AN IMPROVED ALGORITHM FOR SEGMENTATION OF MOTION PICTURE IMAGE SEQUENCES

P. Hillman, J. Hannah, D. Renshaw
University of Edinburgh Department of Electronics and Electrical Engineering, UK

Abstract  Motion picture Special effects often require the segmentation of image sequences into foreground and background. The background and foreground may then be processed separately and recombined, perhaps with additional elements in the scene. Alpha Estimation Systems can perform this segmentation where the background is natural (rather than an artificial background such as a blue screen) but they all require a user to create a hint image: an approximate segmentation of the image which is then refined automatically. A separate hint image is required for each frame of a sequence. This paper presents an algorithm which generates such hint images automatically, so that only a single input is required for an entire sequence. Results are presented which show our algorithm successfully generates hint images where an alternative approach fails.

An adaptation to our previously published alpha estimation system is also shown which is capable of segmenting images even when the foreground is lit from behind.

1 INTRODUCTION

For motion picture special effects, it is often necessary to segment a foreground subject (eg an actor) from the background behind them. Segmentation involves the generation of a matte or alpha channel. The alpha channel can be used to compose the foreground and background together again after independent processing of the two elements, perhaps with some additional elements in the scene which will then appear between the original foreground and background.

Special backgrounds such as bluescreens [1] can be used if the original background behind the subject is not required. The segmentation step (detailed by Vlahos [2]) is simple and robust, and the resultant matte can represent transparencies by outputting shades of grey in the alpha channel. It is a much more complex task to segment an image where the background is arbitrary, and transparent pixels are required.

Various algorithms [3, 4, 5, 6, 7] have been proposed which can generate alpha channels from still images. These all require human interaction in the form of a Hint Image that maps which parts of the image are known foreground and background, and those unknown parts which may be a mixture of the two. These algorithms sample the image in the known areas in order to process the unknown areas. However, for moving image sequences, a different hint image is required for each frame. Drawing such an image for every frame of a long sequence would be too laborious to be practical.

Chuang et al [6] proposed a solution to this problem using Optical Flow [8]. Given a frame \(n\) for which the hint image is known, the optical flow between frame \(n\) and \(n+1\) is generated. This creates a warping matrix which, when applied to the hint image from frame \(n\), generates a hint image for frame \(n+1\). Chuang et al used bidirectional optical flow, so that frame \(n+1\) can be generated from both frame \(n\) and from frame \(n+2\). Where there is a conflict between the two estimates for the hint image, the flow which gives the least error for that pixel is used. At motion picture framerates there can be large amounts of interframe movement, which presents problems to Optical Flow even if multi-resolution techniques are employed.

This paper presents an alternative technique for segmentation of image sequences. Given a hint image from one frame of the sequence, subsequent hint images can be generated automatically. We show that this algorithm can produce hint images in situations when Optical Flow fails. This technique is a modification of one we originally proposed in [3].

The paper is organised as follows. First, a brief summary of our technique for still image alpha estimation is presented, full details of which can be found in [4]. Where backlighting is present in an image, the assumptions made by this algorithm (and most other algorithms) break down. A modification to our algorithm to handle backlighting is then presented. We then present our technique for generating hint images for other frames in a multiframe sequence.

2 STILL IMAGE SEGMENTATION

2.1 User input for alpha channel estimation

A hint image is a rough, hand segmentation of the image, into three types of area: definitely background, definitely foreground, or unknown. The unknown area usually forms a band between the known background and foreground areas, and is the only one which requires processing. Pixels within this area may be part of the background, the foreground, or a combination of the two, and must be assigned an alpha value accordingly. We mark the background, foreground and unknown areas in black, white and grey respectively. Fig. 1 shows an example “hint image” created in this way.

2.2 Alpha estimation

Where there is blurring or transparency in a scene, individual pixels may be a mixture \(\alpha\) of foreground and background. To segment the scene, it is necessary to find the
value of $\alpha$ as well as the foreground colour for every pixel within the unknown area. The output from a segmenter will be a foreground image $C$ and a Matte or alpha channel $\alpha$. Image $C$ contains no trace of the background and is therefore referred to as a Clean foreground image. Pixels in the clean foreground image will be called clean foreground colours. Once $C$ has been found it can be composited into a new background $N$ using the compositing equation

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

for each pixel $ij$ in the image.

Segmentation consists of two steps for each pixel: estimation of the clean foreground colour and the corresponding clean background colour for the given pixel, and then calculation of alpha by inverting the compositing equation and solving for alpha.

