Video Visualization

Introduction

Video tends to be a complex data type when it comes to visualization. There is a clear demand for more automatic processing capabilities as there is so much video material captured nowadays making manual processing impossible.

Image processing techniques combined with suitable visualization approaches can be a useful tool for automatic and semi-automatic ways to process that type of data.

Introduction

There are different ways of visualizing video material. The following slides will introduce a few concepts that were published in the recent past.

Video Visualization

Gareth Daniel, Min Chen,

University of Wales Swansea, UK

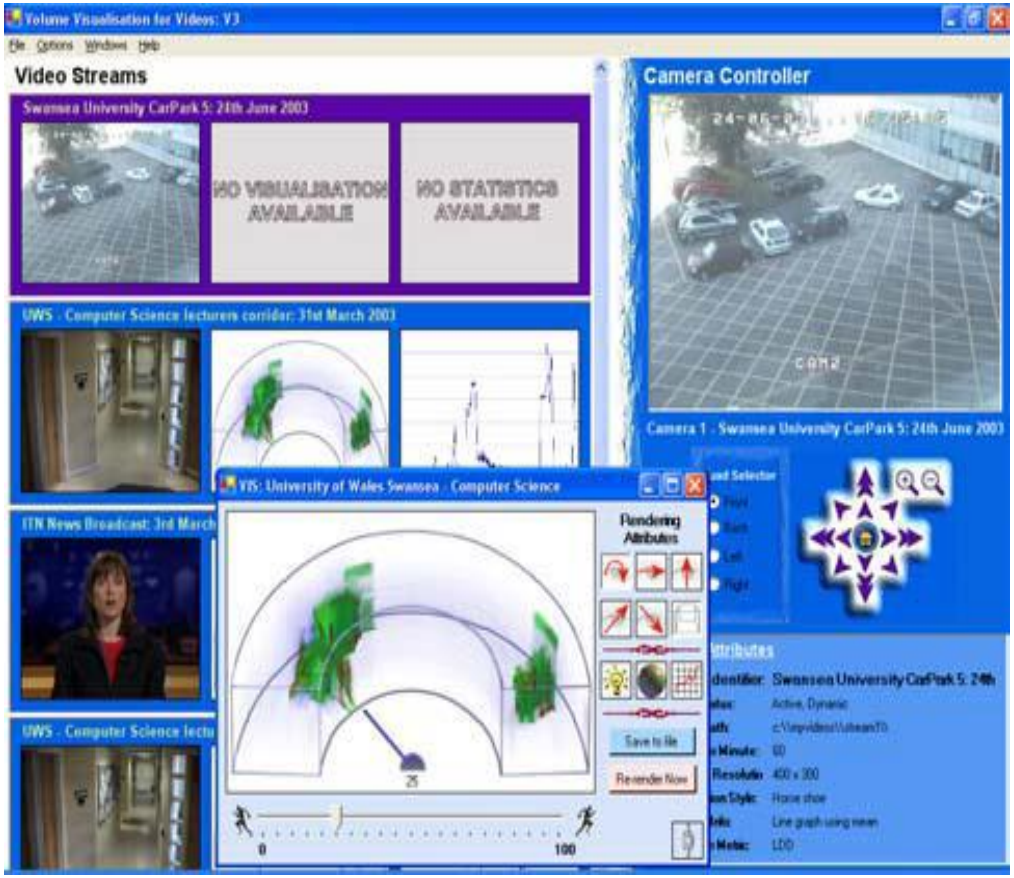
Video visualization pipeline:

video capture ⇒ data communication ⇒ data management ⇒ video processing ⇒ video visualization

Main concept: use volume visualization techniques for proving overview of entire video

