediate segmentation results of newly added modules are linked to the other system results via the rule processor.

A user friendly graphical interface implemented in JAVA 2 serves as communication shell between user and the AM system. The GUI-VSE includes automatic and semi-automatic video segmentation functionalities and other useful tools. In the GUI-VSE, the JAVA code calls the video segmentation functions of the AM via the JAVA Native Interface.

Several experiments on international known test sequences were conducted to assess the performance of the described system. From a subjective point of view, the extracted object segments are a good approximation of the image areas that we try to separate. This assertion is also confirmed when the obtained results were quantitatively evaluated by using the two criteria for objective segmentation evaluation introduced in [7].

Acknowledgements: The research and implementations leading to the presented system was conducted in cooperation with different partners and research institutions involved in the EU Cost211 forum and the BUSMAN project.

REFERENCES

[1] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora, “Image Sequence Analysis for Emerging Interactive Multimedia Services – The European COST211 Framework” IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 7, pp.802-803, November 1998.

[2] M. Bierling, “Displacement Estimation by Hierarchical Blockmatching”, in Proceedings of SPIE Visual Communications and Image Processing 88, pp. 942-951,1998.

[3] E. Izquierdo, “Stereo matching for enhanced telepresence in 3D-videocommunications“, IEEE Transaction on Circuits and Systems for Video Technology, Special issue on Multimedia Technology, Systems and Applications, Vol. 7, No. 4, Aug. 1997, pp. 629-643.

[4] P. Perona, J. Malik, “Scale-Space and Edge Detection Using Anisotropic Diffusion”, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.12. NO.7. July 1990.

[5] J. Stauder, R. Mech, J. Ostermann, “Detection of Moving Cast Shadows for Object Segmentation”, in IEEE Transactions on Multimedia, vol. 1, no. 1, pp 65-76, March 1999.

[6] R. Mech, M.Wollborn. “A Noise Robust Method for Segmentation of Moving Objects in Video Sequences”. IEEE ICASSP’97, pp. 2657-60, 1997.

[7] P. Villegas, X. Marichal, A. Salcedo, “Objective Evaluation of Segmentation Masks in Video Sequences”, Proceedings of WIAMIS 99, Berlin, Germany, May, 1999.