Non-homogeneous Content-driven Video-retargeting

Lior Wolf, Moshe Guttmann, Daniel Cohen-Or
The School of Computer Science, Tel-Aviv University
{wolf,guttm,dcor}@cs.tau.ac.il

Abstract

Video retargeting is the process of transforming an existing video to fit the dimensions of an arbitrary display. A compelling retargeting aims at preserving the viewers' experience by maintaining the information content of important regions in the frame, whilst keeping their aspect ratio.

An efficient algorithm for video retargeting is introduced. It consists of two stages. First, the frame is analyzed to detect the importance of each region in the frame. Then, a transformation that respects the analysis shrinks less important regions more than important ones. Our analysis is fully automatic and based on local saliency, motion detection and object detectors. The performance of the proposed algorithm is demonstrated on a variety of video sequences, and compared to the state of the art in image retargeting.

1. Introduction

With the recent advent of mobile video displays, and their expected proliferation, there is an acute need to display video on a smaller display than originally intended. Two main issues need to be confronted. The first is the need to change the aspect ratio of a video. The second is the need to down-sample the video whilst maintaining enough resolution of objects-of-interest. An example of the first challenge, is the display of wide screen movies on a 4:3 TV screen. Displaying a ball game on a cellular screen is a good example for the need of a smart down-sampling technique, where the ball needs to remain large enough to be easily seen on screen.

The current industry solutions are basic and not very effective. They include: blunt aspect ratio free resizing; cropping the middle of the video; resizing while preserving the aspect ratio by adding black stripes above and below the frame; and keeping the middle of the frame untouched while warping the sides. In fact, it is common nowadays to have printed lines on movie-cameras' screens that mark the region that will be visible in the frame after it would be cropped to the aspect ratio of a regular 4:3 TV screen.

We have developed a method that assigns a saliency score to each pixel in the video. An optimized transformation of the video to a downsized version is then calculated that respects the saliency score. The algorithm is designed to work efficiently in an online manner, ultimately leading to a real-time retargeting of a streaming input video to several output formats. The saliency score is composed of three basic components: spatial gradient magnitude, a face detector, and a block-based motion detector. The optimization stage amounts to solving a sparse linear systems of equations. It considers spatial constraints as well as temporal ones, leading to a smooth temporal user experience.

(a) (b) (c)

Figure 1. An example of retargeting. (a) original frame from the standard benchmark news sequence "Akiyo"; (b) the half width retargeted frame achieved by our method; (c) a conventionally resized frame.

2. Related work

In previous decades, large amount of image processing research, that focused on down-sampling and up-sampling images, accumulated. These classical methods, however, are not "content aware" – they apply the same local operator everywhere across the image, oblivious to the semantics of the image and to the varying importance and sensitivity to distortion of each image region.

Recently, with the ever increasing need to alter the dimensions and aspect ratio of images and videos, the subject of retargeting has regained an increased academic attention, and a number of contributions have been published. Suh *et al*. [8] considered the problem of cropping optimal thumbnails from an input image. Although the task is different from that of retargeting (thumbnails are used for easy access, not to convey the entire content of an image), some of the components in their system have reappeared in later retargeting systems. Most notably, defining an importance measure which is based on both a local low-level saliency

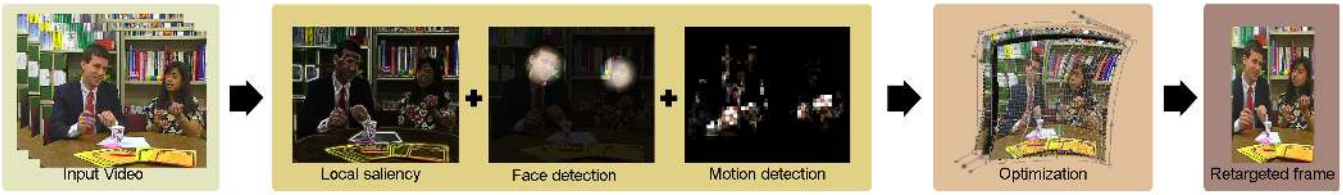


Figure 2. System overview. A saliency score is computed for each frame based on the gradient magnitude, face detection and motion detection. Next, an optimization stage recovers the retargeting warp. The warp is then applied to the original frame.