Our approach for alpha estimation is detailed in [3]. To classify a single pixel $s$ within the unknown area, we extract clusters of nearby known background and foreground points. Observation has shown that these clusters form prolates (cigar shapes) when plotted in RGB colourspace, as shown in Fig. 2. Using Principal Components Analysis, we find the principal axis of each cluster. That is, we find a line $p_0p_1$ through the centre of each cluster. The length of the line is limited by the range of the cluster. The foreground $f$ and background $b$ colours are the closest points on each of these lines to the point $p$, as shown in Fig. 3. Point $q$ is the nearest point on the line $bf$ to $s$. Alpha is then given by the ratio of the length $bs$ to the length $bf$.

2.3 Results

Fig. 4 shows the results of running our algorithm on images with natural backgrounds. While there are problems caused where there is insufficient contrast between background and foreground, the algorithm was capable of successfully resolving the individual strands of hair at the top of the head.
2.4 Other approaches

Other published algorithms for alpha estimation work in a similar fashion: Ruzon and Tomasi [7] split the extracted clusters into a Mixture of Gaussians and, based on the probability distributions of the foreground and background clusters, search for an alpha value that gives a good result using a Bayesian measure. Chuang et al [6] also use Gaussian Mixture Models and a Bayesian formulation, but scan each individual foreground/background cluster combination separately. This simplifies the Bayesian equation making a numerical solution possible. The alpha value they chose is the best result achieved with a Bayesian measure. Finally, Wexler et al [9] solve a Bayesian formulation based on multiple registered images of the foreground with different backgrounds, or where a single background is moving over a static foreground due to parallax.

We have compared our approach to these other hint image based techniques and demonstrated that it produces superior results [4].

3 BACKLIT IMAGES

Actors in scenes are commonly illuminated from behind ("backlit"). This causes a highlight along the edge of the foreground, as shown in Fig. 7(a), taken from the Dragonheart sequence (scanned from a 35mm print of a trailer, 2286 × 1224 pixels, 24bpp, 24fps). Backlighting helps the foreground to stand out (and therefore can help a composite to look more natural) but causes a problem when estimating alpha. Fig. 7(c) shows the result of compositing the alpha channel generated by the algorithm described in section 2.2 over a black background. This shows that the estimated alpha channel is too small - the pixels that were highlighted from the backlight have been classified as background rather than foreground and are missing from the composite scene.

Fig. 5 plots the colour vectors for pixels in a column of Fig. 7(b), with the foreground and background colours marked as crosses. The line connects adjacent pixels between these two end points. If the compositing equation is to hold, these points should be linear combinations of the foreground and background. Thus the intermediate points should lie on a straight line between foreground and background. Instead, the line bends towards a value corresponding to white. The intermediate pixels are a combination of not two but three colours: The clean background, the clean foreground, and a “clean backlight” colour m.

To produce an accurate matte in this case an extension of normal alpha values is required which permits the mixture of three, rather than two colours. Given three clean colours f, b and m, we wish to know for each pixel p how much of each clean colour is required to produce p. Fig. 6 shows the position of these four points.

Rather than using point p directly, the point q is used, which is the nearest point from point p on the plane formed by f, b and m. q is found by calculating the normal n to the plane:

\[ n = (m - f) \times (b - f) \]  
\[ \vec{q} = \frac{n}{|n|} \cdot \frac{(p - f) \cdot n}{|n|} \]  
\[ q = p - \vec{q} \]

Now, the point c must be found. This is the point where the line from b through q intersects with \( \overrightarrow{fm} \). This is found by solving the simultaneous equations of the two lines:

\[ c = f + x m \]
\[ c = b + y q \]

Writing \( u = q - b, v = b - f \) and \( w = m - f \) for short, and equating 5 and 6 gives

\[ f + x w = b + y u \]
\[ x w - y u = v \]

This is an over-specified system, since it has two unknowns in three equations, and can be solved for x using...
just two of the three equations:

\[
x = \frac{u_0 v_1 - u_1 v_0}{w_1 u_0 - w_0 u_1}
\]

(10)

\(x\) can then be substituted into equation (5) in order to find \(c\). Alpha is then found in the normal way

\[
\alpha = \frac{|q - b|}{|c - b|}
\]

(11)

Clean foreground and background colours \(f'\) and \(b'\) are found by adding \(qp\) to \(c\) and \(b\) respectively. Using \(\alpha\) to blend these two colours exactly produces \(p\) as required.

