Sensors **2015**, *15*, 8214-8231; doi:10.3390/s150408214

OPEN ACCESS

sensors

ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Multi-Model Estimation Based Moving Object Detection for Aerial Video

Yanning Zhang, Xiaomin Tong *, Tao Yang and Wenguang Ma

School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China;
E-Mails: zhangyanning_npu@163.com (Y.Z.); yangtaonwpu@163.com (T.Y.);
chogorima@gmail.com (W.M.)

* Author to whom correspondence should be addressed; E-Mail: vision215@mail.nwpu.edu.cn;
Tel.: +86-159-2995-3071; Fax: +86-29-8843-1517.

Academic Editor: Fabrizio Lamberti

Received: 8 December 2014 / Accepted: 30 March 2015 / Published: 8 April 2015

Abstract: With the wide development of UAV (Unmanned Aerial Vehicle) technology, moving target detection for aerial video has become a popular research topic in the computer field. Most of the existing methods are under the registration-detection framework and can only deal with simple background scenes. They tend to go wrong in the complex multi background scenarios, such as viaducts, buildings and trees. In this paper, we break through the single background constraint and perceive the complex scene accurately by automatic estimation of multiple background models. First, we segment the scene into several color blocks and estimate the dense optical flow. Then, we calculate an affine transformation model for each block with large area and merge the consistent models. Finally, we calculate subordinate degree to multi-background models pixel to pixel for all small area blocks. Moving objects are segmented by means of energy optimization method solved via Graph Cuts. The extensive experimental results on public aerial videos show that, due to multi background models estimation, analyzing each pixel's subordinate relationship to multi models by energy minimization, our method can effectively remove buildings, trees and other false alarms and detect moving objects correctly.

Keywords: aerial video; object detection; multi-model estimation; Graph Cuts

1. Introduction

Moving target detection for aerial video is one of the core technologies of UAV (Unmanned Aerial Vehicle) surveillance systems. This technology can be widely applied in military domains such as battlefield reconnaissance and surveillance, positioning and adjustment, damage assessment, electronic warfare, *etc.* Also, it can support civil purposes such as border patrol, nuclear radiation detection, aerial photography, aerial prospecting, disaster monitoring, traffic patrol, security surveillance, *etc.* Due to its wide application, low cost, high cost effectiveness, no risk of casualties, strong survival ability, good maneuvering performance and convenience, moving object detection algorithm for UAV aerial video has become a hot research topic in the computer field. Moving object detection from a UAV is an important research topic crossing image processing and vehicle control. The purpose of this research is to automatically obtain the target position and motion information based on aerial video. This study can not only make UAV's eyes more clear, but also guarantee the advanced processing and applications, such as behavior analysis and importance analysis.

We are faced with core difficulties in moving object detection for aerial video, such as motion mutation caused by UAV fast motion, low resolution noisy images, small target, low contrast, complex background, scale changes and occlusion, *etc.* With UAV development, researchers have proposed many algorithms to solve the above problems. However, most of these methods are under the registration-detection framework, which assumes that scenario only has a single background and will identify all the regions generating parallax error as targets. As a result, tracking failure usually happens in complex scenarios with multiple backgrounds, trees, buildings, *etc.* Therefore, the state of the art solutions in moving object detection cannot satisfy application need and it is developing new technology for complex scenes is necessary.

Automatic estimation of multiple background models for complex scenarios can provide a solution for perceiving the scene accurately. This paper first focuses on automatic estimation of multiple background models for complex scenarios. Then the pixels' motion information and subordinate degrees to multi-background models are analyzed by optical flow. The subordinate degree between a pixel and a background model refers to the degree a pixel and its correspondence fit the background model. Usually, the projection error can be used to measure the subordinate degree. The larger the projection error, the lower the subordinate degree is. Based on the neighborhood information and the subordinate degree, we segment the moving objects via energy minimization [1,2]. Since we estimate multiple background models and perceive complex scenes correctly, our method can detect moving objects accurately under viaducts and other complex backgrounds. Meanwhile, our algorithm can effectively remove buildings, trees and other false alarms and improve the locating precision. In addition, the adoption of energy minimization, which makes use of both the analysis of neighborhood continuity and subordinate degree, can significantly improve segmentation precision.

The rest of this paper is organized as follows. Section 2 summarizes and analyzes the related work in recent years. Section 3 proposes the moving object detection algorithm based on multi-model estimation for aerial video of complex scenarios. The experimental results are reported in Section 4, which demonstrate the accuracy and effectiveness of our approach. Finally, the conclusions are drawn in Section 5.

