KEYFRAMECUT - A FAST GRAPH-CUT BASED ALGORITHM FOR HUMAN VIDEO SEGMENTATION

Viraj Prabhu, Siddhant Jain *
Birla Institute of Technology and Science (BITS), Pilani
Pilani, RJ 333031, India
virajp@vt.edu, siddhaja@adobe.com

ABSTRACT

Segmentation of humans from video is an important problem in computer vision with widespread applications. Various approaches have been proposed that optimize probabilistic models combining object detection with visual cues and propagate the segmentation achieved. We present Keyframecut, a real-time algorithm for human upper body segmentation in video that combines colour, motion, background and shape prior cues along with minimal user interaction and no propagation in a simple graph-based model that is solved using min-cuts to yield accurate foreground segmentation. We demonstrate experimental results with promising accuracy and real-time performance on the Microsoft i2i dataset.

1 INTRODUCTION

The problem of human segmentation in video has attracted extensive research over the past decade owing to its widespread applications. However a general knowledge transferable solution that works across all backgrounds is inherently difficult due to wide variations in parameters such as illumination, pose and colour along the length of the video. This problem of an automatic green screen or chroma keying of humans is of special interest in video conferencing and content authoring scenarios to enable background blurring or substitution for privacy or aesthetic reasons. A real-time implementation can enable these operations on the fly, improving the appeal of the feature.

Our approach takes advantage of the extensibility of graph cut to incorporate several different cues for robust human segmentation in video. The algorithm executes in real time and a single frame segmentation does not rely on the segmentation of its preceding frames. The proposed algorithm combines fast detection schemes with colour, contrast, motion and adaptive background information that greatly reduces the overall running time of algorithm. The multitude of supplementary high and low level cues preserves accuracy and ensures robustness.

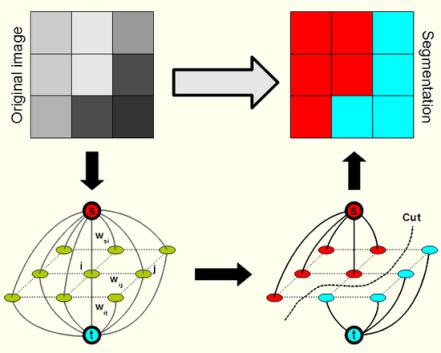
1.1 PRELIMINARIES

Graph cuts are widely used for interactive segmentation and are versatile across a variety of image scenarios.The segmentation is posed as the energy minimization of an image modeled as a graph using min-cuts, and it is possible to incorporate multiple image parameters such as shape, color and motion in the energy equation.

The graph-cut algorithm models an image as a graph, each pixel corresponding to a single node along with an additional source and sink node for the foreground and background, respectively. The user initially labels a few pixels as foreground and background seeds. Edges join each pixel node to its neighboring pixel nodes as well as each pixel node to the source and sink. The edge weights assigned to each type of edge captures the discontinuity in intensity or a combination of factors between neighboring pixels and a notion of dissimilarity of each pixel from the source and sink, and consequently the foreground and background. A disjoint partition of the graph is performed, ensuring that source and sink belong to separate partitions, along edges of minimal total weight using a min-cut algorithm to yield the final segmentation.

*Work done when V.P and S.J were interns at Adobe Systems India

Figure 1: Graph-cut formulation



The energy equation, for image labeling $L = [L_1, L_2...L_n]$ is of the form:

$$E(L) = \beta.B(L) + R(L)$$

Where $B(L)$ is the boundary penalty for neighboring pixels having opposite labels, and $R(L)$ is the region penalty for each pixel belonging to either foreground or background. The above energy expression is used to assign weight to terminal and neighborhood links in the image modeled as a graph, which is solved using min-cuts to produce the desired segmentation.

Grabcut [8] is a graph-cut based segmentation technique that reduces user effort by applying energy minimization iteratively until convergence to an initial foreground bounding box provided by the user.

2 Related Work

The areas of human tracking and video segmentation have attracted extensive research in the past decade. Several approaches have been proposed that combine visual cues with prior information obtained from training [1] or user intervention in probabilistic models that are solved using min-cuts [2].

In videos, segmentation is often propagated as it is unfeasible to provide user defined graph cut seeds for each frame, such as by Bai et al [3] which propagates local classifiers along a frame segmentation using optical flow, which are combined with shape priors and solved using graph cuts. Computing optical flow is computationally expensive and errors tend to accumulate and propagate in videos of long duration without repeated user intervention.

