互動式視訊多媒體拼貼

羅聖傑
國立臺灣大學
forestking@cmlab.csie.ntu.edu.tw

蔡淳宇
國立臺灣大學
apfelpuff@cmlab.csie.ntu.edu.tw

陳維超
順德工業 / 國立臺灣大學
weichao.chen@gmail.com

陳炳宇
國立臺灣大學
robin@ntu.edu.tw

ABSTRACT

隨著人們普遍的使用照相機與攝影機以及網路的普及化，人們自製和分享多媒體影音也更加的容易。由於每個人所擁有以及觀看的多媒體影音內容數量越來越多，種類也越來越複雜，如何有效率地觀看這些多媒體內容也成了一個需要解決的課題。在本論文中，我們基於兩個目的提出了一種以拼貼和互動的方式呈現多媒體影音內容的方法。第一個目的是，由於希望多個視訊多媒體能在同一個畫面呈現，我們希望能更有效的利用畫面的空間，此即每個視訊多媒體不重要的區域可被遮蓋住。第二，我們希望在視訊多媒體播放的同時，可以提供一些互動功能讓使用者在觀看的時候更有參與感。此拼貼和互動的方法分為三個步驟： 一、我們首先對單個影片做分鏡偵測（shot detection）： 由於我們希望畫面上每一個影片都以一個分鏡為單位，以利拼貼的空間利用和視覺效果，此部份我們採用顏色統計差異（color histogram difference）的方法來把每個影片分成多個分鏡。 二、找出每個分鏡中的顯著區域（saliency region）：為了拼貼空間的有效利用，我們對每個影片做一些事前處理，希望保留影片中物體的移動並使重要內容盡量在分鏡的中心播放以利拼貼空間的使用效率。 三、設計互動式隨機拼貼的演算法，使得使用者在觀看多個影片拼貼結果的時候，能夠任意添加、刪除和移動這些影片分鏡。

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI); I.2.10 [**Vision and Scene Understanding**]: Video analysis; I.3.5 [**Computational Geometry and Object Modeling**]: Geometric algorithms, languages, and systems; H.3.2 [**Information Storage**]: File organization

1. INTRODUCTION

The popularity of digital cameras and video recorders put the power of media creation at the hands of millions of people. On the one hand, we can now record our surroundings and share them almost instantly through the Internet. On the other hand, with the sheer amount of the contents we are exposed to nowadays, we now face an increasing burden to categorize, browse, or look at our own personal video and photo collections.

To reduce the effort of managing the ever-growing media collections, many commercial applications started to incorporate search interfaces to retrieve pre-tagged media files. Keyword tags, locations, or face identities are popular methods particularly for Internet applications such as Flickr where people wish to have their uploaded media files noticed and discovered [2]. However, the majority of personal media files remain untagged, and they can become forgotten in the presence of those that are heavily tagged. What we find lacking, therefore, is a new breed of image browser that allows us to quickly rediscover what we already have within our own piles of untagged media store.

There have been various prior research that are useful for our purposes. For example, image collage research such as [19] allow us to generate compact and aesthetically pleasing views of image collections. Image retargeting techniques such as seam carving [20] focus on reducing the size of images without throwing away important contents. Similarly, video summarization methods focus on producing the gist of videos by either reducing their length or by summarizing each of them into one, or a collection, of images [4, 22].

Our goals are related to the research stated above, with several important distinctions. First, ours goal is to create a tool that allows a user to simultaneously display several media files at the same time. As such, the tool should not dictate what is be displayed where. Instead, it ought to respond to user requests such as insertion, deletion, and rearrangement while using the screen real-estate effectively. Second, we wish to reduce the effort and time for a user to browse through video collections, and therefore we are not limited to a specific category of video summarization techniques – we can choose to break up a single video into multiple images or even video segments, and they can all be displayed on the same screen canvas simultaneously. Figure 1 shows a few sample results generated by our system.

