NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
NWU-CS-04-41
August 5, 2004

Automatic Image Retargeting

Vidya Setlur (Northwestern University, USA)
Saeko Takagi (Wakayama University, Japan)
Michael Gleicher (University of Wisconsin, Madison, USA)
Ramesh Raskar (Mitsubishi Electric Research Labs. - MERL, USA)
Bruce Gooch (Northwestern University, USA)

Abstract

We present a non-photorealistic algorithm for automatically retargeting images for a variety of display devices, while preserving the images' important features and qualities. Image manipulation techniques such as linear resizing and cropping work well for images containing a single important object. However, problems such as degradation of image quality and important information loss occur when these techniques have been automatically applied to images with multiple objects. Our algorithm addresses the case of multiple important objects in an image. We first segment the image, and generate an importance map based on both saliency and face detection. Regions are then resized and repositioned to fit within a specified size based on the importance map.

NSF 01416284/0415083

Keywords: image resizing, segmentation, visual attention, cropping, display size

Automatic Image Retargeting

Vidya Setlur* Northwestern University
Saeko Takagi† Wakayama University
Michael Gleicher‡ University of Wisconsin
Ramesh Raskar§ Mitsubishi Electric Research Labs, MERL
Bruce Gooch ¶ Northwestern University

Figure 1: *left) The source image containing three areas of importance, the Pagoda, the waterfall, and the person standing in the bottom right corner. center) The source image retargeted to fit a PDA display. right) The source image retargeted to fit a cell phone display. In the retargeted images, the areas of importance (pagoda, waterfall and person) have moved closer together while retaining key feature relationships.*

Abstract

We present a non-photorealistic algorithm for automatically retargeting images for a variety of display devices, while preserving the images' important features and qualities. Image manipulation techniques such as linear resizing and cropping work well for images containing a single important object. However, problems such as degradation of image quality and important information loss occur when these techniques have been automatically applied to images with multiple objects. Our algorithm addresses the case of multiple important objects in an image. We first segment the image, and generate an importance map based on both saliency and face detection. Regions are then resized and repositioned to fit within a specified size based on the importance map.

CR Categories: I.3.4 [Computer Graphics]: Graphics Utilities—Graphics editors; I.3.8 [Computer Graphics]: Applications; I.4.6 [I.4.6]: Image Processing and Computer Vision—SegmentationRegion growing, partitioning;

1 Introduction

In this paper, we introduce a technique for automatically *retargeting* images, that is, adapting them for display at different sizes and/or aspect ratios. Our method accommodates images with multiple important regions by minimizing the unimportant space between regions. The motivation for this work is the need for tools that allow us to author imagery once, and then automatically *retarget* that imagery for a variety of different display devices.

*e-mail:vidya@cs.northwestern.edu
†e-mail:saeko@sys.wakayama-u.ac.jp
‡e-mail:gleicher@cs.wisc.edu
§e-mail:ramesh@merl.com
¶e-mail:bgooch@cs.northwestern.edu

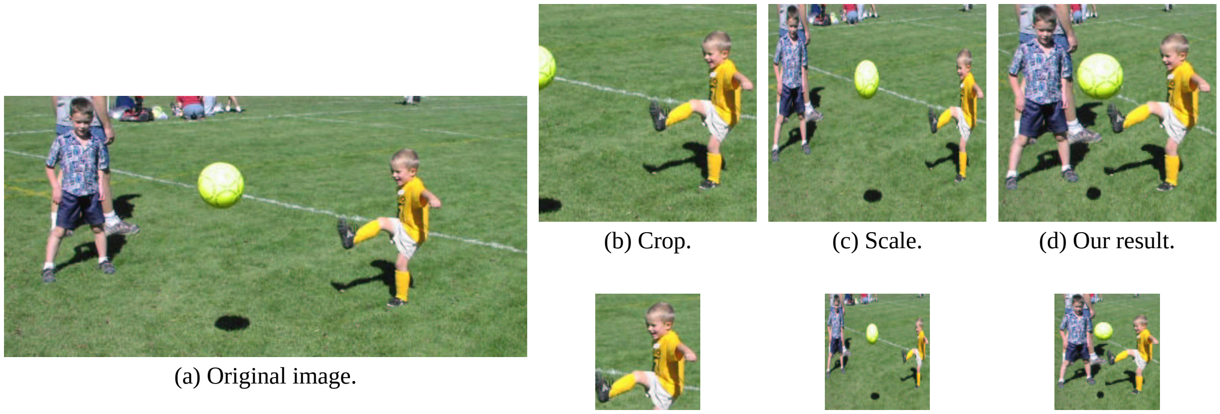
Increasingly, our computing and communications infrastructure is evolving to support images and video. Visual content is becoming more important for sharing, expressing, and exchanging information on devices such as, cell phones and hand-held PCs [Liu et al. 2003], PDAs with video capabilities, home-networked media appliances, and "heads up" informational displays in automobiles and helmets. Image retargeting is also useful for WYSIWYG directory icons for the efficient selection of images from directories and large image databases [Woodruff et al. 2001].

Simply scaling images reduces the size of important features. If there is a single important feature in the image, the image can be cropped and scaled to fit. Images with multiple, important features present a more challenging case for retargeting. In such cases, valuable image area in the target image may be wasted with unimportant regions between important features. For example, in Figure 2 there are important features on both sides of the image and cropping cannot remove the unimportant area between them.