2. Related Work

Moving object detection for aerial video [3] has widely developed in the past few decades. The existing moving object detection algorithms for aerial video mainly include two categories [4,5]: one is the bottom-up method and the other one is the top-down method. The bottom-up method is also named as Data-driven method, which does not rely on prior knowledge and extracts the moving information directly from the image sequences. Top-down method, also named the model-driven algorithm, which relies on the constructed model or prior knowledge, performs the matching computing and solves the posterior probability in image sequences. In matching computing, the moving objects will be detected if the similarity distance is close enough. When computing the posterior probability, the state vector corresponding to the maximum posterior probability will be denoted as the current status of the moving objects.

Using bottom-up method to realize moving object detection for aerial video mainly includes three steps [6–10]. The first step is image matching [11–13], which performs the adjacent frames registration for image sequences. The second step is object detection. Frame difference or background difference is often used to detect change blobs and obtain moving objects after registration. The third step is object classification. There are two tasks in this step. One is to extract the detected moving objects. The other one is to recognize these objects.

The existing bottom-up algorithms for moving object detection include the classic COCOA system [14]. The procedure of this system contains image stabilization, frame difference and block tracking. However, this algorithm often fails in scenario scaling due to the Harris corner-based image stabilization. Cohen *et al.* [15,16] proposed a moving object detection and tracking system. First they aligned the images by estimating the affine transformation model iteratively. Then, the normalized optical flow field was applied for motion detection and the graph representation was constructed to resolve and maintain the dynamic template of moving objects. This system runs fast but it cannot solve the complex scaling scenarios. Ibrahim *et al.* [17] proposed the MODAT framework. Instead of Harris corner, they adopted SIFT (Scale-invariant feature transform) [18] features to fulfill the image matching. However, all of the above methods can only deal with simple background scenes and assume that only the moving objects can cause the parallax error. They tend to go wrong in complex multiple background scenarios, such as viaducts, buildings and trees. Chad *et al.* [19] proposed a moving object detection method for aerial video with low frame rate. They constructed an accurate background model to solve the object detection and the shadow problems. However, the application of this method is restricted because we need to know the camera calibration parameters in advance and start tracking objects manually. Shen *et al.* [20] proposed a moving object detection method for aerial video basing on spatiotemporal saliency. However, this method still cannot overcome the parallax error problem and the false alarm rate is high in complex scenarios. As shown in Figure 1b, false alarms (labeled by the red circles) occurred at buildings and trees when using one affine model to describe the scene. The real objects may be missed due to the inaccurate model estimation.

The top-down method transforms the moving object detection problem to Bayesian prediction. With the known prior probability of the object state, the problem can be solved by estimating the maximum *posteriori* probability continuously after obtaining the new measurement. In other words, Bayesian theory considers the vision-tracking problem as a "best guess" or "deduction" process, and usually adopts the state space approach to achieve vision tracking. The Classical Kalman filter [21] can only handle linear, Gaussian and unimodal situation. However, *posteriori* estimation is often non-linear, non-Gaussian and multimodal in practice. Therefore, EKF (Extended Kalman Filter) [22] is proposed to handle such cases. A particle filter [23] can also solve such non-linear problems.

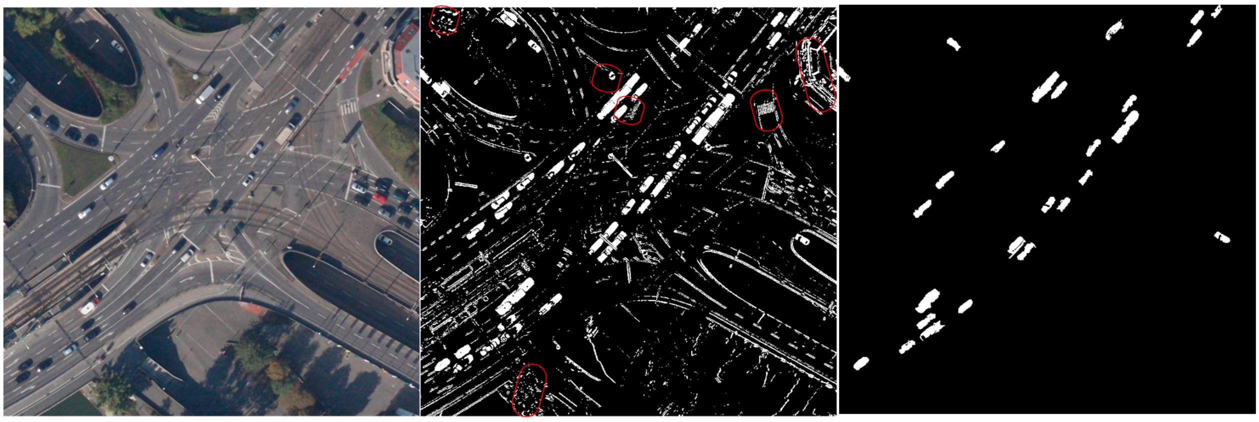
(**a**) (**b**) (**c**)

Figure 1. The comparison of moving detection results by different methods. (**a**) Original image (**b**) Moving detection with false alarms in red circles when using one affine model to describe the scene (**c**) Moving detection by our method.