Approaches such as Hernandez-Vela et al [4] and Ladicky et al [5] that combine object detectors with low level visual information in probabilistic frameworks such as CRFs have demonstrated promising results on combining low and high level cues.

Background subtraction is a classic segmentation technique which traditionally requires the image of a clean plate background. To avoid this inconvenient workflow, alternative techniques have been suggested to train adaptive background mixture models [6] and more recently KaewTraKulPong et al [7] that train a classifier for each pixel and approximate the Gaussians corresponding to background pixels online.

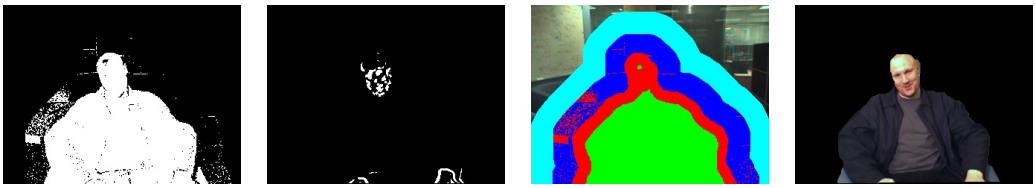
The modeling of shape and pose estimation is a field of extensive research in itself with numerous approaches. However, these approaches require training and are often computationally expensive, both of which are unfeasible for applications of our approach.

3 Proposed Methodology

The algorithm assumes a static background, which can usually be obtained in a content authoring of video conferencing environment. In our approach, a target frame typically has a person hereafter

referred to as the presenter facing the camera, typically with his head, shoulders and arms in the frame.

The user is presented with the first frame of the target video and is required to segment the presenter using GrabCut, by providing a bounding box and optionally labeling a few pixels as seeds to create the first keyframe.



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

Figure 2: (a) to (d) show different cues used in Keyframecut. (a) is the map generated by the colour GMMs, (b) demonstrates the fused motion edge map, (c) are the final seeds given as input to GrabCut, with colours corresponding to probability labels as shown in the legend below. (d) is the final segmentation. Motion hulls are not utilized in this frame segmentation, so they have been excluded.

Table 1: Legend

Color	**Label**
Green	Definite foreground
Red	Probable foreground
Deep blue	Undefined
Cyan	Probable background
Rest	Definite background

3.1 COLOR MODELS

Separate Gaussian Mixture Models are trained on the keyframe as global color models for the foreground and background using k-means with 5 components each. This value has been fixed heuristically and found to work well across the test videos used.

3.2 FACE TRACKING

A face detection algorithm is run to track points along the presenters face. An ellipse is fit to these tracker points and is assigned a high foreground probability. A robust face detector that works with side and front profiles is used. The face ellipse from the preceding frame is used if detection fails in the current frame and the facial ellipse is not allowed to move more than a certain pixel distance - set to half the width and height of the face ellipse along each axis respectively - from its position in the preceding frame, to offset spurious face detections.

3.3 ROUGH SHAPE PRIOR

The centre of the face ellipse is computed and the initial keyframe mask is translated to lie over the detected face, such that the centers of the keyframe face detection and computed face ellipse coincide to create an approximate shape prior. The shape prior is dilated in levels to create bands with pixel foreground probability decreasing with increasing distance from the shape prior. The GMMs trained on the keyframe are used to classify pixels within these bands. Each pixel is assigned a foreground probability by comparing the confidence scores generated by each classifier.

3.4 FUSED MOTION-EDGE MAP

An adaptive background learning model is trained wherein each pixel is modeled as a Gaussian mixture and colors that remain static for longer in the length of the video, the measure of which is computed by the weights of the mixture, are thresholded out as likely static background pixels to create a motion map. The model is updated online to account for illumination variance. A weighted combination of this motion map with a Sobel edge map is used to generate a fused motion edge gray-scale map, assigning higher intensity to edges with strong motion as these are likely to be foreground edges in accordance with the static background assumption made. This map is thresholded and white pixels are assigned high foreground probability.

3.5 MOTION HULLS

Convex hulls of the pixels outside the extended shape prior bands that are marked for motion by the thresholded edge motion map are created and global color models as well as skin color models in the HSV color space are used to classify the pixels within these convex hulls. The motivation is to look for moving hands and attached clothing based on color and motion, which are the most likely candidates for foreground portions that extend outside the shape prior.

3.6 ENERGY MINIMIZATION

Finally a graph cut is performed on the resulting map to produce the segmentation. This entire process is independent of the segmentation of the preceding frames and hence it is possible to seek to a frame directly and obtain its segmentation. Additionally, the reliance on cues that are not computationally expensive reduce the running time while the use of multiple cues preserves the accuracy. The supplementary nature of the cues is validated in the results section.