To assist in generating these increasingly important small images, we introduce a novel method for *Automatic Image Retargeting.* Our goal is to provide *effective* small images by preserving the recognizability of important image features during downsizing. Our retargeting methods handle the cases where there are multiple important features in an image by considering different regions of the image independently. Important features are extracted from the image and re-arranged such that they best fit in the target image. For example, in Figures 1 and 2, the important image elements are brought closer together when the image is downsized; less of the unimportant background is shown so that more space can be used to show the important features.

2 Implementation

Our algorithm takes as input a source image and a specification for the size of the output image. Figure 3 summarizes our algorithm. We first apply the mean-shift algorithm to segment the source im-



(a) Original image. (b) Crop. (c) Scale. (d) Our result. (e) Crop. (f) Scale. (g) Our result.

Figure 2: A comparison between existing image resizing techniques and our automatic image retargeting method. *Note that cropping may eliminate important regions (such as the removal of the boy in blue in images (b) and (e)) and scaling often introduces distortion (the boys become stretched in images (c) and (f)). Our method, shown in the images (d) and (g), tends to move important regions close together while maintaining size and spatial relationships. Our algorithm is able to keep both boys in the image and maintain the relative positions of all shadows.*

age into regions. In Section 4 we discuss the techniques involved in segmenting the image, and combining adjacent regions based on their spatial distribution of color/intensity. In order to identify important regions, we generate an importance map of the source image using saliency and face detection as described in Section 5. If the specified size contains all the important regions, the source image is simply cropped. Otherwise, the important regions are removed from the image, and the resulting "holes" are filled using inpainting, described in Section 6.

The updated background is then resized to fit the input specification. Regions of importance are then "pasted" back onto the updated background based on their importance, and relative topology within the scene. If all the important regions are not able to fit within the new image, they are resized in inverse proportion to their importance. The "pasting" process is covered in detail in Section 7. Our results demonstrate that our non-photorealistic algorithm tends to move noticeable regions closer together while retaining key feature relationships in the image.

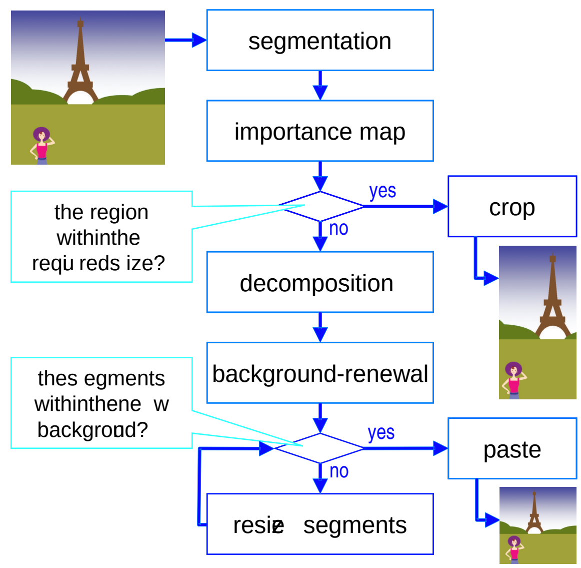


Figure 3: Flowchart of our algorithm.

3 Related Work

Image resizing can be performed manually using standard tools. Commercial products [Adobe n. d.; Gimp n. d.] enable the manual resizing of images using cropping and scaling operations. However, this process is often tedious, especially with large data sets. Also, performing retargeting operations other than simple cropping and scaling requires a great deal of skill and effort.

A few researchers have explored automating image retargeting through automatic cropping processes. For example, Suh et al. [Suh et al. 2003] proposed two techniques for automatic cropping based on using a visual attention model to detect interesting areas in an image. Their first method is based on saliency maps [Itti et al. 1998], while the second is based on face detection [Schneiderman and Kanade 2000]. In both cases the output is a thumbnail, created by cropping and scaling the source image. Neither method can handle cases where there are multiple important features in an image.

Münoz et al. [Münoz et al. 2001] proposed a spline-based algorithm for the enlargement or reduction of digital images with arbitrary (non-integer) scaling factors. Their method is a least-squares approximation of oblique and orthogonal projections for splines.

There has been some work on image resizing that explicitly considers mobile devices. Chen et al. [Chen et al. 2002] developed a system where the most important region is delivered to the client when the screen size is small. However, their method does not extend to images with multiple important regions. Liu et al. [Liu et al. 2003] have also worked on image adaptation in which the user scrolls between 'pages' of an image to view different important regions. The small display size of mobile devices, especially cell phones, makes this method cumbersome. In addition, a user also cannot view all the important regions in a single screen.