The top-down method utilizes the *priori* knowledge to construct a model for the detection problem. Then, the model's correction is verified with the practical image sequences. Since it has a solid theoretical foundation of mathematics and many mathematical tools that can be adopted, the top-down approaches are always the mainstream methods for vision detection. These approaches transform object detection problems to deduction and prediction problems. The assumption is that when the prior knowledge of deduction is correct, the deduction results will be correct. Otherwise, the results may be wrong. Thus, acquiring correct prior knowledge is very important. Existing approaches mostly initialize the objects manually to ensure the correctness of later subsequent detection and location, which is unrealistic in the practical applications. Therefore, in order to detect moving objects automatically for aerial video, reliable detection results from the bottom-up approach should be used as the deduction's *priori* knowledge to achieve a correct prediction.

In this paper, we propose a moving object detection algorithm based on multi-model estimation for aerial video. First, we segment the scene into several color blocks and estimate the dense optical flow. Then, we calculate an affine transformation model for each large area block and merge the consistent models. Finally, Graph Cuts [1,2] is utilized to classify the foreground pixels into different objects. Our method can not only handle the moving object detection in the complex multiple background scenarios with viaducts, but can also remove buildings, trees and other false alarms effectively. As a result, the segmentation and detection precision will be improved.

3. Multi-Model Estimation Based Moving Object Detection

In order to overcome the influence of the complex multiple background scenarios, this paper proposes a moving object detection algorithm for aerial video basing on multi-model estimation.

Firstly, the scene is segmented into several color blocks. Secondly, the affine transformation model between each background region in the current frame and the corresponding region in the previous frame is estimated basing on the dense optical flow. Thirdly, subordinate degree is calculated between each pixel and multiple background models to judge whether the pixel belongs to a moving object or not. Finally, moving objects are segmented by energy optimization method solved via Graph Cuts.

3.1. Algorithm Flow

The flowchart of the proposed framework is shown in Figure 2. Our approach mainly includes four steps: the overall perception of the scene, background model extraction, background region segmentation and moving object detection. First, the overall perception of the scene segments the scene into several color blocks and estimates the dense optical flow. Here, the Mean shift pyramid segmentation method from OPENCV (Open Source Computer Vision Library) is adopted for color blocks segmentation and the Gunnar Farneback algorithm [24] is used for calculating dense optical flow. Second, to confirm the multiple background models included in the scenario, background model extraction calculates the affine transformation models for multiple color blocks and merges the consistent models. Third, the background region segmentation will be transformed to the background and foreground binary classification, multiple background regions and multiple labels classification problem. This problem can be solved by the energy optimization method, which can achieve smooth and continuous global optimal solution. Fourth, after obtaining the foreground regions, we merge the blocks and remove false objects based on the moving consistency and the region proximity. Afterwards, the moving object detection is finished and the accurate detected results are obtained. The background model extraction, background region segmentation and moving object detection are introduced in Section 3.2, 3.3 and 3.4, respectively. The details are as follows.

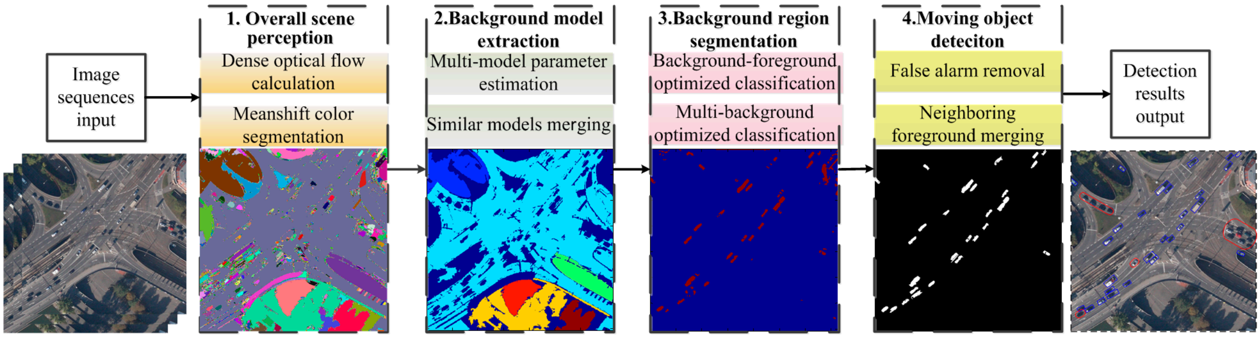
Image sequences input
1. Overall scene perception
Dense optical flow calculation
Meanshift color segmentation
2.Background model extraction
Multi-model parameter estimation
Similar models merging
3.Background region segmentation
Background-foreground optimized classification
Multi-background optimized classification
4.Moving object deteciton
False alarm removal
Neighboring foreground merging
Detection results output

Figure 2. The flowchart of moving object detection based on multi-model estimation.