In the scenario that the users pose changes significantly from his initial shape prior, in which case the user has the option of adding an additional keyframe, which is used to recompute color models and shape prior for the remainder of the video. Once the frame has been segmented, it is then post processed using the techniques described in the following section.

3.7 POST PROCESSING

The binary segmentation mask is first post processed using morphological opening in order to remove salt and pepper noise. A keyframe difference is then performed wherein the intensity of a pixel marked as foreground in the segmented frame is RGB-distance thresholded from the same pixel in the keyframe labeled as background and accordingly reclassified as background. The distance is chosen to account for normal illumination variance, and is fixed heuristically in our approach. This is done to partially overcome the limitations of graph cut in cases of poor color separability around the foreground-background boundary where it tends to include spurious background pixels into the foreground.Once the post processing is complete, the frames are written to the output video file

3.8 IMPLEMENTATION

This proposed algorithm has been implemented on an Intel 2.7 GHz i5 processor with 6GB of RAM, and an average time performance of 350 milliseconds per frame has been observed.

4 RESULTS AND DISCUSSIONS

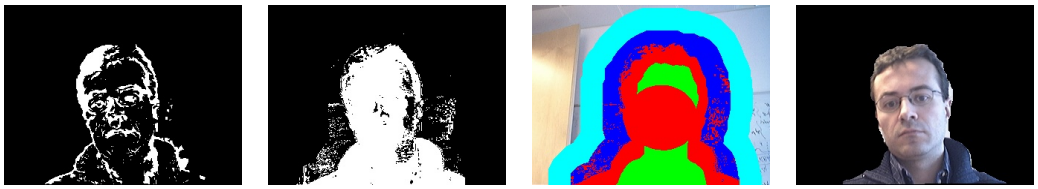
Keyframecut has been tested on the Bilayer Segmentation of Live Video dataset published by Criminisi et al [9], which contains ground truth data in the form of hand labeled segmentation masks for a subset of the total video frames. The metrics computed were segmentation accuracy, precision, recall and execution time. Segmentation accuracy was computed as the percentage of misclassified pixels per frame, averaged over the number of frames.

The algorithm demonstrates consistent performance metrics as shown in Table 2. High accuracy and precision were observed which is a critical attribute of a video conferencing or content authoring

scenario.The execution times for computing each cue was measured and a near-real time execution time of 340ms / frame was observed.

Table 2: Results

Approach	Segmentation Acc.	Precision
Criminisi et-al [1]	0.951	0.959
Ours	0.932	0.941



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

Figure 3: (a) to (d) showing a simpler head segmentation where all three of (a) motion edge (b) colour and (c) face detector (shown by red central ellipse) work well together to produce final segmentation (d).

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

Figure 4: (a) to (e) illustrate the use of (b)motion hulls in addition to (a) colour, face and (c) fused motion edge map to create the final seeds map in (d) which produces the segmentation in (e).

5 Conclusions and Further Work

We have demonstrated Keyframecut, an algorithm for segmenting humans from videos that combines multiple supplementary cues in a simple graph based model to preserve accuracy while still providing real-time performance. We have avoided the use of complex shape modeling techniques to satisfy our real time execution requirement and have proven that the a combination of color, motion and background cues can be employed with a simple shape prior to perform robust segmentation. We have also validated the claim of the cues aiding one another towards a better segmentation, and have demonstrated good results on the Bilayer Segmentation dataset.

6 References

[1] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, Bilayer segmentation of live video, in IEEE Conference on Computer Vision and Pattern Recognition, 2006.

[2] Y. Boykov and M.-P. Jollie. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In Proc. Int. Conf. on Computer Vision, pages CDROM, 2001

[3] Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graph. 28 (2009) 1-11

[4] Hernndez-Vela, Antonio, et al. "Grabcut-based human segmentation in video sequences." Sensors 12.11 (2012): 15376-15393.

[5] Ladick, ubor, et al. "What, where and how many? combining object detectors and crfs." Computer VisionECCV 2010. Springer Berlin Heidelberg, 2010. 424-437.

[6] Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.. Vol. 2. IEEE, 1999.

[7] KaewTraKulPong, Pakorn, and Richard Bowden. "An improved adaptive background mixture model for real-time tracking with shadow detection." Video-based surveillance systems. Springer US, 2002. 135-144.

[8] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309314, 2004.

[9] http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm.