In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '2001)*, Volume I, pages 1063–68, Kauai Hawaii, 11th–13th Dec 2001

Alpha Channel Estimation in High Resolution Images and Image Sequences

P. Hillman J. Hannah D. Renshaw
Department of Electronics and Electrical Engineering, University of Edinburgh
The King's Buildings, Mayfield Road, Edinburgh EH9 3JL Scotland
pmh@ee.ed.ac.uk
Source and result images at http://www.ee.ed.ac.uk/~pmh/alpha

Abstract

For Motion Picture Special Effects, it is often necessary to take a source image of an actor, segment the actor from the unwanted background, and then composite over a new background. The standard approach requires the unwanted background to be a blue screen. While this technique is capable of handling areas where the foreground blends into the background, the physical requirements present many practical problems. This paper presents an algorithm that requires minimal human interaction to segment Motion Picture resolution images and image sequences. We show that it can be used not only to segment badly lit or noisy bluescreen images, but also to segment actors where the background is more varied.

1. Introduction

This paper presents a technique being developed to take Motion Picture resolution still images of actors and to segment the image into foreground (the actor) and unwanted background. The actor can then be composited over a new background. The algorithm works where many pixels are blends of background and foreground (e.g. from motion blur), and removes the background element of these pixels.

Most existing segmentation techniques require filming the subject in front of a blue screen [1]. Software and hardware systems, such as those by Vlahos [2], segment the image by comparing each pixel to the known backing colour.

We present an algorithm that requires some human interaction, but overcomes some of the limitations of blue screen compositing, allowing segmentation of actors from images with more normal backgrounds.

1.1. Previous work

Much work has been undertaken in the investigation of segmenting humans from unwanted backgrounds for areas such as low bandwidth video compression and face recognition. Many techniques rely on the background being darker than the person, and threshold the image intensity (also a common technique in compositing [3]). A similar idea is to produce a reference image of the background with no person present (perhaps by averaging the scene over a long period of time) and taking the difference between that image and each frame [4].

Techniques that consider spatial relationships between pixels might use edge detection and region growing [5]. Edges present problems in Motion Picture images because of motion blur and film grain as well as the high resolution involved in the source images — the edges become so soft it is difficult to locate them precisely. Texture Analysis techniques are also hindered by these problems.

Where edges become very soft, a single pixel may belong to both foreground and background. Its final colour will be some mixture α of a background and 'clean' foreground colour. A segmenter must find the value of α as well as the foreground colour for every pixel within the foreground area. Its output will be a clean foreground image C and a *Matte* or alpha channel α. The image can then be composited into a new background N using the *compositing equation*

$$R_{ij} = C_{ij}\,\alpha_{ij} + (1 - \alpha_{ij})\,N_{ij} \qquad (1)$$

for each pixel ij in the image. This scheme is adequate to compose images where there are blurred edges, but not where there is reflection or refraction. Zongker *et al* [6] developed a system capable of accurately compositing surfaces which are truly transparent and also reflective and refractive, such as a coloured glass. Their technique requires multiple artificial backgrounds and a static foreground.

Segmentation systems which produce foreground images and alpha channels are relatively unusual, and all require a human input. Corel's *KnockOut* package and Adobe's *PhotoShop* Extract Tool both require the user to indicate areas of the image which are definitely foreground and definitely background, and then process the area in

between. Ruzon and Tomasi [7] use a similar setup in their alpha estimation technique. Our approach is similar in that we find alpha values by measuring distances in colour space between two clusters of points sampled from the known background and foreground areas.

Mitsunaga *et al.* [8] developed a system that works with moving images. It requires a hand-generated alpha channel for the first frame. The boundary between foreground and background for the first frame is then known. The boundary in each subsequent frame is found using block matching. Alpha values are generated by calculating the gradient of the image within the boundary. They assume that, within the boundary, the background and foreground are relatively constant so any gradient in the image is due only to a transition between foreground and background. They thus require fairly sharp transitions between foreground and background, or very smooth images.

