Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Non-interactive automatic video segmentation of moving targets*

Yu ZHOU†, An-wen SHEN, Jin-bang XU†‡

(*Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*)

†E-mail: sherry.haku@gmail.com; xujinbang@mail.hust.edu.cn

Received Mar. 15, 2012; Revision accepted Aug. 24, 2012; Crosschecked Sept. 11, 2012

Abstract: Extracting moving targets from video accurately is of great significance in the field of intelligent transport. To some extent, it is related to video segmentation or matting. In this paper, we propose a non-interactive automatic segmentation method for extracting moving targets. First, the motion knowledge in video is detected with orthogonal Gaussian-Hermite moments and the Otsu algorithm, and the knowledge is treated as foreground seeds. Second, the background seeds are generated with distance transformation based on foreground seeds. Third, the foreground and background seeds are treated as extra constraints, and then a mask is generated using graph cuts methods or closed-form solutions. Comparison showed that the closed-form solution based on soft segmentation has a better performance and that the extra constraint has a larger impact on the result than other parameters. Experiments demonstrated that the proposed method can effectively extract moving targets from video in real time.

Key words: Video segmentation, Auto-generated seeds, Cost function, Alpha matte
doi:10.1631/jzus.C1200071 **Document code:** A **CLC number:** TP751.1

1 Introduction

An intelligent transport system (ITS) is a real-time comprehensive information processing system with high accuracy and efficiency, in which a surveillance and information acquisition subsystem is an important part. Image and video processing is necessary for the subsystem and every function of the subsystem is based on the detection, identification, and tracking of moving targets. As a result, video processing algorithms in an ITS should be accurate, real-time, and non-interactive.

The basic task of a traffic surveillance and information acquisition subsystem is to detect the motion knowledge from traffic video. This has been realized through several existing algorithms. Sometimes the subsystem is also required to extract the complete moving targets, including their boundaries. In this case a motion knowledge detection method alone is not enough and techniques involving segmentation are required. Generally speaking, segmentation methods need some extra constraints to distinguish the foreground from the background. Since the motion knowledge can be treated as the extra constraints, extracting moving targets can be treated as a mask generation problem in segmentation methods. As a result, the problem of extracting moving targets can be solved by the methods used in digital matting.

In essence, matting is a special form of segmentation. Studies of matting started in the 1960s and were based on users' input or certain determining rules. The compositing equation of image matting is given as follows (Wang and Cohen, 2007a):

$$I=\alpha F+(1-\alpha)B, \tag{1}$$

in which α is a small positive parameter in [0, 1] and F, B, and I are the foreground image, background image, and observed image (composite colors of pixels), respectively. The method with $\alpha\in\{0, 1\}$ was named 'hard segmentation' (Boykov and Jolly, 2001), and

‡ Corresponding author

* Project (No. 61033003) supported by the National Natural Science Foundation of China

the method with $\alpha \in (0, 1)$ was named 'soft segmentation' (Chuang *et al*., 2001). In this paper, soft segmentation is referred to as matting.

Image matting can also be divided into blue screen matting and natural image matting. Blue screen matting (Smith and Blinn, 1996) is simple and easy to implement, but it is rarely applied due to its high background requirements; natural image matting includes the Bayesian method (Chuang, 2004) and Poisson method (Sun *et al*., 2004).

Most traditional matting or segmentation algorithms require manual image marking as extra constraints of the matting problem. The common marking methods are trimap and scribbles methods. Trimap requires users to outline the foreground boundary from the background accurately, while the scribbles method (Levin *et al*., 2004), based on sketchy line drawings, simplifies the interaction between users and computers. Based on graph cuts, lazy snapping (Li *et al*., 2004) uses different colored lines to indicate foreground and background. Levin *et al*. (2008) presented a closed-form solution for extracting alpha matte from a natural image based on the scribbles. Wang and Cohen (2005) analyzed the defects of trimap, proposed scribbles based on the belief propagation algorithm, and applied them in video matting. Wang and Cohen (2007b) proposed a robust matting method. GrabCut (Rother *et al*., 2004) is also based on graph cuts, but the input method is different as it involves drawing a rectangle around the foreground.

When the matting target is video rather than an image, there are many different ways of performing video matting. Chuang *et al*. (2002) described a video matting approach based upon the Bayesian matting method. The method uses optical flow techniques to flow trimap between user-drawn trimap keyframes, thereby reducing user involvement. Li *et al*. (2010) proposed an algorithm that extends robust matting to video processing and achieves less interaction.

However, the above-mentioned methods require manual contour drawing of the foreground and background. As a result, several non-interactive video matting methods have been proposed. Apostoloff and Fitzgibbon (2004) proposed a method to extract foreground from natural background without interaction, based on Bayesian framework. But it requires a set of images as an extra constraint, and is unsuitable for foreground objects with complex shapes.

McGuire *et al*. (2005) proposed a novel, fully automatic method for pulling a matte using multiple synchronized video streams that share a point of view but differ in their plane of focus. The limitation of this method is that it demands distinct field depths of the target video. Gong *et al*. (2010) implemented a modified background cut algorithm with automatic trimap generation based on Poisson equations. The limitation of this method is that the temporal coherence in the video is used only for background modeling and hence is not fully utilized. Jiang *et al*. (2010) employed stereo motion analysis and presented an unsupervised scheme for stereo video matting without user interaction. But the video must have a multilayer attribute. Lee *et al*. (2010) suggested a temporally coherent approach to video matting to reduce the flickering effect by considering several consecutive frames in obtaining the alpha matte. But the method depends greatly on the quality of the trimaps. Wang *et al*. (2012) proposed a video matting method with real-time automatic trimap generation. But this method has no optimization scheme to save the system from error accumulation and propagation, and thus cannot avoid 'drift' problems for real-time segmentation.

Each of the above methods has its own limitations. For instance, methods with auto-generated trimaps cannot extract multiple objects. Other methods can handle only video sequences with high color saturation, high field depth, multilayer, or simple motion patterns. When the input video sequence is traffic video with low color saturation and complex motion patterns, none of the video matting methods above is appropriate.

Using scribbles, which is more stable than trimap as extra constraints, this paper proposes a non-interactive video matting or hard segmentation algorithm with auto-generated sparse seeds (used as scribbles in manual matting or segmentation). Since the purpose of this research was to extract moving targets from the video, the foreground seeds represented the motion knowledge in the video. The innovation of this paper is that it proposes an automatic method for seed generation. Therefore, interaction is avoided and real-time processing is guaranteed. Moreover, the proposed method has wide applications in processing video sequences with complex motion patterns and low quality.

The detailed algorithm steps in this paper are as follows. First, multi-frame interfusion is performed on the input video sequence with orthogonal Gaussian-Hermite moments (OGHM) and the Otsu algorithm, to obtain the foreground seeds. Second, the distance image of the foreground seeds is calculated, resulting in the background seeds. Third, both seeds are used as extra constraints and the minimal cost function is calculated with graph cuts or a closed-form solution. Finally, the value with the minimal cost function is taken as the mask, and the moving targets are extracted from the input video sequence. Experiment results show that the proposed method is effective in terms of real-time computation and accuracy.