The goal of image retargeting is the ability to capture the "essence" of an image within a smaller form. Jojic et al. [Jojic et al. 2003] proposed a model of image representation, called an *'epitome'*, that attempts to encode this. The epitome of an image is its miniature, condensed version containing most constitutive elements needed to reconstruct the original image. The main idea is to uniquely map every patch in the epitome to a corresponding patch in the original image. This works well when the original im-

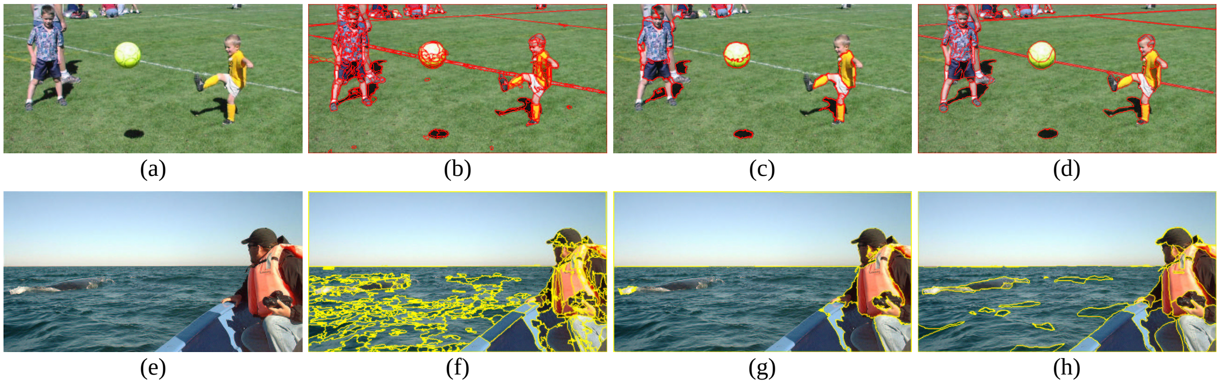


Figure 4: Image segmentation. *a) The original image. b) Applying mean-shift with parameters $h_s = 7$, $h_r = 6$, and $M = 50$. c) Applying mean-shift with parameters $h_s = 32$, $h_r = 30$, and $M = 150$. d) Performing region simplification on (b). e) The original image. f) Applying mean-shift with parameters $h_s = 7$, $h_r = 6$, and $M = 50$. g) Applying mean-shift with parameters $h_s = 22$, $h_r = 22$, and $M = 200$. h) Performing region simplification on (f).*

age has a lot of small, repetitive textures. This technique would not be suitable for obtaining a more comprehensible image where the neighborhoods between important regions are maintained.

Our approach lies in constructing a topologically constrained epitome of an image based on a visual attention model that is both comprehensible and size varying, making the method suitable for display-critical applications.

In order to retarget images with multiple important features we need to re-arrange images. Such re-arrangement has been considered by Balmetti et al. [Balmelli et al. 2002] for generating effective texture maps. Image regions with more frequency content are given more space in the optimized texture map. Unlike the retargeting problem, texture map optimization has a clear utility metric and can create distortions as they will be reversed during the mapping process.

4 Image Segmentation

In order to identify important regions, we require an image segmentation technique. We chose mean-shift image segmentation [Meer and Georgescu 2001] to decompose the given image into homogeneous regions. The advantages of this approach include flexible modeling of the image and noise processes and consequent robustness in segmentation. This technique is a non-linear method for segmentation based on non-parametric density estimation. Here, image data is modeled as clusters of pixels in the combined range-domain space, using kernel based techniques to represent the underlying, multi-modal Probability Density Function (PDF) [Singh and Ahuja 2002]. The pixel clusters or image segments are identified with unique modes of the multi-modal PDF by mapping each pixel to a significant mode using a convergent, iterative process. Let $f(x)$ be the (unknown) PDF underlying a p-dimensional feature space, and x_i the available data points in this space. The mean-shift property is the estimate of the density gradient at location x, proportional to the offset of the mean vector computed in a window, from the center of the window.

It can be formulated as [Christoudias et al. 2002]:

$$\nabla \widehat{f(x)} \sim (ave_{x_i \in S_{h,x}}[x_i] - x) \tag{1}$$

where $S_{h,x}$ is the p-dimensional hyper-sphere with radius h centered on x. Modes are local maxima of the density, i.e., $\nabla f(x) = 0$.

The algorithm proceeds as follows:

1. Choose a radius for the search window $S_{h,x}$.
2. Choose the initial location of the window.
3. Compute the mean of the data point over the window and translate the center of the window to the point.
4. Repeat until translation distance is less than a threshold.

The segmentation is dependent on three parameters, namely spatial radius h_s, color radius h_r, and the minimum number of pixels M that constitute a region for mean shift analysis. If the image deviates from the assumed piecewise constant model, larger values have to be used for h_r and M to discard the effect of local variations in the feature space [Comaniciu and Meer 2002]. For example Figure 4b is over-segmented, whereas increasing h_r and M to larger values causes part of the boy's head on the right to merge with the surrounding grass in Figure 4c. Similarly, in Figure 4g, increasing the parameters h_r and M causes most of the whale to merge with the ocean.

As with other segmentation techniques, it is often difficult to determine the optimal values for each image for the desired result. Instead, we use lower values of h_r and M to obtain highly segmented images containing core clusters. We then merge regions in two steps:

1. If there exist regions completely contained within larger regions, the smaller regions are merged with the outer ones. For example in Figure 4b, the segment fragments present inside the ball are merged together to be Figure 4d.

