Interactive Mosaic Generation for Video Navigation

Kihwan Kim, Irfan Essa, Gregory D. Abowd
College of Computing, Georgia Institute of Technology
Atlanta, GA 30332-0760 USA
{kihwan23,irfan,abowd}@cc.gatech.edu

ABSTRACT

Navigation through large multimedia collections that include videos and images still remains a hard problem. In this paper, we introduce a novel method to visualize and navigate through the collection by creating a mosaic image that represents the compilation. This image is generated by a labeling-based layout algorithm using various sizes of sample tile images from the collection. Each tile represents both the photographs and video files representing scenes selected by matching algorithms. This generated mosaic image provides a new way for thematic video and also visually summarizes the videos. Users can generate these mosaics with some predefined themes and layouts, or base it on the results of their queries. Our approach supports automatic generation of these layouts by using meta-information such as color, time-line and existence of faces or manually generated annotated information from existing systems (e.g., the Family Video Archive).

Categories and Subject Descriptors

H.5.1[Information Interfaces and Presentation]: Multimedia Information Systems – *video;* I.3.8[Computer Graphics]: Applications – *Application;*

General Terms

Algorithms, Design, Experimentation

Keywords

Mosaics, Video Annotation, Video Navigation

1. INTRODUCTION

Mosaic is a form of art created by mixing fragments such as pottery, stone, or colored glass. With recent advancement in computing power and development in computer graphics, many kinds of automatic mosaic generation algorithms are introduced for images and multimedia [6, 7, 9, 10]. These approaches use a small subset of images, called 'tile images', to form a large image relying on layout rules such as matching color, texture, and boundary shape to the large target image. Previous mosaic algorithms have focused on the aesthetic aspects of the result and performance of layout algorithms. Limited attention has been paid to using these usually beautifully generated mosaic images.

Our research starts from the question: how can users see meaningful information through the tile images in a mosaic image and in what ways can they utilize it? Our solution is that users can use it as an interface to deal with large collection of media data. We define 'Interactive Mosaic' as (1) a mosaic that represent some stories or themes which is meaningful to users, (2) an interactive interface that users can follow the stories in it and (3) a media that has semantic relationship between template image and each tile.

The contributions of this paper are the followings: First, we present a mosaic generation tool by which users can create any shape of layout template using various kinds of meta-data and annotated information of scenes when they select tiles. Second, our approach generates the mosaic that represents a visual summary of videos corresponding to certain predefined rules (i.e., scenes in which actors, annotation information and emphasizing index chosen by users exist). By using this summarization, users can easily navigate and browse the videos. Finally, the mosaic can be directly re-generated when users want to change their themes by querying another configuration or emphasizing some area in the mosaic.

Figure 1 shows an example of mosaic image. By selecting a tile image (scene) in the mosaic image(c), user can playback the movie from the scene and see the information about the scene if meta-information is available. It is a simple example of navigating video file such as DVD chapter selection with customized interface. In particular, if users want to generate the mosaic with a specific theme, they can make the mosaic by gathering tiles depending on specific meta-data and conditions. For example, making a mosaic of Christmas tree with baby's images from family videos only requires you to select Christmas tree image and query the name of the baby (assuming such meta-data exists).

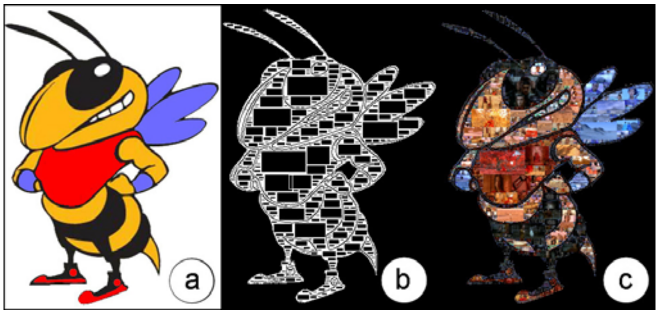


Figure 1 Example of a mosaic made by our approach

(a) Template image (b) Labeled layout image (c) Mosaic image

MM'06, October 23–27, 2006, Santa Barbara, California, USA.
Copyright 2006 ACM 1-59593-447-2/06/0010…$5.00.