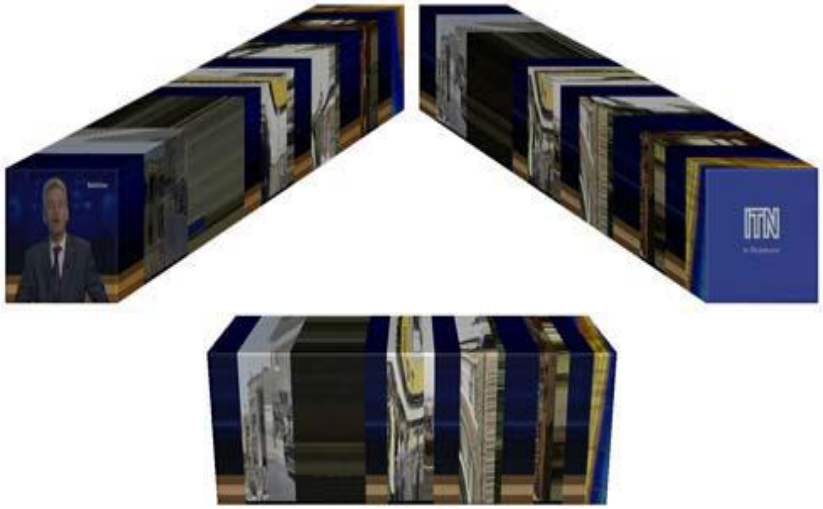
Video Visualization

User interface



Video Visualization

Render video frames as volume



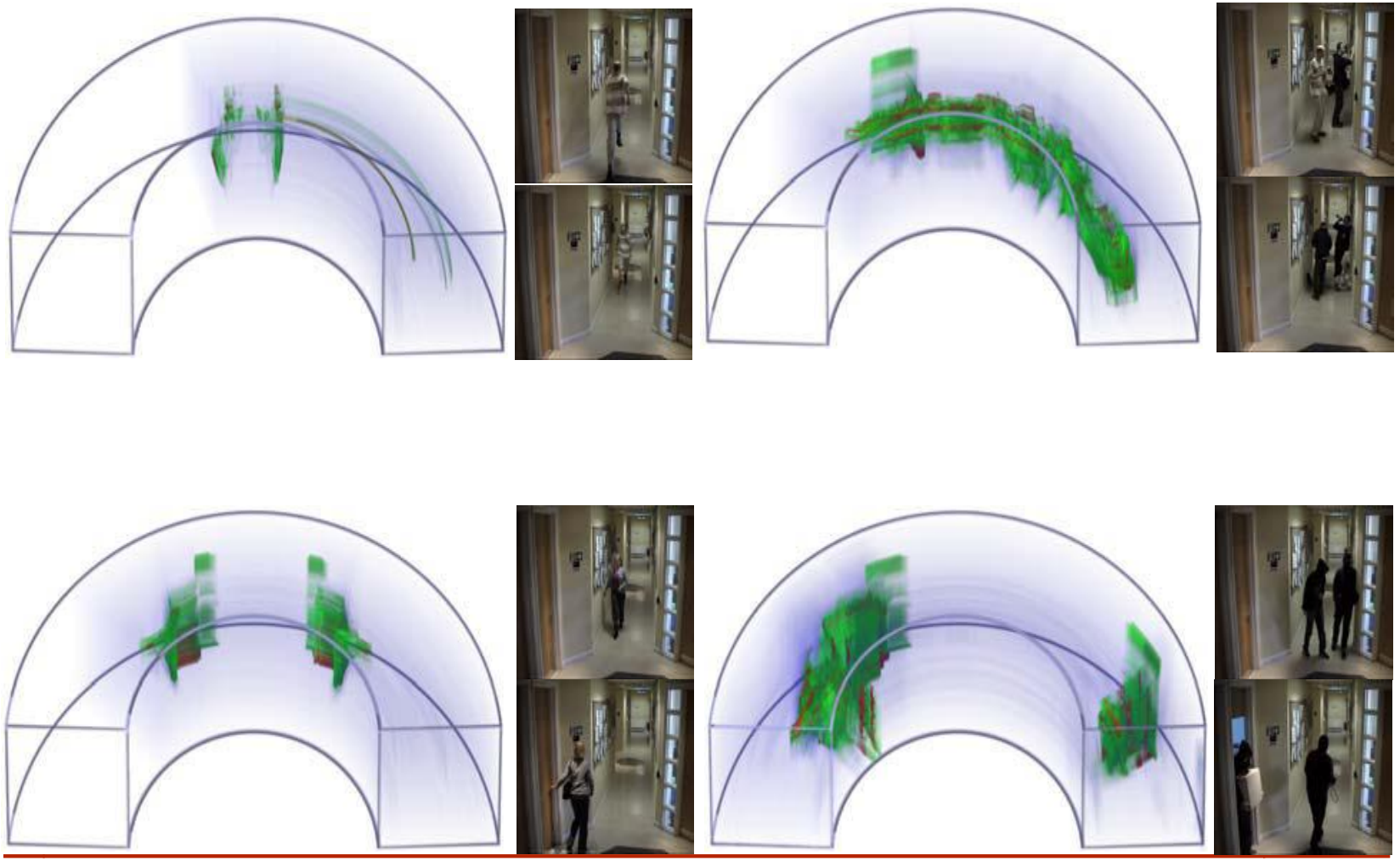
Video Visualization

opacity filters – Each filter creates an opacity feature volume that highlights or de-highlights particular components of an RGB volume. Figure 4 shows a visualization supported by such a filter that determines the opacity of each voxel based on the hue values and edge properties in a small window associated with that voxel.

Video Visualization

change detection filters – Each filter typically creates a feature volume that represents the magnitude of temporal changes in a video volume *V*. This is the focus of the following discussions in the rest of this section.

Video Visualization



Video Visualization

Different measures can be used for additional visualizations:

- Y-DIF(*I*1, *I*2) – *simple difference metric* – It takes two input images, *I*1 and *I*2, and computes a grey-scale output image *O* where each pixel represents the linear distance between the Y-values of two corresponding pixels in *I*1 and *I*2 respectively.
- Y-NMSE(*I*1, *I*2) – *normalized mean squared error metric* – Instead of the linear distance, it computes the squared distance (i.e., error) between the Y values of each pair of corresponding pixels. The name of the metric is inherited from the corresponding statistical indicator that calculates the mean of the squared errors of all pairs of pixels