3.2. Multi-Model Estimation

Estimating accurately the background model parameters of complex scenarios can ensure the correct scene perception, accurate object segmentation and robust object tracking. The current multi-model estimation methods, like JLinkage [25], do not need any prior segmentation information and can classify samples into multiple categories automatically, where each category corresponds to one model. However, this method only adapts to small samples and is unable to solve the big samples like multi-model estimation under complex scenarios. In the aerial video, the background blocks with consistent color often belong to the same background and the background area is much larger than that of objects. Therefore, this paper first segments the scenarios into color blocks and selects the blocks with the large area as the candidate background blocks. Afterwards, an affine transformation model is estimated for each background block.

Let us denote I_t and I_{t+1} as the adjacent two frames. Then, the dense optical flow can be computed by the Gunnar Farneback algorithm [24]. We define OFX_t and OFY_t as transverse and longitudinal optical flow, respectively. The corresponding relationships are as follows:

$$I_t(x,y) \sim I_{t+1}(x',y') \tag{1}$$

$$x' = x + OFX_t(x,y), y' = y + OFY_t(x,y) \tag{2}$$

where $I_t(x,y)$ represents the pixel value in (x,y) of image I_t. $I_{t+1}(x',y')$ is defined as the pixel value in (x',y') of image I_{t+1}. (x,y) and (x',y') form an optical flow pair.

Next, we segment I_t by using Mean shift algorithm, which segments the scene into multiple color blocks based on their color consistency. Then, the blocks whose area is larger than threshold $T_{a\min}$ are selected as background blocks $B_t = \{b_1, b_2, ..., b_{BNum}\}$, where *BNum* represents the number of the color background blocks obtained by segmenting. The color blocks' area set is defined as $A_t = \{a_{b1}, a_{b2}, ..., a_{bBNum}\}$ and a_{bi} is the number of the pixels included in the *i*th background color block. Afterwards, each point in color background blocks and its optical flow point in the next frame obtained by optical flow method [24] form a point pair. Basing on the point pairs in each background block, the affine transformation model between the background block in the current frame and the corresponding region in the next frame is estimated via RANSAC (RANdom SAmple Consensus) method [12].

$$M_t = \{m_1, m_2, ..., m_{BNum}\} \tag{3}$$

$$m_i = \begin{bmatrix} a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 \end{bmatrix} \tag{4}$$

The affine transformation model set M_t is composite of each background block's affine transformation model. The affine transformation model of the *i*th background block is denoted as m_i, including translation, rotation, scaling, cropping and other atomic transformations. a_0 and b_0 represent shift amount between the background block in the current frame and the region in the next frame along the horizontal and vertical direction, respectively. The rest parameters represent composite of scaling, rotation and shearing. The current background blocks segmentation is based on color consistence, so single background may be segmented into several backgrounds due to color inconsistence. For the convenience of later scene analysis, we need to merge multiple background models according to the consistency between different background models. Thus, we define the projection error of the pair of points as follows:

$$Error_i = \left\| (x',y')^T - m_i \cdot (x,y,1)^T \right\|_2 \tag{5}$$

where (x',y') denotes the optical flow point of pixel (x,y) in the consecutive frame. The projection error is the difference in pixels, between two points located in consecutive images that are related by the optical flow. If $Error_i < T_e$, the point pair often belongs to the inliers of the model m_i, otherwise the pair of points is an outlier for the *i*th background block. Then we calculate the connective matrix $R_{BNum \times BNum}$ between the background blocks and the affine models as follows:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1BNum} \\ \vdots & \vdots & \ddots & \vdots \\ r_{BNum1} & r_{BNum2} & \cdots & r_{BNumBNum} \end{bmatrix} \tag{6}$$

$$r_{ij} = \frac{a_{ij}}{a_j} \tag{7}$$

where r_{ij} represents the accordance degree of the *j*th background block b_j to the *i*th model m_i. aI_{ij} is the number of inliers belonging to model m_i in the *j*th background block b_j. $r_{ij} = \frac{aI_{ij}}{a_{bj}}$, denotes the j th background block's rate of the inliers to model m_i. If $r_{ij} > T_r$ and $r_{ji} > T_r$, the i th background block b_i and the *j*th background block b_j are from the same background plane and can be combined to one background model. Thus, we update B_t and M_t, st. $b_i = b_i \cup b_j$, and meanwhile delete b_j and m_j.