2. RELATED WORK

Current mosaic algorithms emphasize the use of different packing (layout), matching and shapes of tiles to generate mosaics. Photomosaic [9] divides the original images into certain set of rectangles with same sized tiles and calculates the average color in each grid area. However, their approach is limited to square tiles on a grid. Hausner [6] shows the mosaic algorithm using tile-positioning-based Centroidal Voronoi Diagram(CVD) and applies direction field with packing similar-shaped objects. Kim and Pellacini [7] took tile and container image as an arbitrarily shaped object in packing and suggested optimal search algorithm to pack. Finally Smith and Liu [10] applied area-based CVD in packing, and they showed animated sequence where each sequence is made by mosaic. As mentioned above, each of their approach is focused on beautiful-looking results and on the performance of the algorithm while ours covers not only aesthetics but also usefulness of the mosaic image. However, in our approach, there exist some trade-offs between aesthetic aspects and usefulness. For usefulness, We have some constraints (1) Each tile image should be an image or scene from video stream (2) Tile image should not be rotated or cropped (3)Each tile's shape should not be changed but size can be re-sized as long as its aspect ratio remains the same.

There have been active researches in the area of navigating and summarizing video because of huge increase in amount of individual archive. Arman [2] focused on browsing video contents by analyzing key-frames but only presented them along different frames by scoring similarity. Some approaches summarized the videos by categorizing frames by semantic events and visual similarity and packing them with differently sized frames in comic book styles [3,11]. Yeung et al [13] suggested pictorial summaries of videos using clustering techniques. They created pictorial summaries of video voting by "dominant score" in each frame and determined the size along with such voting. There are also some approaches dealing with aesthetic and usability by making collage style layout for summarizing video frame or images. Fogarty [5] made image collages by defining heuristic rules for making layouts and Diakopoulos [4] suggested Photo Collage authoring tools which uses meta-data in making layouts with dynamic query mechanism. Our approach is extended from their work in the sense of generating story-driven layout with annotation information. In particular, our approach has been adapted and tested to use meta-information from annotations made by the Family Video Archive system (FVA, Abowd et al. [1]) and this data directly influences the choice of weighting parameters in the matching step and the labeling step. Mosaic can be seen as another way of making layout for media summary. However, unlike other layout approach, mosaic allows users to make layout in many ways with keeping the layout image has meaningful semantics (i.e., logos) and it leaves many ways for users to summarize videos.

3. THE APPROACH

The overall data flow for our approach is shown in Figure 2. At first, the template image is segmented by color. The segmented image is now the container image for our mosaic. Once the container is made, each segmented area is divided into various sizes of rectangle by packing algorithm. These rectangles are modified later by additional manipulation if user wants to change the distribution of tiles. After tiling is performed, each tile area is labeled by number, size, position and color distribution of original template image. The feature selection process gathers annotation information and attributes of each frame or image from media collection by analyzing them. This feature data is made by two methods. The first method acquires the features automatically in Video streams such as movie file, DVD and huge sets of image files. By automatic annotation, we can get the information about mean and variance of color and attributes such as face existence [12], frame number, physical file location of the video file or images. The second method is using a system such as the Family Video Archive (FVA) [1] to get annotation information manually. By FVA, we can manually tag the annotation information such as names of people appeared in video scene, time and date, back ground history of the film and etc. During the matching step, each tile area in the result of the packing is replaced by an image from the media collection. This image is selected by first considering the annotation information and feature attributes of candidates extracted from feature selection step and then calculating similarity of color distribution. After the mosaic image is made, users can navigate the video or image files. Moreover, users can customize the mosaic for their own way by manipulating the allocation of tiles or selection of tile images.

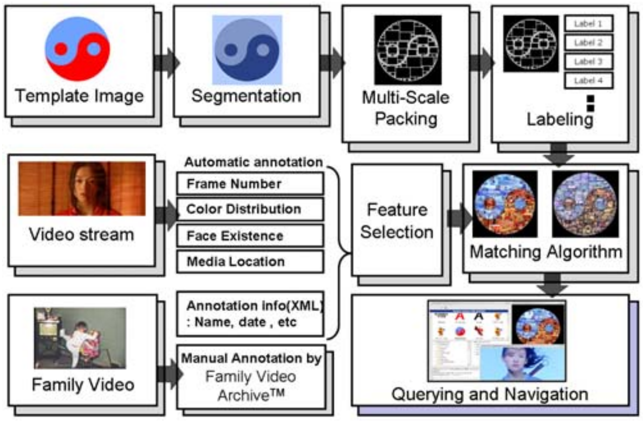


Figure 2 Application diagram

3.1 Packing and Labeling Algorithm

Our packing algorithm does not use deformable tiles, but instead fits various sizes of tiles that keep the original aspect ratio. This makes easier for users to recognize the tiles and select scenes. Initially, there are two look-up tables where the dimension of the tables is the same as that of the template image. Elements of the first table are filled with the value $C_{(i,j)}$,which denotes segment number of template image in *(i, j)th* pixel. This table is defined as "Segment table". Second one is "Flag table" and it is set to one initially. When tile area is determined, every occupied element in flag table is set to be zero. Once users determine the maximum tile size as H_{max} by W_{max} with certain aspect ratio, the iteration starts with the size. In each iteration, H and W are decreased with keeping the ratio of rectangle until the rectangle becomes a pixel. In N_{th} iteration, we keep moving the window, which is H_N by W_N sized rectangle through the segment table. In each step of moving the window, algorithm checks the following condition (1):