2. RELATED WORK

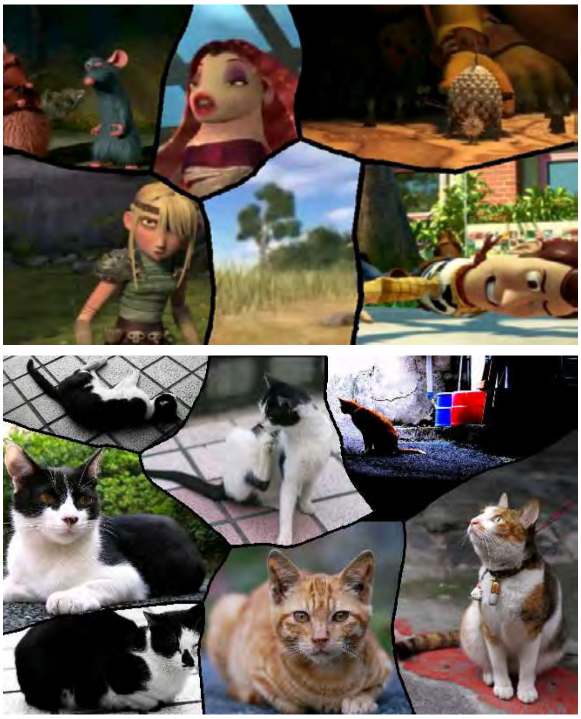


Figure 1: Example assemblages generated by our system, including a set of cartoon movie trailers (upper) and photos on cats (lower). These are generated automatically and can be interactively manipulated by the users.

Automatic Image Collage. An image collage refers to an image created from an assemblage of a collection of images. A variety of automatic image collage techniques have been developed both for research and commercial purposes. Google's Picasa[1], for example, incorporates a feature that generates collages of complete input images. It also provides different composition styles to the users. Atkins [3] proposed an efficient method of organizing images in a page. Wang *et al.* [24] presented picture collage, which optimizes the layout of rectangular images to maximize the portion of salient regions in the result. Battiato *et al.* [5] improved the result of picture collage by exploiting semantic information to compute the saliency. Rother *et al.* [19] presented AutoCollage, a method for constructing a seamless collage from input images.

Video Summarization. Truong and Venkatesh [22] provided an excellent review of video summarization techniques, which are divided into two classes, namely video skims and still image summaries. Video skims generate a shorter summary video to summarize the whole video, while still image summaries extract a number of keyframes from a video to pack the summary image. For video skims, Christel *et al.* [7] presented studies that measure effectiveness of video skim techniques. Divakaran *et al.* [9] devised a method to adjust video framerates by analyzing temporal motion activity, and speed up parts of the video with less activity. Peker and Divakaran [18] used motion activity as well as various semantic cues such as face, skin color, or speech to control the video playback rate.

The second class of techniques, image summaries, is related to image collage and similar to our applications in several respects. Zhu *et al.* [27] proposed the video booklet system, which extracts a number of thumbnails from a video, and then reshaped by a set of predefined shape templates. Wang *et al.* [25] presented video collage, which blends the selected images to produce seamless video summary. Yang *et al.* [26] and Mei *et al.* [17] both extended the seamless blending to generate arbitrary shape collages. Correa and Ma [8] presented a method to interactively generate seamless video summaries. Barnes *et al.* [4] proposed a method to automatically generate video tapestries that allow for continuous panning. Chiu *et al.* [6] generated non-triangular layout to effectively summarize the salient regions. Kang *et al.* [13] proposed the space-time video montage.

Our Contributions. Our goals are similar to automatic image collage in that we wish to preserve salient regions on the canvas. Additionally, we have to incorporate both images and videos in our input, and automatically generate a collage that we can efficiently edit and manipulate afterwards. This means we have to forgo some of the more time-consuming techniques such as graph-cut, Poisson blending and Markov chain optimizations used by several image collage research. In the end, ours is an incremental algorithm that iteratively computes locally optimal collage configurations. We also choose to play the videos as-is without skimming, and we do not consider this a drawback because we can readily transform our input videos with any suitable skimming techniques. Specifically, our contributions are as follows.

- We propose a novel method for analyzing temporal-spatial salient regions of a video,
- a method for extracting the temporal-spatial salient regions while removing apparent camera or object motions,
- an efficient, greedy technique for packing a collection of irregularly-shaped visual media, and
- a scheme to iteratively optimize the packing when it is disturbed.