Very little academic work has been undertaken in the accurate segmentation of motion picture resolution images (>4 megapixel, 14 bits per channel) where many pixels in the image are blends of foreground and background.

2. The approach

2.1. User setup

In order that some parts of the image can be identified as the foreground, and to reduce computational load, some level of human interaction is used. The user is required to draw a 'hint image'. This is a greyscale image which is black where there is known background, white where there is known foreground, and grey over the area where there is a transition between background and foreground. The pixels marked as grey are the only ones that require processing. It is critically important that the white and black areas do not cover parts of the image which are not entirely foreground or background respectively, as this will cause poor results in that area.

2.2. Algorithm

Our algorithm takes the input and its corresponding 'hint image' and finds all the pixels along the border of the grey area. Pixels which are black along this border are marked as *background edge* pixels, and similarly the white pixels are marked as *foreground edge* pixels.

Small neighbourhoods of pixels within the unknown area are processed in turn, by scanning the image for the next unclassified pixel. All unclassified pixels within a fixed radius r_c of this next pixel are collected into a set U. The n_b background edge pixels nearest in the image to the centre of set U are collected to form a background set B. Every known background pixel within a radius r_b of any of these collected background edge pixels is added to the background set. The same procedure is used to generate a foreground set F. The points from which the pixels are selected are shown in Figure 1.

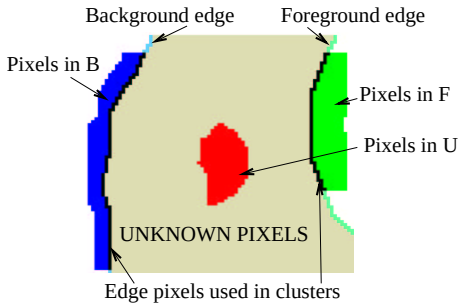


Figure 1. Pixels used to form background, foreground and unknown sets

In our approach, colours are considered as vectors in 3D RGB space. The RGB model is used rather than a perceptual colour model. Most image sensors — including film — are sensitive to Red Green and Blue. The averaging effect of motion blur (the major cause of blended pixels in images) is therefore best described in RGB space.

A blended pixel $\mathbf{p}$ from the set of unknown pixels U will be a blend of a clean foreground $\mathbf{f}$ and a clean background colour $\mathbf{b}$. We assume that $\mathbf{f}$ is within the foreground cluster and $\mathbf{b}$ within the background cluster extracted from the image. The pixel $\mathbf{p}$ will lie on the line $\overrightarrow{\mathbf{bf}}$, and α will be the ratio $|\overrightarrow{\mathbf{bp}}|/|\overrightarrow{\mathbf{bf}}|$

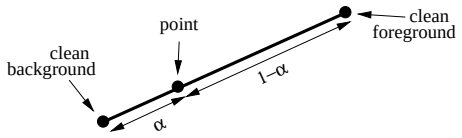


Figure 2. The α value of a point is the distance of the point along the line between its clean foreground and background colours in colour space

Even assuming that the clean foreground and background colours are within the clusters selected, there is no way to determine which two colours are the 'real' clean colours. Smith and Blinn [1] show that this is the case even with a perfect bluescreen and propose a technique which can perfectly resolve the correct clean foreground colour provided the scene is photographed with two different coloured backgrounds. This is implausible with human actors,

and so the clean foreground and background colours must be approximated.