3.3. Background Segmentation Based on Graph Cuts

We define the set of points that do not belong to the large background region as Ω. Then points of Ω can be judged as background region points or not based on the existing multiple background models. This paper proposes an energy minimization based algorithm for optimized classification. First, we define the scenario points belonging to $l = BNum + 1$ categories, where $BNum$ is the number of background models. We need to define and solve a label function $f : \Omega \to L$ where $L = \{0, 1, 2, ..., BNum\}$ are all the possible category labels for all the points in Ω. Label $i > 0$ corresponds to the background pixels, which are located in the *i*th background region. Label 0 corresponds to no background models, but corresponds to the foreground pixels. Given a pixel *p*, if $f(p) > 0$, it belongs to background region. Otherwise if $f(p) = 0$, this pixel belongs to foreground region. Energy function is as below:

$$E(f) = E_d(f) + E_s(f) \tag{8}$$

where data term E_d represents the sum of classification cost of the points in Ω classified into different labels. The smooth term is a regularizer that encourages the neighboring pixels to share the same label. Therefore, the classification problem is transformed to minimizing $E(f)$ and finding corresponding solution. However, minimizing $E(f)$ directly is very difficult because the above classification problem is the coupling of foreground and background, and background and background classification. This paper decomposes the above problem into two optimized solution modules $f = \{fs, fc\}$: (1) optimizing fs for background segmentation; (2) optimizing fc for classifying different background categories. In the first module, in order to segment the background regions, we transform this optimized classification problem to solve the binary energy minimization. If a pixel belongs to background, its label is 0, otherwise 1. The energy function includes a one variable data term and pairwise smoothing terms, where data term represents the cost of labeling the pixels to the background. The smoothing term corresponds to the continuous smoothness prior of the background region. The Graph Cuts [1,2] is adopted for optimizing and solving energy minimization problem. In the second module, the problem of classifying background points into different background models is transformed to a multi-labeling energy minimization problem, which can also be solved via Graph Cuts [1,2]. The data term of energy function represents the cost of tagging the points with the background labels. The smoothing term represents the background regions' continuity constraint.

3.3.1. Optimal Segmentation of Background Region

Following the above analysis, we need to seek a labeling function $fs : \Omega \to Ls$, $fs : \Omega \to Ls$. The background energy function is defined as follows:

$$E(fs) = E_d(fs) + E_s(fs) \tag{9}$$

- Data term

If a point belongs to the background region, it should be an inlier of one background model and its projection error corresponding to background model should be small, otherwise this point belongs to the foreground region and is the outlier to all the background models. Therefore, we choose the projection error to define the data term $E_d(fs)$:

$$E_d(fs) = \sum_{p\in\Omega} \left| fs(p) - (1 - Inl(p)) \right| \tag{10}$$

$$Inl(p) = \begin{cases} 1 & \sum_{i=1}^{BNum} IsI_i(p) > 0 \\ 0 & otherwise \end{cases} \tag{11}$$

$$IsI_i(p) = \begin{cases} 1 & Error_i(p) < T_e \\ 0 & otherwise \end{cases} \tag{12}$$

where $IsI_i(p)$ represents pixel p's inlier property projected in the model m_i. If the property is 1, this pixel belongs to the inliers, otherwise the outliers. $Inl(p)$ represents pixel p's background property. If property is 1, this pixel belongs to the background region, otherwise the foreground region. The penalty is given when pixel p is classified to the foreground point and $Inl(p) = 1$. The classified cost is not 0 and $fs(p) - (1 - Inl(p)) = 1$. Similarly, the classification penalty will also be given when the pixel p is classified to the background point and $Inl(p) = 0$.

- Smooth term

Smooth term $E_s(fs)$ is a regularizer that encourages the overall labeling smoothly [1,2]. The prior is that two neighboring pixels have a higher probability to be classified as background points together or foreground points together. Here, we adopt the standard four-connected neighborhood system and penalize the fact if two neighboring pixels' labels are different.