in two images. In addition, the Y-component of each input image is normalized based on its mean value and standard deviation prior to the computation of mean squared errors. This may reduce the luminous difference caused by different lighting and atmospheric conditions.

Video Visualization

- IQ-DIF(*I*1, *I*2) – *color difference metric* – It computes the angle between the IQ vectors of the two corresponding pixels in *I*1 and *I*2, and sets the corresponding pixel value in *O* to the angle. It gives a result similar to that obtained by computing the hue difference in the HSV space.
- Y-LDD(*I*1, *I*2) – *linear dependence detector (LDD)* – We use the improved version proposed by [Durucan and Ebrahimi 2001b]. This is an illumination invariant change detector, and it examines the changes from a reference image $R = I2$ to another image *I*1 using a small window. Given a $k \times k$ window centered at (x, y), for each of the two input images, we place the Y values of all pixels in the window into a vector, that is, vector $M = (m1, m2, \ldots)$ for image *I*1, and $N = (n1, n2, \ldots)$ for *R*. We normally increase the values in *M* and *N* by one to ensure no zero component is in either vector. Based on the algebraic properties of the two vectors, the level of dependence between the two vectors can be measured by:

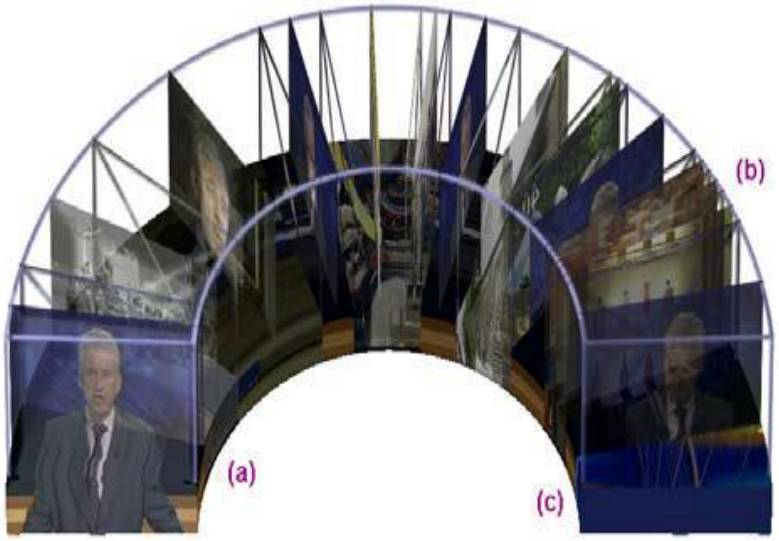
$$c = \left(\frac{1}{k^2}\sum_{i=1}^{k^2}\frac{m_i}{n_i}\right) - \frac{1}{k^2}\sum_{i=1}^{k^2}\frac{m_i}{n_i}$$