2. Similar adjacent regions are merged based on their spatial distribution of color/intensity. In order to perform color analysis of the image, the image must be present in perceptually uniform space. We use a two step process to convert to CIE to account for the dependence of color appearance on spatial structure [Fleet and Heeger 1997]. We first convert the input RGB image to the L$\alpha\beta$ opponent color space [Mirmehdi and Petrou 2000] that consists of three color planes, O_1, O_2, O_3, representing luminance, red-green, and the blue-yellow planes separately. Then each of the planes is smoothed directly by applying Gaussian kernels. These kernels are the sums of Gaussian functions with different values of standard deviation σ, computed according to:

$$\frac{1}{m}\Sigma_i \frac{w_i}{n_i} e^{\frac{x^2+y^2}{\sigma_i^2}} \tag{2}$$

We use the values of (w_i, a weighting term, and the standard deviation σ_i) from the work by Mirmehdi and Petrou [Mirmehdi and Petrou 2000]. The authors performed psychovisual measurements on human subjects to obtain these values. n_i is used to normalize the sum of the matrix elements of each Gaussian kernel, and m normalizes the sum of the elements of the final matrix to 1.

Once the kernels are applied to the image, we carry out color measurement [DeCarlo and Santella 2002] in CIE-Luv, a perceptually uniform space. The pixels representing each segmented region are used to form 3D color histograms that contain a range of values for each of the color components [Pietikainen et al. 2001]. We then compute a color dissimilarity measure called histogram intersection [Swain and Ballard 1991] to determine color similarity between regions. In Figure 4h, similar colored water segments are merged. We prune the number of possibilities by calculating similarity only with adjacent regions.

A **dual graph** is then created to store the global information of the image, where a node of the graph corresponds to a region. An edge of the graph indicates that the two regions are adjacent. The histogram information is also stored for each node, and is used later in the retargeting process.

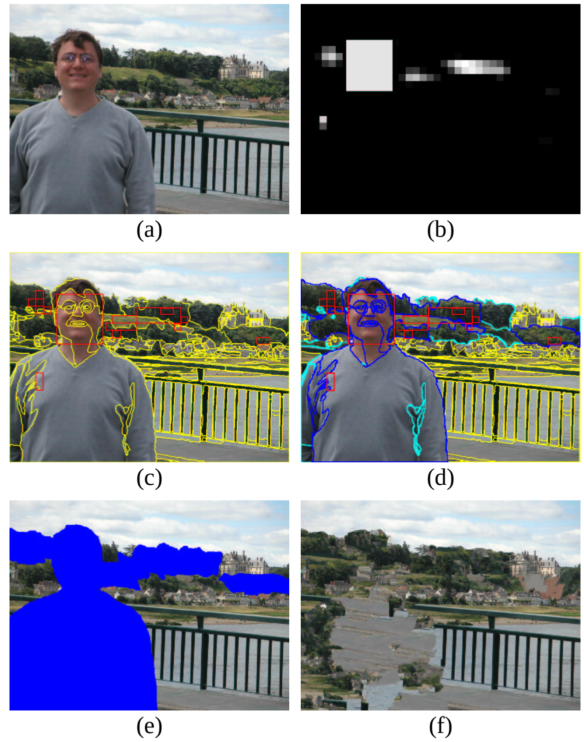


Figure 5: Importance map generation. *a) The original image. b) Corresponding importance map. c) Mapping importance map (shown in red) to segmented image. d) Determining Regions-Of-Importance (ROI). The regions containing importance map values are outlined in blue, and the adjacent regions with similar color histogram values are outlined in cyan. The blue and cyan outlined regions together form the ROI. e) Masked areas shown in blue. f) After inpainting.*

5 Importance Map

The importance map, which is the attention model for the image, is built up by combining a series of importance measures. This allows our system to be adapted to differing image creation goals, and to be easily extensible. For example, a semi-automatic retargeting application could be constructed by extending the importance map to include a user input importance map. Such a map could be created by users clicking on image regions that are important to them.

We use a visual attention model defined by Chen et al. [Chen et al. 2003] to populate the importance map. The visual attention model for an image is defined as a set of importance objects.

$$IO_i = (ROI_i, IV_i, MPS_i), 1 \leq i \leq N \quad (3)$$

where, IO_i is the i^{th} importance object within the range, ROI_i is the Region-Of-Importance of IO_i, IV_i is the importance value of the IO_i, MPS_i is the *minimal perceptible size* of IO_i, and N is the total number of importance objects in the image.

Each entry in the importance map corresponds to an *importance object* (IO). Generally, most perceptible information of an image lies in these 'areas of importance'. Each IO is assigned three attributes, namely *Region-Of-Importance*, *importance value* (IV), and *minimal perceptible size* (MPS). The *ROI* could correspond to a particular object or to a spatial region in the image. The importance rectangles from the importance map are mapped onto the segmented image as shown in Figure5c. The *ROI* is calculated by finding all the regions that either contain or are a part of these importance rectangles (outlined in blue in Figure 5d), and their similarly colored adjacent regions (outlined in cyan in Figure 5d. These adjacent regions are determined using the histogram information from the dual graph calculated during segmentation. The *IV* is a quantified value of the importance of an attention object and is also an indicator of the weight of a particular attention object's contribution to the overall information conveyed by the image. During the image retargeting process, there may be a need to resize an attention object to accommodate it within the specified size. *MPS* is a value assigned to an *IO* which indicates the minimum resolution allowed during retargeting. If the specified size is smaller than any of the *MPS* values in the importance map, the image is retargeted to the next higher acceptable size. This helps prevent degradation of image quality, and maintains the informativeness of the image. We predefine the *MPS* to be 25×30 pixels, which is the smallest resolution to show the face region without severely degrading its perceptibility [Ramos and Hemami 1996].

