Background Removal Tool for Object Movies

Jin-Ren Su[1], Shang-Ru Tsai [1], Yu-Pao Tsai[1,2], and Yi-Ping Hung[1,2*]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.

*Email: hung@csie.ntu.edu.tw, Tel: +886-2-23625336-433

Abstract

The background removal of object movies is an important issue while it is integrated into a real or a virtual scene. In this paper, we introduce a semi-automatic tool for removing background of object movies. Our tool automatically removes the background based on the characteristics of object movie, and allows the user intervention to give a guide for improving the segmentation result of automatic process. Once the user intervention occurs in some image frames, the automatic process propagates the correct information to all image frames, and updates the intermediate segmentation result. Even through the user interaction is allowed, there have some pixels can not be determined to a part of object movie or the background due to those pixels may be composites of the foreground and the background. For those pixels, we estimate the alpha value of them to obtain a better result. The experimental results show that this tool is simple to use and the user is able to obtain more accurate results with limited user interaction.

Keywords:

Object movie, Background removal, Video object segmentation, MAP, Alpha estimation , User intervention.

1. Introduction

To construct a realistic environment is an import research topic in computer graphics. Image-based object/scene is more photorealistic than geometry-based, and the rendering time is independent of the complexity of the object/scene. Object movie (OM) is composed of a set of 2D images taken around a 3D object. It is an image-based representation of a 3D object, and has widely used to enhance perception of subtle information for better understanding the shape, texture and other characteristics of the 3D object[4, 7, 3]. When the object movies are integrated into a real or a virtual scene, the background removal of object movies is necessary.

Since video object coding is one of the most important functionalities proposed by MPEG4 standard, there are many researchers have proposed the segmentation methods for video objects. Among them, some are unsupervised[5,14,15,17]. However, unsupervised methods may not be able to always extract the exact video object as desired[2,6,10,11,18]. To solve this problem, some researchers propose semi-automatic methods by allowing user interaction to guide the segmentation. However, few of lectures are reported to segment the object movies.

An OM is a special case of videos, and it has some characteristics: the colour of background is relatively static and uniform compared with the foreground object. In this paper, we treat the segmentation problem as a labeling problem, and utilize these characteristics of OMs to make an interactive tool.

1.1 Overview of Our Tool

Our goal is to assign every pixel a label, which can be "Foreground", "Background", and "Uncertain" (abbreviated as "F", "B", and "U") for a given OM. Some notations about OM that we will refer in this paper are as the following:

$$\begin{aligned} OM &= \left\{ f_{tp} \mid 0^\circ \le t \le 90^\circ, 0^\circ \le p \le 360^\circ \right\} \\ &= \left\{ S_t \mid 0^\circ \le t \le 90^\circ \right\} \end{aligned}$$

$$S_T = \left\{ f_{tp} \mid 0^\circ \le p \le 360^\circ, t = T \right\}$$

f_{tp} : the image frame with tilt angle t and pan angle p

where S_T is defined as an equi-tilt set which is a sub-sequence of the OM with the same tilt angle and continuous pan angles.

As the flowchart of our tool shown in Figure 1, the proposed segmentation method for OMs consists of three main stages: the initial labeling, the label updating (include the MAP labeling and the temporal labeling), and the alpha estimation. In the initial labeling, we extract the definite foreground and background pixels. In the label updating, the pixels assigned "U" label after the initial labeling stage will be updated by spatial and temporal information based on the extracted foreground and background.

After the label updating, the intermediate segmentation result may have some misclassified pixels. In our system, user intervention is allowed to modify the misclassified pixels through provided user interactive interface. After modification, the system will reenter the label updating stage to obtain a more accurate result.

After user intervention, the most pixels are classified to the foreground or the background except the pixels that may be composite of the foreground and the background. Then, the system will enter the alpha estimation stage. In the alpha estimation stage, we adopt the method proposed by Chuang et al.[1] to assign an alpha value to pixels labeled as "U". After the alpha estimation, we can remove background and get a better segmentation result with alpha information.