$$\sum_{j=0}^{H_N} \sum_{i=0}^{W_N} C_{(i,\,j)} \cdot flag_{(i,\,j)} = H_N \cdot W_N \cdot C_{(left,\,top)} \tag{1}$$

If (1) is true, a H_N by W_N rectangle with the top-left corner coordinate as *(left,top)* can be packed inside the segmented area and assigned a labeling number. We repeat this step for a given iteration and pack as many tiles of same size as possible.

However, in most cases, we do not need to calculate (1) by checking all pixels in the window.

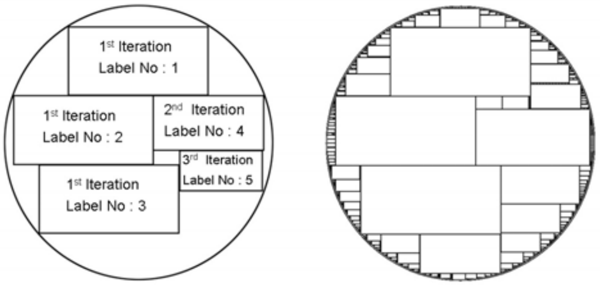


Figure 3 Examples of Packing and Labeling

We can safely pack a rectangle if the summation of $C_{(i,\ j)}{\cdot}flag_{(i,\ j)}$ along the boundary is equal to $(2{\times}(H_N + W_N) - 4){\cdot}C_{(left\ ,top)}$, where $2{\times}(H_N + W_N) - 4$ is the number of boundary pixels in N_{th} candidate window. Because the size of the rectangle is decreasing after each iteration, no previously packed rectangles can possibly exist in the interior of the window area in consideration. This approach can speed up the labeling process. However, this approach cannot be used when users manipulate the mosaic by changing packing layout manually after first packing process is done. Figure 3 shows the examples of packing after the third iteration (left) and full iterations (right). After the third iteration is over, total labeling number became five because three same size tiles are determined at first iteration.

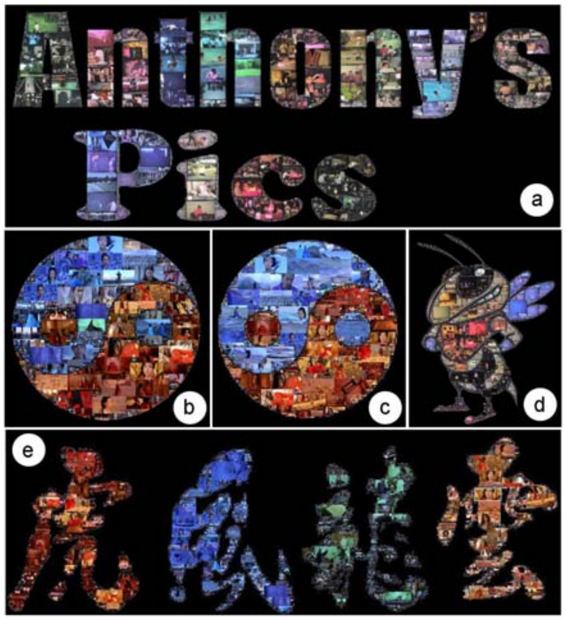


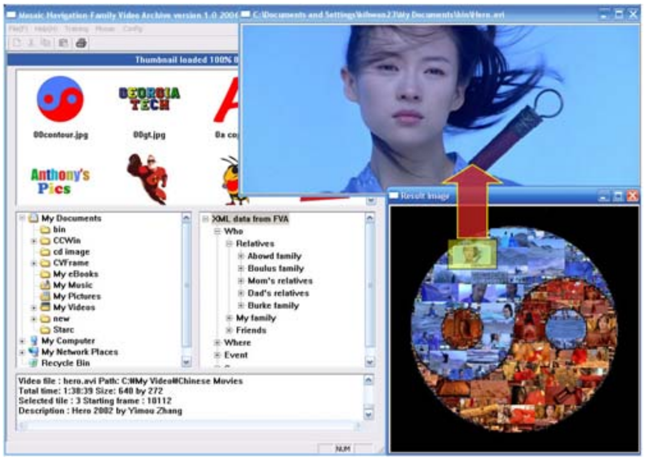
Figure 4 Examples of matching results corresponding to various filtering and constraints