(a) (b)

Figure 3. (a) original, (b) retargeted frame (half width). Note that a cropped window cannot achieve the same frame area utilization.

measures, and on high level object detectors.

An extended cropping mechanism, applied to video is presented by Liu and Gleicher [3]. Compared to our system their system is not designed to be an online system (they use the whole shot), and is limited to displaying a sequence of cropped windows. Figure 3 demonstrates that cropping alone is sub-optimal.

A non-photorealistic solution for retargeting stills is proposed by Setlur *et al*. [7]. Their system is based on separating foreground from background, and relies heavily on their capability to solve this separation problem.

Recently, two systems introduce photorealistic solutions for content-aware remapping of still images. Gal *et al*. [2] introduce a method to modify an arbitrarily image warp while preserving the shape of important regions by constraining their deformation to be a similarity or a rigidity transformation. Although their method differs from ours, both methods define the mapping as the solution of a linear system of equations. Nevertheless, in Gal *et al*.'s method the important regions are manually marked by the user. A different approach is proposed by Avidan and Shamir [1], where the retargeting is applied by reducing the width (or the height) of the image by one pixel at a time, through deleting a vertical (or horizontal) connected paths of low importance pixels. While they show excellent results on images, their method seems to be overly expensive to be extended to video and their solution is greedy and discrete. Unlike the above two, our work is designed for video, and in particular for video streaming.

3. Overview

Our system, described in Figure 2, consists of two main stages. A computation of the saliency matrix and a mapping calculation stage. Then, the retargeted frame is rendered by a forward-mapping technique.

Given a new frame, we compute a per-pixel importance measure. This measure (see Sec. 4) is a combination of three factors: a simple, gradient based, local saliency; an off-the-shelf face detector; and a high-end motion detector.

The optimization of the mapping function from the source resolution to the target resolution is set through a linear systems of equations. Each pixel (i,j) at each frame t is associated with two variable $x_{i,j,t}$, $y_{i,j,t}$ that determine its location on the retargeted frame. We optimize for horizontal warps and for vertical warps separately, using the same technique. The horizontal post-warp location is first constrained to have the same coordinates as the warp of the pixel just below it $x_{i,j+1,t}$, and the pixel just before it $x_{i,j,t-1}$. Then it is constrained to have a distance of one from the warping of its left neighbor $x_{i-1,j,t}$.

For obvious reasons, it is impossible to satisfy all of the constraints and yet fit into smaller retargeting dimensions. To these space preserving constraints, we add weight in proportion to the pixel's importance value. A pixel with high importance is mapped to a distance of approximately one from its left neighbor, while a pixel of less saliency is mapped closer to its neighbor. Time smoothness is also taken into consideration, in order to generate a continuous natural-looking video.

Our algorithm is designed for video streaming. Therefore, time smoothness and motion analysis considerations are limited to the previous frames only. Such considerations need only apply to frames of the same shot (a sequence of video frames taken from a continuous viewpoint).

The proposed system automatically breaks a long video into a sequence of shots using a simple online algorithm, similar to the one shown in [6], where the block matching operation is replaced with the efficient algorithm of [5].First, motion estimation is applied on each macro-block (16×16 pixels). A shot boundary is detected wherever the number of blocks for which the motion estimation fails exceeds a threshold.

4. Importance determination

We define the content preservation weight matrix:

$$S = \min(S_E + \sum_i S_F^i + S_{MD}, \mathbf{1}) \quad (1)$$

Each entry in the matrix represents the saliency of a single pixel in the source frame I. Values range between 0 and 1, where zero values are, content wise, non-important pixels.

Local saliency We employ the simplest measure for local information content in the frame. Namely, we use the L_2-Norm of the gradient: $S_E = ((\frac{\partial}{\partial x} I)^2 + (\frac{\partial}{\partial y} I)^2)^{1/2}$.

Face detection Human perception is highly sensitive to perspective changes in faces, more specifically to frontal portraits. In order to avoid deforming frontal portraits we employ the Viola and Jones face detection mechanism [9].