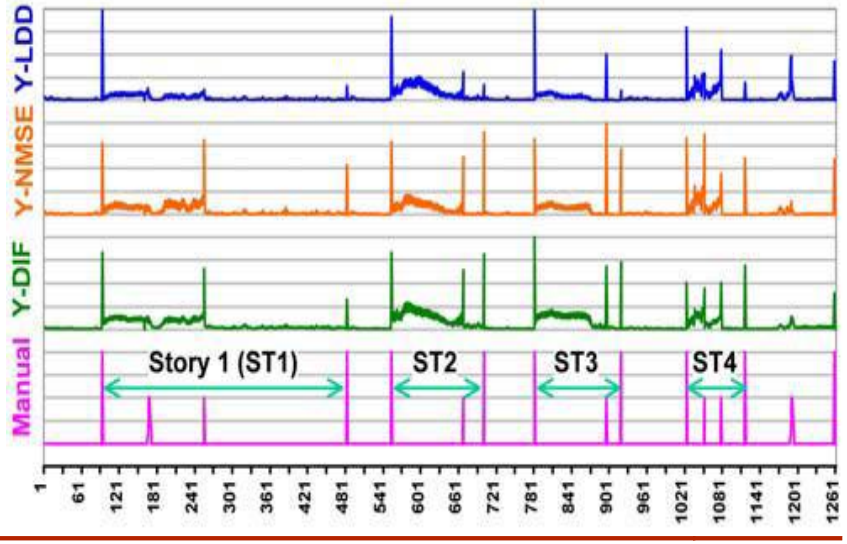
When $c=0$, Y-LDD detects no changes; when $c<0$, it detects illumination changes; when $c > 0$, it detects other changes, including object changes.

Video Visualization

Results:

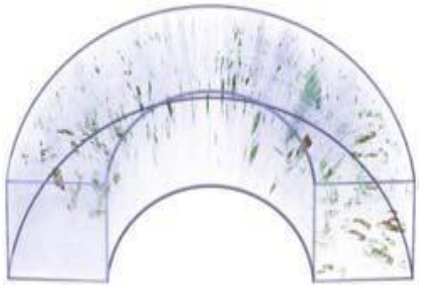
Line graphs depicting the change-detection results produced by Y-LDD, Y-NMSE, Y-DIF, and human decision. A visualization (based on Y-NMSE) of transition frames and representative images can help identify errors in the numerical results.