Ruzon and Tomasi [7] use a similar technique to ours in order to extract clusters of foreground and background points close to a set of unknown pixels. They quantise each pair of foreground and background clusters into sets of subclusters $\{(\mathbf{v_1}, \sigma_{\mathbf{v}_1}, x_1),\ (\mathbf{v_2}, \sigma_{\mathbf{v}_2}, x_2), \ldots, (\mathbf{v_n}, \sigma_{\mathbf{v}_n}, x_n)\}$ and $\{(\mathbf{w}_1, \sigma_{\mathbf{w}_1}, y_1),\ (\mathbf{w}_2, \sigma_{\mathbf{w}_2}, y_2), \ldots, (\mathbf{w}_m, \sigma_{\mathbf{w}_m}, y_m)\}$ respectively, where $\mathbf{v}$ and $\mathbf{w}$ are the subcluster means, σ are their variances, and x_i and y_i are the proportion of points within the set allocated to subcluster i. They assume that the clean foreground colour is some linear combination of the foreground subcluster means, and calculate the colour and α by examining pairs of subclusters in foreground and background. Each pair of $\mathbf{v_p}$ and $\mathbf{w_q}$ yields a probability distribution of likely α values, according to the values of σ_p,σ_q and $x_p \cdot y_q$. The likelihood of a particular α value is found by summing the probability for that α value over each sensible pair p, q. The α value chosen is the one that yields the highest total probability.

The iterative search for the best alpha value and the vector quantisation system they use [9] are computationally expensive for very large images. Approximating the subclusters as Spherical Gaussians (where the covariance matrix is a scalar multiple of the identity matrix) is inaccurate in high resolution images.

Our procedure is based on the observation that the clusters tend to be prolate (cigar shaped) in RGB space: The pixels form the same basic colour with varying degrees of illumination, or they are part of a transition between two colours. Using Principal Components Analysis [9], the principal axis through the centre of each cluster is found. The limits of the cluster along this line are calculated by PCA transforming the set and finding the range (r_{min}, r_{max}) of the transformed set along the principal axis. The mean $[\mu_1, \mu_2, \mu_3]$ in each dimension of PCA space is also calculated. The transformed end points $\mathbf{p_{min}}$=$[r_{min}, \mu_2, \mu_3]$ and $\mathbf{p_{max}}$=$[r_{max}, \mu_2, \mu_3]$ are then inverse transformed back into RGB space to form $\mathbf{p_0}$ and $\mathbf{p_1}$. Fig 3 shows a cluster, and the line $\overrightarrow{\mathbf{p_0 p_1}}$. The range of the set is used rather than the variance because the clusters are very rarely symmetric. The range calculation is performed for both foreground and background, giving two lines $\overrightarrow{\mathbf{p_0 p_1}}_f$ and $\overrightarrow{\mathbf{p_0 p_1}}_b$, as shown in Fig 4.

It is now assumed that every pixel $\mathbf{s}$ in the set of unknown pixels U is composed of a clean background colour close to a colour $\mathbf{b}$, and a clean foreground close to a colour $\mathbf{f}$, where $\mathbf{b}$ is a point on the line $\overrightarrow{\mathbf{p_0 p_1}}_b$ and $\mathbf{f}$ lies on the line $\overrightarrow{\mathbf{p_0 p_1}}_f$. $\mathbf{f}$ and $\mathbf{b}$ are therefore the estimated clean foreground and background colours. The most appropriate points to choose for $\mathbf{b}$ and $\mathbf{f}$ are those points closest to $\mathbf{s}$, formed by finding

$$q = \frac{(\mathbf{s} - \mathbf{p_0}) \cdot (\mathbf{p_1} - \mathbf{p_0})}{|\mathbf{p_1} - \mathbf{p_0}|^2} \tag{2}$$

for foreground q_f and background q_b. If q_f and q_b are limited to the range $(0, 1)$, then

$$\mathbf{b} = \mathbf{p_{1_b}} q_b + \mathbf{p_{0_b}} (1 - q_b) \tag{3}$$

$$\mathbf{f} = \mathbf{p_{1_f}} q_f + \mathbf{p_{0_f}} (1 - q_f) \tag{4}$$

gives points $\mathbf{f}$ and $\mathbf{b}$ constrained to lie between $\mathbf{p_0}$ and $\mathbf{p_1}$. The alpha value can then be calculated using

$$\alpha = \frac{(\mathbf{s} - \mathbf{b}) \cdot (\mathbf{f} - \mathbf{b})}{|\mathbf{f} - \mathbf{b}|^2} \tag{5}$$

again limiting the result to lie on the range $(0, 1)$.