$$E_s(fs) = \sum_{p\in\Omega, q\in N_p} S_{p,q}(fs(p), fs(q)) \tag{13}$$

$$S_{p,q}(fs(p), fs(q)) = \min(\tau_s, \beta(p,q) \cdot \left| fs(p) - fs(q) \right|) \tag{14}$$

$$\beta(p,q) = h(\left|\min_{i=1}^{BNum}(Error_i(p)) - \min_{i=1}^{BNum}(Error_i(q))\right|) \tag{15}$$

where τ_s and $\beta(p,q)$ represent the maximum value and the weight of the smooth term, respectively. N_p is the four neighborhoods of pixel p. $h(\cdot)$ is the weight function. The weight function is a decreasing function because a big penalty should be given when the neighboring pixels are classified to different categories if their minimum projection errors are similar. When the minimum projection errors of two pixels are more similar, the weight is bigger and the smooth constraint is stronger. The inverse function $h(\cdot)$ is selected to achieve smooth constraint. If two neighboring pixels p and q share the same label, then $fs(p) = fs(q)$, $S_{p,q}(fs(p), fs(q)) = 0$. That is to say, the smooth cost is 0. Otherwise, if the neighboring pixels p and q are labeled with different tags, then the smooth cost $S_{p,q}(fs(p), fs(q)) > 0$. Just as defined in Equation (15), the closer the minimum projection errors of the two neighboring pixels, the bigger the smooth cost of labeling them with different tags.

Based on the design of above data term and smooth term, Graph Cuts is adopted to solve the minimization problem of $E(fs)$. Afterwards, background segmentation result is obtained.

3.3.2. Optimal Classification of Different Backgrounds

Denoting $\Omega_b = \{p :| fs(p) = 0\}$ as the set of points classified as the background pixels in Ω. We need to seek the labeling function $fc : \Omega_b \to Lc$, $Lc = \{1, 2, ..., BNum\}$. Similarly, we adopt the energy minimization framework for solving fc. The energy minimization of the background classification is defined as follows:

$$E(fc) = E_d(fc) + E_s(fc) \tag{16}$$

- Data term

Data term should reflect the subordinate degree between background pixel and multi-background models, and achieve minimum value if the pixel belongs to someone model. Projection error can satisfy above requests. Therefore, we define the cost function by using projection error as follows:

$$E_d(fc) = \sum_{p \in \Omega_b} Errop_{fc(p)}(p) \tag{17}$$

- Smooth term

Smooth term $E_s(fs)$ is a regularizer that encourages the overall labeling is smooth [1,2]. Similar with Section 3.3.1, we adopt the standard four-connected neighborhood system and penalize if the labels of two neighboring pixels are different.

$$E_s(fc) = \sum_{p \in \Omega_b, q \in N_p} S_{p,q}(fc(p), fc(q)) \tag{18}$$

$$S_{p,q}(fc(p), fc(q)) = \begin{cases} \min(\tau_s, \beta(p,q)) & fc(p) \neq fc(q) \\ 0 & otherwise \end{cases} \tag{19}$$

where τ_s and $\beta(p,q)$ have the similar definitions to Section 3.3.1. We also adopt Graph Cuts to minimize $E(fc)$ and obtain the segmentation of different backgrounds. The points of Ω_b are classified to the corresponding background blocks according to classified results of the label function fc.

$$b_i = b_i \cup \{\forall p \in \Omega_b, p :| fc(p) = i\} \tag{20}$$

3.4. Moving Object Detection

The pixels classified as foreground pixels may come from true moving object, and may also belong to false alarms of parallax error caused by buildings and others. How to distinguish these two category points is the key of segmenting moving object accurately. As we know, when a moving object is compensated by the background model, the parallax error only causes by the object itself, which represents the absolute motion vector of the object. Then the object motion between two neighboring frames is approximately the linear motion. As a result, the motion vectors of the inliers belonging to one object are similar. As shown in Figure 3, the motion vectors of true object in the red bounding box are similar. In contrast, the buildings do not belong to any background and all the existing background models cannot compensate the parallax error caused by the platform motion. Therefore, no matter if it is compensated by any one of the background models, parallax error distributes without dissimilarity, as the false alarm in the blue box of Figure 3.

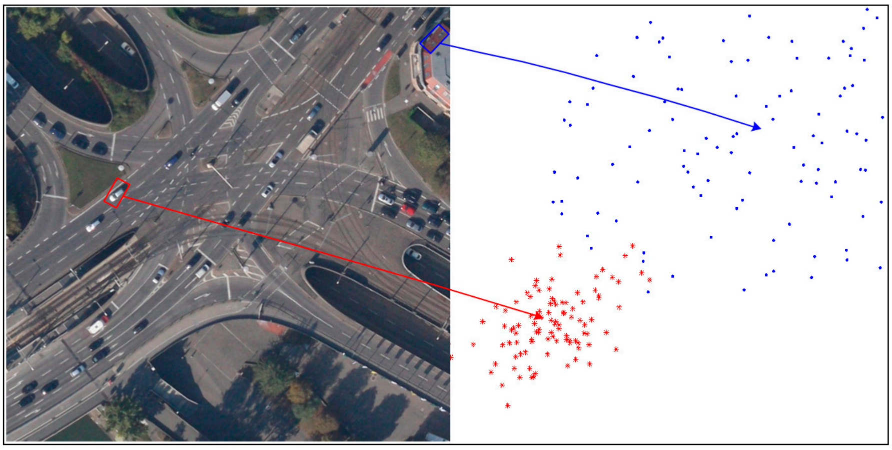

Figure 3. The distribution of motion vectors in blocks.

