AUTOMATIC FOREGROUND EXTRACTION IN VIDEO

Haoqian Wang, Bowen Deng, Kai Li, Yongbing Zhang and Lei Zhang

Shenzhen Key Laboratory of Broadband Network and Multimedia
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
wanghaoqian@tsinghua.edu.cn

ABSTRACT

This paper presents an automatic and efficient system for extracting dynamic objects of interest from videos. We take advantage of a saliency map and an optimization-based segmentation algorithm to extract the foreground objects automatically in some key frames. Then, the segmentation results in those key frames are propagated to other frames via an error map-based propagation scheme. Finally, a Bayesian matting-based refinement approach is employed to to handle the topology changes. Experiments show that our system is able to generate high quality results at a low computation cost.

Index Terms— Foreground extraction, foreground object

1. INTRODUCTION

Foreground extraction refers to the problem of extracting foreground objects from images and videos. It is one of the fundamental problems in image processing field, and has attracted intensive attentions from both academia and industry.

Although tremendous progress has been made in the field of video segmentation/matting in the past two decades, previous approaches mainly focus on accuracy while fail to take efficiency into consideration. On one hand, most matting systems rely heavily an accurate trimap (an initial segmentation result), which is always obtained though intensive user interaction, such as some video cutout systems [1, 2, 3, 4]. On the other hand, user interaction is often required to refine the matting details in the non-key frames when using a frame-by-frame propagation strategy. The bilayer segmentation system [5], for example, employs much interaction in local refinement when a good matte cannot be produced by propagating the results from key frames to non-key frames.

Therefore, in this paper we aim at developing an automatic and efficient system for accurately extracting foreground objects from videos. The balance between accuracy and efficiency is sought through a set of techniques. First, a saliency map is used to initialize the foreground region in some key frames. Then the foreground object is extracted by a GrabCut-based optimization scheme. With the help of salient object detection and optimization-based segmentation, the extraction results in key frames are generated automatically. Secondly, we propose an intelligent strategy to automatically propagate the accurate segmentation results through an error map and refine the details in each intermediate frame. The propagation unit consists of two parts: the color learning model and motion learning model, which model the color and topology changes, respectively. Once the changes are too big, the local refinement unit is used to handle the occlusion, dis-occlusion, and motion blur.

Experimental results show that our proposed system has not only a low computation cost and requires no user interaction, but also comparable accuracy with state-of-the-art algorithms.

1.1. Related Works

This paper is related to research on video matting, especially video object cutout. A survey on video matting can be found in [6]. Generally, various image matting methods, such as Bayesian Matting [7], Poisson Matting [8], Random Walk Matting [9], Closed-form Matting [10], Easy Matting [11], and Robust Matting [12], can be extended to videos by adopting a two-step framework [2]. Recently, a tri-level propagation approach [13] is proposed to deal with regions with topology changes. Video object cutout systems, such as [1, 2, 14], combine the motion and color models together to propagate well-segmented key frame results to non-key frames. However, they usually require intensive user interaction to refine the regions with similar appearance between the foreground and background.

2. SALIENCY-BASED SEGMENTATION

We first show how to obtain the accurate segmentations on some key frames in this section, and then propagate the results from key frames to in-between frames in Section 3. Key frames are typically sampled at ten-frame intervals, but the sampling rate may vary according to object motion. For slower moving or deforming objects, a lower sampling rate

This work was partially supported by the National Science Foundation of China under Grant U0935001, U1301257 and 61170195, and by the Basic Research Plan in Shenzhen City under Grant JC201105201110A.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

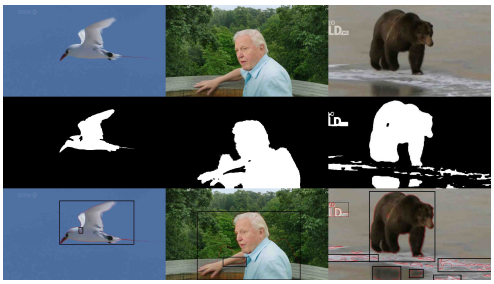

Fig. 1. Salient object detection results. First row: video frames; second row: generated saliency maps: third row: bounding boxes of the salient regions.

may be used.

