AttentionAttractor: efficient video stream similarity query processing in real time

Ira Assent, Ralph Krieger, Thomas Seidl
RWTH Aachen University
Data management and exploration group
52056 Aachen
{assent,krieger,seidl}@cs.rwth-aachen.de

Abstract

In a current project, customers are attracted by a video streaming application. A video camera records people passing by, and a monitor shows an alienated version of the setting accordingly. The idea is to replace the image on the video screen by a mosaic of similar images to draw their attention to the location. For successful implementation, several aspects are of key importance: the images chosen in the mosaic should be similar enough for easy recognition, and the result of the alienation should be computed fast enough for display on the screen in real time.

1. Introduction

Getting potential customers’ attention is growing increasingly difficult. People are to a large extend overwhelmed by standard sales promotion. Customers are drawn in by new technologies which demonstrate novel possibilities. In a current project, we alienate the video image captured by a shop camera and project the result onto a screen for passers-by. This ‘mosaic’ of video stream data is a challenge for state of the art technology. This project requires real-time computation of the images to be feasible, while also requiring recognizable similarity results. Brief:

- correspondence with human perception of similarity
- real time responses
- scalability
- robustness to small changes in incoming video frames

2. Similarity search

It is important that the similarity model produces convincing results for human eyes. The degree of similarity refinement should be adjustable according to the degree of change in the video stream. Whenever changes occur only locally or to a small degree, more time can be spent on computing similarity for other tiles. If all tiles of one video image are processed before the mosaic is updated, the delay in response times is infeasible. We therefore progressively compute results and display them immediately (Fig. 1). Tiles should not be simply processed in the order arriving from left to right, top-down, as the result is less likely to immediately reflect the areas of greatest change. These areas, however, are noticed first by human spectators.

3. System architecture

To guarantee performance for interactive mosaic video the AttentionAttractor uses three priority queues (Fig. 2).
For human perception, it is important that the tile currently shown on screen is not significantly different from the current actual tile. Since the tiles in a video stream constantly change it is important to efficiently measure the dissimilarity between tiles. Thus, instead of only using the time consuming lower bounds. The ‘Video Module’ extracts the tiles from the live video stream. The extracted tiles are inserted into the video priority queue according to degree of change and passed on to the image database module. The ‘Image Database Module’ consecutively receives tiles from the ‘Video Module’. For each tile the dissimilarity between the tile and its corresponding counterpart from the ‘Presentation Module’ is measured. The tile is added to the database priority queue. The database queue is used as input for the query processor which seeks the most similar image from the image database. The job of the ‘Presentation Module’ is to update the mosaic tiles. Thus it receives new tiles from the ‘Image Database Module’, measures the dissimilarity between the new tile and the displayed counterpart and updates the mosaic priority queue. The tile with the highest priority is finally updated first. Each module of the AttentionAttractor can use a different distance function for determining the priority of tiles(Fig. 3).

The core of the AttentionAttractor is the ‘Query Processor’. It consists of three components: the query cache, an index structure and the image database. The cache is important for storing tiles queried repeatedly like those in the background. If the cache does not contain a tile, an R*-Tree [2] is searched for the nearest neighbor. It is well researched that the performance of spatial index structures degrades rapidly with higher dimensions. Thus we use the KNOP multistep approach which uses a ranking query to retrieve candidates from the index [4]. After a candidate is retrieved the query processor updates the priority distance for the tile queried. Before the query processor continues with the KNOP refinement step the queue is checked if another tile now has a higher priority. If so the query is suspended and stored with the current tile while the query processor continues with the tile now having the highest priority. Thus, KNOP allows for interrupting the similarity search for one tile in favor of one with higher priority. Even though a suspended candidate is not the correct nearest neighbor it is nevertheless a good approximation. It may be later refined further until completely refined or replaced by more recent changes.

4 Demonstration Summary

The demonstration system developed in our project shows the aspects of video stream similarity search as described above. The video camera image will be displayed along with its alienated version. The priority queues will be visualized and the underlying similarity search results will be demonstrated. By tracking the number of tiles processed and the ratios of the refinement steps that have to be taken, the efficiency and effectivity of the system are shown.

References