The detector returns a list of detected faces. Each detected face i has a 2D center coordinate F_p^i and a radius F_r^i. The face detection score of each pixel is a function of the distance of that pixel from the face's center: $D_i(x,y) = \|F_p^i - (x,y)\|_2$, and is given by the cubic function:

$$\hat{S}_{Fi}(x,y) = \max(1 - \frac{-D_i(x,y)^3 + .5 * D_i(x,y)^2}{-(F_r^i)^3 + .5 * (F_r^i)^2}, \mathbf{0}) \quad (2)$$

This function, which ranges between 0 and 1, is used to weight the importance of the face as an almost constant function with a drastic fall near the end of the face. This allows some flexibility at the edges of the face whilst avoiding face deformation.

We further introduce a rescaling measure,

$$F_{rn}^i = \frac{F_r^i}{\max(C_{Width}, C_{Height})}$$
$$S_F^i(x,y) = \hat{S}_F^i(x,y)(1 - 2.5 * (F_{rn}^i)^4 - 1.5 * (F_{rn}^i)^2) \quad (3)$$

used to rescale the general saliency of a detected face in relation to the area it occupies in a $C_{Width} \times C_{Height}$ pixels frame. A 1 factor is used where the size of the face is relatively small, while extremely large faces tend to be ignored. The above prevents a distorted zooming effect, i.e. retargeting of the frame such that it is mostly occupied by the detected face.

Since, as stated below, when shrinking the width of an image, we demand smoothness over the columns, a detected face also prevents thinning the regions below it. Therefore, human bodies are shrunk less, as necessitated.

Motion detection Moving objects in video draw most of the viewers' attention and are content-wise important. By using a motion detection mechanism we mange to retarget the video while preserving the temporal context.

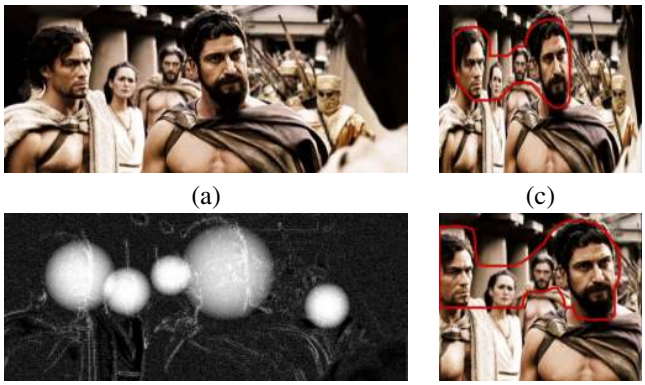


Figure 4. Retargeting examples of a frame (a) from the movie "300" with and without face detection. (b) the gradient map, with the faces detected imposed. (c) the result of retargeting to half the width without face detection. (d) the result of retargeting with face detection. The result of the whole shot compared to bicubic interpolation is available in the supplemental material.

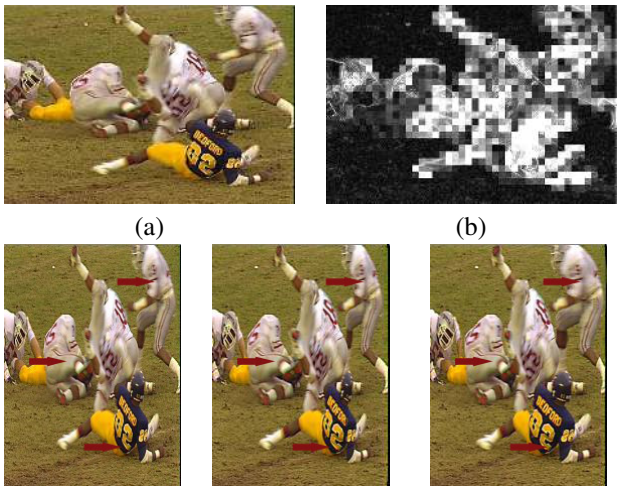


Figure 5. Retargeting a frame taken from the MPEG/ITU-T committee benchmark video "football" (a). (b) The saliency map which contains the motion component. (c) the result of bicubic interpolation to half the width. (d) retargeting without motion based saliency. (e) the result of retargeting with the full saliency map.

The second motion detector suggested in [4] is implemented. The selected algorithm is efficient and effective.The image is divided into 8×8 blocks. The motion based saliency $S_{MD}(x,y)$ is set to one if the block containing (x,y) has motion, and zero otherwise.

As can be seen in Figure 5 the moving objects gain saliency, thus seizing a larger area in the retargeted video.

