

Fast Content Aware Video Retargeting

Awadhesh Srivastava

Department of Information Technology
Krishna Institute of Engineering & Technology
Ghaziabad

K.K. Biswas

Department of Computer Science & Engineering
Indian Institute of Technology Delhi

ABSTRACT

When a video is displayed on a smaller screen than originally intended, some of the information in the video is necessarily lost. In this paper, we introduce Video Retargeting that adapts video to better suit the target display, minimizing the important information lost. We can remove uninteresting part from video like image with some important modifications to retarget it using seam carving. Instead of removing seam from individual frames we extract seam-surface from the space-time volume. To calculate this surface we use seam carving in association with motion projection with lesser algorithmic complexity.

General Terms

Video, retargeting, screens, layout.

Key words

Dynamic programming, Seam surface, video-frames.

1. INTRODUCTION

With the advancement of technology it is easier to take picture or video in higher resolution, but the displaying these pictures or videos are limited by multiple screen sizes. There are very few methods proposed to display video on various screen size. While cropping may cut some important information from the video, whereas scaling will change the dimension of objects which may lead to unpleasant experience.

Existing work in video retargeting can be divided into two broad categories: cropping and resizing. Cropping uses a sliding window to pan through a scene, which works like a virtual camera focus on salient regions. Resizing adjusts the frames in a non homogeneous manner by either squeezing less salient regions or removing the seams with minimized energy. Michael Rubinstein et. al. [1] proposed very good but computationally expensive scheme to retarget the video, on the other hand you must have complete video sequence to retarget for their approach. Seam-surface is a surface in video cube which is 8-connected not only in XY plane but also in time domain. Removing such surfaces leave very less artifacts. In cropping, Fan [3] and Wang [9] extracted the regions of interest (ROIs) and sent output videos to users adaptively with a “display path”. Liu et al. [4] convert a high resolution film to a normal resolution, keeping the original camera movement by using heuristic penalties. Likewise, a sliding window was employed by F. Megino [5], D. Thomas [8] to create the effects of pan, tilt and zoom from still images. When a single window cannot cover two separate objects in one frame, intermittent black padding is usually applied, which disturbs most viewers. Overall, the cropping methods introduce pseudocamera movements that compromise the original intent of the photographer.

In terms of resizing techniques, Setlur first introduced bi-layer segmentation for scaling the filled-in background and removed objects respectively [6]. This approach is limited as it heavily depends on the segmentation results, which could break the relative proportion between objects. Nonhomogeneous resizing, i.e., shrinking less important regions more, was adopted by Wolf et al. [11], but the pixel-wise mapping approach suffers high computational complexity. Recently, Wang et al. [10] presented an image resizing method with a scale-and-stretch mesh, which computes an optimal scaling factor for each region by combining visual attention and gradient map. This method is only designed for images, and a bended grid may distort the structure of complex backgrounds. Feng Liu et. Al. [12] proposed automatic pan and scan in which they find most salient part of video frames using face detection and motion detection but videos like football match players are hardly headed to camera, in that case face detection will fail. [13] Yu-Shuen Wang et.al. used a grid based approach with scale-and-stretch optimization technique to retarget the video. Nam et. al. [14] done the video retargeting using DCT with motion vectors for mobile screens.

Our seam surface removal technique is different from [1]. To calculate seam-surface, [1] are using graph-cut method to calculate a min-cut using their specially constructed graph, which has as many nodes as the pixels in the video and edges are almost 8-fold of nodes. It is quite large graph and complexity of calculating a cut in this graph is either $O(N^2E^2)$ or $O(N^2\sqrt{E})$ depending upon Labeling algorithm by Edmonds or preflow-push algorithm by Goldberg respectively, where n is number of nodes and e is number of edges in the graph.

2. MOTION PROJECTION MATRIX

We have used this metric to measure motion of objects within a video segment or in entire video sequence. This metric would show where it is high motion, and where it is low motion in the video. this information help us to decide which area in the video is less salient in terms of motion and so uninteresting part of video can be extracted fig 1. If a video V have F frames, and each frame f_i has of size $M \times N$, then motion projection matrix P will be calculated as below

$$diff_i = |f_i(m, n) - f_{i+1}(m, n)|$$

$$P(m, n) = \max(diff_1, diff_2, \dots, diff_F)$$

Where $i = 1, 2, \dots, F$; $m = 1, 2, \dots, M$; $n = 1, 2, \dots, N$;

