QIMERA: A SOFTWARE PLATFORM FOR VIDEO OBJECT SEGMENTATION AND TRACKING

N. O'CONNOR[1], T. ADAMEK, S. SAV, N. MURPHY AND S. MARLOW

Centre for Digital Video Processing
Dublin City University, Ireland
E-mail: [1]Noel.OConnor@dcu.ie

In this paper we present an overview of an ongoing collaborative project in the field of video object segmentation and tracking. The objective of the project is to develop a flexible modular software architecture that can be used as test-bed for segmentation algorithms. The background to the project is described, as is the first version of the software system itself. Some sample results for the first segmentation algorithm developed using the system are presented and directions for future work are discussed

1. Introduction

The QIMERA[1] project was initiated in March 2002 by a group of researchers sharing a common interest in video object segmentation and tracking. They decided to form a voluntary (i.e. non-funded) project between themselves and to invite participation from other researchers working in the field. The project currently consists of five members who work remotely over email.

The objective of the QIMERA project is to develop a flexible modular software architecture for video object segmentation and tracking facilitating multiple configurations of analysis algorithms and supporting user interaction when necessary. The goal is to develop a system into which individual analysis tools can be easily integrated in order to test their efficiency/accuracy. The system should not be tied to one particular segmentation algorithm, but rather should be configurable depending on the type of segmentation problem to be addressed and the analysis tools available. An architecture for such a system has been proposed and an initial software implementation of the key

[1]The name QIMERA is derived from the Spanish word '*quimera*' ('*chimera*' in English) meaning a fantastic fabrication of the mind, especially an unrealistic dream. Project members felt this to be a suitable description for the very challenging problem of segmentation.
Qimera web page: http://www.qimera.org

components is available. A segmentation algorithm that uses the available components has been developed.

In this paper we present a high level overview of the system, a description of the first algorithm implemented, and some sample results. It should be noted that the algorithm described is the first attempt to actually use the QIMERA platform, and as such it is quite crude.

2. System Overview

The QIMERA platform consists of graphical user interface (GUI) and a system core. The system is designed so that the GUI and the core can be decoupled. In this way, the core could run on one platform whilst the GUI runs on a different (remote) platform.

The system core consists of a set of configurable *Analysis Modules* communicating with the GUI and each other. The *Analysis Modules* are designed to group different approaches to specific image/video analysis tasks. The idea is that, for example, many different approaches to colour segmentation could be grouped in a single Colour Analysis Module. Thus, each Analysis Module can consist of one or more individual analysis tools that could either work together or in competition with each other. In order to produce an output object segmentation, the results of a number of different Analysis Modules need to be combined – e.g. combining the results of independent colour segmentation and motion segmentation processes. The task of scheduling when results should be combined and actually carrying out the inference process is performed by the *Inference Engine*. This structure is illustrated by a screen shot of the System Configuration Interface in Figure1.

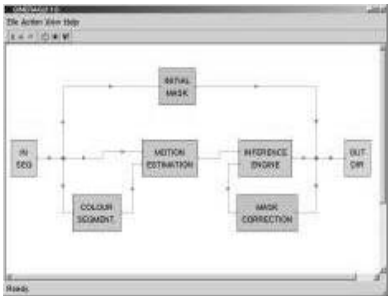

Figure 1. System Configuration Interface

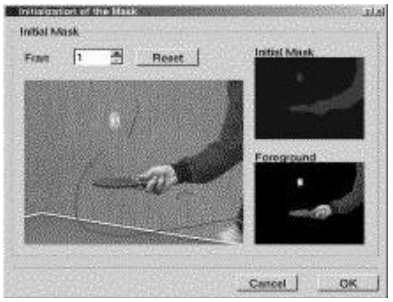

Figure 2. User Interaction Interface

Each module depicted in Figure 1 has two components: a GUI for user interaction (setting of the initial parameters) and a processing component that implements the module. A brief description of each module in presented below:

- *input (output) module* – assists the user in the selection of the input video sequence (output results)

- *initial mask module* – assists the user in the grouping of the segmented regions in the initial frame to form an initial object segmentation mask
- *colour segmentation module* – segments each frame into uniform colour regions.
- *motion estimation module* – estimates the motion of regions/objects.
- *inference engine* – combines the results of other modules
- *mask correction module* – allows the user to correct the object mask, if required, at any frame during the tracking stage

3. A Semi-automatic Segmentation Algorithm using QIMERA

In order to instantiate an initial version of the platform, individual analysis tools were developed and integrated into the relevant modules as outlined in Figure 1. Using these analysis tools, the system was used to develop an approach to semi-automatic segmentation. In this section we describe this approach. It should be noted that this approach is only representative of the type of approach that could be developed with this system.

3.1. *User interaction*

In order to mark objects, the user draws two coloured *scribbles* [1] on the initial frame: over the foreground and background parts of the image. Since the image is pre-segmented into uniform colour regions, the user's scribbles specify a number of regions that are then classified as background or foreground. If both scribbles touch the same region a conflict occurs for that region. In this case the region is considered unselected. The scribble interface is depicted in Figure 2.