The backlight colour can be estimated geometrically from the pixel values. It will be a point \(m\) such that \(f', b\) and \(m\) enclose virtually every point within a small area of the transition region. It may also be possible to estimate the point using a lighting model. However, since the clean foreground colour tends to be a uniform colour and intensity across the whole image, it can be easily estimated and entered by hand.

Fig. 7(d) shows the result of using the backlight alpha estimation algorithm on a small area the frame (Fig. 7(b)). The backlight colour in this case was set to twice the intensity of full brightness in the frame. This clearly shows a more accurate segmentation, as the highlighted area has been correctly detected as foreground.

4 MOVING IMAGE SEQUENCES

For motion picture special effects it is necessary to segment moving images rather than single frames. It would be possible to segment an entire moving image sequence by requiring a human operator to generate a hint image for every separate frame of the sequence and segmenting them as independent still frames. In order to segment sequences with little extra input, a system that generates hint images for subsequent frames has been developed.

Chuang et al use optical flow to update their hint images. Given a frame \(n\) for which an alpha channel has already been produced, the hint image for frame \(n+1\) can be generated from the motion vectors between the two frames. If pixel \(xy\) in frame \(n\) has corresponding motion vector \((\delta_x, \delta_y)\), then pixel \((x + \delta_x, y + \delta_y)\) is marked as background if the alpha channel for \(xy\) is 0, foreground if the alpha channel for \(xy\) is 1, and unknown otherwise. By generating this data from two known frames (e.g frame \(n+2\) as well as frame \(n\) for frame \(n+1\)), and fusing together the two results, reliability is improved. Black’s Robust Optical Flow algorithm [8] was used to estimate the optical flow.

While Optical Flow is very successful when resolution is relatively low and interframe movement is small, it is not always suitable for segmenting motion picture sequences. The high resolution makes processing impractically slow (much slower than modifying the hint images by hand) and large interframe movements cause the optical flow to fail. Even if the bidirectional technique described by Chuang et al is used it is not always possible to generate motion vectors in areas which become unoccluded between frames. For example, Figs. 9 and 13 show cases where the hint image generated by optical flow is incorrect. In these cases an alternative technique is required.

Our assumption that the background and foreground are different colours, which is required for the alpha channel estimation described in section 2.2, can be update hint images. A probability based colour classifier can be used to generate a hint image for frame \(p\) given an alpha channel for frame \(n\). Frame \(n\) is split into three regions: those for which the alpha channel is zero (i.e. background), those for which the alpha channel is one (foreground) and all other values (similar to the unknown area). Blocks of size \(m \times m\) pixels (typically \(128 \times 128\)) are then processed and a Mixture of Gaussians found for the distribution of pixel values in each block for each region: The RGB colour values are quantised [10] to derive a set of \(N\) means \(\mu\) (typically about ten). Each vector is then assigned to the subcluster which contains the most similar mean. The result is a set of subclusters \(\{(n_0, \mu_0, \Sigma_0), (n_1, \mu_1, \Sigma_1), \ldots, (n_M, \mu_M, \Sigma_M)\}\) where \(n\) is the number of pixels in the subcluster, \(\mu\) is the mean and \(\Sigma\) the covariance matrix. The sets are cached as they are used several times during processing.

To create an alpha channel for frame \(p\), blocks \(C\) of \(m \times m\) pixels are processed together. A search aperture is selected which must be large enough to cover all movement. In the results presented here, the aperture was \(384 \times 384\), i.e. \(3\times3\) blocks of 128 pixels. The sets of clusters for each of the blocks covered are merged. The result is three sets \(F, B\) and \(U\) for foreground, background and unknown areas respectively.

For each pixel \(P\) in block \(C\), the foreground, background and unknown area probabilities are found:

\[
p_f = \frac{1}{\omega |F|} \sum_{j \in F} n_{F_j} (\Sigma_{F_j}^{-1})^\frac{1}{2} e^{-\frac{(p-\mu_{F_j})^T \Sigma_{F_j}^{-1} (p-\mu_{F_j})}{2}}
\]

(12)

where \(|F|\) indicates the total number of pixels in the set and \(P\) is the colour of the pixel under classification. \(p_b\) and \(p_u\) are found in a similar manner. In the event that one of the regions has no pixels, the probability for that region is set to zero. \(\omega\) is a weighting factor which ensures that all probabilities sum to one.