2.1. Salient Region Detection

In this subsection, we show how to extract the initial rough foreground region with a salient region detection algorithm [15].

Some key frames are typically sampled every T frames (a typical value of 10 is used in our system). The sampling rate can also vary according to the object motion. Then we use the center surround method [16] to detect the salient region in key frames. Specifically, the saliency value of a pixel x in image I is given by the weighted distance of the histograms in the central and surrounding regions containing x:

$$f(x, I) \propto \sum_{\{y|x\in R^\star(y)\}} \omega_{xy}\chi^2(R^\star(y), R_s^\star(y)), \qquad (1)$$

where R_s is the surrounding region with the same area of neighbor region R, $R^\star(y)$ is the most distinct rectangle centered at pixel y:

$$R^\star(y) = \arg\max_{R(y)} \chi^2(R(y), R_s(y)), \qquad (2)$$

where $\omega_{xy} = \exp(-\frac{||x-y||_2^2}{2\sigma_y^2})$ is the Gaussian weight function parameterized by variance σ_y^2, and $\chi^2(x, y) = \frac{1}{2}\sum_i \frac{(x_i-y_i)^2}{x_i+y_i}$ is the Chi-square distance.

Afterwards, a morphological opening operation (the dilation of the erosion) is applied to the saliency map to remove noise, especially small objects from the foreground. Then the salient region is regarded as the initial foreground object. As shown in Figure 1, a bounding box of the salient region is computed to indicate the foreground region. If there are multiple rectangles, we just select the largest one.

2.2. Optimization-based Segmentation

After obtaining the rough foreground region in Section 2.1, we employ an optimization-based framework to segment the foreground object more accurately, where temporal coherence between frames is also taken into consideration.

Before introducing the segmentation details, we formulate the problem as follows. Consider a graph $F = V \cup E$, where V is a set of nodes (*i.e.*, pixels) in the frames, E contains two kinds of edges: E_I connecting nodes within a frame, and E_B connecting nodes between adjacent key frames [1]. Mathematically, video object extraction is viewed as a soft labeling problem. We use an energy minimization formulation similar to GrabCut [17] to solve this problem.

First, given the initial foreground region, two types of Gaussian Mixture Models (GMMs) are constructed, one for the background and the other for the foreground. Then we label each pixel x with $\alpha \in [0, 1]$ by minimizing the following energy function:

$$\sum_{x\in E_I} D_I(\alpha, k, z; \theta) + \sum_{x\in E_B} D_B(\alpha, k, z; \theta) + \lambda D_\varphi, \qquad (3)$$

where D_I and D_B represent the cost of edges in frames and between frames, respectively.

$$D_{\{I,B\}} = -\log p(z_x|\alpha_x, k_x; \theta) - \log w(\alpha_x, k_x), \qquad (4)$$

where $p(\cdot)$ is a Gaussian distribution parameterized by θ, $w(\cdot)$ is the mixture weight, and x is the pixel index, z_x is the pixel color, k_x is the most likely GMM component for each pixel, and θ is learned from image data z. The smooth term D_φ with weight λ ensures the temporal coherence across frames:

$$D_\varphi = \sum_{x\in\Omega} \varphi(|\nabla C|)|\nabla\alpha|^2, \qquad (5)$$

where ∇C represents the gradient of the observation image, Ω is the unknown region, function $\varphi(\cdot) \propto \frac{1}{\max(\cdot,\varepsilon)}$ is used to modulate the smoothness term, and ε is a small positive number preventing $\varphi(\cdot)$ from becoming infinite.

3. PROPAGATION AND REFINEMENT

In this section, we show how to propagate the accurate extraction results of key frames in Section 2 to in-between frames, and refine the unmatched regions to obtain the final results.