3.2. *Tracking strategy*

Once the initial mask is constructed, the foreground object is segmented and updated for each frame in the video sequence in the following manner (similar to the approach outlined in [2]):

- Each frame is segmented into uniform regions.
- For each region in the current frame a two-parameter motion projection is estimated.
- The regions that are projected inside the previous foreground mask are selected in the foreground mask for the current frame. The regions for which the backward projection extends outside the previous foreground mask are labeled as outliers and their association to foreground or background is further investigated.

The details of the modules that implement the above algorithm are described in the following.

3.3. *Colour segmentation module*

The colour segmentation module partitions each image in the video sequence into uniform colour regions. For this we use a modified Recursive Shortest Spanning Tree (RSST) algorithm [3]. Our modification is based on the observation that sometimes this approach merges regions with very different colors because of the strong penalty for joining large regions (originally introduced to improve the spatial continuity of the final regions). The problem is particularly visible when a small number of final regions are desired. To make the segmentation more insensitive to small changes in illumination due to shadows etc. we use the HSV colour space. The algorithm operates in two stages:

- The first stage is identical with the original RSST algorithm. It iteratively merges regions (two regions per iteration) according to the distance calculated using the colour features and region size. The process stops when the desired number of regions is obtained. Because of the problem outlined above the specified number of regions should not be smaller than 255.
- The second stage continues to merge regions, but a new formula is used to calculate the distance between two regions:

$$d(r_i, r_j) = \frac{1}{4} \cdot | saturation_i - saturation_j | + \frac{3}{4} \cdot | hue_i - hue_j | \qquad (1)$$

where:

$d(r_i, r_j)$ is the distance between regions *i* and *j*.

The above formula does not discourage large regions (since spatial continuity was obtained in the first stage). The hue operations are computed modulo 360^o. The region merging stops when all distances exceed a predefined threshold.

3.4. *Initial Mask Module*

The user drawn scribbles classify the regions they intersect into background and foreground respectively. Not all the regions in the initial image are intersected by one of the scribbles[2] therefore they are not explicitly assigned to background or foreground during user interaction. The unclassified regions are labeled iteratively. For all unlabeled regions, the distances to the available objects (foreground/background) are calculated as:

$$d_{ur}(l) = \frac{\min[d(ur, lr)]}{\sum d(ur, lr)} \qquad (2)$$

where:

$d_{ur}(l)$ is the distance to the closest classified region

$\min[d(ur, lr)]$ - is the minimum distance to each labeled region.

[2] It is generally desirable to keep user interaction to a minimum.

$d(ur, lr)$ - is the distance to a labeled region.

The distance measure incorporates the colour and spatial features of the regions. At every iteration the region which has the smallest distance is assigned to the foreground or background respectively. The assignation of the regions ends when all regions are labeled

3.5. *Motion Estimation Module*

Motion estimation is performed on regions using backward motion projection. Every region in the current frame is backward projected into the previous frame based on a colour metric and using a two parameter motion model.

3.6. *Inference Engine*

The inference engine assigns the background/foreground labels to the current image regions based on the results of the motion estimation module. The decision rules are the following:

- The regions projected completely inside the previous frame foreground mask are assigned to the foreground.
- For regions that are only partial projected inside the previous frame foreground mask (outlier regions) the pixels inside and respectively outside the mask are identified and the colour distance between each partition and the surrounding regions is computed according to equation (2). The ratio of the distances is computed. The region is a classified as background or foreground if the ration exceeds a given threshold used to define the robustness of the tracking procedure.

4. Conclusions and Future Work

The tracking results obtained with the algorithm described in the previous section are illustrated in Figure 3. The results indicate that the algorithm is a suitable starting point for work on supervised object segmentation and tracking. Future work on this particular algorithm will focus on developing a fully automatic mode of the algorithm, refining the process of motion estimation and investigating a statistical approach to combining individual segmentation results to produce a final object segmentation.

Future work in the QIMERA project itself will focus on collaborative work on a more sophisticated inference engine, integrating additional analysis tools, adding segmentation evaluation metrics and providing an enhanced interface for module communication and integration.

Figure 3. Segmentation results for the MPEG-4 test sequences: “Foreman”, “Mother and daughter”, “Table tennis”, at frames 1, 14, 23.

Acknowledgments

This material is based upon work supported by the IST programme of the EU in the project IST-2000-32795 SCHEMA. The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

References

[1] N. O'Connor, S. Marlow, "Supervised semantic object segmentation and tracking via EM-based estimation of mixture density parameters", Proceedings NMBIA'98 (Springer-Verlag), pp. 121-126, Glasgow, July 1998.

[2] F. Marques, B. Margotegui, F. Mayer, "Tracking areas of interest for content-based functionalities in segmentation-based video schemes", IEEE International Conference on Accoustic, Speech and Signal Processing, vol. 2, pp. 1224-1227, May 1996.

[3] E. Tuncel, L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modelling", IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no.5, August 2000.