2.3. Clean foreground colour

In order to composite correctly, the clean foreground colour is needed as well as α. Recombining $\mathbf{f}$ and $\mathbf{b}$ (the estimated clean foreground and background colours) using the calculated value for alpha according to Equation 1, produces a point $\mathbf{q}$ which is the closest point on the line to the original pixel $\mathbf{s}$. By adding the vector $\overrightarrow{\mathbf{qs}}$ to the estimated foreground point $\mathbf{f}$, a better estimate $\mathbf{f}'$ of the clean foreground colour can be generated.

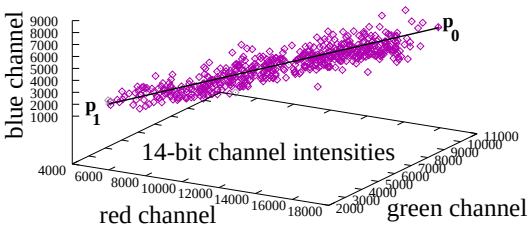


Figure 3. Cluster of points in RGB space with the line $\overrightarrow{\mathbf{p_0 p_1}}$

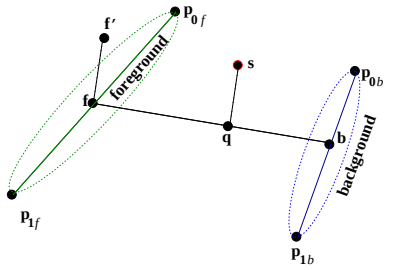


Figure 4. Position of points used to classify in colour space

(a) Original frame (b) *Keylight* package (c) *KnockOut* package (d) Our algorithm

Figure 5. Mattes from Frame 30 of CFC's *Rachael* sequence

2.4. Alpha correction

It has been assumed that the clean foreground and background colours lie very close to the principal axis $\overrightarrow{\mathbf{p_0p_1}}$ of the two clusters. For clusters with large variances, this can lead to an anomaly: pixels in the unknown area may be closer to the foreground axis than some pixels within the foreground cluster. These pixels will not be assigned $\alpha = 1$ if they lie between the principal axes of foreground and background. This can result in two identically coloured pixels (one within the foreground cluster and one in the unknown region) being assigned different alpha values. If the pixels are adjacent, a visible artifact will result.

To avoid this problem, the pixels in each cluster which are closest in colour space to the point being classified $\mathbf{s}$ are found. Each of these closest pixels are classified in the same way as $\mathbf{s}$ to give corrections, c_f for foreground and c_b for background. α for $\mathbf{s}$ can be then be adjusted using $\alpha' = (\alpha - c_b)/(c_f - c_b)$.

It is possible to prescale every value of c_b and c_f by some constants, which is equivalent to the *Matte Density* setting in the Ultimatte bluescreen package.

This stage also provides indication that the clusters are too similar: If c_b is too big, or c_f is too small, or if $c_b > c_f$ it is decided that the clusters are too similar to accurately classify this pixel — they will probably appear to overlap in colour space. Pixels which cannot be classified because of this similarity can be marked with a special alpha value, and must be classified by hand or using an alternative technique.