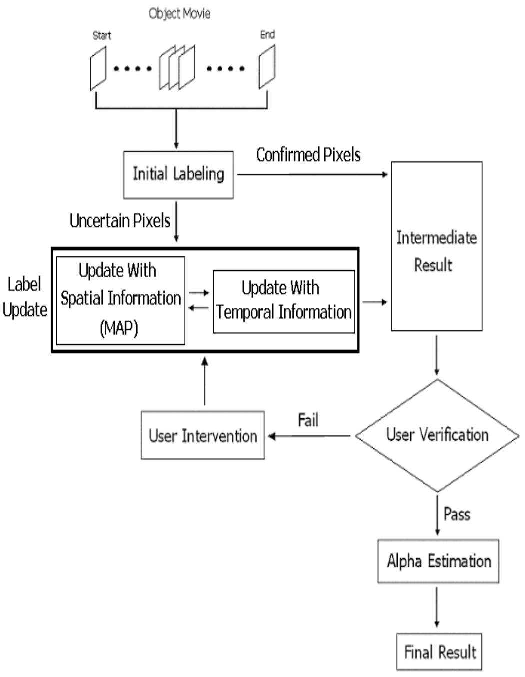


Figure 1. The flowchart of the proposed tool.

2. Initial Labeling

The aim of this stage is to decide the labels of pixels that can be classified as "F" or "B" very definitely, while keep the other uncertain pixels to be decided at following stages. Since the background of an equi-tilt set is relatively static and uniform-colored as compared with foreground, we focused on extracting background pixels at first and then found the definite foreground pixels from the other pixels based on the color different from the means of the extracted background pixels.

In this stage, we perform a pixel-based labeling for every pixel of an equi-tilt set (S_T) mainly by using their color information. From our observation, if the color variation of the pixels with the same image position in an equi-tilt set is relatively large, it should not be a background pixel. To suppress the effect due to image noise, we measure the color variation for all pixels in frame $f_{T,p}$ by calculating the block different from the neighboring frames $f_{T,p-1}$ and $f_{T,p+1}$. To deal with lighting change, we first subtract the color mean of block for every pixel within the block before calculating the block different. If the measures of the pixels the same coordinates in S_T are very small, those pixels will be labeled as "B". Otherwise, the pixel will be labeled to "U".

After that, we collect the color information (here we use LUV color space) of those pixels labeled as "B" and model the color information as a Gaussian distribution. Then we classify the other pixels labeled as "U" into either "F" or "U" with a strict threshold to make sure that only pixels very unlike the background will have chance to be labeled as "F".

3. Label Updating

In this stage, the aim is to classify the labels of pixels labeled as "U" to foreground or background as far as possible. That is, we will further label these uncertain pixels with methods making use of both spatial (intra-image) and temporal (inter-image) information.

Since all pixels of every working frame are classified into three clusters: "F", "B", and "U" before, we only deal with pixels classified as "U" in label updating stage. That is, only pixels labeled as "U" before would be relabeled as "F", "B", or still "U" in the label updating stage, but the other pixels will not be allowed to change their labels. The label updating stage consists of two main processes: the spatial updating and the temporal updating.

The spatial updating process followed by the temporal process will be iterated until it converges. That is, the label updating stage will repeat itself until there is no label been updated.

3.1 Spatial Updating

In this process, we reduce the pixel classification problem of a single image into a spatial graph labeling problem. Our method is to construct a graph for every frame of an OM and apply MAP (*Maximum a Posteriori*) method to achieve a optimal labeling set of the graph. To reduce complexity and save time, a vertex of a graph on which we perform MAP labeling could represent either a watershed region of which all the pixels have the same label or a pixel. There is an edge between two nodes only if they are spatially neighbored in the image. Since the graph of every working frame is built, we apply MAP method on each of them. In this paper, we model the observation term by taking advantage of color information of neighboring vertices and model the prior term as a MRF (Markov Random Field) [8, 9, 13] in which we apply our prior knowledge of 2-clique relations in the spatial graph. Because global optimization for MAP is not possible, we adopt ICM (Iterative Conditional Modes) algorithm to achieve local optimization in our implementation.