3. ALGORITHM

We design our media assemblage techniques based on a few visual guidelines. First, we wish to reduce visual complexities while navigating these media. This means the non-essential parts of a video can be covered up or eliminated in the assemblage. We also plan to support interactive operations such as addition, deletion, and rearrangement, and while the assemblage would change during these operations, its layout should stabilize and stay static soon after these operations are complete so as to minimize disturbances while playing individual videos within the assemblage. For this purpose, the essential, or salient, region of a video needs to

[1] http://picasa.google.com/

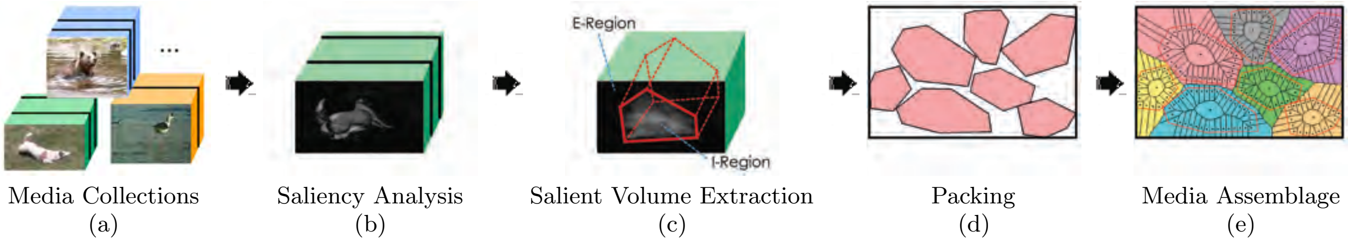


Figure 2: The overview of our assemblage system. (a) The media collections with shots are detected. (b) The saliency analysis is performed on every frame. (c) The salient volumes are extracted in every shot. (d) These media are packed into the canvas. (e) The final assemblage is generated.

have a static outline throughout its timeline. To ensure efficient extraction of the salient regions, we need to be aware of the camera motions for videos where the subject matters are moving.

Starting with a set of videos and photos $\mathbf{M}$, our system computes a configuration $\mathbf{X} = \{p_i, R_i\}_{i=1}^{|\mathbf{M}|}$, where p_i is the position and R_i is the high salient region of the i-th element of $\mathbf{M}$. Our goal is to find the optimal configuration $\mathbf{X}^*$ subject to a packing energy E,

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} E(\mathbf{X}). \tag{1}$$

Figure 2 illustrates the steps of our algorithm. We treat photos as static videos, and perform the process in two stages, namely *media analysis* (Figure 2(a-c)) and *media packing* (Figure 2(d-e)). The *media analysis* stage is a preprocessing step designed to discover informative regions within each individual videos. Given an input video, we use color histogram to split it into a number of video shots. Then, for each individual shot i, we compute temporal-spatial saliency of its frames (Figure 2(b)) and extract the region of interest R_i by considering the saliency distribution within the shot (Figure 2(c)). The *media packing* stage combines all the input media and efficiently packs them together. We first compute an initial packing by greedily minimizing the blank region on the canvas, followed by an iterative process that adjusts the packing configuration $\mathbf{X}$ according to the packing energy E (Figure 2(d)). Finally, the system decides what regions are visible for every media and generates a final assemblage (Figure 2(e)). In the rest of this section, each step of the algorithm will be described in more details.

3.1 Video Shot Detection

As stated before, we would like the salient boundaries of each individual element in the final assemblage to stay fixed while playing back videos. For this purpose, we want each element in $\mathbf{M}$ to be as coherent as possible, and ideally each element should consist of only one single shot. After experimenting with several possible methods, we found the method by Lienhart [14] to be effective for our application. This technique measures color histogram differences between two adjacent frames and declares a shot boundary when a large color discontinuity occurs.

3.2 Saliency Analysis

Psychological studies show that visual signals contrast such as motion and color are likely to attract people's visual attentions [21]. We adopt similar visual attention formulation by Liu *et al.* [15] to compute a saliency map per frame per video, and emphasize the salient regions in the final assemblage in order to utilize the 2D canvas more efficiently. In this method, the saliency of each pixel p is calculated as a weighted sum of the motion contrast saliency (S_M), the image saliency (S_I) and the face saliency (S_F), as follows

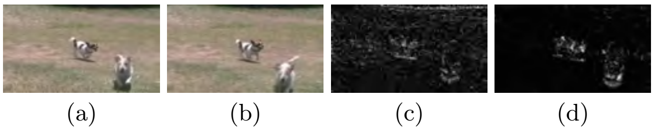


Figure 3: The examples of saliency analysis on the video frame. (a) The input video frame i. (b) The input video frame $i+1$. (c) The original motion magnitude of frame i. (d) The motion contrast magnitude of frame i.

$$S(p) = w_M S_M(p) + w_I S_I(p) + w_F S_F(p). \tag{2}$$

We use $w_M = w_I = w_F = 1/3$ in our implementation. Our method differs in that we use a simple panning motion model to preserve the styles of the original shot, and we adopt a different image saliency measure. An example of this process is shown in Figure 4.