3.2 Matching using Meta-information

In the matching step, the algorithm votes for each tile to determine most appropriate tile image both for users and aesthetic result of the mosaic. In this step, users can provide conditions which determine information and attributes to be considered. First, users filter image candidates according to the provided constraints, such as face existence, physical location, owner's name and name of appearance. Then, the algorithm searches the best image by calculating the distance between a candidate image and the template image. This distance is a weighted sum consisting of each component of RGB and HSV, standard deviation of intensity, frame number off-set and time line. Some of our matching results are shown in Figure 4. Figure 4-(a) is the output of collecting tile images as a result of querying word "Anthony" as person appeared in the scene and "1960 – 1980" as date tag. (b) is the result of filtering "face existence" through the movie "Hero(2002)",which is performed by face detection algorithm using Viola's algorithm [12]. (c) is only using color information on the "Hero" and (d) is output of making Buzz – the mascot of Georgia Tech - image by date constraints as "1950 – 1980" from every family videos. (e) is the result of using color and frame number on "Hero(2002)".

3.3 Navigation and Summarization

Once the mosaic is generated, users can navigate and browse their media. In our application, user can select any tile images in the mosaic and select the navigation menus in popup. The navigation menu is consists of (1) Watch scene, (2) Playback from this scene, (3) Find nearby frames and (4) Display the annotation information of this scene. (1) is watching only one original frame or scene from the file and (2) playbacks video file starting from selected scene. When users click a tile image in the mosaic and select the (3), the application shows all earlier and later frames appeared in the mosaic within the user-specified frame offset of the selected image. (4) is the menu that users can see every annotation information of selected scene including physical location of the media. For example, if users filtered by face-existence constraint, they can navigate the mosaic formed from the scenes that contain people. In this case, users can selectively see the scenes that their favorite actors on it or can see people in the family videos. Figure 5 shows some examples how users can navigate their own rich media by mosaic.

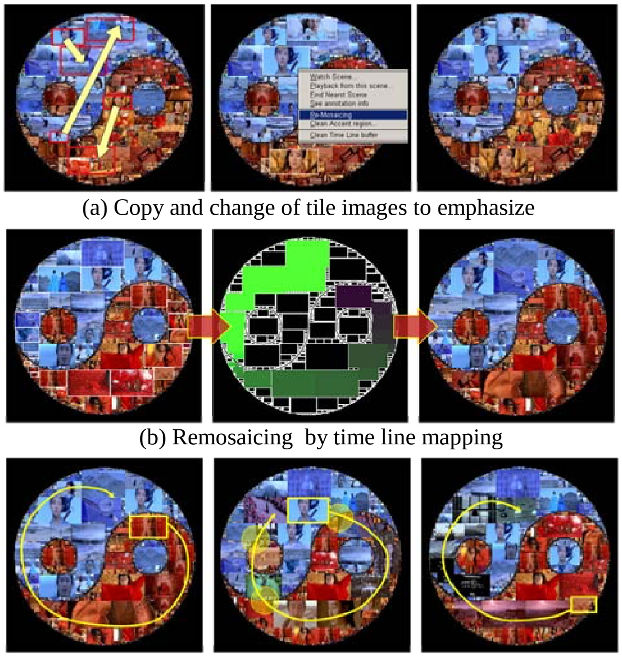


Video file : hero.avi Path: C:\My Video\Chinese Movies
Total time: 1:38:39 Size: 640 by 272
Selected tile : 3 Starting frame : 10112
Description : Hero 2002 by Yimou Zhang

Figure 5 Navigation and browsing videos

3.4 Remixing mosaic

Remixing or re-editing mosaic is useful when users want to customize the mosaic using other conditions, weights on matching step or emphasize some tiles. Figure 6-(a) shows how to change the size and position of some tile images to emphasize it. Figure 6-(b) shows that users can reshape certain area following to timeline rule by assigning timeline mapping – darker area is associated with earlier frame while lighter area is later frame. In non-annotated video stream, this time line sequence follows frame number or elapsed time while it follows date information of each frame in annotated videos. So once time line is applied in the area, the weighted-parameters in matching step is changed to make timeline parameters be prior to color distribution in voting. Figure 6-(c) shows some results of time-line formulation. However, middle and right images of Figure 6-(c) shows somewhat undesirable result while left image shows desirable one. The middle image has three abrupt color changes (red to blue or blue to red) so that some tile images on the time-line has inadequate color distribution than others because it considers one more constraint during matching step. The right image of Figure 6-(c) starts with almost at the end of stream. Thus, later part of time-line has some mismatched tile images due to lack of frame samples at the end of the stream.