\(P\) is assigned to the region which corresponds to the highest probability measure. Multiple source frames can be used to make the hint image more accurate. The probabilities for each reference frame are calculated independently and the pixel assigned to whichever region has the highest probability across all reference frames.
The resultant hint image can be noisy and often has an unknown area which is slightly too small. This occurs because the distribution of the unknown area overlaps with the foreground and background clusters. To rectify this problem, small regions are removed and the unknown area dilated slightly. The result is a clean hint image which can be used to segment the frame.

4.1 Results

With the short Teddy sequence, a hand drawn hint image was generated from frame 1 and processed to generate an alpha channel, as shown in Fig. 8. Black’s optical flow algorithm fails for this image sequence, due to the aperture problem: the scene is too uniform for the optical flow to resolve to the correct motion in the scene. Fig. 9 shows the incorrect hint image generated for frame 2 of the Teddy sequence using Optical flow. Clearly, the grey area in the hint image — which should overlap the boundary between the bear and the background — is not correctly positioned. Fig. 10 shows the hint image generated using our algorithm and the output alpha channel and composited image generated using our estimation algorithm using this hint image. The results for frame 3 are shown in Fig. 11. While these results are not perfect (note the disoccluded table surface on the left right hand side of the image which is classified as foreground in frame 3) they are a significant improvement over the use of Optical Flow.

Optical Flow also fails on the Dragonheart sequence. Fig. 12 shows the result for frame 1 of the sequence, generated using our still image alpha estimation algorithm. Fig. 13 shows the poor hint image subsequently generated by Optical Flow for frame 15 of the sequence.

We then generated hint images for this sequence using a multiple source frame approach. First, the hint image for the final frame of the sequence (frame 21) was estimated from frame 1. Since the two frames are so dissimilar, there were areas which required minor correction by hand, as shown in Fig. 14. This was the only other frame that required such modification and this task is much quicker than generating an entire hint image by hand. The intermediate frames were then generated using both frame 1 and frame 21. These frames can be processed in parallel since there are no interdependencies. Figs. 15 to 17 show the results for frames 5, 10 and 15 of the sequence. Backlit alpha estimation was used for alpha estimation in this sequence, since the background is brighter than the foreground and backlighting is present.

When the new segmented foreground is composited over a new background and the video sequence viewed, some sparking can be observed: individual pixels appear and disappear in subsequent frames causing a flickering effect at the edges. This is partly due to the noise sensitive nature of the backlight alpha algorithm. There are also minor artifacts visible around the horse’s saddle in the intermediate frames. These could easily be corrected by a special effects artist.

5 CONCLUSIONS AND FUTURE WORK

This paper has presented an algorithm for automatically generating hint images for a multiframe image sequence. Using this algorithm, only a small number of hint images need to be generated by hand in order to segment an image sequence, with only two of the 21 frames in the Dragonheart sequence being created by hand. Since Optical Flow fails on this sequence the only alternative to our technique would be to create every hint image manually. Our technique operates much faster than Optical Flow, since our algorithm is an order of magnitude faster than optical flow computation and can be run in parallel. Thus, there may be cases where Optical Flow succeeds but it is still preferable to use our technique. Therefore, it would be a valuable tool in semi-automatic image segmentation for motion picture special effects and post production.

Future work will investigate the possibility of simultaneous classification of multiple frames in order to reduce sparkling by enforcing temporal consistency of alpha values.

Acknowledgements Peter Hillman was supported by EPSRC studentship number 99303086 and award GR/S06578/01.

References

Figure 7: Result of using backlight alpha estimation

Figure 8: Results of running our still image algorithm on frame 1 of the Teddy sequence
Figure 9: Hint image for frame 2 of the Teddy sequence is incorrect when estimated using Optical Flow


Figure 10: Results for frame 2 of the Teddy sequence.
Figure 11: Results for frame 3 of the *Teddy* sequence

Figure 12: Results for frame 1 of the *Dragonheart* sequence

Figure 13: Detail of incorrect hint image for Frame 15 of the *Dragonheart* sequence generated using Optical Flow
Figure 14: Results for frame 21 of the Dragonheart sequence

(a) Hint image automatically generated from frame 1

(b) This hint image hand modified

(c) Generated alpha channel

(d) New composite image

Figure 15: Results for frame 5 of the Dragonheart sequence

(a) Hint image automatically built from frames 1 and 21 for frame 5

(b) Generated alpha channel

(c) New composite image

Figure 15: Results for frame 5 of the Dragonheart sequence
Figure 16: Results for frame 10 of the *Dragonheart* sequence.

Figure 17: Results for frame 15 of the *Dragonheart* sequence.