3.1.1 Construction of 2D Spatial Graphs

To reduce the pixel classification problem of an image into a graph labeling problem, we first build a 2D graph for every working frame. For a graph completely built from pixels of an image, the time complexity and used memory will be a fatal problem at MAP labeling process. For a graph completely built from watershed regions of an image, the classified results may have many misclassified regions since a watershed region may contain background pixels and foreground pixels at the same time. To come to a compromise, we build a graph in which a vertex could represent either a watershed region or a pixel.

So we first apply watershed segmentation proposed by Vincent and Soille [16] on every frame of an OM, and if all pixels of a watershed region have the same label, the watershed region itself will represent a vertex in the graph of the working frame, otherwise the watershed region will be split into pixels each of which represent a vertex in the graph of the working frame. Here we adopt topographic simplification followed by watershed segmentation to let the watershed regions as small as possible. And there is an edge between two vertices only if they are spatially neighbored in the image. An example of graph building is shown in Figure 2.

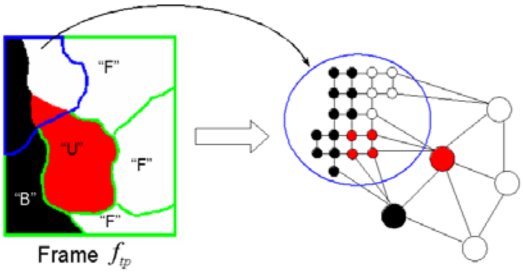


Figure 2. An example of the construction of a 2D spatial graph.

3.1.2 MAP Labeling

After the graph of each working frame is built, we apply MAP method on the graph to get an optimal labeling set of the graph. According to the Bayes rule, the MAP can be computed by using the following formulation:

$$\max_L P(L \mid M) \propto \max_L P(M \mid L)P(L) \tag{1}$$

L : the labeling set of the graph
M : the measurement

As shown above, to maximize the posteriori probability is to maximize the observation term $P(M|L)$ and the prior term $P(L)$. We will discuss how to model the observation term and the prior term at section 3.1.3 and 3.1.4 respectively.

3.1.3 The Observation Term

On modeling the observation term, we take advantage of the color information of neighboring vertices in the graph. Here, we adopt LUV color space instead of common RGB color space, because LUV is a uniform color space.

For a vertex currently labeled as “F”, we find a fixed number of its neighboring vertices that are labeled as “F” and perform color quantization[12] on the set of vertices. After color quantization, the set of vertices are separated into several clusters. Then we compute the Euclidean distances between the LUV color vector of the focused vertex and the mean LUV color vectors that belong to different clusters respectively. If the smallest Euclidean distance is small enough, the focused vertex tends to be labeled as “F”. For a vertex currently labeled as “B”, the process is very similar except that we don’t perform color quantization for the color of the background because it is assumed to be a single and uniform. For a vertex currently labeled as “U”, we just assign a constant as the observation term. Notice that if the number of a vertex’s neighboring vertices is less than the fixed number, we will expand the search range to its neighboring vertices’ neighboring vertices until we collect enough samples.