According to the above analysis, we will first calculate the motion vectors of foreground blocks compensated by the background model, and determine the moving objects by analyzing similarity of the motion vectors. The final foreground color blocks $Ob_t = \{ob_1, ob_2, ..., ob_{ObNum}\}$ can be obtained by integrating the foreground-background segmentation in Section 3.3 and the color segmentation in Section 3.2. Here, $ObNum$ is the color block number of the current foreground regions. We count the labeling set of the background models surrounding each color block and denote the labeling set as $MOb_t = \{mob_1, mob_2, ..., mob_{ObNum}\}$, where $mob_j = \{i :| \exists p \in ob_j, q \in b_i, p \in N_q\}$. The number of labels in mob_j is $MNum_j$. The motion vector of pixel $p(x, y)$ after compensated by model m_i is defined as follows:

$$\vec{v} = (x', y')^T - m_i \cdot (x, y, 1)^T \tag{21}$$

For each color block ob_j, we calculate the motion vector set compensated by each model, and count the mean value and variance of Gaussian distribution of the motion vector set. If the variance is small enough (T_σ is the variance threshold), this block is a moving block, otherwise a false object. The details of the procedure are in Algorithm 1.

Algorithm 1 False Object Removal Algorithm

```
Input: background model $T_\sigma$, motion block set $Ob_t$ and model label set $MOb_t$.
Output: motion blocks with false alarms removed.
FOR $j = 1 : ObNum$ DO
    FOR $i = 1 : MNum_j$ DO
        Computing the motion vectors compensated by corresponding models for the ith label, which belongs to labels of pixel $mob_j$ in $ob_j$
        Counting the mean value $\overrightarrow{\mu_{i,j}}$ and variance $\overrightarrow{\sigma_{i,j}^2}$ of these motion vectors.
    END
    If $\min_i |\overrightarrow{\sigma_{i,j}^2}| > T_\sigma$ or the area of $ob_j$ is larger than threshold $T_{a\max}$, $ob_j$ is the false alarm block and should be removed, otherwise $ob_j$ is background block and we define the label of minimum variance model as $j_m$. Then the background model label that $ob_j$ belongs to is also set to $j_m$. The moving speed of block $\overrightarrow{v_j} = \overrightarrow{\mu_{j_m,j}}$ and the corresponding variance $\overrightarrow{\sigma_j^2} = \overrightarrow{\sigma_{j_m,j}^2}$.
END
```

Therefore, we can obtain the foreground blocks by removing false alarm and updating Ob_t and $ObNum$. However, these blocks are segmented using the color consistence. Since the object color may be inconsistent, sometimes an object will be segmented into several blocks. To overcome this drawback, we need to merge these foreground blocks. We calculate the adjacent matrix $Nb_{ObNum \times ObNum} = \{nb_{jk}\}$ between moving blocks, where $nb_{jk} = 1$ represents that the *j*th and *k*th foreground block are neighborhood, *i.e.*, $\exists p \in ob_j, q \in ob_k, q \in N_p$. $nb_{jk} = 0$ indicates that two blocks are not neighborhood. Next we calculate the area sum matrix $A_{ObNum \times ObNum} = \{a_{jk}\}$ of moving blocks. If the sum of *j*th foreground block area and the *k*th foreground area $a_j + a_k < T_{a\max}$, then $a_{jk} = 1$, otherwise 0. We compute the speed similarity matrix $V_{ObNum \times ObNum} = \{v_{jk}\}$. If the background model label of *j*th and the *k*th foreground block are same as well as $\left|\overrightarrow{v_j} - \overrightarrow{v_k}\right|^2 < \left|\overrightarrow{\sigma_j}\right| \cdot \left|\overrightarrow{\sigma_k}\right|$, then $v_{jk} = 1$, otherwise 0.

If $\exists j, k$, $st.$ $n_{jk} \cdot a_{jk} \cdot v_{jk} = 1$, then we consider that the *j*th block and the *k*th foreground block belong to the same object and merge them to one. The new moving speed of the union foreground block is $\frac{a_j \cdot \overrightarrow{v_j} + a_k \cdot \overrightarrow{v_k}}{a_j + a_k}$ and the corresponding variance is recalculated. Nb, A and V also need to be recalculated.

Afterwards, merging is repeated until no foreground blocks can be merged. The final merged results are the moving object detection results.