5. Optimization

We cast the problem of finding the optimal mapping between the source image and the retargeted image as a sparse

linear system of equations that we solve in a least squares manner. A more natural formalization is to cast the problem as a constrained linear system. This way one can guarantee that no pixel falls out of bounds and that the mapping preserves the order of the pixels along the scan lines in the image. However, the solution to the unconstrained system is more efficient, and, in practice, the mappings recovered using the unconstrained systems of equations do not contain noticeable artifacts due to changes in the order of the pixels.

In the retargeting process a pixel (i,j) in frame t of the video is being mapped into a pixel in frame t of the output video with some computed location $(x_{i,j,t}, y_{i,j,t})$. Hence, there is twice the number of variables ($x_{i,j,t}$ and $y_{i,j,t}$) to solve for then the number of pixels in the input video. We compute the y variables separately from the computation of the x variables, using the same linear method (described below). The mapping computation is done one frame at a time (see below), and so our system of equations has (approximately) the same number of unknowns as the number of pixels in one input frame.

Consider the problem of recovering the new x-axis locations $x_{i,j,t}$ of pixels (i,j), $i = 1..C_{Width}, j = i..C_{Height}$, in frames $t = 1..C_{Duration}$. The problem of determining $y_{i,j,t}$ is the transpose of this problem and is solved in a similar manner. Also, consider first the more applicable problem, in which we would like to shrink the frame, i.e., to map the frame to a narrower frame with width $C_{TargetWidth} < C_{Width}$. The expanding problem is similar, though its goal is more application dependent (a detailed discussion of expansion is omitted for brevity).

There are four types of constraints. First, we constrain each pixel to be at a fixed distance from its left and right neighbors. Second, each pixel needs to be mapped to a location similar to the one of its upper and lower neighbors. Third, the mapping of a pixel at time t needs to be similar to the mapping of the same pixel at time $t + 1$. The forth constraint fits the warped locations to the dimensions of the target video frames.

Importance modeling. If a pixel is not "important" it can be mapped close to its left and right neighbors consequently blending with them. An "important" pixel, however, needs to be mapped far from its neighbors, thus a region of important pixels is best mapped into a region of a similar size. We formulate these insights into equations stating that every pixel should be mapped at a horizontal distance of 1 from its left and right neighbors. These equations are weighted such that equations associated with pixels with higher importance-score are more influential on the final solution. The first type of equations is therefore:

$$\begin{aligned} S_{i,j,t}(x_{i,j,t} - x_{i-1,j,t}) &= S_{i,j,t} \\ S_{i,j,t}(x_{i+1,j,t} - x_{i,j,t}) &= S_{i,j,t}, \end{aligned} \tag{4}$$

Where S is the saliency matrix of Eq. 1, except the time index appears explicitly.

Boundary substitutions. In order to make the retargeted image fit in the new dimensions the first pixel in each row of the frame $(1, j, t)$ is mapped to the first row in the retargeted video, i.e., $\forall j, \forall t \;\; x_{1,j,t} := 1$. Similarly, the last pixel of each row is mapped to the boundary of the remapped frame: $x_{C_{Width},j,t} := C_{TargetWidth}$.

Spatial and time smoothness It is important to have each column of pixels in the input image mapped within the boundaries of a narrow strip in the retargeted image. Otherwise, the image looks jagged and distorted. These type of constraint are weighted uniformly, and take the form:

$$W^s(x_{i,j,t} - x_{i,j+1,t}) = 0 \tag{5}$$

In our system $W^s = 1$. In order to prevent drifting, we also add a similar constraint that states that the first and the last pixels of each column have a similar displacement.

$$W^s(x_{i,1,t} - x_{i,C_{Height},t}) = 0 \tag{6}$$

The mapping also has to be continuous between adjacent frames, as stated bellow:

$$W^t_{i,j,t}(x_{i,j,t} - x_{i,j,t-1}) = 0, \tag{7}$$

where, in order to prevent distortion of faces, the weighting depends on the face detector saliency map $W^t = 0.2(1 + S_F)$. Note that in our system, we work in an on-line mode, which means that we do not build a system of equations for the whole shot. Instead we compute the mapping for each frame given the previous frame's computed mapping. This limited-horizon online time-smoothing method is good enough for our purpose and, as can be seen in Figure 6 can improve results significantly.