Motion Contrast Saliency. Moving objects should be assigned higher saliency values because humans are particularly good at perceiving them. This behavior is encoded as motion contrast saliency as shown in Figure 3. First, we use Lucas-Kanade method [16] to analyze the relative motion between two adjacent frames, and then approximate a global camera motion by using a voting scheme where the motion vectors are used to vote both on a consensus motion direction and magnitude. The motion contrast is then obtained by subtracting the original motion vector with the global camera motion and normalized to $0 \sim 1$ into motion contrast saliency.

Image Saliency. There exist various methods to measure image saliency based on low-level feature contrast [12, 11, 1, 10]. We choose the approach by Achanta *et al.* [1] which calculates the saliency of each pixel based on its color and luminance differences with respect to its neighbors. Figure 4(c) shows the image saliency results.

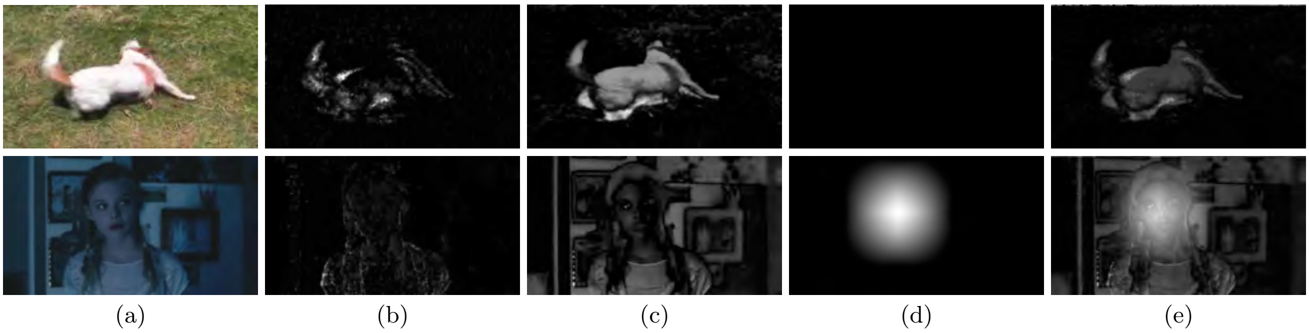


Figure 4: Examples of our saliency analysis process. (a) The input video frames. (b) The motion contrast saliency S_M. (c) The image saliency S_I. (d) The face saliency S_F. (e) The combined saliency S.

Face Saliency. To emphasize the importance of human faces, we detect the presence of faces with the methods proposed by Viola [23]. The face saliency value is then calculated by applying a Gaussian attenuation function surrounding the area of the detected faces. Figure 4(d) shows the image saliency results.

3.3 Salient Volume Extraction

Given the saliency map of each individual frame, our next goal is to extract volumes of the video for the following packing stages. For a video shot where a single salient object moves at a constant speed, we may compute the cumulated saliency S_c along a temporal skew (x, y),

$$S_c(i,j) = \sum_f S_f(i + fx, j + fy), \qquad (3)$$

where $S_f(i, j)$ is the saliency value of the f-th frame at pixel index (i, j). An optimal direction (x, y) is where the cumulated saliency values are concentrated in a region as small as possible. To determine this direction, for each frame we iteratively select the pixels with highest saliency values until the sum of these values exceeds half of the total saliency values within this frame, and construct a bounding box using the selected pixels. Then, we fit a least-square line over the centers of the bounding boxes over time, and use the line direction as the optimal direction to align the video volume, accumulate the saliency values, and determine the region that should be preserved in the final assemblage.

To calculate this region, we again select those pixels with highest cumulated saliency value until the sum of these values reaches a predefined threshold of say, in our case, 50% of the sum of cumulated saliency from all the pixels. We then construct this *I-Region* R_i from the convex hull of the selected pixels. The region outside the *I-Region* is defined as the external region *E-Region*, whose pixels can be discarded when a tighter packing is desirable.