5.1 Image Attention Model

We use the saliency-based image attention model [Itti et al. 1998] to generate the first contribution to our importance map. The saliency model is used to extract attended locations in complex scenes based on a biologically plausible architecture. The technique uses Gaussian pyramids to compute several 'feature maps' for three low level features: color C, intensity I, and orientation O, which represent the visual scene. Such feature extraction is achieved through linear filtering for the given feature type, followed by a center-surround operation which extracts local spatial discontinuities for each feature type. Spatial discontinuity locations are then combined into a unique 'saliency map' represented as:

$$S = \frac{1}{3}(N(I) + N(C) + N(O)) \quad (4)$$

where N denotes normalization.

The two-dimensional topographical saliency map is used to determine the importance values within the original image. We binarize the saliency map to find the *ROIs*. The *IV* can be calculated as:

$$IV_{saliency} = \sum_{(i,j \in R)} B_{i,j} \cdot W^{i,j}{}_{saliency} \quad (5)$$

1/3	1/3	1/3	
1	2	1	3/12
4	8	4	4/12
1	2	1	5/12

Figure 6: Positional weight for face detection.

$B_{i,j}$ denotes the gray-scale value of pixel (i, j) in the saliency map. Since people pay more attention to the region near the center of an image, a normalized Gaussian template centered at the image is used to assign the positional weight ${W^{i,j}}_{saliency}$ [Chen et al. 2002].

5.2 Face Attention Model

Images of people are popular as well as important in many application areas. However, saliency map generation relies only on low-level features, and it might not be able to recognize faces correctly. The face is a highly important characteristic of human beings, and dominant faces in images certainly attract viewers' attention. Therefore, we use a face attention model proposed by [Ma et al. 2002; Chen et al. 2003], in addition to the image attention model. By applying face detection [Schneiderman and Kanade n. d.; Niblack et al. 1993], we obtain information about faces in the image such as position, region, and pose. The size and position of a face usually reflects its importance. Hence, the importance value in this model is calculated:

$$IV_{face} = \sqrt{Area_{face}} \cdot W^i_{pos} \qquad (6)$$

where $Area_{face}$ denotes the size of the detected face region, and W^i_{pos} is the weight of its position as defined in Figure 6 and $i \in [0, 8]$ is the index of the position [Ma et al. 2002].

The visual attention and face detection models are integrated together to get importance information. Currently, we adopt a linear combination to implement fusion scheme due to its effectiveness and simplicity. With such a scheme, each attention model should be normalized to [0, 1]. The IV of each IO is normalized to [0, 1] and the final IV is computed according to the equation. We normalize importance values from both attention models and compute the importance value IV for each importance object IO as:

$$IV_i = w_k \cdot \overline{IV^k_i} \qquad (7)$$

where w_k is the weight of the model k, and $\overline{IV^k_i}$ is the normalized importance value of the IO_i detected in model k. Since the important regions largely depend on the saliency map, sometimes our system does not identify an object that might be considered important by the viewer. In Figure 5c, the white castle in the background is not recognized to be important. By allowing the user to mouse-click on important regions, we can expect better results. Moreover, the importance map construction is flexible, and can accommodate other attention models as desired.

6 Inpainting

First we remove the objects of importance from the source image. Regions-Of-Importance (ROI) in the Importance Map may encompass several of the segmented regions in the source image. Each segmented region (R) is represented as a node in the dual graph, which was constructed during the segmentation stage. We obtain nodes encompassing the ROI by traversing the dual graph and selecting adjacent nodes, if such nodes exist. This traversal is restricted to nodes within a pre-specified distance from each ROI.

Table 1: Parameters used in Equation 9.

ω_k	: The set of offsets considered when calculating the similarity of two texture patches.
ω_o	: The set of locations containing pixels in the output image.
$O(s)$	: The pixel in the output image at location s.
$L(s)$	: The location of the pixel in the input image that is in the output image at location s.
$K(u)$	: Weighting given to a particular offset u.
$A(s), B(s)$	: Two uniformly random functions.

Once the 'objects of importance' are detected, we remove the corresponding regions from the source image and an inpainting technique is used to fill in the resulting gaps. Inpainting allows us to perform image manipulations on the background and important regions without introducing visual artifacts. We reconstruct the missing information, filling texture using the inpainting algorithm of Harrison [Harrison 2001]. He describes a method to reconstruct an image with the same texture as the given input image by successively adding pixels from the input image in a particular order. The procedure is capable of reproducing large features even with the interaction of neighboring pixels, and transfers large complex features of the input to the output image. It avoids decomposing the input image into a feature set, and could reproduce a variety of textures, making it suitable for our purpose. We note that this method could easily be replaced by other techniques [Pérez et al. 2003; Drori et al. 2003].