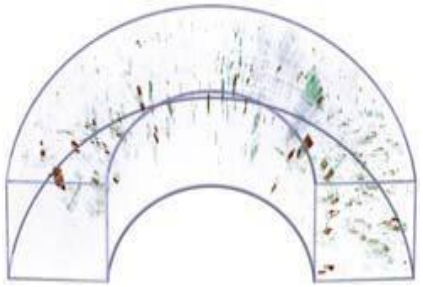


Video Visualization

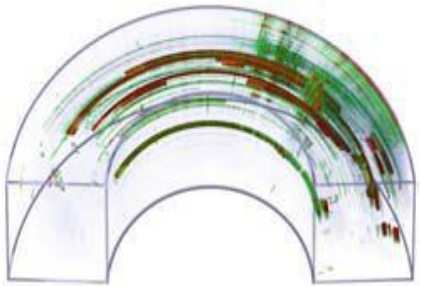
Results



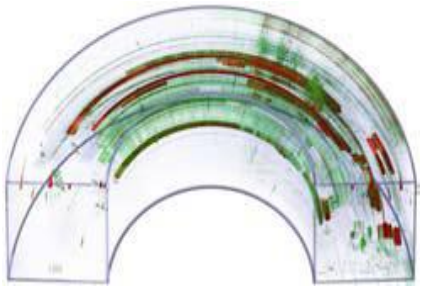
relative, Y-LDD



relative, Y-MNSE



absolute, Y-LDD



absolute, Y-MNSE

MediaMetro

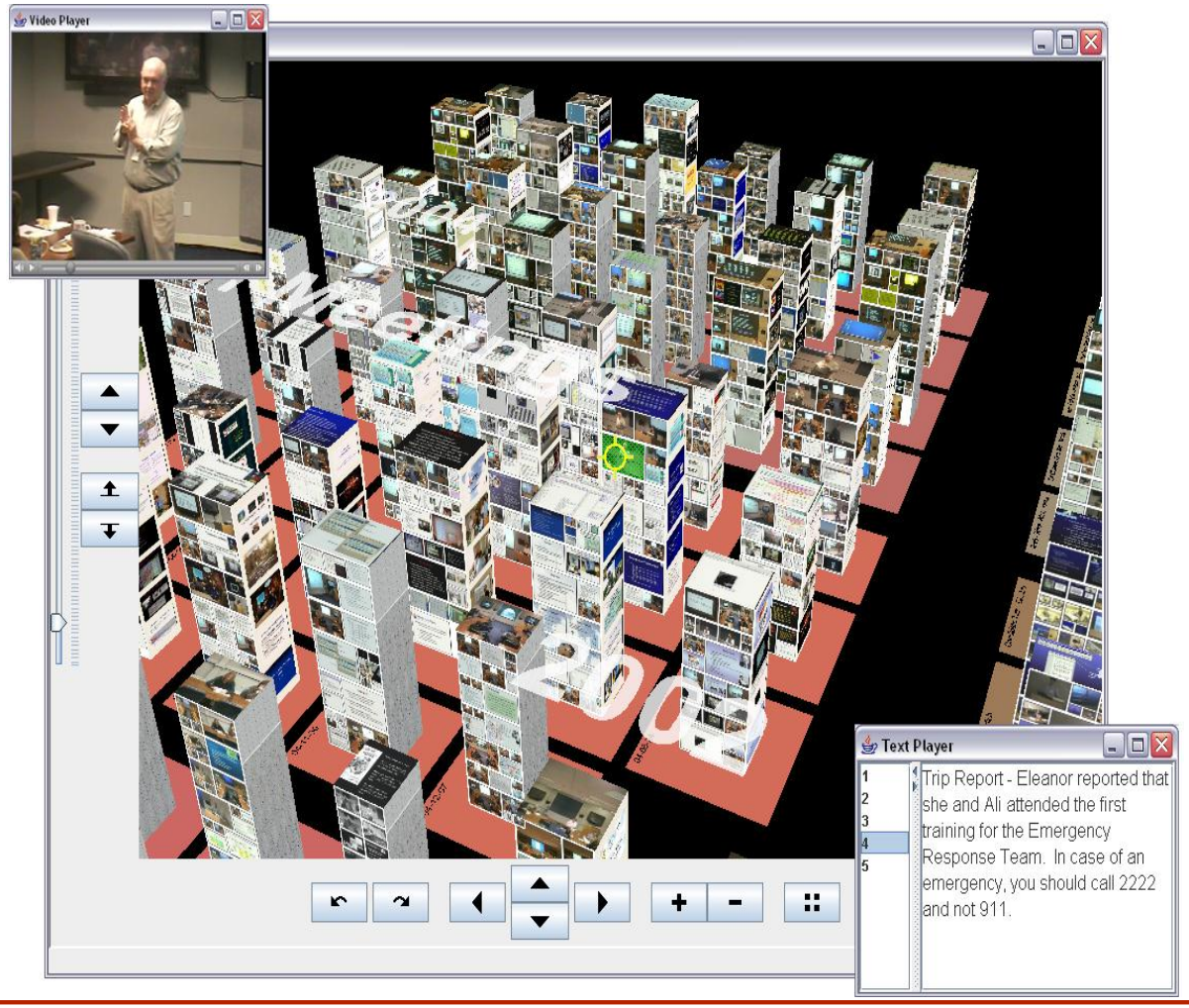
MediaMetro: Browsing Multimedia Document Collections with a 3D City Metaphor