2.5. Multi-classification

Pixels are classified in small, non overlapping sets. If the foreground or background is detailed (i.e. non uniform) then two adjacent sets of pixels can be classified using pairs of clusters in very different positions. This will lead to detectable lines in the image. This can be avoided by multiply classifying pixels. Every n^{th} row and column is examined for a pixel within the unclassified region, forming the set U out of all points within a radius r_c of the scanned pixel, whether or not they have been previously classified. Values of n and r_c are chosen such that a single pixel will be classified more than once. The final α value is calculated by forming a weighted average of the calculated α values. The weight for each classification is proportional to $\overrightarrow{\mathbf{fb}}/\overrightarrow{\mathbf{qs}}$ (all the weights used in classifying a single pixel sum to 1). This scheme favours cluster pairs positioned such that $\mathbf{s}$ lies close to the line $\overrightarrow{\mathbf{fb}}$, indicating that the $\mathbf{f}$ and $\mathbf{b}$ are likely clean foreground and background colours.

3. Results

We first compare our algorithm to standard bluescreen systems. Fig. 5(a) shows a frame from the Rachael sequence, which contains large amounts of film grain noise. Fig. 5(b) shows the result of processing this image with Keylight, a professional bluescreen package. The background area in this sequence is affected by the film grain. Our result is shown in Fig. 5(d). Here, $r_c = 30$, $n_b = 30$, $n_f = 40$, $r_b = 4$, $r_f = 10$ (Our algorithm is not greatly sensitive to changes in these parameters). Single classification of each pixel has produced a satisfactory result. The area between the cardigan and curtain proved challenging, as the colours are very similar, but performance in other places is very good, even preserving the strand of hair by the mouth. Noise has not affected the background area. Fig 5(c) shows the matte produced by Corel KnockOut. This image has a noisy background, and has not accurately segmented the fingers or the area where the cardigan is in shadow.

While our algorithm performs very well even on noisy bluescreen images, we would like to use more normal backgrounds. Fig. 6(b) shows the result of segmenting the Gema image (Fig. 6(a)) using Corel KnockOut. There are many missed areas of foreground around the T-shirt and the top of the head, and much noise in the background. Fig. 6(c) shows the result of using our algorithm in multi-pass mode. This image contains fewer artifacts, with a much improved performance around the top of the head. The background and foreground in this area are very similar in colour. The background area is still slightly noisy. These pixels were

(a) Original frame (b) *KnockOut* package (c) Our algorithm

Figure 6. Results with *Gema* test image

detected as being badly classified, but the result of classification has been shown for comparison.

Full colour results (including the clean foreground images) are available from the authors' webpage.

4. Moving image sequences

Our objective is to avoid the need for user input for every frame of an image sequence. One possibility would be to use block matching to locate the unknown area in the current frame, similar to the approach taken by Mitsunaga *et al* [8]. The amount of inter-frame movement in our sequences, and the amount of motion blur resulting, makes this technique difficult to apply to Motion Picture images.

Instead, a different approach has been taken. The alpha channel generated from the previous frame can be used to assist with the segmentation of the current frame. The first step is to segment roughly the current frame, by using a K nearest neighbour classifier.

For each pixel in the previous and current frames, the mean colour of a $t \times t$ block centred around the pixel is calculated. For each of the blocks in the current frame, all blocks in the previous frame within a fixed aperture are scanned. The K nearest blocks to the current block (*i.e.* those that have the smallest difference in colour space) are found. Any block in the previous frame which is not entirely background or entirely foreground is ignored.

If more than three quarters of the K nearest blocks came from an area which was previously classified as entirely background or foreground, and the difference between this block and the nearest block is less than some threshold, the pixel is marked as background or foreground respectively. Otherwise, the block is likely to be a combination of background and foreground and is marked as unknown. The aperture used is dependent on the amount of inter-frame movement. This segmentation results in a noisy ternary segmentation of the current frame. Any pixel which borders the background and foreground is also marked as unknown, and the unknown area is dilated.