In fact, we model the observation probability as a Gaussian distribution of which the covariance matrix is an identity matrix. Notice that a graph vertex can represent either a single pixel or a watershed region, so we should weight the collected LUV color vectors by its area and distance. That is, when evaluating a vertex’s observation term, we will weight the collected neighboring vertices by their number of pixels and by the distance between the vertex and the collected neighboring vertices. Then the evaluation equation of the observation term can be shown as following:

$$P\left(I_i \mid \{l_i = k\} \cup \overline{L_i}\right) = \begin{cases} \dfrac{1}{K_F} \exp\left[- \left\| I_i - \mu_i^c (\text{"}F_c\text{"}) \right\|^2 \right] & ; k = \text{"}F_c\text{"}, c = 1,2,....,C \\ \dfrac{1}{K_B} \exp\left[- \left\| I_i - \mu_i (\text{"}B\text{"}) \right\|^2 \right] & ; k = \text{"}B\text{"} \\ \dfrac{1}{K_u} & ; k = \text{"}U\text{"} \end{cases}$$

where

$$\mu_i(\text{"}k\text{"}) = \frac{\sum_{j \in N_i, l_j = \text{"}k\text{"}} \omega_{ij} I_j}{\sum_{j \in N_i, l_j = \text{"}k\text{"}} \omega_{ij}}$$

$$\omega_{ij} = Area_j * e^{-\text{distance}_{ij}}$$

$$\overline{L_i} = L - \{l_i\},$$

where C is the number of clusters of foreground after color quantization, $\frac{1}{K_F}, \frac{1}{K_B}, \frac{1}{K_U}$ are normalizing constants, I_i is the LUV vector of vertex *i* and K_U is a constant.

3.1.4 The Prior Term

In this paper we model the prior term as a MRF (Markov Random Field)[8,13,錯誤! 找不到參照來源。] and apply our prior knowledge of 2-clique relations in the spatial graph. Markov random field theory is a branch of probability theory for analyzing the spatial or contextual dependencies of physical phenomena. It is used in visual labeling to establish probabilistic distributions of interacting labels. The Hammersley-Clifford theorem states that a Markov random field *F* on a set *S* with respect to the neighboring system *N* is also a Gibbs random field on *S* with respect to *N*. And a configuration *f* (here means a labeling set of the graph vertices) of a Gibbs random field obeys a Gibbs distribution. A Gibbs distribution takes the following form:

$$P(f) = \frac{e^{-\frac{1}{T}U(f)}}{Z}$$

$$Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)}$$

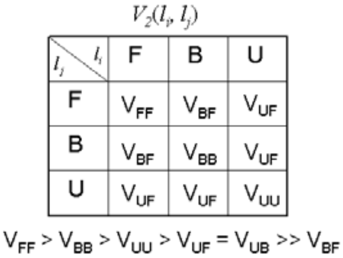
Where *Z* is a normalizing factor which is a constant for all the configurations, so there is no need to compute the value of *Z*; *T* is the temperature, which is generally assumed to be 1; *U*(*f*) is the energy function which is the sum of clique potentials $V_c(f)$ over all possible cliques *C*.

In our case, we only use cliques of size two to define the energy function. Note that a vertex in the graph can represent either a single pixel or a watershed region. So for one vertex, we weight its 2-clique energies by the area of its neighboring vertices when evaluating its prior term. Therefore, our energy function is defined as:

$$U(L) = \sum_{i \in V} \left[\frac{\sum_{j \in N_i} \omega_j V_2(l_i, l_j)}{\sum_{j \in N_i} \omega_j} \right]$$

$$\omega_j = \text{Area}_j$$

To apply our prior knowledge of 2-clique relations, we construct a relation table as shown in Figure 4-3.