3.4 Packing

After extracting the salient volumes, we pack the media set $\mathbf{M}$ by following a few criteria. First, we wish to use the canvas space efficiently. Second, salient regions of the media should never be occluded. Third, the canvas should observe the aspect ratios of the display devices. With these goals in mind, our objective is to find an optimal configuration $\mathbf{X}^*$ without occluding each of the *I-Regions* in $\mathbf{M}$. This packing problem, unfortunately, is a NP-complete problem, and we propose to approximate the optimal solution by a two-stage heuristic. First, we use a greedy algorithm to initialize a layout configuration. Then, this configuration is iteratively optimized to reach a local minimum.

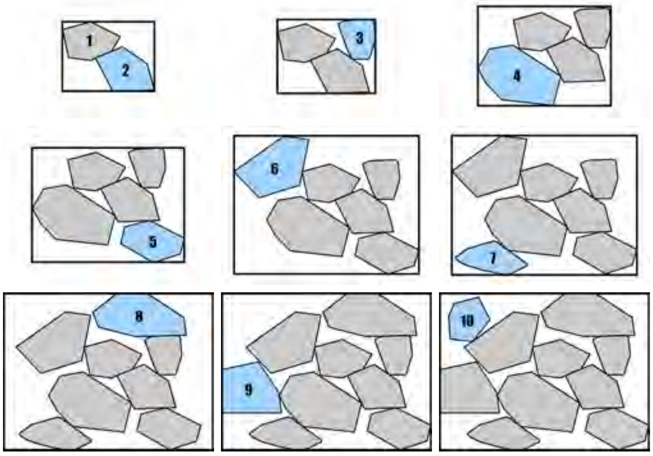


Figure 5: An initial packing example.

3.4.1 Packing Initialization

As shown in Figure 5, the packing initialization process is as follows. The first *I-Region* is first placed at the center of the canvas. Then, we place each remaining *I-Region* R_i radially around the canvas center while ensuring no overlap between R_i and all other *I-Regions* already on the canvas. We pick an optimal direction that minimizes the empty space while respecting the aspect ration of the canvas. Figure 6 shows two examples of the packing initialization under two different pre-selected aspect ratios of 4:3 and 3:1, respectively.

3.4.2 Packing Optimization

After initialization, we iteratively optimize for the configuration $\mathbf{X}^*$ by randomly selecting an *I-Region* and moving it toward a unit direction that reduces the packing energy E by the greatest amount. Each step of this process is guaranteed to reduce the packing energy, and we repeat the process until it stabilize to a local minima. Based on the packing criteria described before, we design our energy function based on a combination of penalty measures on empty space, *I-Region*

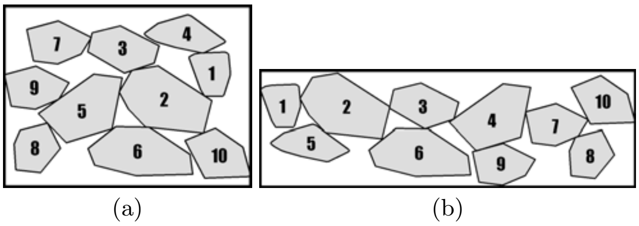


Figure 6: The results of packing initialization with different preferred aspect ratio: (a) 4:3; (b) 3:1.

occlusion, and aspect ratio deviation, as follows

$$E = E_{es}^{\alpha} + kE_{occ} \tag{4}$$

where the occlusion weight k is set to 1×10^5 in our implementation to ensure that the *I-Regions* never get occluded. We now describe each of the energy terms (E_{es}, α, and E_{occ}) in the following paragraphs.

Empty Space Penalty. We define the empty spaces on the canvas as regions that are not covered by any *I-Region*. This energy term represents the percentage of empty spaces on the canvas, as follows

$$E_{es} = \frac{Area(R_B - \cup_i R_i)}{Area(R_B)}, \tag{5}$$

where R_B is the bounding box formed by all *I-Regions* R_i.

***I-Region* Occlusion Penalty.** This energy term penalizes coverage of salient video regions, and is simply defined as the total areas of covered *I-Regions*,

$$E_{occ} = \sum_i Area(R_i) - Area(\cup_i R_i). \tag{6}$$

Aspect Ratio Deviation Penalty. The optimal packing should respect an aspect ratio specified by the user. This can be described by the following term

$$\alpha = \frac{1}{(q_c - q_p)^2 + \varepsilon} \tag{7}$$

where q_c is the aspect ratio of the bounding box R_B, q_p is the desired aspect ratio, and we use a small number $\varepsilon = 10^{-6}$ to set an upper bound for α. Since the magnitude of the empty space penalty E_{es} is always less than 1, the first term in Equation 4 becomes very small once we approach the desired aspect ratio.