2 Automatic generation of foreground and background seeds

2.1 Temporally coherent moving target discovery

In this study, OGHM is used for seed generation. Within a series of consecutive frames of the input video sequence, the changing of one pixel on a temporal axis can be regarded as a numerical signal. Since OGHM is essentially a temporally coherent high-pass filter, the high frequency components are preserved and the low frequency components are filtered out. Thus, the dynamic pixels and the stationary pixels are distinct.

In a video sequence $\{I(x,y,t)\}_{t=0,1,2,\cdots}$, for each spatial pixel (x,y) in temporal axis t, the nth order OGHM is defined as follows (Wu and Shen, 2004):

$$M_n(t,I(x,y,t))=\int_{-\infty}^{+\infty} I(x,y,t+v)g(v,\sigma)H_n(v/\sigma)\mathrm{d}v, \tag{2}$$

where $g(v,\sigma)$ is a Gaussian function with standard deviation σ, and $H_n(v/\sigma)$ is the scaled Hermite polynomial function of order n. Within the domain of definition $(-\infty,+\infty)$, it can be written as

$$H_n(v)=(-1)^n\exp(v^2)\frac{\mathrm{d}^n}{\mathrm{d}v^n}\exp(-v^2). \tag{3}$$

According to the characteristics of orthogonal Hermite polynomial and Gaussian functions, OGHM has the following properties:

$$\begin{aligned} M_n(t,f(x,y,t)) &= \sum_{i=0}^{n} a_i \frac{\mathrm{d}^i g(t,\sigma)*I(x,y,t)}{\mathrm{d}t^i} \\ &= \left(\sum_{i=0}^{n} a_i \frac{\mathrm{d}^i g(t,\sigma)}{\mathrm{d}t^i}\right)*I(x,y,t) \\ &= F(t,\sigma)*I(x,y,t). \end{aligned} \tag{4}$$

In Eq. (4), $F(t,\sigma)$ is used as the mask of the input signal, where a_i depends on σ only. According to Gaussian filtering theory, OGHM includes the majority of useful information when we use a mask of size $l=10\sigma+1$.

To detect the moving targets in a video sequence with the OGHM method, the convolution computation with $F(t,\sigma)$ and $I(x,y,t)$ is first performed, resulting in the 'moment image', represented by $M_n(x,y)$.

Since $M_n(x,y)$ is a gray-scale image, to generate foreground seeds, binarization is necessary for $M_n(x,y)$. In this study, we adopt the Otsu algorithm to calculate the threshold for binarization. The output binary image is represented by $B(x,y)$. In $B(x,y)$, pixel '1' indicates the motion knowledge.

Fig. 1 shows the process of moving target detection with the above processes. Fig. 1a is the middle

(a) (b) (c)

Fig. 1 Moving target detection in the video with orthogonal Gaussian-Hermite moments (OGHM): (a) middle frame of the consecutive frames; (b) middle image by OGHM; (c) binary image by the Otsu algorithm

frame of the input video sequence, Fig. 1b is the moment image $M_n(x, y)$, and Fig. 1c is the binary image $B(x, y)$.

2.2 Foreground and background seeds

Binary image $B(x, y)$ shows the motion knowledge of the input video as well. However, because of interference in the traffic video, there is noise in $B(x, y)$. If $B(x, y)$ is directly used as foreground seeds, the segmentation results may not be satisfying. Therefore, $B(x, y)$ should be optimized with morphological processing. The optimized binary image can be regarded as 'foreground seeds' (Fig. 2a).

The seeds (Li *et al*., 2004; Levin *et al*., 2008) usually include two types: foreground seeds and background seeds (indicated by the pixels on the colored lines in Figs. 4a and 4b, respectively). While Fig. 2a indicates the foreground seeds only, the background seeds still need to be generated automatically. Distance transformation and threshold segmentation are performed to realize background seed generation.

First, an image with the same size as the foreground seeds (Fig. 2a) is created, where the foreground seed pixels are set to '0' and non-seed pixels to infinity. Second, starting from the pixels with value '0', the image is passed over with the distant transform mask (Fig. 3). According to Eq. (5), each non-seed pixel is assigned a distance value:

$$\begin{aligned} V_{ij} = \min(&V_{i-1,j-1}+d_2, V_{i-1,j}+d_1, V_{i-1,j+1}+d_2,\\ &V_{i,j-1}+d_1, V_{i+1,j-1}+d_2, V_{i+1,j}+d_1,\\ &V_{i+1,j+1}+d_2, V_{i,j+1}+d_1). \end{aligned} \quad (5)$$

In this work, we set d_1=1 and d_2=2. The result is called a 'distance image' (Fig. 2b). The pixels further away from the foreground seed pixels have higher intensity, and they are assigned larger distance values. Third, the distance image is binary segmented using a predefined threshold to generate a binary image (Fig. 2c). The pixels with value '1' (those in the white area) are far away from the foreground seeds (Fig. 2a); thus, they can be viewed as the background seed pixels. As a result, we define Fig. 2c as the background seeds.

d_2	d_1	d_2
d_1	0	d_1
d_2	d_1	d_2

Fig. 3 Mask of distance transformation

3 Extraction of foreground targets

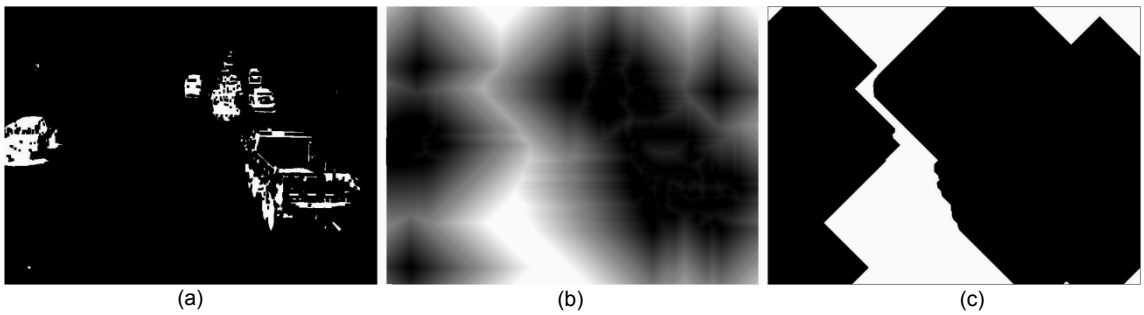
3.1 Segmentation based on graph cuts

Distinguishing foreground from background can be regarded as labeling on the label field. In this study, it is also regarded as solving the Markov random field (MRF) problem (Li, 1995). According to probability theory, the problem of finding the optimal segmentation is equivalent to finding the maximum probability configuration of the MRF, which can also be treated as a cost minimization problem.

In other words, the solution is to find the labeling set that minimizes the following cost function:

$$E(L) = \lambda_1 \sum_{p \in P} D_p(L_p) + \lambda_2 \sum_{(p,q) \in N} F_{p,q}(L_p, L_q), \quad (6)$$

where $L=\{L_1, L_2, \cdots, L_{|P|}\}$ is the set of labeling values, P is the set of all nodes in the image, N is the set of all arcs connecting the adjacent nodes, p and q are the



(a) (b) (c)

Fig. 2 Background seed automatic generation based on foreground seeds: (a) foreground seeds; (b) distance image; (c) background seeds

basic elements of the image, either the pixel or the region, and L_p and L_q are corresponding labeling values of p and q in the label field, respectively. $D_p(L_p)$ is the data energy, denoting the cost when the label of node p is L_p, and $F_{p,q}(L_p, L_q)$ is the smoothness energy, denoting the cost when the labels of adjacent nodes p and q are L_p and L_q, respectively. λ_1 and λ_2 are weights to balance the two terms in Eq. (6).

Based on the MRF model and the min-cut/max-flow theory in graph theory, the graph cuts algorithm was originally proposed by Boykov *et al*. (2001). Graph cuts can be used to process not only the basic pixels but also the pre-segmented regions. The processing of pre-segmented regions is much faster than that of basic pixels, and less prone to cavities. The process of graph cuts segmentation based on pixels or regions is shown in Fig. 4.

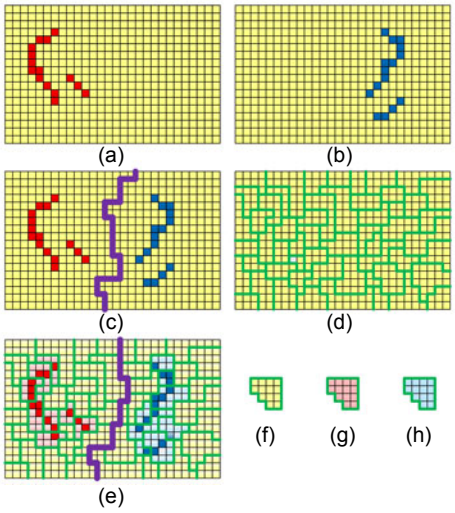


Fig. 4 The process of graph cut segmentation based on pixels or regions: (a) foreground seeds; (b) background seeds; (c) graph cuts based on pixels; (d) pre-segmentation image; (e) graph cuts based on regions; (f) uncertain region; (g) foreground seed region; (h) background seed region

The time complexity of graph cuts is very high when using elementary pixels as processing units (Fig. 4c). To improve the real-time performance, the original image is pre-segmented (Fig. 4d). The basic unit of pre-segmentation is the region. The regions containing seed pixels are named seed regions (Figs. 4g and 4h) and the regions not containing seed pixels are named uncertain regions (Fig. 4f). In fact, the processes of segmentation are essentially the same whether the basic unit is a pixel or a region. Pre-segmentation methods include feature clustering, region growing, and watershed algorithms (Vincent and Soille, 1991). A watershed algorithm is adopted in this study.

Color $C(i)$ is defined as the mean color of a region. The foreground seed regions, the background seed regions, and uncertain regions are labeled F, B, and U, respectively. The mean color sets of the three regions are designated as $\{C_F\}$, $\{C_B\}$, and $\{C_U\}$, respectively. After $\{C_F\}$ and $\{C_B\}$ are color clustered using the K-means method, the results of clustering are denoted as $\{K_n^F\}$ and $\{K_m^B\}$.

For each region p, the minimum distance from its color $C(p)$ to the foreground clusters can be calculated as $d_p^F = \min \| C(p) - K_n^F \|$, and to the background clusters as $d_p^B = \min \| C(p) - K_m^B \|$. Therefore, the data energy $D_p(L_p)$ in Eq. (6) can be calculated as follows:

$$\begin{cases} D_p(L_p=1) = \begin{cases} 0, & p \in F, \\ \infty, & p \in B, \\ \dfrac{d_p^F}{d_p^F + d_p^B}, & p \in U, \end{cases} \\ D_p(L_p=0) = \begin{cases} \infty, & p \in F, \\ 0, & p \in B, \\ \dfrac{d_p^B}{d_p^F + d_p^B}, & p \in U. \end{cases} \end{cases} \tag{7}$$

$C_{pq}=\|C(p)-C(q)\|^2$ is weighted by the shared boundary length between regions p and q. Therefore, the smoothness energy $F_{p,q}(L_p, L_q)$ in Eq. (6) is calculated as follows:

$$F_{p,q}(L_p, L_q) = \frac{|L_p - L_q|}{C_{pq} + 1}. \tag{8}$$

The results of Eqs. (7) and (8) can be substituted into Eq. (6), and thus the minimum energy can be obtained.

3.2 Matting based on closed-form solution

The matting problem is based on some extra constraints. Therefore, it is assumed that both F and B in Eq. (1) are approximately constant over a small

window around each pixel. In other words, F and B are assumed to be locally smooth. Under this assumption, Eq. (1) can be rewritten to express α as a linear function:

$$\alpha_i \approx aI+b \quad \forall i \in w, \tag{9}$$

where $a=1/(F-B)$, $b=-B/(F-B)$, and w is a small image window. The goal of matting is to find the values of α, a, and b which minimize the cost function:

$$J(\alpha,a,b)=\sum_{j\in I}\left(\sum_{i\in w_j}(\alpha_i-a_jI_i-b_j)^2+\varepsilon a_j^2\right), \tag{10}$$

where w_j is a small window around pixel j. $a_j=0$ means that α is constant over the jth window.

Eq. (10) includes a regularization term on a. This term is added to provide numerical stability. In this work, ε is a constant whose value is 0.0001.

Since a window is placed around each pixel, the windows w_j in Eq. (10) overlap. This property enables the propagation of information between neighboring pixels. However, as Eq. (10) has three variables α, a, and b, with $3N$ unknown numbers for an image with N pixels, no definite solutions exist.

a and b are selected to minimize Eq. (10), so that $J(\alpha, a, b)$ becomes a unary function of variable α, defined as $J(\alpha)$:

$$J(\alpha)=\min_{a,b}J(\alpha,a,b). \tag{11}$$

Then $J(\alpha)=\alpha^{\mathrm{T}}L\alpha$, where L is an $N\times N$ matrix, whose (i, j)th entry is

$$L_{ij}=\sum_{k|(i,j)\in w_k}\left\{\delta_{ij}-\frac{1}{|w_k|}\cdot\left[1+\frac{1}{\varepsilon/|w_k|+\sigma_k^2}(I_i-\mu_k)(I_j-\mu_k)\right]\right\}, \tag{12}$$

where δ_{ij} is the Kronecker delta, μ_k and σ_k^2 are the mean and variance of the intensities in the window w_k around pixel k, respectively, and $|w_k|$ is the number of pixels in this window. In this work we use windows of 3×3 pixels.

From Eq. (12) we can see that a and b of the cost function in Eq. (10) can be eliminated, leaving a cost with N unknown numbers, which is also the α value of the pixels.

To extract an α matte according to the seeds, the following equation needs to be solved:

$$\alpha=\arg\min(\alpha^{\mathrm{T}}L\alpha), \quad \alpha_i=s_i \ \forall i\in S, \tag{13}$$

where S is a set of seed pixels and s_i is the value indicated by the seeds.

Let α^* denote the true α matte. If F and B satisfy the line model in every local window w_k, and if the extra constraints S are consistent with α^*, then α^* is an optimal solution for Eq. (13), where ε in L is 0 (Eq. (12)). Since $\varepsilon=0$, if the color line model is satisfied in every window w_k, it follows from the Eq. (10) that $J(\alpha^*, a, b)=0$, and therefore $J(\alpha^*)=\alpha^{*\mathrm{T}}L\alpha^*=0$.

3.3 Mask generation

The theory in Section 3.1 refers to Boykov and Jolly (2001) and Li *et al*. (2004). The segmentation method based on graph cuts is hard segmentation, so the mask is a binary image. From Eqs. (7) and (8), we can see that the labeling value L_p of an arbitrary region p can be only either 0 or 1. According to Eqs. (7) and (8), to minimize the cost $E(L)$, we use the min-cut/max-flow theory (Boykov and Kolmogorov, 2004). The result is noted as L, which is also the mask.

The theory in Section 3.2 refers to Levin *et al*. (2008). Since α matte is a gray-scale image, binarization is required, the threshold of which is determined according to the actual situation.

The mask can also be noted as L.

When the mask is generated, it is multiplied by the original image $I(x, y)$:

$$M(x, y)=I(x, y)*L, \tag{14}$$

where $M(x, y)$ is the segmentation result of moving targets in the video sequence.

4 Flow chart and algorithm process of the proposed method

The process of the proposed method is summarized in Fig. 5. First, several consecutive frames are sampled from the input video sequence. The sample

frames are processed with OGHM, Ostu binarization, and distance transformation, resulting in the foreground and background seeds. Second, the mask is generated with graph cuts or closed-form solution using foreground seeds and background seeds as extra constraints.

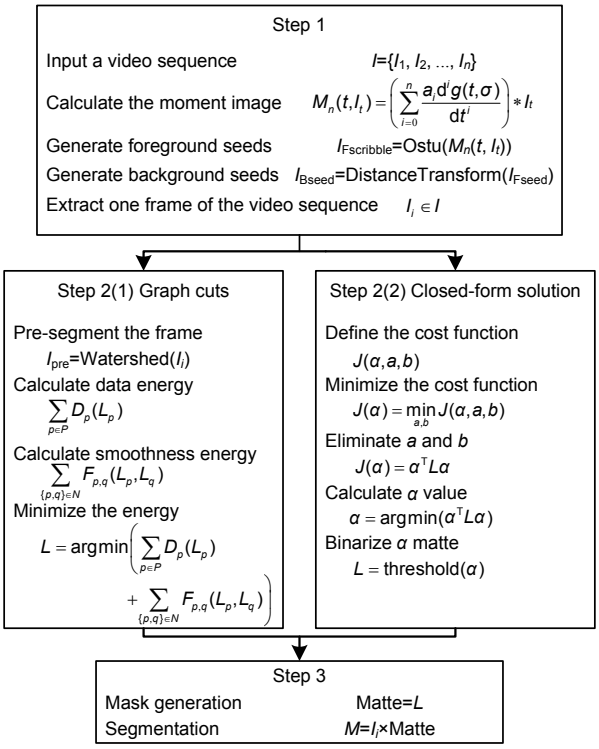


Fig. 5 Algorithm process of the proposed method

5 Experimental results

The proposed method was compared with other video matting methods, including a method depending on auto-generated trimap. Then the methods are compared with each other in terms of their accuracy rate, detection rate, and Jaccard index.

As shown in Fig. 6, $M_{00}\cup M_{01}\cup M_{10}\cup M_{11}$ indicates the gross pixel numbers of an image, $M_{10}\cup M_{11}$ is the pixel number of the template area, $M_{01}\cup M_{11}$ is the pixel number of the segmentation result area, and M_{11} indicates the number of pixels in the intersection of the template area and segmentation result area. The accuracy rate, detection rate, and Jaccard index are defined as

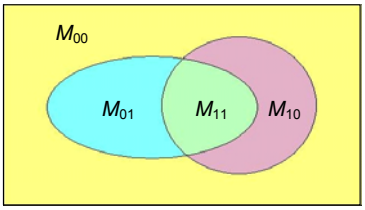


Fig. 6 Relationship between the template area and the segmentation result area

$$\begin{cases}\text{Accuracy\_Rate}=\dfrac{M_{11}}{M_{11}\cup M_{10}},\\ \text{Detection\_Rate}=\dfrac{M_{11}}{M_{11}\cup M_{01}},\\ \text{Jaccard\_Index}=\dfrac{M_{11}}{M_{11}\cup M_{10}\cup M_{01}}.\end{cases} \quad (15)$$

These three criteria have different meanings, but they all range from 0 to 1, and a larger value represents a better result from the segmentation algorithm.

5.1 Results of manual scribbles and auto-generated seeds

Figs. 7a and 7b show the process of segmentation when the extra constraints are manual scribbles. Fig. 7c shows the process of segmentation using the proposed method. Since the generation mechanisms of the two extra constraints are not the same, no quantitative comparison is made. Fig. 7 proves that the proposed method based on auto-generated seeds is capable of achieving similar performance to methods based on manual interaction.

5.2 From auto-generated seeds to extracted mask

Fig. 8 shows the progress and results of the proposed method. The input of the proposed method is a video sequence in which Fig. 8a is one frame. Fig. 8b is the binary image from OGHM and morphological processing with the input video sequence, which is used as the foreground seeds in segmentation. Fig. 8c is the binary image when the foreground seeds are processed with distance transformation, which is used as the background seeds in segmentation. Figs. 8b and 8c are used as the extra constraints of the subsequent process. Fig. 8d is the mask generated with graph cuts. Fig. 8e is the matte generated with a closed-form solution. Fig. 8f is the mask generated by binarization from Fig. 8e.

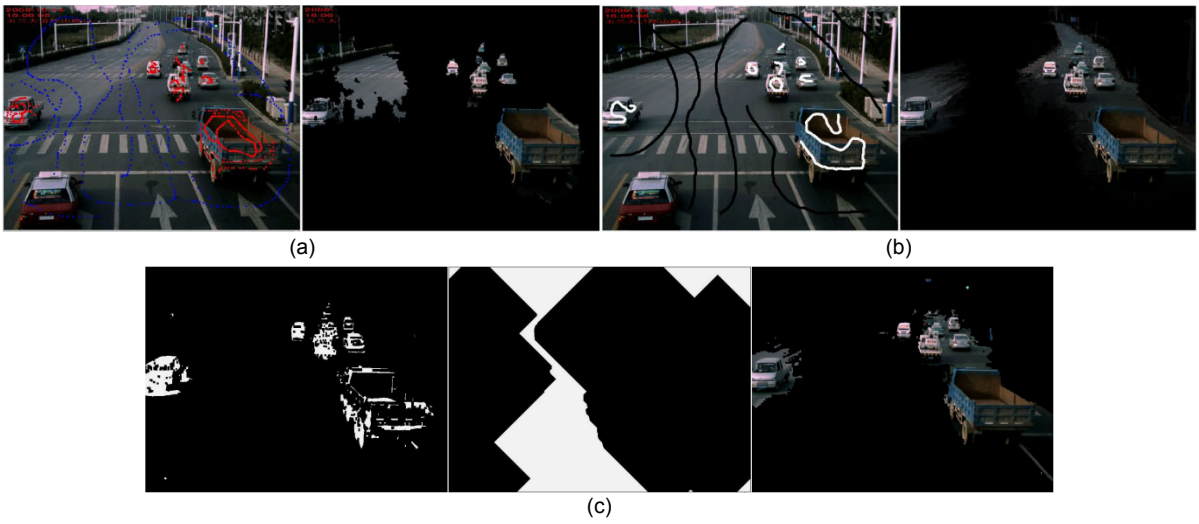


Fig. 7 Manual scribbles compared with auto-generated seeds: (a) lazy-snapping based on manual scribbles; (b) closed-form solution based on manual scribbles; (c) the proposed method

In (a) and (b), the figures on the left are the manual inputs and those on the right are the segmentation results; in (c), the two figures on the left are the foreground seeds and background seeds, respectively

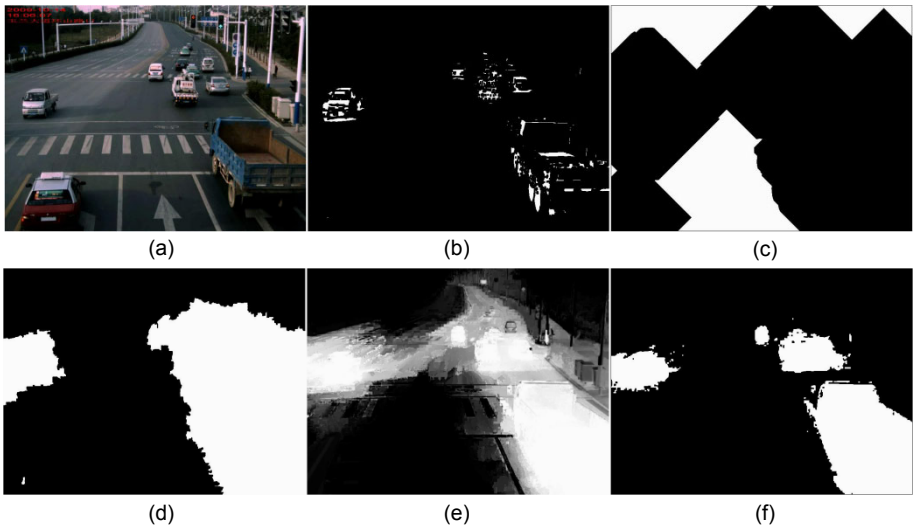


Fig. 8 Moving target extraction based on the proposed method: (a) one frame of the video sequence; (b) foreground seeds; (c) background seeds; (d) extracted mask based on graph cuts; (e) extracted matte based on a closed-form solution; (f) mask binarization of the matte

5.3 Comparison of different algorithms

Fig. 9 displays the non-interactive segmentation results of moving objects in a traffic video clip with different methods. The eight figures in Fig. 9a are frames 21, 26, 31, 36, 41, 46, 51, and 56; the figures in Fig. 9b are the corresponding template masks. The figures in Fig. 9c demonstrate the experiment results based on auto-generated trimap (Gong *et al*., 2010). The figures in Fig. 9d are the masks from the closed-form solution, while those in Fig. 9e are the masks from the graph cuts method. The accuracy rate, detection rate, and Jaccard index of the frames in Fig. 9 are listed in Table 1.

5.4 Analysis of parametric influence

With the same foreground and background seeds, we compared the influence of the parameters, such as the number of means in the *K*-means algorithm and the number of means in the *K*-means algorithm and the ratio of λ_1 to λ_2 in Eq. (6). The masks in these experiments are shown in Fig. 10.

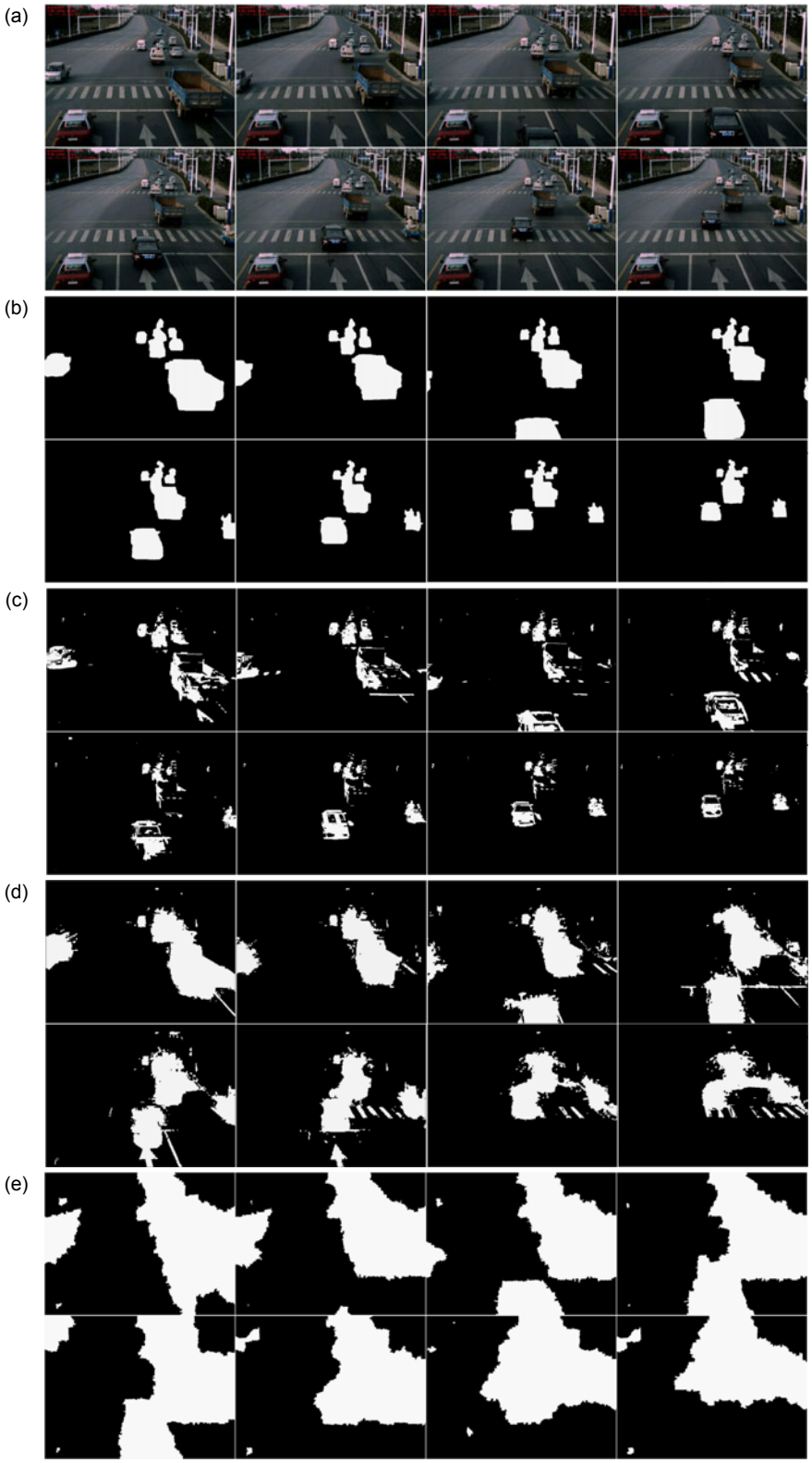


Fig. 9 Video segmentation results from using the proposed method: (a) original frames; (b) template mask; (c) mask calculated by auto-generated trimap; (d) mask calculated by a closed-form solution; (e) mask calculated by graph cuts

Table 1 The evaluation indices of Figs. 9c–9e

Frame No.	Accuracy rate			Detection rate			Jaccard index		
	AGT	CFS	GC	AGT	CFS	GC	AGT	CFS	GC
21	0.889	0.746	0.346	0.538	0.938	0.998	0.504	0.711	0.356
26	0.808	0.783	0.306	0.466	0.904	0.996	0.420	0.723	0.310
31	0.836	0.661	0.290	0.475	0.919	0.996	0.434	0.625	0.289
36	0.835	0.616	0.298	0.434	0.907	0.997	0.400	0.579	0.298
41	0.865	0.595	0.219	0.445	0.930	0.998	0.416	0.570	0.219
46	0.882	0.552	0.219	0.524	0.944	0.998	0.490	0.535	0.219
51	0.849	0.500	0.158	0.486	0.941	0.999	0.447	0.485	0.158
56	0.863	0.424	0.154	0.456	0.939	0.999	0.425	0.413	0.154

AGT: mask calculated by auto-generated trimap (Fig. 9c); CFS: mask calculated by closed-form solution (Fig. 9d); GC: mask calculated by graph cuts (Fig. 9e)

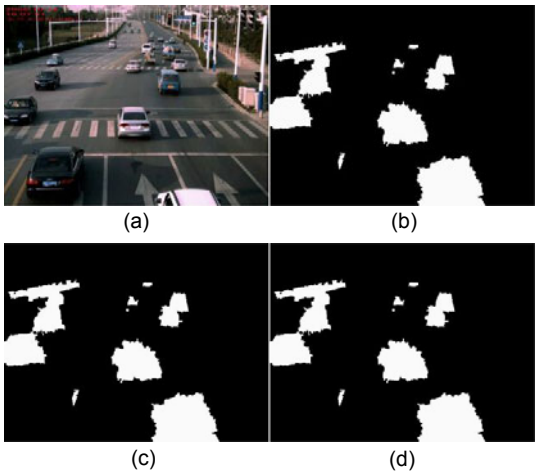


Fig. 10 Masks with different parameters but the same seeds: (a) original frame; (b) mean=5, λ_2/λ_1=60; (c) mean=60, λ_2/λ_1=60; (d) mean=5, λ_2/λ_1=1

With other parameters unchanged, the influence of the seeds was compared. Background seeds were generated from the distance image using the threshold segmentation (Fig. 2). The threshold varies with background seeds. Fig. 11 shows masks with different thresholds. Fig. 12 shows the trends of the evaluation indices in relation to the threshold.

To verify the universality of the proposed method and to obtain a set of empirical parameters, three different video clips were processed with the proposed method. The mean accuracy of each video clip was calculated. The three video clips were taken at the same location but the speed, flow, density, and color of the moving targets were different. According to Fig. 12, the threshold in distance transformation was selected as 40. The detection results from using different thresholds for binarizing the alpha matte are shown in Fig. 13. The threshold calculated with the Otsu method is denoted as T_{otsu}. A number denoted by T was added to T_{otsu} forming the threshold used in generation of the mask.

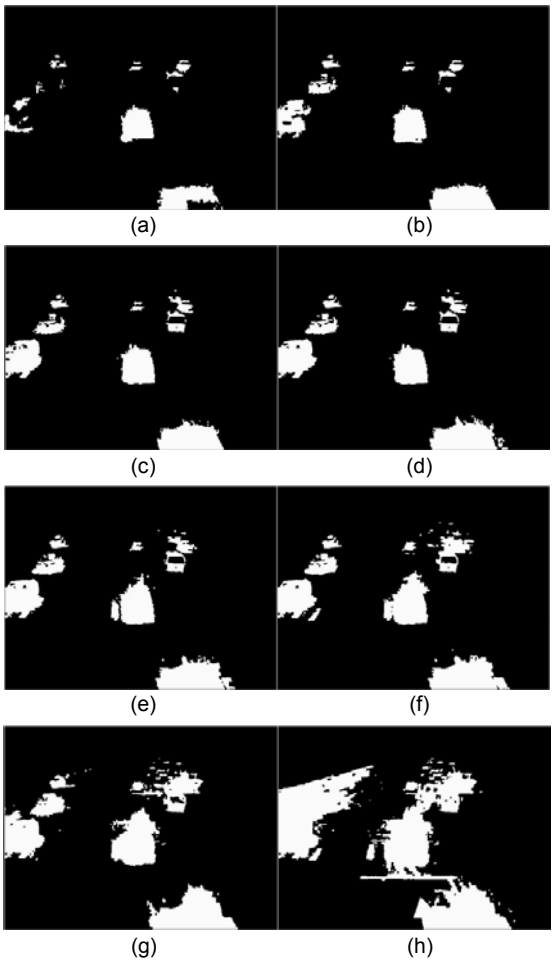


Fig. 11 Masks used in distance transform with the threshold being 10 (a), 20 (b), 30 (c), 40 (d), 50 (e), 60 (f), 90 (g), and 120 (h)

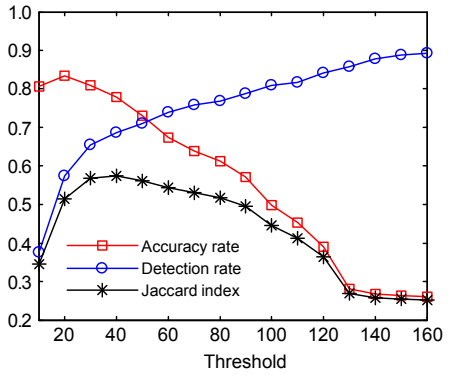


Fig. 12 Trends of the evaluation indices with threshold

The mean accuracy of the video clips differed markedly (Table 2), so the mean accuracy is closely related to the features of the moving targets. For the same video clip, the influence of T on the mean accuracy was relatively small. The average processing time per frame of each video clip is listed in the last column of Table 2.

6 Discussions

6.1 Selection of optimal parameters

Figs. 10b–10d are almost the same, so when the foreground and background seeds remain unchanged, other parameters have little impact on the result. From Figs. 11 and 12 we can see that as the threshold increases, the detection rate increases while its gradient decreases, and the accuracy rate decreases. There is a peak in the Jaccard index. Thus, the threshold with the peak Jaccard index is often selected as the threshold required for auto-generated background seeds.

From the analysis above we can see that, as the extra constraints, the seeds have the largest impact on the segmentation results. Therefore, for a segmentation method based on auto-generated seeds, the core of the method is how to make the seeds better represent the foreground and background.

The optimal threshold in distance transformation is 40 (Fig. 12). The best estimated threshold for binarizing the alpha matte was T=0.25 (Table 2). With this threshold, the mean accuracy reaches a local maximum.

6.2 Auto-generated trimap compared with auto-generated seeds

Comparisons of the results in Table 1 show that the detection rate of the method based on auto-generated trimap was very low while the accuracy rate was very high. This indicates that the moving targets are segmented accurately by this method, but much of the motion knowledge is lost due to the limitations of trimap. In contrast, the detection rate of the method based on auto-generated seeds was much higher than that of the method based on the auto-generated trimap. The method based on auto-generated seeds preserves most of the motion knowledge in the original video sequence.

6.3 Soft segmentation compared with hard segmentation

In the process of moving target extraction based on auto-generated seeds, we adopt a closed-form solution based on soft segmentation, and graph cuts based on hard segmentation in generating the final mask. Hard segmentation had a very high detection rate and a very low accuracy rate, resulting in a low Jaccard index (Table 1). This shows that hard segmentation preserves all the motion knowledge of the video sequence, but also keeps a lot of redundant information.

Sometimes moving targets are not detected when using soft segmentation (Fig. 10). This never happens when using hard segmentation, indicating that hard segmentation is more stable than soft segmentation.

With the combination of auto-generated seeds and closed-form solution, the resulting Jaccard index (Table 1) is obviously higher than that of other video segmentation methods. This indicates that the proposed method is suitable for low quality video sequences with multiple moving targets, and thus is more expandable than other video segmentation methods. The proposed method may also be suitable for other applications which need extraction of moving targets.

6.4 Real-time processing

The last column of Table 2 shows the average processing time per frame of the three video clips. The processing speed is lower than 30 frames/s, which is not enough for real-time processing. However, the proposed method was implemented with MATLAB which is based on interpretive execution. If the proposed method is implemented on a high-performance hardware platform with Visual C++ based on complied execution, the processing speed can be improved and real-time processing can be realized.

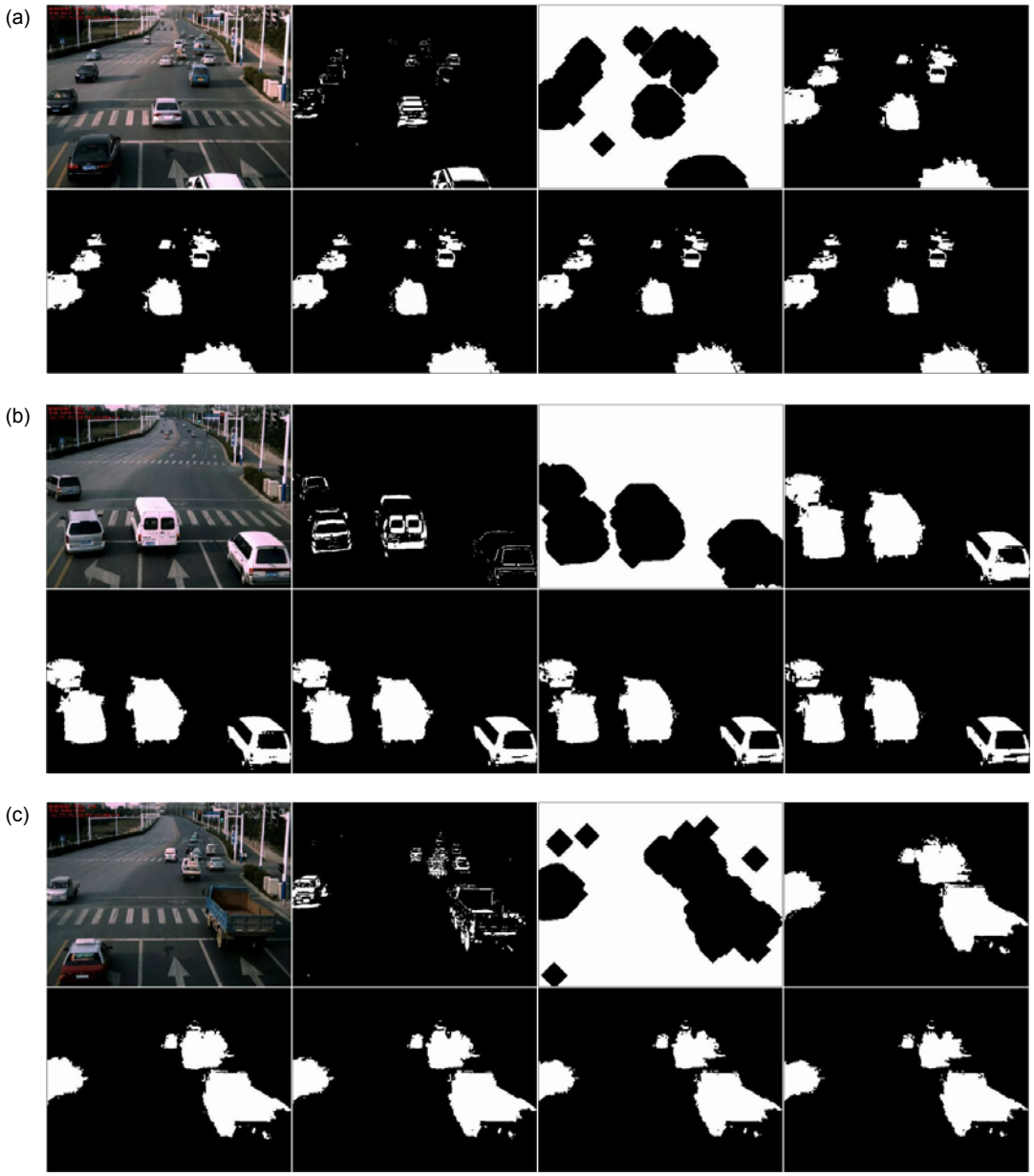


Fig. 13 Results of video clip 1 (a), video clip 2 (b), and video clip 3 (c)