The image inpainting method involves two stages. First, pixel interrelationships are analyzed. This evaluates the extent to which each pixel constrains values taken by neighborhood pixels. To compare individual pixels, the sum of the absolute values of the differences in each color component is used:

$$d((r_1, g_1, b_1), (r_2, g_2, b_2)) = |r_2 - r_1| + |g_2 - g_1| + |b_2 - b_1| \qquad (8)$$

In order to measure how closely patches from the input image match a patch in the output image, a weighted Manhattan (city block) distance function is used:

$$D(s,t) = \varepsilon \mid A(s) - B(s) \mid + \sum_{u \in \Omega_K, t+u \in \Omega_O} K(u) d(I(s+u), O(t+s)) \qquad (9)$$

The parameters used in Equation 9 are shown in Table 1.

The second stage involves adding pixels to the initial blank output image until it is filled. The order in which pixels are added determines the quality of the texture synthesis. Priorities are assigned to each location in the output image, with highest priority given to locations highly constrained by their neighboring pixels. Then the following algorithm is used:

While there are still empty locations in the output,

1. Find the empty location with highest priority
2. Choose a pixel from the input image to place in that location
3. Update neighbors based on new pixel value.

Once we generate the new background, we scale it down to the target size using linear resizing methods. Figure 7 shows the masks in the image, and the results after inpainting is applied. Removal of large regions from the image sometimes leads to some artifacts as shown in Figure 5f. Since the removed objects' centroid positions are maintained while pasting back onto the updated background, most of these artifacts are minimized in the final image.

(a) Original image. (b) Mask areas shown in blue. (c)After inpainting.

Figure 7: Inpainting. *Demonstrating masking, removal of important objects, and inpainting.*

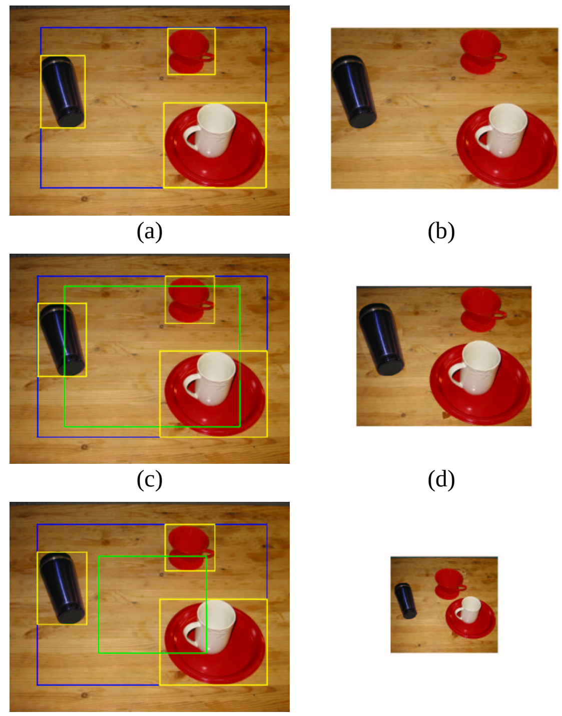
7 Pasting

After removing the important objects and filling in the 'holes' using inpainting, we composite the importance objects (*IO*) onto the resized background. As the background is resized to the specified size, the centroids of the important regions also shift. We apply a set of placement rules to each object prior to pasting them on the updated background (target area T).

1. If all importance objects (*IO*) fit inside the specified target size, then automatic cropping is performed.
2. else, shrink *ROI* in inverse proportion to their importance until they fit inside the specified image size.

These rules are illustrated in Figure 8. A bounding box is created for each object in the importance map. When the bounding boxes are within the specified size, automatic cropping is performed to remove the area outside the bounding boxes as shown in Figure 8a. If the *IO* do not fit, offsets are calculated for each bounding box.

In order to keep the retargeted image consistent with the original, it is important to ensure that the object is pasted on approximately the same textural color. The dual graph G computed during the mean shift segmentation discussed in Section 4, contains the region information for the objects regarding adjacency, and histograms. We use this information to check the colors of the surrounding areas. While pasting the object onto a sub-region of the target area T, we compute color similarity between that sub-region and the regions adjacent to the object. We use the same color dissimilarity measure [Swain and Ballard 1991] that we used during segmentation. If these values match up, they are pasted about their new centroid positions as in Figure 8b. Otherwise, the *ROI* of the object is scaled down until it satisfies this condition. The scaling factors are calculated such that the aspect ratio of the *ROI* is preserved. If two bounding boxes intersect, the *ROI*s will overlap when they are pasted, as shown in Figure 8c. Then the least important of the objects in contention is scaled. For scaling to occur, we check if the resized *ROI* satisfies the minimal permissible spatial area (*MPS*) of the *IO*.



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

Figure 8: Rules of placement and pasting. *Each pair of images illustrates the relative bounding boxes and the resulting image. The blue bounding box is the ideal automatic crop, the yellow bounding boxes contains the importance objects and the green bounding box is the retargeting size. Objects are first moved as close as possible. If the bounding boxes (shown in yellow) intersect, then the objects are shrunk until they fit the retargeting size.*

8 Results and Discussion

The results indicate that retargeting tends to preserve the recognizability of important objects, when compared to traditional resizing techniques. Our method has worked reasonably well in many cases, especially when important objects are far apart from each other. Though we might be deforming the original image, we are able to better allocate the source images' important features in the target images. In Figure 9, a crop of the images allows only one object to be included (whale, a single person, castle). Scaling on the other hand leads to distortion of the images. By using our method, we tend to retain important objects, yet minimizing their overall distortion.