Patrick Chiu, Andreas Girgensohn, Surapong Lertsithichai, Wolf Polak, Frank Shipman

The MediaMetro application provides an interactive 3D visualization of multimedia document collections using a city metaphor. The directories are mapped to city layouts using algorithms similar to treemaps. Each multimedia document is represented by a building and visual summaries of the different constituent media types are rendered onto the sides of the building. From videos, Manga storyboards with keyframe images are created and shown on the façade; from slides and text, thumbnail images are produced and subsampled for display on the building sides. The images resemble windows on a building and can be selected for media playback. To support more facile navigation between high overviews and low detail views, a novel swooping technique was developed that combines altitude and tilt changes with zeroing in on a target.

MediaMetro

Results



News Video Retrieval via Visualization

Large-Scale News Video Retrieval via Visualization

Hangzai Luo Jianping Fan Yuli Gao Shin'ichi Satoh William Ribarsky, UNC Charlotte

As the content of everyday news reports is unpredictable, keyword based news search engine cannot provide effective services to audiences because the audiences may not be able to figure out proper keywords to search. In this paper, a novel framework is proposed to help audiences browse and retrieve news video clips without the need of keywords. Interesting keyframes and keywords are automatically extracted from news video clips and visually represented according to their interestingness and informativeness measurement. A computational approach is also developed to quantify the interestingness measurement of video clips. The keyframes and keywords are carefully organized so that the audiences can find news stories of interest at first glance.

News Video Retrieval via Visualization

Results

MPCA Subspace

Visualization and Clustering of Crowd Video Content in

MPCA Subspace

Haiping Lu, How-Lung Eng, Myo Thida, Konstantinos N. Plataniotis

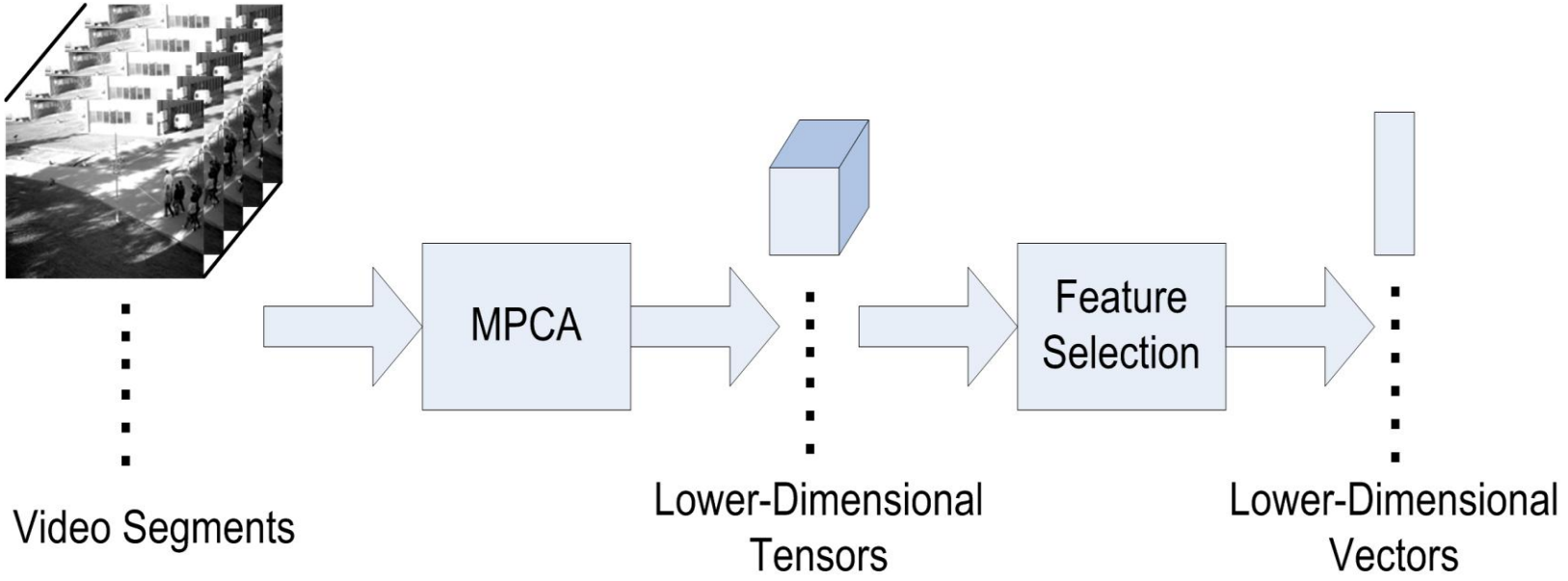
This approach uses multilinear principal component analysis (MPCA). In contrast to feature-point-based approach and frame-based dimensionality reduction approach, the proposed method maps each short video segment to a point in MPCA subspace to take temporal information into account naturally through tensorial epresentations. Specically, MPCA projects each short segment of a video to a low-dimensional tensor rst.