The four figures in the first row of (a–c) are a random frame in the video clip, foreground seeds, background seeds, and the masks respectively, with T equal to 0.15. The four figures in the second row of (a–c) are the masks with T equal to 0.20, 0.25, 0.30, and 0.35, respectively

Table 2 The evaluation indices of the three videos in Fig. 13

Video No.	Total number of frames	Mean accuracy					Average processing time (s)
		T=0.15	T=0.20	T=0.25	T=0.30	T=0.35	
1 (Fig. 13a)	51	0.575	0.583	0.585	0.583	0.574	0.78
2 (Fig. 13b)	75	0.673	0.702	0.710	0.696	0.687	0.82
3 (Fig. 13c)	97	0.714	0.725	0.736	0.732	0.731	0.85

7 Conclusions

An automatic segmentation method is proposed for the moving targets in a complex video. The procedures of the proposed method are as follows. First, the moving targets in a video sequence are detected using OGHM and used as foreground seeds after binarization. Second, background seeds are generated with distance transform and threshold segmentation based on the foreground seeds. Third, one frame in the middle of the video sequence is selected as the original image. Finally, with the two types of seeds combined, the final mask is generated with graph cuts and a closed-form solution. Experiment showed that when the threshold to binarize the alpha matte is properly selected, the closed-form solution based on soft segmentation is better than graph cuts based on hard segmentation.

Compared with traditional video segmentation methods, the proposed method achieves satisfactory segmentation results without human-computer interaction, thus realizing non-interactive moving target extraction, and providing a good resolution for the automatic detection of moving targets in video sequences. Moreover, the proposed method achieves a better Jaccard index than the traditional methods.

Experimental results also revealed some problems. First, for low contrast images, segmentation results based on color features only were not satisfactory. Therefore, the proposed method is less effective when the color contrast between vehicles and the background is very low. Second, as a result of an automatic process, the seeds may be generated in unexpected positions, which are not as accurate as manual scribbles in separating the foreground from the background, thus introducing noise. The noise may cause differences between the actual target and the results of mask generation.

Further research is needed to investigate and strengthen the proposed method in two aspects to achieve a higher Jaccard index: first, more features (such as the geometric shapes of vehicles) could be added to the proposed method to make it more suitable for the segmentation of traffic images; second, the process of seed generation should be optimized. For example, local feature extraction for moving target detection could be combined with OGHM to separate the foreground from the background more accurately.

References

Apostoloff, N., Fitzgibbon, A., 2004. Bayesian Video Matting Using Learnt Image Priors. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.407-414. [doi:10.1109/CVPR.2004.1315061]

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut-max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(9):1124-1137. [doi:10.1109/TPAMI.2004.60]

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**(11):1222-1239. [doi:10.1109/34.969114]

Boykov, Y.Y., Jolly, M.P., 2001. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Object in ND Images. Proc. 8th IEEE Int. Conf. on Computer Vision, p.105-112. [doi:10.1109/ICCV.2001.937505]

Chuang, Y., 2004. New Models and Methods for Matting and Compositing. PhD Thesis, University of Washington, Washington D.C., USA.

Chuang, Y., Curless, B., Salesin, D.H., Szeliski, R., 2001. A Bayesian Approach to Digital Matting. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.264-271. [doi:10.1109/CVPR.2001.990970]

Chuang, Y., Agarwala, A., Curless, B., Salesin, D.H., Szeliski, R., 2002. Video matting of complex scenes. *ACM Trans. Graph.*, **21**(3):243-248. [doi:10.1145/566654.566572]

Gong, M.L., Wang, L., Yang, R.G., Yang, Y.H., 2010. Real-Time Video Matting Using Multichannel Poisson Equations. Proc. Graphics Interface, p.89-96.

Jiang, M., Crookes, D., Chen, M., 2010. Multi-layer Stereo Video Matting: Video Matting. Proc. Int. Conf. on Multimedia, p.1163-1166. [doi:10.1145/1873951.1874177]

Lee, S.Y., Yoon, J.C., Lee, I.K., 2010. Temporally coherent video matting. *Graph. Models*, **72**(3):25-33. [doi:10.1016/j.gmod.2010.03.001]

Levin, A., Lischinski, D., Weiss, Y., 2004. Colorization using optimization. *ACM Trans. Graph.*, **23**(3):689-694. [doi:10.1145/1015706.1015780]

Levin, A., Lischinski, D., Weiss, Y., 2008. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(2):228-242. [doi:10.1109/TPAMI.2007.1177]

Li, S.Z., 1995. Markov Random Field Models in Computer Vision. Springer Verlag, New York, USA, p.1-10.

Li, W., Han, G.Q., Gu, Y.C., Zhang, X.Y., Zhang, S.K., 2010. Robust video matting algorithm. *J. Appl. Res. Comput.*, **27**(1):358-360, 376. [doi:10.3969/j.issn.1001-3695.2010.01.107]

Li, Y., Sun, J., Tang, C.K., Shum, H.Y., 2004. Lazy snapping. *ACM Trans. Graph.*, **23**(3):303-308. [doi:10.1145/1015706.1015719]

McGuire, M., Matusik, W., Pfister, H., Hughes, J.F., Durand, F., 2005. Defocus video matting. *ACM Trans. Graph.*, **24**(3):567-576. [doi:10.1145/1073204.1073231]

Rother, Y., Kolmogorov, V., Blake, A., 2004. "GrabCut"—

interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, **23**(3):309-314. [doi:10.1145/1015706.1015720]

Smith, A.R., Blinn, J.F., 1996. Blue Screen Matting. Proc. 23rd Annual Conf. on Computer Graphics and Interactive Techniques, p.259-268. [doi:10.1145/237170.237263]

Sun, J., Jia, J., Tang, C.K., Shum, H.Y., 2004. Poisson matting. *ACM Trans. Graph.*, **23**(3):315-321. [doi:10.1145/1015706.1015721]

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, **13**(6):583-598. [doi:10.1109/34.87344]

Wang, J., Cohen, M.F., 2005. An Iterative Optimization Approach for Unified Image Segmentation and Matting. 10th IEEE Int. Conf. on Computer Vision, p.936-943. [doi:10.1109/ICCV.2005.37]

Wang, J., Cohen, M.F., 2007a. Image and video matting: a survey. *Found. Trends Comput. Graph. Vis.*, **3**(2):97-175. [doi:10.1561/0600000019]

Wang, J., Cohen, M.F., 2007b. Optimized Color Sampling for Robust Matting. IEEE Conf. on Computer Vision and Pattern Recognition, p.1-8. [doi:10.1109/CVPR.2007.383006]

Wang, L., Gong, M.L., Zhang, C.X., Yang, R.G., Zhang, C., Yang, Y.H., 2012. Automatic real-time video matting using time-of-flight camera and multichannel Poisson equations. *Int. J. Comput. Vis.*, **97**(1):104-121. [doi:10.1007/s11263-011-0471-x]

Wu, Y., Shen, J., 2004. Moving object detection using orthogonal Gaussian Hermite moments. *SPIE*, **5308**:841-849. [doi:10.1117/12.525427]