Since our method is completely automatic, it may not be optimal if an important feature is on a similarly textured background, such as a single face in a crowd of people. Emotional connectedness between objects is another aspect that our system cannot address at this point. We believe that by making the system semi-automatic, the user can designate emotionally important objects in an image. Our algorithm is non-photorealistic and may not maintain semantic relationships among the objects in the retargeted image. For example, our system does no establish semantic correlation between objects and their shadows. In Figure 2, the shape of the ball's shadow in the retargeted image is not consistent with the shape of the ball. This is because our system identifies the ball and not its shadow to

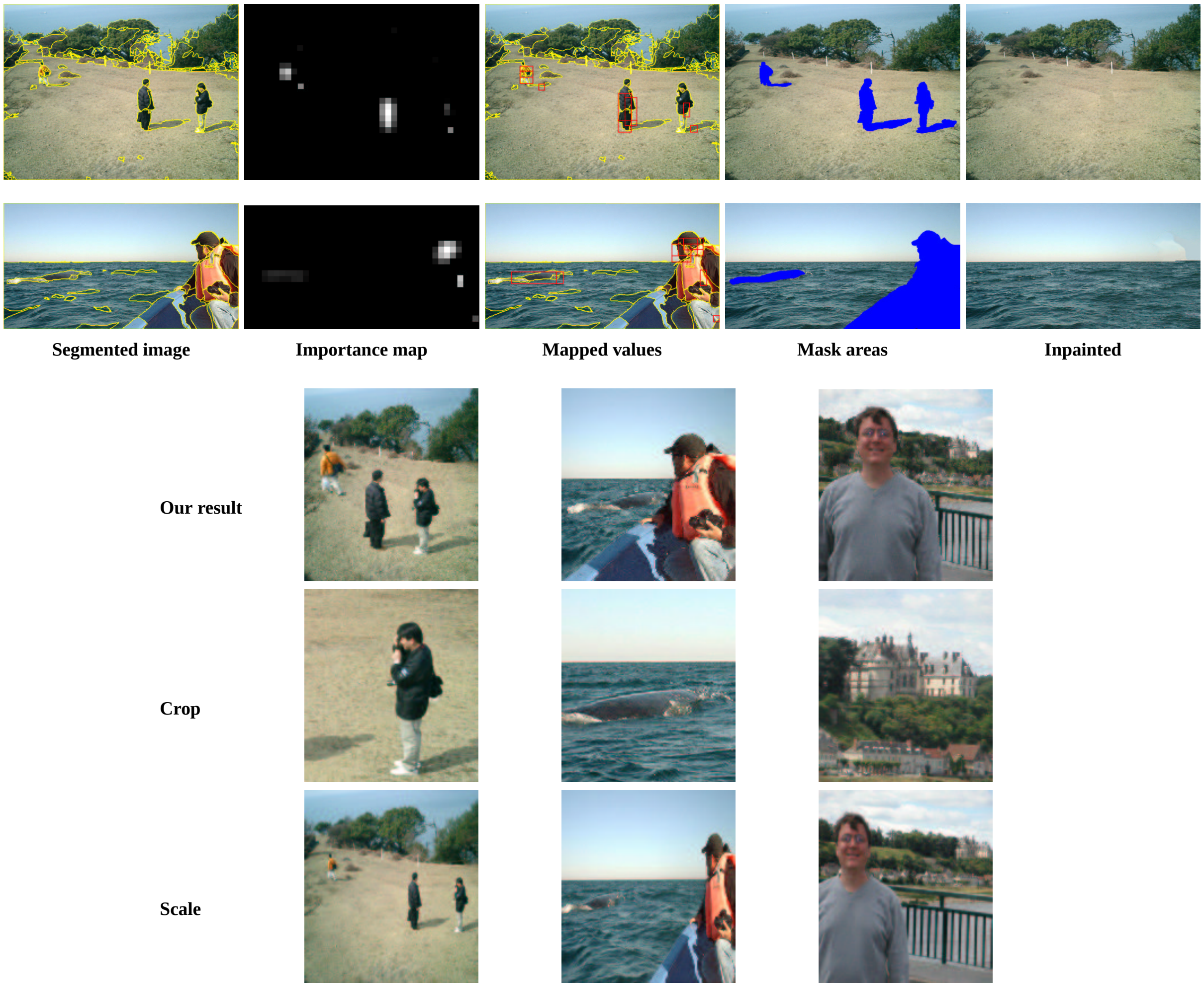


Figure 9: A comparison between existing image resizing techniques and our automatic image retargeting method. *Note that cropping may eliminate important regions and scaling often introduces distortion. Our method tends to move important regions close together while maintaining the size, aspect ratio, and spatial relationships of important regions of the image. In all of these examples, the images are retargeted to a cell phone size display. The original images for each column set are Figure 6a, Figure 4e, and Figure 5a respectively.*

be important, leading to the resizing of the shadow along with the background.

9 Conclusion

Images are a powerful mode of communication. Because of this, our computing infrastructure is constantly evolving to deliver higher quality imagery. Ubiquitous high-speed networking provides imagery to home theater screens, cellular phone displays, networked PDAs, and even displays embedded in refrigerators, elevators and airplane seats.