MPCA Subspace

A few MPCA features are then selected according to the variance captured as the nal representation. Thus, a video is visualized as a trajectory in MPCA subspace. The trajectory generated enables visual interpretation of video content in a compact space as well as visual clustering of video events. The proposed method is evaluated on the PETS 2009 datasets through comparison with three existing methods for video visualization. The MPCA visualization shows superior performance in clustering segments of the same event as well as identifying the transitions between events.

MPCA Subspace

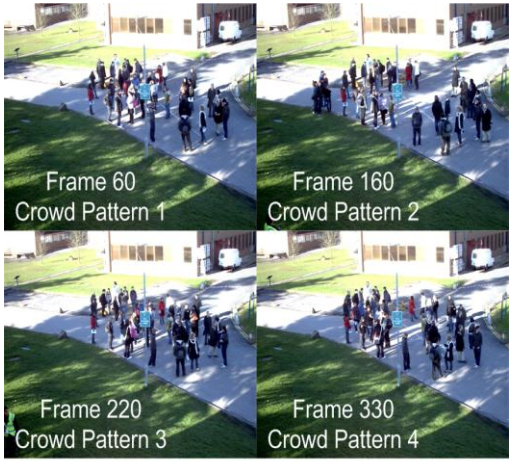
MPCA is a multilinear subspace learning method that extracts features directly from tensorial representation of multi-dimensional objects. MPCA can be used for gait recognition by representing each half cycle of gait silhouette sequences as a third-order tensor. Here, we apply MPCA to crowd video analysis by extracting MPCA features from raw video sequences.