There are two alternative ways of processing these unknown pixels. One way is to find background and foreground edge pixels in this new segmentation, and use these to form clusters for classification. However, this segmentation is not accurate and our algorithm will fail if the known background or foreground region contains a blended pixel, as this blended colour will be considered to be a clean colour. An alternative is to classify each pixel $\mathbf{s}$ in the unknown region of the current frame using clusters extracted from the **previous** frame, again using the alpha channel to identify these regions. For this method, edge pixels in the previous frame must be located as before. Now, for each foreground edge pixel in the previous frame, the sets F and B are found as before, and all unknown pixels in the current frame within a large radius r_c are processed.

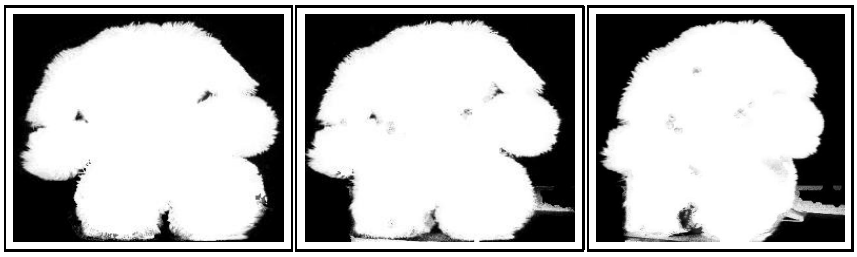
4.1. Results

Figure 7(d) was generated using our standard single frame approach with image 7(a). The remaining mattes were produced using the multi-frame technique. The mattes have been cropped in order to show them more clearly. In this sequence, the table top has a very similar colour to the soft toy, which has caused the streak visible in the right hand side of the image. Inter-frame movement is $\approx$ 170 pixels.

5. Conclusions and future work

We have presented a technique for segmenting subjects with soft edges from motion picture resolution images using limited human interaction, and extended this technique to allow classification of subsequent frames with little or no further user-input. Our algorithm appears to outperform commercially available packages on images with non-uniform backgrounds. As expected, in areas where the

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



(d) Output 1 (e) Output 2 (f) Output 3

Figure 7. Results with *Teddy* test sequence

background and foreground are too similarly coloured, performance is degraded.

Future work will look at these difficult areas and attempt to use spatial as well as colour-based information to assist the classifier. Where the actor is moving differently to the background, it may be possible to detect that certain areas of the image are background because they do not move with the actor. Automatic generation of the hint image for the first frame of a sequence will also be investigated. This will remove all need for human interaction, making the system suitable for live television and MPEG-IV content based compression.

Acknowledgements

Peter Hillman is supported by EPSRC studentship number 99303086 The *Rachael* sequence provided by the Computer Film Company, London.

References

[1] A. R. Smith and J. F. Blinn, "Blue screen matting," *Computer Graphics: Proceedings of the ACS*, pp. 259–268, 1996.

[2] P. Vlahos, "Electronic composite photography with colour control." United States patent number 4,007,487, February 1977.

[3] R. Brinkmann, *The Art and Science of Digital Compositing*. Morgan Kaufmann, 1999.

[4] P. I. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," in *Proceedings of the 6th British Machine Vision Conference*, pp. 347–356, 1995.

[5] A. Moghaddamzadeh and N. Bourbakis, "A fuzzy technique for segmentation of color images," in *Proceedings of the third IEEE Conference on Fuzzy Systems. IEEE World Congress on Computer Intelligence*, pp. 83–88, 1994.

[6] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin, "Environment matting and compositing," in *Computer Graphics: Proceedings of SIGGRAPH '99*, pp. 205–214, 1999.

[7] M. Ruzon and C. Tomasi, "Alpha estimation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 18–25, June 2000.

[8] T.Mitsunaga, Y. Yokoyama, and T.Totsuka, "Autokey: Human assisted key extraction," in *Computer Graphics: Proceedings of SIGGRAPH '95*, pp. 265–272, August 1995.

[9] M. Orchard and C. Bouman, "Color quantization of images," *IEEE Transactions on Signal Processing*, vol. 12, pp. 2677–90, December 1991.