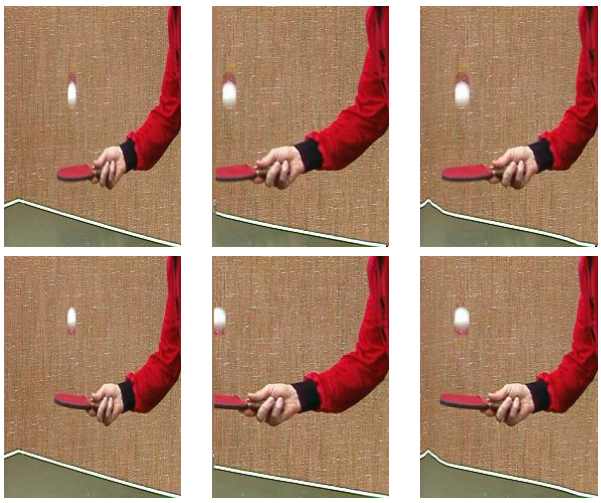
6. Results

Altering the aspect ratio. Examples of aspect ratio altering are exhibited in Figure 7 and in other figures throughout this manuscript. See the accompanied supplemental material for retargeted videos. The format of the retargeted videos is as follows: each frame is divided into three sub frames. The bottom one is the original video frame. The top right sub-frame is the result of applying bicubic interpolation to obtain a new frame of half the input width. The top-left sub-frame is our retargeted result.

While our algorithm does not explicitly crop frames, whenever the unimportant regions in the frame lie away from the frame's center, an implicit cropping is created. See, for example, the retargeting result of the sequence Akiyo (Figure 1). Many pixels at the left and right sides

(a) (b) (c)

Figure 7. (a) "Paris" sequence, from the MPEG/ITU-T standard benchmark, retargeted to half the frame's width; (b) Retargeting a frame taken from the motion picture "The Departed"; (c) Retargeting a frame from the video "Top 100 moments in NBA history". The original frame is shown at the bottom of each triplet, a bicubic interpolation is shown on the top-right. Our retargeting method (top left of each triplet) prevents much of the thinning effect caused by the rescaling and preserves details.



(a) (b) (c)

Figure 6. Retargeting examples of frames from the benchmark video "tennis" with and without time smoothness. The top row depicts the retargeting results of frame 10, and the bottom of frame 15. Retargeting results are shown for bicubic interpolation (a), frame by frame retargeting (b), and time smoothed retargeting (c). Time smoothing prevents the video from "jumping around".

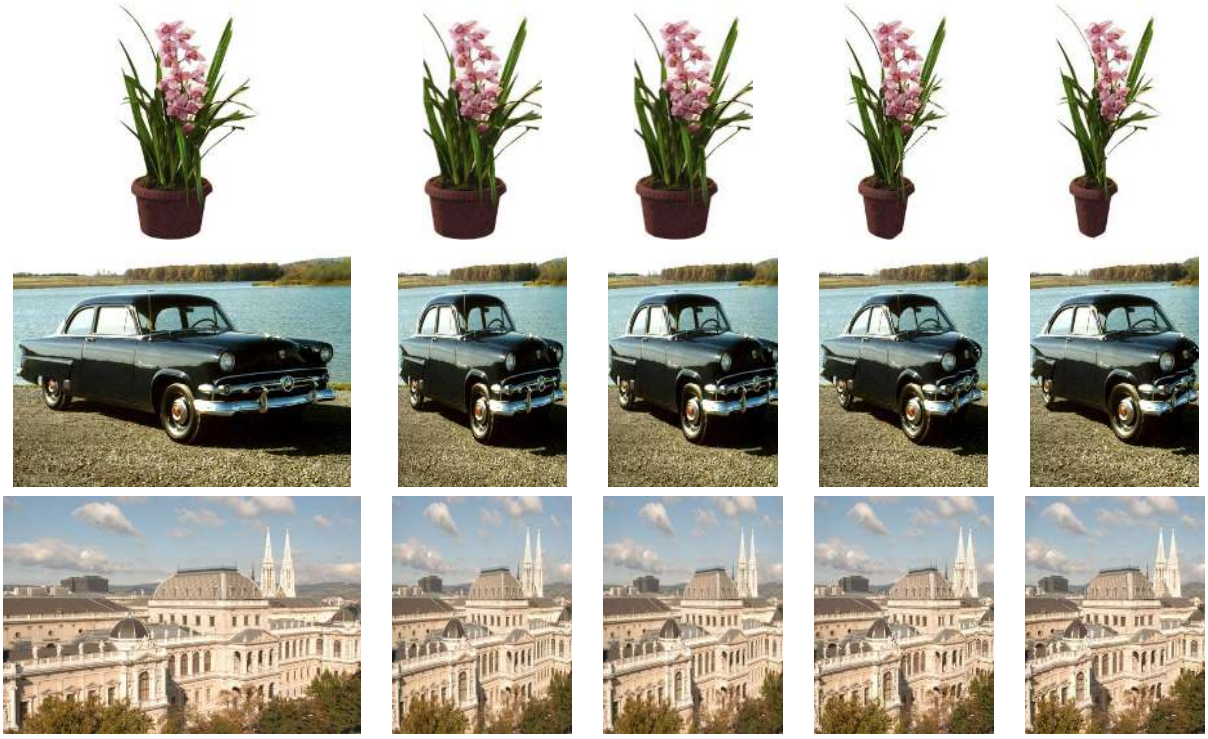
("The Departed") ("Akiyo")