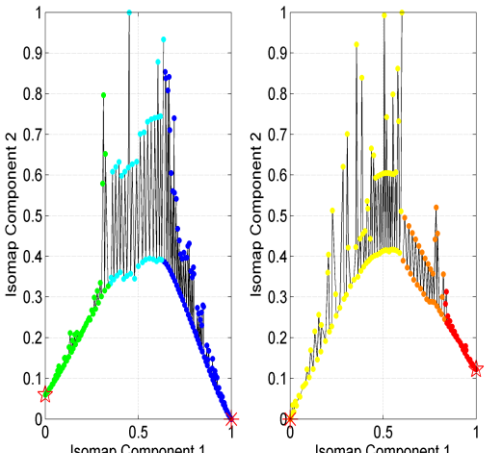


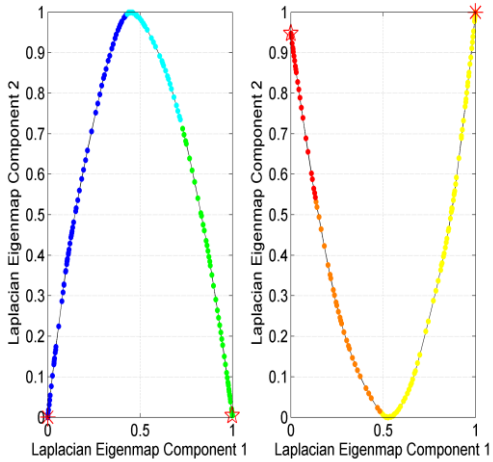
MPCA Subspace

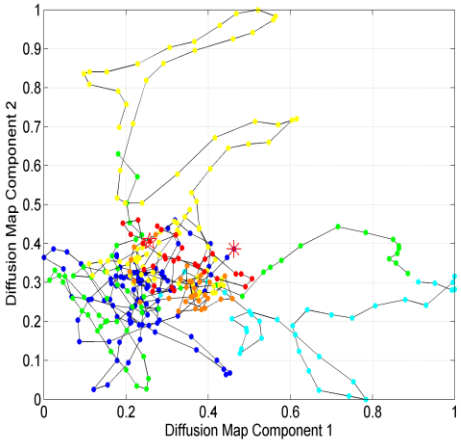
Results

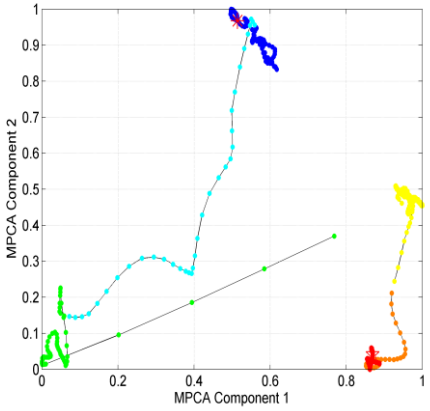
- Crowd Pattern 1 (Blue)
- Transition: Pattern 1 to 2 (Cyan)
- Crowd Pattern 2 (Green)
- Crowd Pattern 3 (Yellow)
- Transition: Pattern 3 to 4 (Orange)
- Crowd Pattern 4 (Red)











Video Summagator

Video Summagator: An Interface for Video Summarization and Navigation

Cuong Nguyen, Yuzhen Niu, and Feng Liu

Video Summagator (*VS*) provides a volume-based interface for video summarization and navigation. *VS* models a video as a space-time cube and visualizes the video cube using real-time volume rendering techniques. *VS* empowers a user to interactively manipulate the video cube. We show that *VS* can quickly summarize both the static and dynamic video content by visualizing the space-time information in 3D. We demonstrate that *VS* enables a user to quickly look into the video cube, understand the content, and navigate to the content of interest.

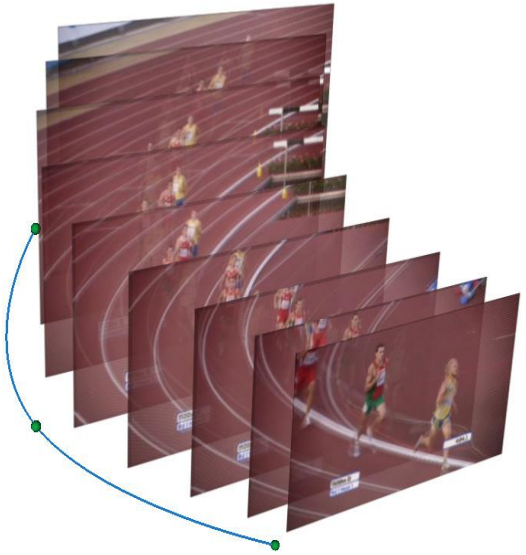
Video Summagator

Video Summagator visualizes a video in 3D, allowing a user to look into the video cube, and enables rapid visualization and navigation

Video Summagator

Video cube deformation

VS supports a user to globally shear the video cube to create a panoramic summarization (left). The user can also use a spline interface to deform the cube to better arrange the video content in the 3D space. For the example in (right), the user define three control points (in green) to define a new spline to deform the cube.



Video Summagator

Results