Our vision is that increasingly ubiquitous displays can provide people with information when and where they need it, provide more effective channels of inter-personal communication, deliver educational media, and provide entertainment. Achieving this vision requires providing imagery for a variety of display devices.

This work in automatic image retargeting is a first step in that direction. We have demonstrated an algorithm that allows a user to author imagery once, and then automatically *retarget* that imagery for an assortment of display devices. Our results demonstrate that our method tends to minimize the loss of detail and distortion. In addition, the algorithm moves significant regions closer together while retaining key feature relationships in the image. One can imagine a variety of retargeting applications such as: entertainment images for cellular phones, training images for PDAs, and status information for "heads up" displays.

References

ADOBE. www.adobe.com.

BALMELLI, L., TAUBIN, G., AND BERNARDINI, F. 2002. Space-optimized texture maps. *Computer Graphics Forum 21*, 3, 411–420.

CHEN, L., XIE, X., FAN, X., MA, W.-Y., ZHANG, H.-J., AND ZHOU, H. 2002. A visual attention model for adapting images on small displays. Tech. Rep. MSR-TR-2002-125, Microsoft Research, Nov.

CHEN, L.-Q., XIE, X., MA, W.-Y., AND ZHOU, H.-Q. 2003. Image adaptation based on attention model for small-form factor devices. In *Proceeding of the 9th International Conference on Multimedia Modeling*, IEEE, 421–439.

CHRISTOUDIAS, C. M., GEORGESCU, B., AND MEER, P. 2002. Synergism in low level vision. In *Proceedings of 16th International Conference on Pattern Recognition - Volume 4*, IAPR, 150–155.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May), 603–619.

DECARLO, D., AND SANTELLA, A., 2002. Stylization and abstraction of photographs.

DRORI, I., COHEN-OR, D., AND YESHURUN, H. 2003. Fragment-based image completion. *ACM Transactions on Graphics 22*, 3 (July), 303–312.

FLEET, D. J., AND HEEGER, D. J. 1997. Embedding invisible information in color images. In *IEEE Signal Processing Society 1997 International Conference on Image Processing (ICIP'97)*, 532–535.

GIMP. www.gimp.org.

HARRISON, P. 2001. A non-hierarchical procedure for re-synthesis of complex textures. In *Proceedings of WSCG 2001*, 190–197.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (Nov.), 1254–1259.

JOJIC, N., FREY, B., AND KANNAN, A. 2003. Epitomic analysis of appearance and shape. In *Proceedings of 9th IEEE International Conference on Computer Vision*, IEEE, 34–41.

LIU, H., XIE, X., MA, W.-Y., AND ZHANG, H.-J. 2003. Automatic browsing of large pictures on mobile devices. In *Proceedings of the eleventh ACM international conference on Multimedia*, ACM, 148–155.

MA, Y.-F., LU, L., ZHANG, H.-J., AND LI, M. 2002. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, ACM Press, 533–542.

MEER, P., AND GEORGESCU, B. 2001. Edge detection with embedded confidence. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 12 (Dec.), 1351–1365.

MIRMEHDI, M., AND PETROU, M. 2000. Segmentation of color textures. In *Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 142–159.

MÜNOZ, A., BLU, T., AND UNSER, M. 2001. Least-squares image resizing using finite differences. *IEEE Transactions on Image Processing 10*, 9 (sep), 1365–1378.

NIBLACK, W., BARBER, R., EQUITZ, W., FLICKNER, M., GLASMAN, E. H., PETKOVIC, D., YANKER, P., FALOUTSOS, C., AND TAUBIN, G. 1993. The qbic project: Querying images by content, using color, texture, and shape. In *Proceeding of the SPIE, Vol. 1908, Storage and Retrieval of Image and Video Databases*, SPIE, 173–187.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Transactions on Graphics 22*, 3 (July), 313–318.

PIETIKAINEN, M., MAENPAA, T., AND VIERTOLA, J. 2001. Color texture classification with color histograms and local binary patterns. *Proc. 2nd International Workshop on Texture Analysis and Synthesis, Copenhagen, Denmark* (June), 109–112.

RAMOS, M. G., AND HEMAMI, S. S. 1996. Edge-adaptive jpeg image compression. In *Proceeding of the SPIE, Vol. 2727, Visual Communications and Image Processing '96*, SPIE.

SCHNEIDERMAN, H., AND KANADE, T. Face detection demonstration. robotics institute. www.vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi.

SCHNEIDERMAN, H., AND KANADE, T. 2000. A statistical model for 3d object detection applied to faces and cars. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition - Volume 1*, IEEE, 746–751.

SINGH, M. K., AND AHUJA, N. 2002. Mean-shift segmentation with wavelet-based bandwidth selection. In *Proceedings of the sixth IEEE Computer Vision Conference*, IEEE, 43–50.

SUH, B., LING, H., BEDERSON, B. B., AND JACOBS, D. W. 2003. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ACM, 11–99.

SWAIN, M., AND BALLARD, D. 1991. Color indexing. *International Journal on Computer Vision 7*, 1, 11–32.

WOODRUFF, A., FAULRING, A., ROSENHOLTZ, R., MORRSION, J., AND PIROLLI, P. 2001. Using thumbnails to search the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 198–205.