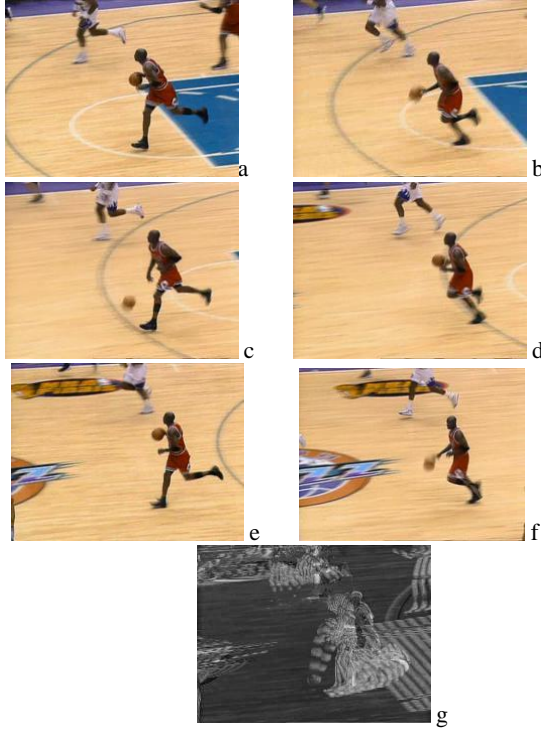


Fig. 1 (a)-(f) intermediate frames of a video sequence with moving camera, (g) projection matrix of (a)-(f).

2.1 Seam Removing

Seam is a 8-connected pixel path from one end of the image to the opposite end, If we have a image of size $m \times n$ then a vertical seam S is mathematically define as

$$s = \{x(i), i\}_{i=1}^{i=n}, s. t. \forall |x(i) - x(i-1)| \leq 1 \quad \dots (1)$$

Where x is a mapping $x: [1 \dots n] \rightarrow [1 \dots m]$. We remove seams by using a dynamic programming approach [2]. We create a memoization matrix C , as cost matrix, for the variance matrix V , Matrix C 's top row is same as the top row of V . Next, we traverse the image from the second row to the last row and compute the cumulative minimum cost for all possible 8-connected pixels for each entry. The other elements are computed as follows

$$C(0, n) = V(0, n);$$

$$C(m, n) = V(m, n) + \text{Min}[C(m-1, n-1), C(m-1, n), C(m-1, n+1); \quad \dots (2)$$

This gives the cost of all 8 connected pixel paths, from which the path with the lowest cost can be figured out. This block path is removed from the image, which is resized by pushing to left all pixels lying on the right hand side of this path. This process is repeated L times to remove L seams from the image.

3. VIDEO RETARGETING

3.1 Splitting the video in time domain.

We divide complete video sequence into segments in time domain (fig. 2) and apply following procedure on every segment. Calculate projection matrix P for given segment, compute cost matrix C for middle frame f_{mid} of same segment, superimpose P on C , further calculate seams S for f_{mid} , then remove

S from all the frames in given segment. This process can be applied to on-line video since there is no need of all the frames of video at any instance of time.

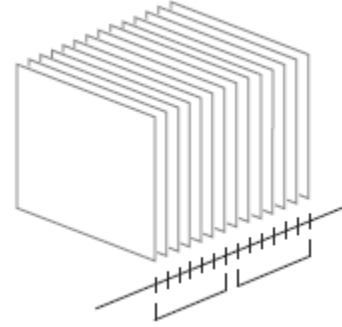


Fig. 2 Segmented video

PROCEDURE APPROACH-1

```

FOR each segment
  Calculate P
  Compute C for  $f_{\text{mid}}$ 
   $C = C + P$ 
  Calculate S for  $f_{\text{mid}}$ 
  Remove S from all frames in segment
END-FOR

```

This is very simple scheme and also very effective for video which are static in nature (by static we mean objects have not frequent motion) like interview, or news reading. Since movement of objects is very limited so the uninteresting part in video V will remain almost same w.r.t. time. In this approach we calculate uninteresting part once and remove it from whole segment. This approach will not work on video which have lot of motion or camera is moving, for those videos we suggested another scheme which is presented in section 3.2.



Fig. 3: (a) original frame from the standard benchmark news sequence Akiyo (b) retargeted frame by Approach 1

3.2 Overlapped Splitting of Video

In previous approach there is no relation between video segments for calculating uninteresting part in any segment, which can introduce sudden change (jitter effect) in retargeted video between two segments. Sudden change between two isolated segments can be reduced by our proposed approach, which will reduce the magnitude of jitter effect significantly. In this approach we divide the entire video into overlapped segments, so every overlapped segment is consist of an isolated segment which is not overlapped with any other segments which is depicted in fig. 4. We are splitting in overlapped fashion because now we are not only considering the motion of object within isolated segment but also outside the isolated segment. In this approach motion projection matrix contains the motion information of isolated segment and overlapped segment as well.

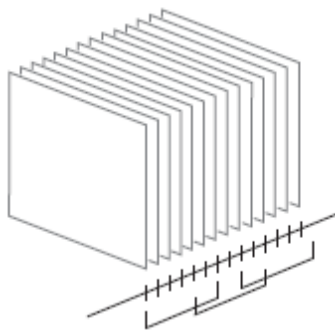


Fig. 4: overlapped segments

In this approach every segment has three parts one isolated part (p2) and two overlapped parts (p1 and p3). Isolated part p2 is not common to any other segment, while overlapped parts p1 and p3 are common to previous and next segments respectively as shown in fig. 4. We calculate projection matrix P for given segment, calculate cost matrix C for middle frame f_{mid} of isolated part p2 of same segment. Superimpose P on C, further calculate seams S for C and remove it from isolated part p2 and first overlapped part p1 (not from second overlapped part p2, since seams will be removed from this part in next segment seam removal.) as shown in fig. 5.

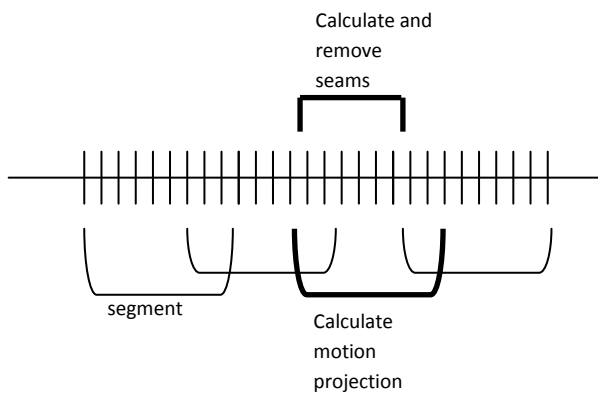


Fig. 5: processing of segments

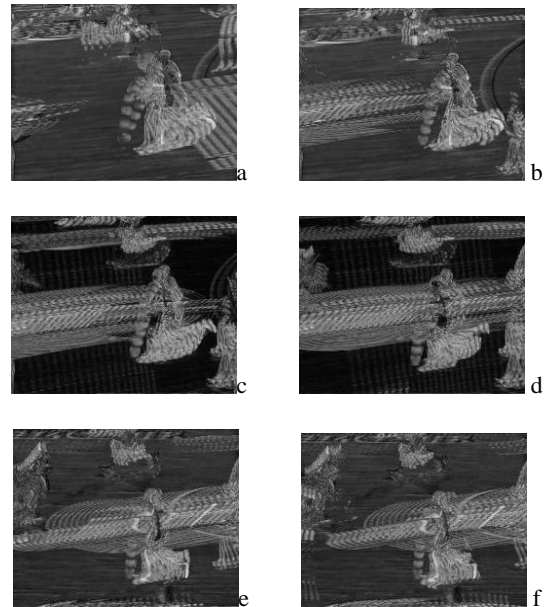


Fig. 6: (a)-(f) projection matrix of overlapped video segments

Projection matrix for overlapped video segments are shown in fig. 6. Algorithm for approach 2 is as follows

PROCEDURE APPROACH-2

```

FOR each segment (p1+ p2+ p3)
    Calculate P
    Compute C for  $f_{mid}$  of p2
     $C = C + P$ 
    Calculate S for  $f_{mid}$ 
    Remove S from p1 and p2
END-FOR
    
```

This scheme performs better than previous approach because we are providing space for smooth transition from one segment to next one in form of overlapped calculation of projection matrix. This approach can also work in on-line processing environment.

3.3 Seam Surface Removal

Partitioning the video and then process the segments for retargeting can introduce jitter in retargeted video, because more or less we are processing segments individually. Removal of a surface from the video cube cannot introduce any jitter in retargeted video. This surface would be uninteresting part of video in terms of objects in a frame and motion of object among the frames.

If seam s_1 of frame f_1 is 8-connected to seam s_2 of frame f_2 , s_2 is 8-connected to s_3 and so on, then all the seams collectively make a seam-surface, which is 8-connected to all 3-dimension by its construction.

To calculate such surface we are using dynamic programming as well as greedy algorithms. Calculation of seam in a frame is done by dynamic programming [2], while extending this seam to a surface we are using greedy approach. Since greedy algorithm chooses best solution locally, it sometimes may introduce artifacts in video, but generally it gives good results. Removing such surfaces leave very less artifacts than previous schemes. Seam surface can be calculated as below

PROCEDURE APPROACH-3

```

    Compute cost matrix for  $f_1$ 
    Superimpose motion projection matrix over  $f_1$ 
    Calculate seam  $s_1$  for  $f_1$ 
    Remove seam  $s_1$  for  $f_1$ 
    FOR  $i = 2$  to  $F$ 
    Compute cost matrix  $C$  for  $f_i$ 
    Superimpose motion vector  $P$  over 8-connected region of  $s_{i-1}$ 
    Calculate  $s_i$  for  $f_i$  such that  $s_i$  is 8-connected to  $s_{i-1}$ 
    Remove seam  $s_i$  for  $f_i$ 
    END_FOR
    
```



Fig. 7: Red line represents intersection of seam surface with frame

4. SPEED-UP THE PROCEDURE

Removing a surface from the space-time volume is a costly operation, it requires shifting of all those pixels to the one position left, which is right side of seam-surface, so we calculate and remove 3 seam-surfaces at a time, to reduce overall shifting of pixels to retarget the video. To do this we calculate a seam-surface, apply high weights to this surface and calculate next seam-surface. Due to applying high weights to previous surface no point will be intersected with next surface.

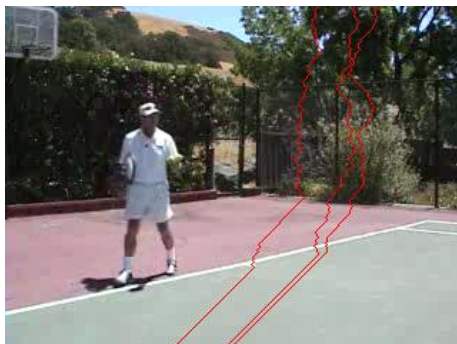


Fig. 8: A frame with 3 simultaneously calculated seams

5. FRAME FORMAT

To process the video we have to convert it into frames, and we have to take care of proper frame format, by proper we mean that it should be less compressed format (.ppm), it should not high compressed format like (.jpg). Effect of frame format is shown in fig. 9. Highly compressed format have lot of noise in frames and seam removal will bring those noise close to each other so retargeted frames have much more noise than original frames, in other word we can say noise density get higher in retargeted frames. Retargeted video

quality will be degraded if we use highly compressed frame format, so we are using pnm format.



a.



b.

Fig. 9: (a) JPEG format, (b) PNM format

6. CONCLUSION

We have segmented the video and process each segment individually, further segmented video in overlapped fashion and then process all segments with the help of motion projection matrix. There still exists jitter effect a bit. To achieve jitter free video we suggested seam-surface removal from the entire video with less computational cost. We also have mentioned the effect of frame format for video processing.

7. REFERENCES

- [1] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. In *Siggraph*, 2008.
- [2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. In *ACM Trans. Graph*, 2007.
- [3] X. Fan, X. Xie, H.-Q. Zhou, and W.-Y. Ma. Looking into video frames on small displays. In *ACM Multimedia*, 2003.
- [4] F. Liu and M. Gleicher. Video retargeting automating pan and scan. In *ACM Multimedia*, 2006.
- [5] F. Megino, J. Sanchez, and V. Lopez. Virtual camera tools for an image2video application. In *International Workshop on Image Analysis for Multimedia Interactive Services*, 2008.
- [6] V. Setlur and S. Takagi. Automatic image retargeting. In *MUM*, 2005.
- [7] C. Tao, J. Jia, and H. Sun. Active window oriented dynamic video retargeting. In *Workshop On Dynamical Vision, ICCV*, 2007.

- [8] D. Thomas, D. Philippe, and N. Hermann. Pan, zoom, scan time-coherent, trained automatic video cropping. In *CVPR*, 2008.
- [9] J .Wang, M. J. Reinders, R. L. Lagendijk, J. Lindenberg, and M. S.Kankanhalli. Video content representation on tiny devices. In *ICME*, 2004.
- [10] Y. S. Wang, C. L. Tai, O. Sorkine, and T. Y. Lee. Optimized scale-and-stretch for image resizing. In *ACM Trans. Graph*, 2008.
- [11] L.Wolf, M. Guttmann, and D. Cohen-Or. Non-homogeneous content-driven video retargeting. In *ICCV*, 2007.
- [12] Feng Liu, Michael Gleicher: Video retargeting: automating pan and scan. *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, 2006
- [13] Yu-Shuen Wang, Hongbo Fu, Olga Sorkine, Tong-Yee Lee, Hans-Peter Seidel : Motion-Aware Temporal Coherence for Video Resizing, *ACM Transection on graphics (TOG)-Proceedings of ACM Asia 2009*, Vol 28 Issue 5 December 2009
- [14] Hyeong-Min Nam, Keun-Yung Byun, Jae-Yun Jeong, Kang-Sun Choi, Sung-Jea Ko: Low Complexity Content-Aware Video Retargeting for Mobile Devices. *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 1, FEBRUARY 2010