3.1. Error Map-based Propagation

Inspired by previous frame-to-frame propagation strategy in [4], we design our propagation algorithm as follows. We first detect both the Harris corner points and SURF [18] points in previous key frame I_i, and track them in non-key frame I_t by using a standard KLT tracker [19]. To remove the global motion, a homograph matrix between I_i and I_t is estimated with the RANSAC method [20]. Then, the forward error map E_t^f is computed as the color difference between the aligned previous key frame and frame I_t. The backward error map E_t^b

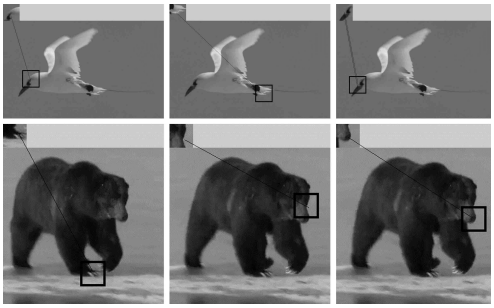

Fig. 2. The propagation process. The marked regions in each image is matched precisely with those in the key frames.

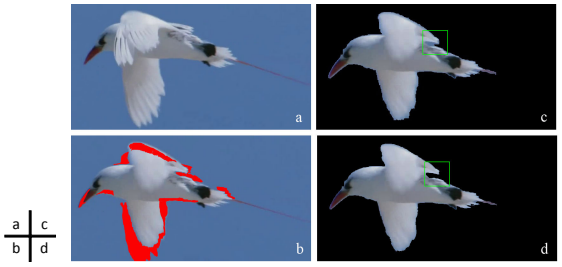

Fig. 3. The refinement process. (a) The key frame. (b) An intermediate frame with the error map marked in red. (c) the coarse result after propagation. (d) the refined result.

between frame I_t and the following key frame I_{i+1} is computed in a similar way. We combine E_t^f and E_t^b together to obtain the final error map E_t as in [4]. Connected pixels in E_t forms a connect region set C_t^r. Pixels whose connected region area is greater than a threshold e ($e = 30$ in our system) are regarded as topology change region, which is refined in Section 3.2.

In terms of pixels with a lower region area in C_t^r, which usually means the topology changes are small, we compute their mattes through m nearest $N * N$ windows ($N = 9$ and $m = 10$ in our system) located near the corresponding pixel position in I_i. The foreground color probability $p_f(x)$ and background color probability $p_b(x)$ for pixel x are computed as:

$$p_f(x) = \exp(-\frac{\Sigma_{i=1}^{m}||l_{t+1}(x) - l_t(f_i)||^2}{m \cdot \delta}), \quad (6)$$

$$p_b(x) = \exp(-\frac{\Sigma_{i=1}^{m}||l_{t+1}(x) - l_t(b_i)||^2}{m \cdot \delta}). \quad (7)$$

where $l(x)$ is the color values of x in RGB space, and m is the number of pixels whose color value are closet to $l(x)$.

The normalized foreground probability $p(x)$ is defined as:

$$p(x) = \frac{p_f(x)}{p_f(x) + p_b(x)} \quad (8)$$

With the computed foreground probability $p(x)$, pixel x is label as foreground if $p(x) > 0.5$; otherwise x is background. One coarse segmentation result is shown in Figure 3.

In addition, we find that pixels who move a lot with respect to the global motion may fit a local motion model. Therefore, it is not necessary to use the complex local refinement for segmenting these pixels. We can estimate a local homography matrix for a local region. Then, Equation (6), (7), and (8) are used to estimate the segmentation result for a region whose local movement is small.

3.2. Local Refinement

The propagation approach described in Section 3.1 does not process the regions where topology changes a lot, *i.e.*, whose color difference in the error map is big. From our observation, three situations may result in topology changes: occlusions, dis-occlusions, and motion blur caused by fast moving. Occlusions happen when features which appear in previous frame are not presented in current frame, and the other way around in dis-occlusions. Occlusions and dis-occlusions do not cause color contrast changes, while motion blur does. If these three situations are not detected, we use the Bayesian matting [7] to refine these local regions.

In terms of occlusions, we refine them via following steps:

1. Decide whether this region contains foreground boundary or not. If it is inside the foreground, go to step 2; otherwise, go to step 3.
2. Decide whether the occlusion causes background exposure by comparing with the background color model in Section 3.1. If it does, assign the exposed pixels to background.
3. Consider pixels in a narrow band around the current foreground boundary, and pixels in the original $m * m$ window. Use previous GMMs to compute the probabilities that pixels belongs to background or foreground. Since the boundary is clear and solid, there is no need to use the complicated Bayesian matting method.