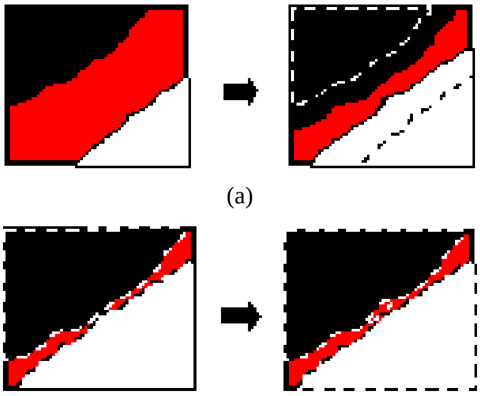
$V_2(l_i, l_j)$

l_j \ l_i	F	B	U
F	V_{FF}	V_{BF}	V_{UF}
B	V_{BF}	V_{BB}	V_{UF}
U	V_{UF}	V_{UF}	V_{UU}

$V_{FF} > V_{BB} > V_{UU} > V_{UF} = V_{UB} >> V_{BF}$

Figure 3. The 2-clique relation table for the evaluation of $V_2(l_i, l_j)$.

We prefer the “F-F” relation more than other relations because most of the background pixels have been extracted after previous processes. By setting the “F-F” cliques a higher energy than “B-B” cliques, the “F-F” cliques will have a higher possibility of occurrence. That is, the foreground will expand more rapidly toward the border than the background, which is shown in Figure 3-4(a). And the “U-U” cliques have a higher energy than “U-F” and “U-B” cliques because pixels near the object border should be left undecided until alpha estimation stage. The “B-F” relation is our most unwanted relation because the border between foreground and background, i.e. the object contour, should also be kept undecided until alpha estimation stage, which is shown in Figure 4(a) and 4(b).


(a)

(b)

Figure 4. (a) By setting “F-F” cliques the highest energy, the foreground will expand more rapidly than the background. (b) By setting “B-F” cliques the lowest energy, the object border will remain uncertain during the “Label Updating” stage.

3.2 Temporal Updating

After the “Label Updating Using Spatial Information” stage, there are still some uncertain pixels that cannot be decided to be either foreground or background. The goal of this process is to assign a reliable label to these uncertain pixels based on the temporal information (inter-image information). We first apply motion estimation to every uncertain pixel by block matching on its neighboring frames within certain range. Afterward we filter out those unreliable motion vectors since the motion estimation for certain pixels may be erroneous. Then we can find the linked pixel-list of every uncertain pixel, which we call a “worm”. That is, a “worm” of an uncertain pixel *i* consists of pixels that can be reached from *i* through estimated motion vectors. As shown in Figure 4-5, the worm of i_1 is a linked pixel-list emitted from i_1, so as the worm of i_2. The two ends of a worm will terminate at the pixels that don’t have reliable motion vectors. For example, the length of the worm of i_2 shown in Figure 5 is two.

After the worm of an uncertain pixel *i* is constructed, we will assign a label to *i* based on the label information of its worm. There are only three conditions: first, if all the labels of the worm are “Uncertain”, the label of the pixel *i* is remained “U”. Second, if the labels of the worm are either “U” or “F”(“B”) and at least one label is “F”(“B”), then we will assign the label “F”(“B”) to the focused pixel *i*. Third, if the labels of the worm have both the label “F” and “B”, that is, a contradiction, then the label of the pixel *i* is remained “U”. Here we adopt sequential updating, that is, this process will be executed frame by frame. Global updating of the process is left as future work.

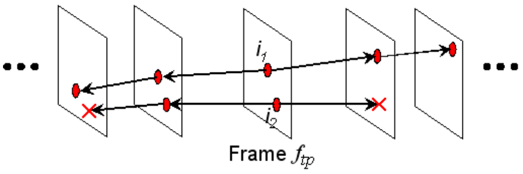


Figure 5. The worms of the uncertain pixels i_1 and i_2

4. Experiments

In this section, we show some experimental results to demonstrate our tool. The first experimental result shown in Figure 6 is generated by our tool without any user interaction. Due to the limitation of paper length, some of frames are shown in this paper. The first row shows the original images, and the second row shows the *trinary labeled images*, which red, white, and black pixels indicate label “U”, “F”, and “B”, after the initial labeling stage. The third row show the trinary labeled images after "Label Updating", and forth row shows the composites of the final segmented images and blue backgrounds.

Figure 7 illustrates the user intervention is involved based on the automatic segmentation result shown in Figure 6. Figure 7(a) and (b) show the selected trinary images and segmented images from image 6. We label some uncertain pixels to be "F" as shown in Figure 7(c). The two circles mark the areas have be modified. The modified pixels propagate the corrected information to other frames such that the segmented result shown in Figure 7(d) looks better.

Two another experimental results are showed in Figure 8 and Figure 9 for the "Vessel" sequence and the "Flower" sequence. For the "Vessel" sequence, the user intervention is not required due to it has uniform background in comparison with the "Winnie" sequence, and the segmented result is acceptable. In contrast with the "Vessel" sequence, the color of the pot in the "Flower" sequence is similar to the background, such that it is not easy to be segmented without any user intervention, as shown in Figure 9.

5. Conclusion and Future Work

In this paper, we introduce an interactive tool for removing background of OMs. The segmentation problem is treated as a labeling problem. The initial labeling stage and labeling updating automatically remove the background of OMs, and allows the user intervention to achieve the more accurate segmented result. After modification, the system will reenter the label updating stage to obtain a more accurate result, and the correct information can be propagated to whole OM. Afterward the alpha estimation is applied to obtain a better segmentation result.

The experimental results show that this tool is simple to use and the user is able to obtain more accurate results with limited user interaction. Our tool is very useful when one want to integrate object movies into a new background to construct a photorealistic environment.

There are several directions that may be able to improve this work. If the background is captured in preprocessing, it can be used in our initial labeling stage and labeling updating to significantly improve the segmentation result. In order to speedup the performance, we plan to introduce a hierarchical scheme into our tool.

References

1. Y.-Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, "A Bayesian Approach to Digital Matting",
   CVPR 2001, December 2001, Kauai, Hawaii, Vol.II, pp.264-271.
2. C. Gu and Lee, M. C., "Semi-automatic segmentation and tracking of semantic video objects," *IEEE Transaction on Circuits and Systems for Video Technology*, 8(5), pp. 572-584, 1998.
3. http://www.apple.com/quicktime/qtvr/
4. Y.-P. Hung, C.-S. Chen, Y.-P. Tsai, S.-W. Lin, "Augmenting Panoramas with Object Movies by Generating Novel Views with Disparity-Based View Morphing," Journal of Visualization and Computer Animation, Special Issue on Hallucinating the Real World from Real Images, Vol. 13, pp. 237-247, 2002.
5. Y.-P. Hung, Y.-P. Tsai, C.-C. Lai, "A Bayesian Approach to Video Object Segmentation via Merging 3D Watershed Volumes," *Proceedings of International Conference on Pattern Recognition* (ICPR02), Quebec, Canada, Vol. 3, August 2002.
6. S. Jehan-Besson, M. Barlaud, and G. Aubert, "Video Object Segmentation Using Eulerian

Region-Based Active Contours," *Proceedings of International Conference on Computer Vision 2001 (ICCV '01),* pp. 353-361, 2001.

7. A. Katayama, Y. Sakagawa, H. Yamamoto, and H. Tamura, "Shading and Shadow-Casting in Image-Based Rendering without Geometric Models," *Conference Abstracts and Applications* (SIGGRAPH '99), 1999, pp. 275.

8. S. Z. Li, "Modeling Image Analysis Problems Using Markov Random Fields," in C. R. Rao and D. N. Shanbhag (ed), *Stochastic Processes: Modeling and Simulation*, *Handbook of Statistics*, Vol. 20, Elsevier Science, 2001.

9. S. Z. Li, Markov Random Field Modeling in Computer Vision, Springer-Verlag, Tokyo, 1995.

10. H. Luo and A. Eleftheriadis, "An interactive authoring system for video object segmentation and annotation," *Signal Processing: Image Communication*, Vol. 17,p559-572, 2001.

11. B. Marcotegui, P. Correia, F. Marques, R. Mech, R. Rosa, M. Wollborn, F. Zanoguera "A Video Object Generator Tool Allowing Friendly User Interaction" *Proceedings of IEEE International Conference of Image Processing, Kobe (Japan),* October 1999

12. M.T. Orchard, C.A. Bouman, "Color Quantization of Image, " IEEE Tran. on Signal Proc. 39(12), 1991

13. I. Patras, E. Hendriks, and I. Lagendijk, "Video Segmentation by MAP Labeling of Watershed Segments," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, no. 3, pp. 326-332, 2001.

14. I. Patras, E. Hendriks, and I. Lagendijk, "Video Segmentation by MAP Labeling of Watershed Segments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, no. 3, pp. 326-332, 2001.

15. J. Shi and J. Malik, "Motion Segmentation and Tracking Using Normalized Cuts," *Proceedings of IEEE International Conference on Computer Vision*, pp. 1154-1160, Bombay, India, 1998.

16. L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, no. 6, pp. 583-598, 1991.

17. D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, no. 5, pp. 539-546, 1998.

18. H. Zhong, L. Wenyin, S. Li, "Interactive Tracker – A Semiautomatic System for Video Object Segmentation," Proceedings of IEEE International Conference on Multimedia and Expo (ICME '01), pp. 645-648, 2001.

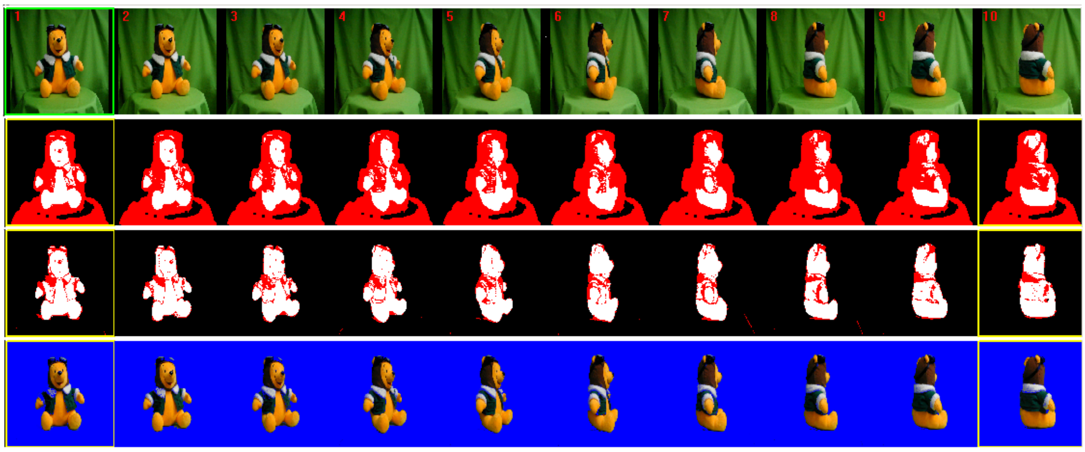
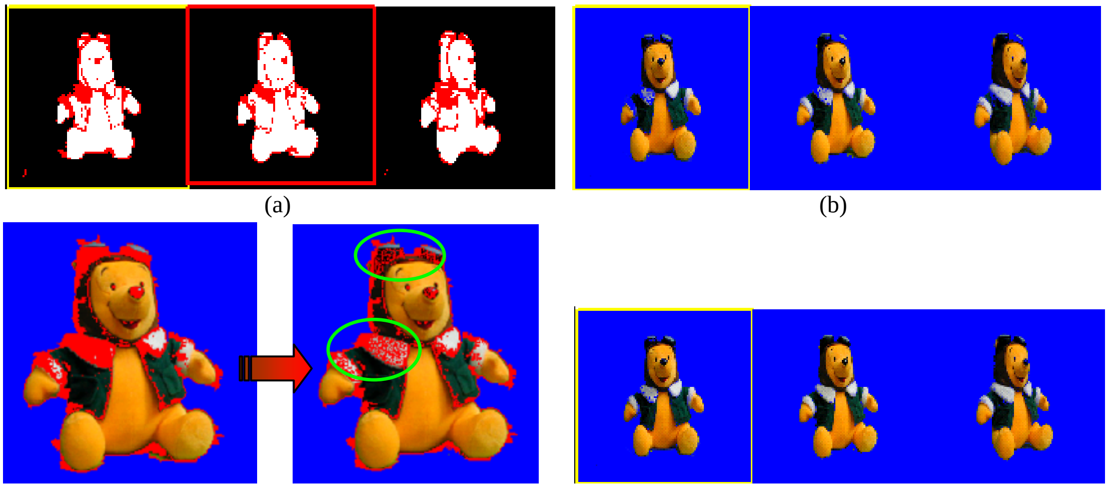


Figure 6. The segmentation result of the “Winnie” sequence.



(a)

(b)

(c)

(d)

Figure 7. Illustration of user intervention. (a) Trinary labeled images of selected three frames after the label updating stage. (b) The final result after estimating alpha value and compositing to a blue background without user interaction. (c) Apply user interaction on the meddle frame of the selected three frames. (The green circles indicate the modified area after the user interaction) (d) The final result with user interaction

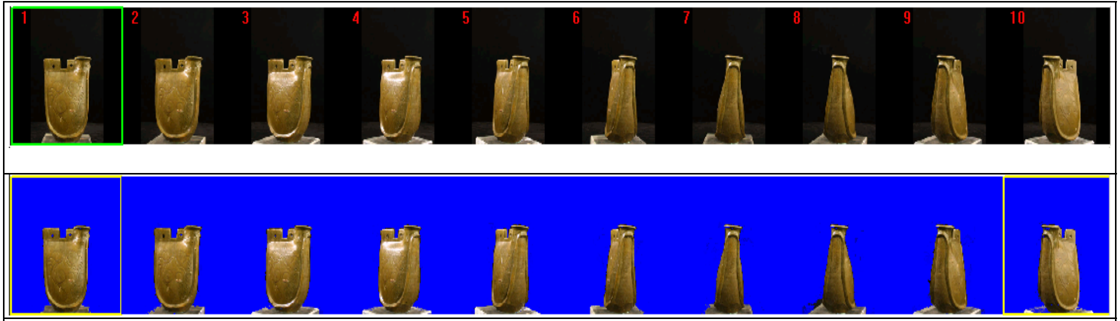


Figure 8. The segmentation result of the “Vessel” sequence.

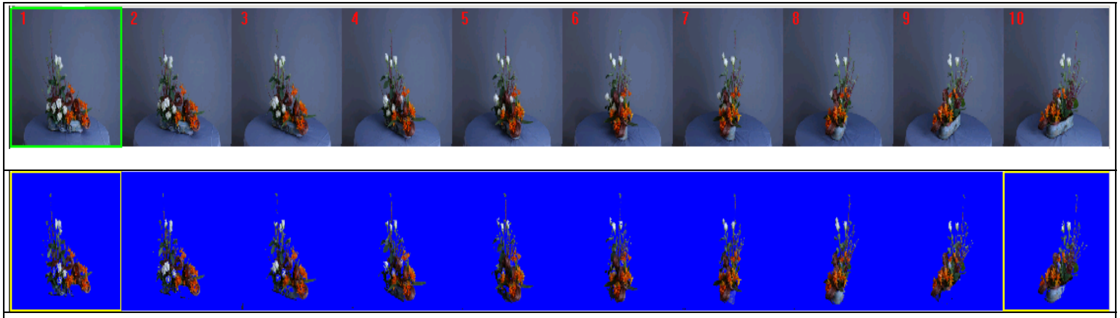


Figure 9. The segmentation results of the “Flower” sequence.