3.5 Media Assemblage

Now that we have a layout configuration of the media set, we may begin generating an assemblage on the canvas. The choice of rendering styles is a rather artistic one. For example, AutoCollage adopts a seamless blending style between adjacent images [19]. For our purposes, we need to clearly distinct media boundaries for playback and interaction purposes. Simply rendering all *I-Regions* may suffice, but we would like to fill up as much empty spaces as possible by rendering non-essential *E-Regions*.

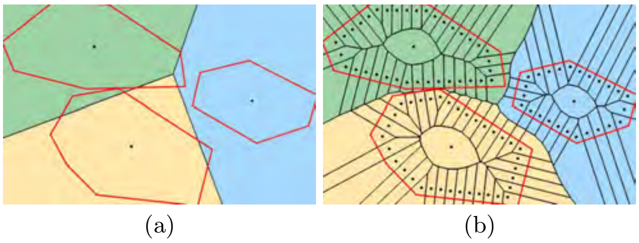


Figure 7: Comparison of assemblage methods. The red polygons are *I-Regions*. (a) A straightforward Voronoi segmentation. Notice some *I-Regions* are eroded in this example. (b) Our approximated region-based Voronoi approach.

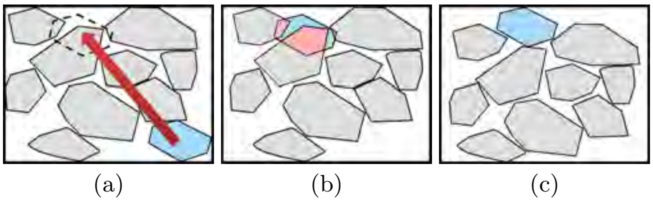


Figure 8: Interacting with the assemblage. The user drags the blue region (a), causing several salient regions to become occluded (b). Our system iteratively refines the assemblage and resolves the problem in just a few iterations (c).

A straightforward approach is to segment the canvas using complete *I-Regions* as Voronoi sites. As this method proved to be too slow to run at interactive rate, we approximate this algorithm by sampling a number of Voronoi sites along the bounaries of the *I-Regions*. This approach, in addition to its speed advantages, has an additional benefit where we can control the smoothness of Voronoi region boundaries by changing the sampling rate of the sites. Figure 7 shows the assemblage using our approach (b) compared to a simple Voronoi segmentation (a). Notice that the sites are sampled a short distance away from the *I-Region* boundaries to prevent nearby regions from eroding into each other.

4. RESULTS

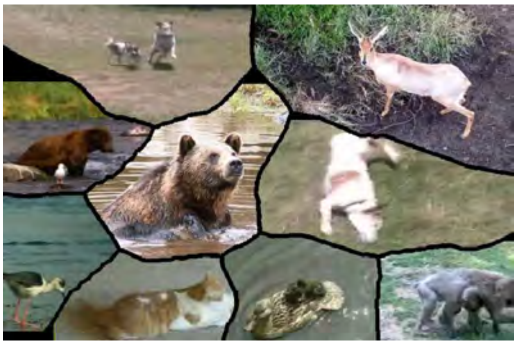


Figure 9: An assemblage of animal videos.

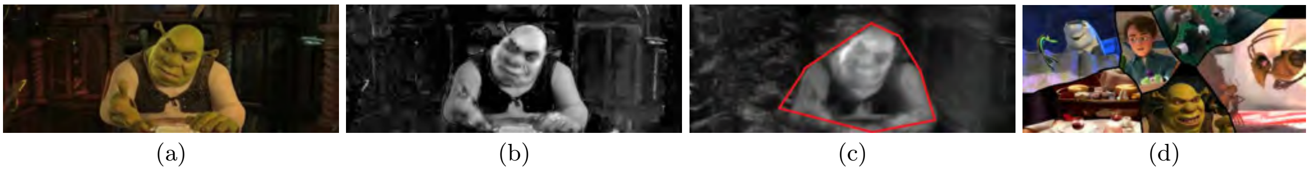


Figure 10: Assembly of a collection of movie trailers, including (a) one frame from a movie trailers, (b) its saliency map, and (c) the extracted volume and the *I-Region* (red polygon). (d) shows the final assemblage of the collection.