Dis-occlusions are processed in a similar way. The only difference lies in step 2, which is to decide whether the dis-occlusions cause foreground exposure by comparing with the foreground color model, and label the exposed pixels as foreground.

Local refinement in the blurred region is difficult to deal with. In such regions, GMMs do not work due to the similar color distribution between foreground and background. However, we can perform segmentation in these blurred regions by finding the corresponding regions in key frames. Some regions in key frame I_i corresponding to the blurred region in frame I_t can be roughly found according to the KLT tracking result. Then, we use a Gaussian kernel to blur these regions. By comparing with the blurred region in frame I_t, we regard the region in key frame I_i with a lowest color difference as a reference for segmentation. An example of local refinement

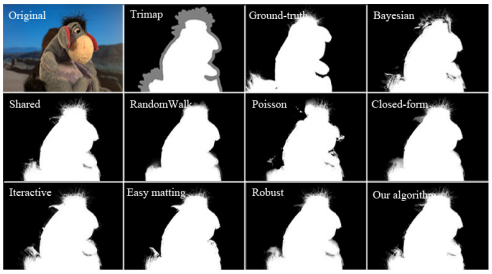

Fig. 5. One comparison example for different methods.

is shown in Figure 3.

4. EXPERIMENTS

We have conducted extensive experiments on a PC with 3.3GHz CPU and 4GB memory. All the input video sequences are downloaded from the Docume channel on Youku website. As shown in Figure 4, we can see that our system is able to achieve accurate extraction results. Table 1 shows the computation cost at each step of the four videos. The total processing time per frame is about $0.2 \sim 0.4$ second. If only Bayesian matting is applied to the whole video, it will take $50 \sim 100$ seconds to obtain a matte for each frame. The average compute times compared with different matting methods are shown in Table 2. Therefore, our system is of efficiency. It is possible to achieve a real-time performance via GPU programming. In addition, the whole system runs automatically and does not require any user input.

	#1	#2	#3	#4
Width	1280	640	512	1280
Height	720	352	288	720
Frames	98	155	160	82
Key frames	10	15	15	8
Saliency map(ms)	61.1	35.4	28.1	58.3
Segmentation(ms)	153.1	63.2	57.1	98.7
Refinement(ms)	97.2	36.4	28.6	74.4
Propagation(ms)	71.4	24.8	21.8	61.6

Table 1. Computation cost at each step for four videos.

We also provide a quantitative evaluation on matting accuracy. Eight algorithms on 24 test images are compared with our approach on the same trimaps. We compute the Mean Squared Error (MSE) between the estimated mattes and the groundtruth in each image. The evaluation result contains two indicators: accuracy and robustness. The accuracy represents the best case while the robustness represents the worst case, which are reflected by the minimum and maximum MSE across all images, respectively. The detailed evaluation results and the average compute times of different matting methods are shown in Table 2. Figure 5 shows the extracted mattes obtained by different approaches for the donkey image. In summary, our algorithm is demonstrated to achieve the comparable accuracy efficiently.

Matting methods	Accuracy (Min MSE)	Robustness (Max MSE)	Compute times(s)
Bayesian [7]	2.31	3.14	82.1
Shared [21]	0.88	1.30	0.065
Random walk [9]	2.40	2.91	4
Poisson [8]	5.94	6.50	25.7
Closed-form [10]	0.93	1.36	12.3
Iteractive BP [22]	1.48	2.23	320
Easy [11]	3.09	4.46	167.3
Robust [12]	0.89	1.58	2.09
Ours	1.04	1.34	0.24

Table 2. The quantitative evaluation on matting results.

5. CONCLUSIONS

This paper presents an automatic and nearly real-time video foreground extraction system with no user interaction requirement. To achieve this goal, we first incorporate the saliency detection and optimization-based segmentation to obtain the foreground object automatically. Then, Bayesian matting is used to refine the details after propagating the segmentation results in key frames to other frames. Experiments have shown the efficiency, accuracy, and robustness of our system. However, for now the proposed system has difficulty in handling videos with complex background. In the future, we will extend our framework to videos under illumination changes and complex background.

6. REFERENCES

[1] Yin Li, Jian Sun, and Heung-Yeung Shum, "Video object cut and paste," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 595–600, 2005.

[2] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro, "Video snapcut: robust video object cutout using localized classifiers," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 70:1–70:11, 2009.

[3] Jue Wang, Pravin Bhat, R. Alex Colburn, Maneesh Agrawala, and Michael F. Cohen, "Interactive video cutout," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 585–594, 2005.

[4] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski, "Video matting of complex scenes," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 243–248, 2002.

[5] Antonio Criminisi, Geoffrey Cross, Andrew Blake, and Vladimir Kolmogorov, "Bilayer segmentation of live

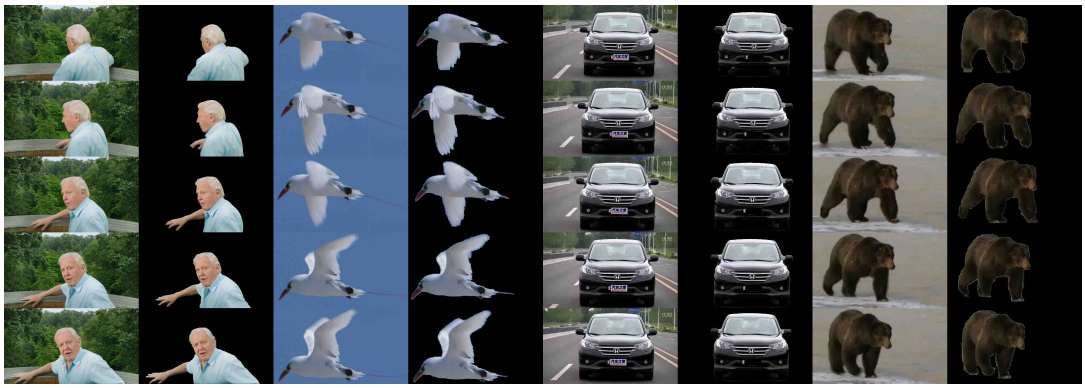

Fig. 4. Four extraction results. The original images and extraction results are shown in the left and right, respectively.

video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 53–60.

[6] Jue Wang and Michael F. Cohen, "Image and video matting: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, 2007.

[7] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski, "A bayesian approach to digital matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, vol. 2, pp. 264–271.

[8] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum, "Poisson matting," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 315–321, 2004.

[9] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann, "Random walks for interactive alpha-matting," in *IASTED International Conference on Visualization, Imaging and Image Processing*, 2005, pp. 423–429.

[10] Anat Levin, Dani Lischinski, and Yair Weiss, "A closed form solution to natural image matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 61–68.

[11] Yu Guan, Wei Chen, Xiao Liang, Ziang Ding, and Qunsheng Peng, "Easy matting - a stroke based approach for continuous image matting," *Computer Graphics Forum*, vol. 25, no. 3, pp. 567–576, 2006.

[12] Jue Wang and Michael F. Cohen, "Optimized color sampling for robust matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[13] Jinlong Ju, Jue Wang, Yebin Liu, Haoqian Wang, and Qionghai Dai, "A progressive tri-level segmentation approach for topology-change-aware video matting," *Computer Graphics Forum*, vol. 32, no. 7, pp. 245–253, 2013.

[14] Xue Bai, Jue Wang, and Guillermo Sapiro, "Dynamic color flow: a motion-adaptive color model for object segmentation in video," in *European Conference on Computer Vision*, 2010, pp. 617–630.

[15] Hongliang Li and King N Ngan, "Automatic video segmentation and tracking for content-based applications," *Communications Magazine, IEEE*, vol. 45, no. 1, pp. 27–33, 2007.

[16] Laurent Itti, Christof Koch, and Ernst Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[17] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[18] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.

[19] Jianbo Shi and Carlo Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[20] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] Eduardo S. L. Gastal and Manuel M. Oliveira, "Shared sampling for real-time alpha matting," *Computer Graphics Forum*, vol. 29, no. 2, pp. 575–584, 2010.

[22] Jue Wang and Michael F. Cohen, "An iterative optimization approach for unified image segmentation and matting," in *IEEE International Conference on Computer Vision*, 2005, vol. 2, pp. 936–943.