UNSUPERVISED MERGER DETECTION AND MITIGATION IN STILL IMAGES USING FREQUENCY AND COLOR CONTENT ANALYSIS

Serene Banerjee and Brian L. Evans

Embedded Signal Processing Laboratory, Center for Perceptual Systems
The University of Texas at Austin, Austin, TX 78712-1084 USA
{serene,bevans}@ece.utexas.edu

ABSTRACT

When taking pictures, professional photographers apply photographic composition rules, e.g. avoidance of mergers. A merger occurs when equally focused foreground and background regions appear to merge as one object. This paper presents an unsupervised algorithm that (a) detects the main subject, (b) detects background objects merging with the main subject, and (c) reduces the visibility of merging background objects. Detection of the main subject requires automated adjustment of camera settings. The rest of the algorithm does not adjust or use the camera settings. The algorithm does not make assumptions about the scene setting (indoor/outdoor) or content. The algorithm is amenable to implementation on a fixed-point processor.

1. INTRODUCTION

Developing automated methods for improving photograph composition during image acquisition, e.g. for digital still cameras, can be helpful to amateur photographers. The two required key steps are: (1) automated segmentation of the main subject [1, 2, 3] and (2) automation of selected photographic composition rules [4]. This paper automates one of the photographic composition rules to avoid mergers. A merger occurs when equally focused foreground and background regions merge as one object. Fig. 1(a) shows an example of a merger where the trees appear to grow out of the main subject's head. Other examples include a horizontal line shooting through the subject's ears, and a knee or elbow extending from the frame edge.

In photography, the three-dimensional world is mapped to a two-dimensional picture. Professional photographers change camera settings, so that the main subject is in focus, while the objects in the background that merge with the main subject are blurred [5]. This preserves the sense of distance between the objects in the photograph. This paper presents an algorithm that automatically identifies the main subject and the background object that merges with the main subject. The background object is then blurred. The approach could be extended to identify more than one background object merging with the main subject.

2. BACKGROUND

The main subject can be in focus, while the background is blurred by diffused light, with the autofocus filter and appropriate camera exposure settings. We utilize this frequency difference between the main subject and the background to segment the main subject [3]. Our algorithm [3] involves adaptively filtering the image to enhancing the edges of the main subject, detecting the sharpened edges, and closing the boundary of the detected edges, in 13 multiplies, 9 shifts, 4 adds, and 6 byte memory accesses per pixel. It has lower complexity than Won, Pyan and Gray's iterative approach [2], and does not require training as does the Bayes net approach by Luo, Etz, Singhal, and Gray [1].

(a) (b)

Figure 1: Examples of (a) a merger of the main subject, the man, with the trees in the background (in color) and (b) the detected main subject mask in (a).

The rest of the paper assumes that the main subject mask has been detected. The subsequent tasks will be to segment the background and identify which background object merges with the main subject. The merging background object is then blurred. The software and color images for this paper are available at

www.ece.utexas.edu/~bevans/papers/2004/mergerDetection/

3. MERGER DETECTION AND MITIGATION

The generated main subject mask [3] divides the picture into foreground and background regions. Fig. 1(b) is the generated main subject mask for Fig. 1(a). The goals will be to segment the background, identify merging objects, and blur the picture. The formulation of the steps follow.

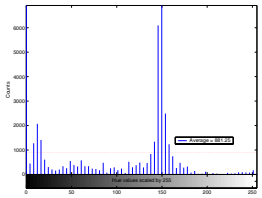

Figure 2: Histogram of the hue values for the background of Fig. 1(a), showing the average and peaks.

(a) Object 1 (b) Object 2 (c) Object 3

Figure 3: Some of the background objects (in color) for Fig. 1(a) identified by the color background segmentation.

3.1. Background segmentation

The color information is used for segmentation of the background objects. The red, green, and blue (RGB) image provided by the camera is transformed to the hue channel found in the hue, saturation, value (HSV) space. In HSV space, hue corresponds to color perception, saturation provides a purity measure, and value provides the intensity. A histogram in the hue space is then utilized for segmentation of the background region. Although hue does not model the color perception of the human visual system as accurately as CIELab, it is chosen because the transformation from RGB to hue has lower implementation complexity.

Let the hue values be on the interval $[0, 255]$ and broken into m-bins. The discrete probability distribution for hue values belonging to each bin is

$$P(\text{hue}_m) = \frac{c(\text{hue}_m)}{T_c} \tag{1}$$

where $c(\text{hue}_m)$ is the count corresponding to each bin and T_c is the total count of values in all bins. By modeling the background picture as a Gaussian mixture of hue values, the task is to further segment these m-bins into n-groups, where each group will identify a different object.

The term $\frac{T_c}{m}$ gives the average of the hue values. Any hue value above this average is marked as a dominant hue. Based on the available dominant hues, the n-groups are determined automatically, where each group contains only one dominant hue. Each group boundary lies halfway between two of the dominant hues. This ensures that the local maximums of the probability distribution, $P(\text{hue}_m)$ is captured in each group. Pixels with hue values falling in each of the identified n-groups form different background objects.

For the proposed algorithm, m is chosen to be 64, as it is assumed that a difference in four hue levels (i.e., 256/64 levels) would correspond to approximately the same perceived color. Fig. 2 shows the color histogram for the hue values with the average and the peaks for the background of Fig. 1(a). Based on the color histogram and the average value, $n = 10$ background objects are automatically identified for Fig. 1(a). Fig. 3 shows three of these identified background objects.

3.2. Merger detection

Based on the background segmentation, the background image can be modeled as a linear combination of the background objects. Thus,

$$S_b = \sum_{i=1}^{n} O_i \tag{2}$$

where S_b is the background image and O_i are the identified n background objects. Now, one or more of these background objects may merge with the main subject. We chose the background object that has the largest high frequency energy and is touching the main subject mask.

To automatically identify the merged object, each object O_i is transformed to a feature space representation, Ω_i, where $\Omega_i \in \Gamma$. Γ is defined as a weighted sum of the high frequencies contained in the spatial region of each object. High frequency coefficients are obtained from the first level of the two-dimensional Gaussian pyramid [6] of the intensity image. Gaussian pyramids are localized in space. The Gaussian pyramid could be replaced with a Laplacian pyramid, for an extra subtraction per pixel.

The high frequency coefficients are weighted with the inverse of the distance in space to the main subject mask. To compute the inverse distance transform, the distance transform coefficients are stored as a grayscale image, and are subtracted from 255 before multiplication with the high frequency coefficients. This assigns more penalty to the higher frequencies closer to the main subject. Fig. 4(a) and (b) show the Euclidean distance transform [7, 8] coefficients and high frequency coefficients obtained from the first level of the Gaussian pyramid, respectively. In Section 4, we will reduce the implementation complexity of the inverse distance measure computation.

An object O_i is detected to be merged with the main subject if its feature space representation, Ω_i, is more than a threshold. This threshold could be selected by the user. This paper presents an unsupervised approach in which the object O_i yielding the maximum value of the feature space representation, Ω_i, is identified to be the merged object. This unsupervised approach detects the object that produces the strongest merger and blurs the produced artifact. For Fig. 1(a), the tree object shown in Fig. 3(b), produces the maximum of the weighted sum of high frequencies, identifying that the tree merges with the main subject.

3.3. Selective blurring

The detected merged object, O_i^*, has feature a space representation, Ω_i^*. To reduce the effect of the merger, Ω_i^* needs to be reduced. As Ω_i is the weighted sum of the high frequencies, the high frequency coefficients are masked when

the image is reconstructed from the Gaussian pyramid representation. In Fig. 1(a), the high frequency coefficients of the first level of the Gaussian pyramid are masked out using the approximate shape of the detected tree object. The resulting image is shown in Fig. 5. To increase the amount of smoothing, masking can be extended to higher levels of the Gaussian pyramid decomposition.

4. IMPLEMENTATION COMPLEXITY

The proposed algorithm is shown in Fig. 6. The original RGB image of dimension $N \times M$ requires $3NM$ grayscale pixels (8 bits per grayscale pixel) without compression. The main subject is detected with 13 multiplies, 9 shifts, 4 adds, and 6 byte memory accesses [3]. The output binary main subject mask requires NM bits.

Background segmentation starts with a conversion from RGB to hue. The hue value calculation uses an intermediate variable, H', which is in the interval $[-255, 1275]$ and can be represented by a 12-bit signed integer. The pseudocode for the conversion follows:

```
min = min(R, G, B);
max = max(R, G, B);
δ = max - min;
if (R == max) H' = G-B; (within yellow & magenta)
else if (G == max) H' = 2δ+B-R; (within cyan & yellow)
else H' = 4δ+R-G; (within magenta & cyan)
H = (H' + 255) >> 3;
```

In the worst case, the conversion to hue requires 2 shifts, 3 adds, 6 compares, and 4 byte memory accesses per pixel. The histogram and thresholding requires 1 add and 1 compare per pixel. The hue values are stored in NM pixels, and a buffer of $NM \log_2 n$ bits stores the information of the segmented objects. As the number of objects, n, will practically be less than 2^8, the segmented objects' information can be overwritten on the buffer storing the hue values.

The intensity Gaussian pyramid first converts the color image to an intensity image by either

$$I = (R + G + B)/3 \text{ or } I = (R + 2G + B)/4 \qquad (3)$$

The former step requiring 2 adds and 1 multiply is suitable for programmable digital signal processors. For a hardware implementation, we could use the later, which requires 2 adds, a shift left by one bit (multiplication by 2) and a shift right by two bits (division by 4). Shifts can be used because the RGB values are non-negative. The intensity image is stored in NM pixels. Any level of the Gaussian pyramid can be computed by convolving the grayscale image with a 3×3 filter with power-of-two coefficients, which requires 9 shifts, 8 adds and 4 byte memory accesses per pixel. The 9 reads in image values to compute the convolution can be stored in registers in order to reduce the number of memory reads to 3 per pixel. The first level of coefficients are stored in NM pixels, and the intensity image may be overwritten in a sequential implementation of Fig. 6.

The inverse distance transform could be determined from the Euclidean distance transform [7, 8] by subtracting its value from 255. In this case, the inverse distance transform would be computationally intensive. We propose an approximate, lower complexity, inverse distance measure. Along each row (column) the distance of each "off" pixel from the nearest "on" one is computed and a ramp function is generated. The maximum of the horizontal (row) distance and the vertical (column) distance is taken as the distance from the nearest "on" pixel. In order to assign more penalty to the high frequency coefficients close to the main subject, the pixels closer to the main subject mask have a higher weight. The weights are stored in NM pixels. The distance measure requires 2 adds, 1 compare, and 2 byte memory accesses per pixel.

(a)

(b)

Figure 4: (a) The Euclidean distance transform coefficients and (b) the high frequency coefficients from the first level of the Gaussian pyramid for Fig. 1(a). The background object is detected to be merged if it yields the maximum of the weighted sum of (a) and (b).

Figure 5: The detected merged region is processed in the frequency domain to reduce the effect of the merger. The blurred trees induce a sense of distance. Image is in color.

For each background object, the intensity Gaussian pyramid coefficients are weighted by the inverse distance transform coefficients and summed. The background object with the highest sum is chosen as the background merging object, and the corresponding background object mask is output. The background object mask can be stored in the main subject mask buffer so as to reuse memory. All totaled, 1 multiply, 1 add, and 1 compare are required per pixel.

In the final step, the color Gaussian pyramid and reconstruction only have to be applied to those pixels in the binary mask input. For each pixel in the binary mask input, the first level of the color Gaussian pyramid transformation is calculated separably for each RGB planes. For each

color plane, 9 shifts, 8 adds, and 3 byte memory accesses are required for a 3×3 filter kernel. The high frequency coefficients for the merging background object are masked with 1 compare and 1 memory access per pixel. The output (merger reduced) image takes 9 shifts, 8 adds, 1 compare, and 1 byte memory access per pixel, and would be stored in $3NM$ pixels.

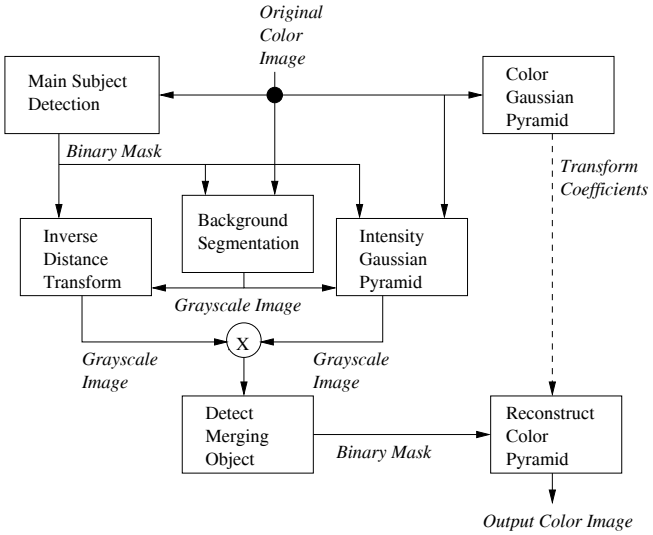

Figure 6: Proposed merger reduction algorithm for an original $N \times M$ color image. Storage is $3NM$ grayscale pixels (bytes) for the original, NM bits for a mask, and $3NM$ grayscale pixels (bytes) for the output (merger reduced) image. For a parallel implementation of the subsystems, an additional storage of $2NM$ pixels (bytes) is needed.

The computational requirements for each block in Fig. 6 are given in Table 1. All the blocks, except the main subject detection and color Gaussian pyramid/reconstruction, work only on the background image. Hence, the complexity will depend on the percentage of background pixels in the image.

The proposed algorithm was tested on several pictures. The merger reduced image for Fig. 7(a) is shown in Fig. 7(b). The background trees merging with the bird are blurred out, inducing a sense of distance.

5. CONCLUSION

This paper presents an unsupervised algorithm for automatic merger detection and mitigation when taking photographs in digital still cameras. The performance of the color based segmentation will be limited for highly textured backgrounds, which may require texture segmentation instead. Alternately, merger detection could be used to warn the user of a possible merger.

6. REFERENCES

[1] J. Luo, S. P. Etz, A. Singhal, and R. T. Gray, "Performance-Scalable Computational Approach to Main Subject Detection in Photographs," in *Proc. SPIE Conf. on Human Vision and Electronic Imaging*, Jan. 2001, vol. 4299, pp. 494–505.

Block	$\times$	$<<$	$+$	$\geq$	m
Detect main subject	13	9	4		6
Segment background	0	2	4	7	4
Intensity Gaussian pyramid	1	9	10		4
Inverse distance transform			2	1	2
Detect merging object	1		1	1	1
Color Gaussian pyramid		27	24		9
Reconstruct pyramid	1	27	24		3
Total	**16**	**74**	**69**	**9**	**29**

Table 1: Per pixel implementation complexity of the proposed algorithm in number of multiplications ($\times$), shifts ($<<$), additions ($+$), comparisons ($\geq$), and byte memory accesses (m). Detect main subject is applied to the entire image. The last two steps are only applied to the merging background object. The other steps are applied only to the background.

(a) Original image (b) Merger reduced

Figure 7: The proposed algorithm reduces the effect of the merger of the tree with the bird. The blurred trees in the processed image are distinguishable as a separate object from the main subject, i.e. the bird. Images are in color.

[2] C. S. Won, K. Pyan, and R. M. Gray, "Automatic Object Segmentation in Images with Low Depth of Field," in *Proc. IEEE Int. Conf. on Image Proc.*, Sept. 2002, pp. 805–808.

[3] S. Banerjee and B. L. Evans, "A Novel Gradient Induced Main Subject Segmentation Algorithm for Digital Still Cameras," in *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, 2003, to appear.

[4] S. Banerjee and B. L. Evans, "Unsupervised Automation of Photographic Composition Rules in Digital Still Cameras," in *Proc. SPIE Conf. on Sensors, Color, Cameras, and Systems for Digital Photography VI*, 2004, to appear.

[5] Kodak, *How to Take Good Pictures: A Photo Guide by Kodak*, Ballantine, Sept. 1995.

[6] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable Multiscale Transforms," *IEEE Trans. on Information Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.

[7] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, "Linear Time Euclidean Distance Transform Algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 529–533, May 1995.

[8] O. Cuisenaire and B. Macq, "Fast and Exact Signed Euclidean Distance Transformation with Linear Complexity," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Mar. 1999, vol. 6, pp. 3293–3296.