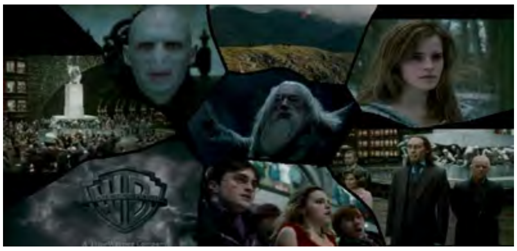


Figure 11: Summarization of a movie trailer. Regions of videos in this assemblage are different video shots come from the same trailer.

In this section we present our results as well as several applications. Figure 10 shows the intermediate results of summarizing a collection of movie trailers, including saliency maps (b), *I-Region* (c), and final media assemblage (d). Our system creates interactive assemblages that users can interact with via insertion, deletion, and dragging operations, and our system automatically adjust the layout to the next optimal configuration after these operations. Figure 8 shows an example of dragging and dropping of a media file, and our system's response to refine the configuration.

In addition to these manipulation operations, our system can be used for a variety of applications such as video collection summarization, single video summarization, personal media folder visualization, and interactive video board. We now discuss a few of the possibilities as follows.

Media Collection Presentation Our system can provide a presentation of a set of video collections and dynamically play these videos. A user can preview all these videos simultaneously on a single screen and then modify the presentation as she sees fit. Figure 10(d) and Figure 9 are a few example of video collection summarization.

Single Video Summarization. Our system can summarize a single video by assembling each individual shot onto a canvas. Unlike traditional video summarizations, which select some key frames and summarize them with still image collage, our approach can play all shots simultaneously, or sequentially with time overlaps, and the users can get a quick temporal review through this dynamic summarization. Figure 11 shows an example of summarizing a video.

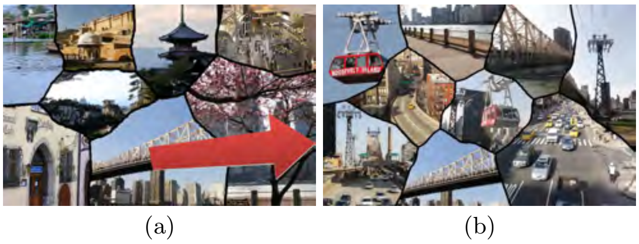


Figure 12: A media file browser application. (a) Preview of a folder with travel photos and videos. After clicking on the bridge photo, the system brings up an assemblage of media files related to the trip to New York (b).

Personal Media File Browser. Figure 12 shows an example of visualizing a collection of media related to traveling. As a user browses through this collection in the root folder, she can choose to preview representative photos and videos from different trips (a). She can click on one of the interest region on the assemblage which brings up media files from the sub-folder, presented as another assemblage (b).

4.1 Comparisons

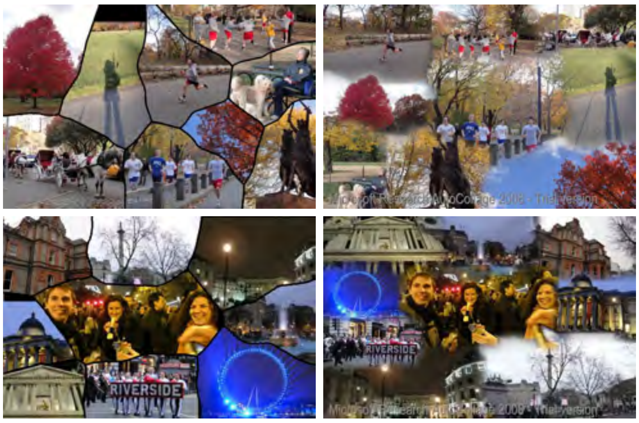
Here we present a qualitative comparison between AutoCollage [19] and our approach. AutoCollage operates on images, and we generate both of the results using the same photo sets. Figure 13 shows the comparisons on two different photo sets. The most obvious difference is an aesthetic choice where AutoCollage generates seamless collages while ours have clear boundaries around the photos. Both AutoCollage and our system are completely automatic. However, users can interactively adjust and refine the assemblage generated by our system, and this is not possible with AutoCollage.