("300") ("Football")

Figure 8. Down-sizing results. The bicubic interpolation is shown at the bottom of each pair. Our retargeting method (top) applies a non-homogenous zooms to the objects of interest.

of the input frames are mapped into the first and last few columns of the retargeted frames, hence disappearing.

Down-sizing results The down-sampling results (preserving the aspect ratio) are exhibited in Figure 8. The x-axis and the y-axis warps were computed independently on the original frame and then applied together to produce the output frames. As can be seen, there is a strong zooming-in effect in our results, as necessitated by the need to display large enough objects on a small screen.



(a) (b) (c) (d) (e)

Figure 9. Comparison with the Seam Carving technique [1], for both the L1-norm saliency advocated by [1], and the L2-norm we use in our work. (a) Original, (b) our retarget using L1-norm to half the width, (c) retarget using L2-norm, (d) Seam Carving using L1-Norm, (e) Seam Carving using L2-Norm. Our method is more robust to the Saliency measure selection (little difference between (a) and (b)). Due to the continuous nature of our method, it also creates less jagged lines (note the jagged lines in the Ford images and on the buildings on columns (d) and (e)). The advantages of optimizing the entire mapping at once are also visible, for example, when examining the flowerpot.

Comparison with Seam Carving [1]. Figure 9 shows a side by side comparison of our method and the method of Avidan and Shamir [1], where the two algorithms use the same importance measures. As can be seen, our method does not suffer from artifacts as much as the Seam Carving method does. We contribute this to the discrete and greedy nature of the Seam Carving method, while tends to uniformly distribute the error across the whole image.

7. Summary

We introduce a method for video retargeting, where instead of cropping the frames, we shrink them while respecting the salient regions and maintaining the user experience. The proposed system is efficient and the optimization stage consists of solving a sparse $N \times N$ system, where N is the number of pixels in each frame. The method is well adapted to batch applications, but is designed for streaming video since it computes the warp of a given frame based on a small time-neighborhood only, and it is fast enough to avoid delays.

Acknowledgments

This research is supported by the Israel Science Foundation (grants No. 1440/06, 1214/06) and the Colton Foundation.

References

[1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *SIGGRAPH*, 2007. Accepted. Available at www.faculty.idc.ac.il/arik/. 2, 6

[2] R. Gal, O. Sorkine, and D. Cohen-Or. Feature-aware texturing. In *Proc. Eurographics Symp. on Rendering*, 2006. 2

[3] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *ACM Multimedia*, 2006. 2

[4] S.-C. Liu, C.-W. Fu, and S. Chang. Statistical change detection with moments under time-varying illumination. *IEEE Trans. on Image Processing*, 1998. 3

[5] J. Lu and M. Liou. A simple and efficient search algorithm for block-matching motion estimation. *IEEE Trans. Circuits and Systems*, 1997. 2

[6] J. Meng, Y. Juan, and S.-F. Chang. Scene change detection in an MPEG-compressed video sequence. In *Digital Video Compression: Algorithms and Technologies*, 1995. 2

[7] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch. Automatic image retargeting. In *Proc. int. conf. Mobile and Ubiquitous Multimedia*, 2005. 2

[8] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *Proc. Sym. on User interface software and technology*, 2003. 1

[9] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 2004. 3