(a) Copy and change of tile images to emphasize

(b) Remosaicing by time line mapping

(c) Examples of time line formulatioin

Figure 6 Remixing mosaic

4. RESULTS

The result images under various constraints and filters are shown in Figure 1~Figure 6. As shown in Figure 4-(a), the template image which represents the name of person has tile images that have his pictures in certain period. This result shows us that the semantic meaning of each tile can be representative of template image. 4-(b) also shows us the mosaic which is summarized by the scenes where actors appeared. We also presented navigation interface in the mosaic image in Figure 5 where users can select the scenes in the movie. Finally, we also showed another approach of customizing mosaic when users need additional manipulation on the mosaic in Figure 6. In the training step, we trained 800 key-frames from the movie “Hero(2002)” for Figure 4-(b),(c),(e) and 335 family video files with overall 1600 key-frame images for Figure 4-(a),(d). Clearly, in aesthetic view, the output image is relatively less beautiful than that of using only color information when we use more conditions to make mosaic (Figure 4-(b) and (c)). But outputs still show desirable results following the definition of interactive mosaic. They give more chances for users to customize the mosaic which contains themes and summarization of videos. They can also be an interactive interface for users to navigate rich media.

5. FUTURE WORK and CONCLUSION

In this paper, we introduced an approach for navigating and summarizing videos through mosaic generation algorithm. We discuss how users can use this mosaic image in navigating media with additional annotation information. However, the annotation information to make themes for our approach has a room for further improvement. Our future work will include other techniques to help classify scenes in video streams during the feature selection step, so that users can gather more information about the scenes automatically. We could also extend our work to make a zoomable mosaic interface so that users can navigate more easily.

6. ACKNOWLEDGEMENTS

The authors are very grateful to Dongryeol Lee for his help in evaluating algorithm and application and to Nick Diakopoulos for his feedback and support during the test.

7. REFERENCES

[1] Abowd, G. D., M. Gauger, et al. The Family Video Archive: an annotation and browsing environment for home movies. *In Proc. of the ACM international conference on Multimedia,* 2003, 1-8.

[2] Arman, F., R.Depommier, et al. Content-based Browsing of Video Sequences. *In Proc. of the ACM Multimedia*, 1994, 1-8.

[3] Boreczky, J., A. Girgensohn, et al. An interactive comic book presentation for exploring video. *In Proceedings of the SIGCHI* 2000, 185-192.

[4] Diakopoulos, N. and I. Essa. Mediating Photo Collage Authoring. *In ACM UIST'05*, 2005, 183-186

[5] Fogarty, J., J. Forlizzi, et al. Aesthetic information collages: generating decorative displays that contain information. *In ACM UIST 2001*, 2001, 141-150.

[6] Hausner, A. Simulating Decorative Mosaics. *In Proceedings of ACM SIGGRAPH '01*, (Los Angeles, CA), 2001, 573-580.

[7] Kim, J. and F. Pellacini. Jigsaw Image Mosaics. *In Proceedings of ACM SIGGRAPH '02*, (San Antonio, TX) ,2002, 657-664.

[8] Shipman, F., A. Girgensohn, et al. Generation of interactive multi-level video summaries. *In Proc. of the ACM Multimedia*, 2005, 392-401

[9] Silvers,R and Hawley, M. Photomosaics, Henry Holt and Co., 1997

[10] Smith, K., Y. Liu, et al. Animosaics. *In Proc. of the ACM SIGGRAPH/ Eurographics symposium on Computer Animation,* 2005, 201-208

[11] Uchihashi, S., J. Foote, et al. Video Manga: generating semantically meaningful video summaries. *In Proc. of the ACM international conference on Multimedia*, 1999 383-392.

[12] Viola, P. and M. Jones. Rapid object detection using a boosted cascade of simple features. *In Proc. of the International Conference on Computer Vision and Pattern Recognition*, 2001, 511-518.

[13] Yeung, M. M. and B. L. Yeo. Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content. *IEEE Trans. Circuits and Sys. for Video Techonology*, IEEE Circuits and Systems Society, NJ 1997. 71-78