4.2 Video Results

Readers are strongly recommended to watch our video results via the following anonymous link:
http://www.youtube.com/watch?v=9tzazErlLCE

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a dynamic media assemblage method for summarizing and presenting visual media interactively. We analyze the temporal-spatial salient regions within each shot for more efficient packing. Our energy function and iterative optimization process guarantees occlusion-free packing of salient media regions while ensuring appropriate canvas aspect ratio. We also showed that our method can be applied to many applications, such as



(a) (b)

Figure 13: Compare media compilation to AutoCollage. Top row: Central Park. Bottom row: New Year's Eve 2011. Column (a): our results. Column (b): AutoCollage.

image and video collection presentation, single video summarization, and hierarchical media browser.

In our current implementation, the packing algorithm does not respect any user-specified order, and we are working on packing algorithms that take the order into consideration. Furthermore, our algorithm, while being fairly interactive, may get stuck on local minima, and we would like to see a better re-initialization scheme when this happens. Finally, unlike a traditional file browser, a media assemblage is inherently limited by the size of its canvas, and therefore we plan to introduce methods that allow panning and scrolling as well as smart hierarchical layout within the assemblage. With this, it becomes possible to jump seamlessly from file browsers, media previewer and media assemblage browsers, and thus endowing users with more choices of managing their ever-growing media collections.

6. REFERENCES

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[2] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *ACM CHI 2007 Conference Proceedings*, pages 971–980, 2007.

[3] B. Atkins. Adaptive photo collection page layout. In *Proceedings of the 2004 IEEE International Conference on Image Processing*, volume 5, pages 2897 – 2900, 2004.

[4] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics*, 29(4):89:1–89:9, 2010. (SIGGRAPH 2010 Conference Proceedings).

[5] S. Battiato, G. Ciocca, F. Gasparini, G. Puglisi, and R. Schettini. Smart photo sticking. In *Proceedings of the 5th International Workshop on Adaptive Multimedia Retrieval*, pages 211–223, 2007.

[6] P. Chiu, A. Girgensohn, and Q. Liu. Stained-glass visualization for highly condensed video summaries. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, volume 3, pages 2059 – 2062, 2004.

[7] M. G. Christel, M. A. Smith, C. R. Taylor, and D. B. Winkler. Evolving video skims into useful multimedia abstractions. In *ACM CHI 1998 Conference Proceedings*, pages 171–178, 1998.

[8] C. D. Correa and K.-L. Ma. Dynamic video narratives. *ACM Transactions on Graphics*, 29(4):88:1–88:9, 2010. (SIGGRAPH 2010 Conference Proceedings).

[9] A. Divakaran, K. A. Peker, and H. Sun. Constant pace skimming and temporal sub-sampling of video using motion activity. In *Proceedings of the 2001 IEEE International Conference on Image Processing*, volume 3, pages 414–417, 2001.

[10] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2376–2383, 2010.

[11] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[12] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

[13] H.-W. Kang, X.-Q. Chen, Y. Matsushita, and X. Tang. Space-time video montage. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1331–1338, 2006.

[14] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 290–301, 1999.

[15] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *ACM Multimedia 2006 Conference Proceedings*, pages 241–250, 2006.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.

[17] T. Mei, B. Yang, S.-Q. Yang, and X.-S. Hua. Video collage: presenting a video sequence using a single image. *The Visual Computer*, 25:39–51, 2009.

[18] K. A. Peker and A. Divakaran. An extended framework for adaptive playback-based video summarization. In *Proceedings of SPIE Internet Multimedia Management Systems IV*, volume 5242, pages 26–33, 2003.

[19] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. AutoCollage. *ACM Transactions on Graphics*, 25(3):847–852, 2006. (SIGGRAPH 2006 Conference Proceedings).

[20] A. Shamir and S. Avidan. Seam carving for media retargeting. *Communications of the ACM*, 52:77–85,

2009.

[21] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[22] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3:3:1–3:37, 2007.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[24] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum. Picture collage. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 347–354, 2006.

[25] T. Wang, T. Mei, X.-S. Hua, X.-L. Liu, and H.-Q. Zhou. Video collage: A novel presentation of video sequence. In *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo*, pages 1479 –1482, 2007.

[26] B. Yang, T. Mei, L.-F. Sun, S.-Q. Yang, and X.-S. Hua. Free-shaped video collage. In *Proceedings of the 14th International Multimedia Modeling Conference*, pages 175–185, 2008.

[27] C.-Z. Zhu, X.-S. Hua, T. Mei, and X.-Q. Wu. Video booklet: a natural video searching and browsing interface. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 113–120, 2005.