4. Experimental Results and Analysis

In order to evaluate the proposed multi-model estimation based moving object detection algorithm for aerial video, we perform the comparison experiments on the public DAPAR VIVID (Defense Advanced Research Projects Agency, Video Verification of Identity program) and KIT AIS (Karlsruher Institut für Technologie Aerial Image Sequences) Data Set databases. In DAPAR VIVID database [26], the EgTest01 dataset contains many moving cars but the background is relatively simple. In KIT AIS Data Set [27], shooting frame rate is 1FPS and it includes viaducts, overpasses, buildings, trees and other complex scenarios, which is very challenging for the moving object detection algorithms of aerial video. The configuration of the computer used in our experiments is CPU Intel(R) Core(TM) 2 Duo 2.66 GHz, RAM 2.0 G. It takes about 4 s to process each frame for 724 × 708 image sequences. The most time-consuming step is the Mean shift segmentation, which takes about 3.5 s per frame. Dense optical flow calculation takes about 0.25 s and Graph cuts takes about 0.25 s. We have not done any acceleration. For practical application, parallel computing and other fast calculation method can be used to accelerate the segmentation and detection procedure. Our approach involves several parameters, including mainly the background threshold $T_{a\min}$, the object area threshold $T_{a\max}$, the projection error threshold T_e, the variance threshold T_σ and the smooth threshold τ_s. The color blocks with area larger than $T_{a\min}$ are considered to be background blocks. The smaller $T_{a\min}$ is set, the more background models we get, and the more complicated the multi-model estimation step is. The bigger $T_{a\min}$ is set, the more likely we miss some background models. In our experiments, we set $T_{a\min} = 6400$ to get a balance between the complexity and model number. $T_{a\max}$ is the max threshold for the object area. If it is set too small, then true object will be considered as small background blocks. Otherwise, the objects close to each other would be considered as one with large value for $T_{a\max}$. In our experiments, we set $T_{a\max} = 800$ to detect vehicles on the road. T_e is the projection error threshold. If a pixel's projection error for a given affine model is bigger than T_e, then it is considered to be an outlier for the model. Otherwise, if its projection error is smaller than T_e, it is an inlier for this model. The smaller T_e can bring more outliners and meanwhile cause more false alarms. The bigger T_e sometimes makes the algorithm miss true moving pixels. For the balance of false alarms and missing, we set $T_e = \sqrt{3}$ in our experiment. The variance threshold T_σ determines which foreground blocks are true object blocks and which blocks are false alarms. The smaller the value of T_σ, the fewer false alarms we detect and meanwhile the more likely we miss the true moving object. The larger T_σ would cause more false alarms. We set $T_\sigma = 4$ in our experiment for the best performance. The smooth threshold τ_s defines the max smooth cost of labeling two neighboring pixels with different tags. The larger τ_s brings a smoother labeling map and object missing is more likely to occur. The small τ_s decreases the smoothing effect. We set $\tau_s = 4$ in our experiments.

The detection method in [14] is the most representative method in which Harris features are abstracted for registration and frame difference is used to detect moving objects. Shen *et al.* [20] proposed a moving object detection method for aerial video based on spatiotemporal saliency. This method can accurately handle moving target detection under simple scenarios. However, it has not adopted multiple background analysis for the scenarios, and detection missing and false alarms will happen frequently in complex scenarios. As there are no published codes for the approach in [14,20] on the web, we implement these two approaches for comparison.

We compare our algorithm with the method in [14] and Shen [20] on the StuttgartCrossroad01 dataset of KIT AIS Data Set. The results are shown in Figure 4. This dataset contains overpasses and multiple background complex scenarios as well as complex elements, such as trees and shadow, which will influence the detection results. All the factors will bring substantial challenge to the detection algorithms. In Figure 4, the images from top to bottom show the detected results of the 1st, 5th, 9th and 12th frames. The images from left to right are separately, the detection results of this paper, the segmentation results of this paper, the detection results by [14], the detection results of Shen [20], and the ground truth. In the first column of Figure 4, the objects in blue boxes are the detection results of this paper. The objects in red boxes are stationary targets. The detection results show our approach can segment and detect moving objects accurately in the complex background situation of overpasses. Since both of the approaches, in [14,20], cannot perceive multiple background of the scenario and cannot obtain accurately background information, the situations such as inaccurate moving segmentation and false alarms will happen. We can see these situations in the third and fourth column of Figure 4. The blue bounding boxes show the detected objects. The yellow boxes show the false detection and missing targets. Although, the method in [20] performs better than the method in [14], false alarms and inaccurate detections occur frequently in both of these two methods. The ground truth published on the web marks all the vehicles in the scene, including both moving objects and stationary vehicles.

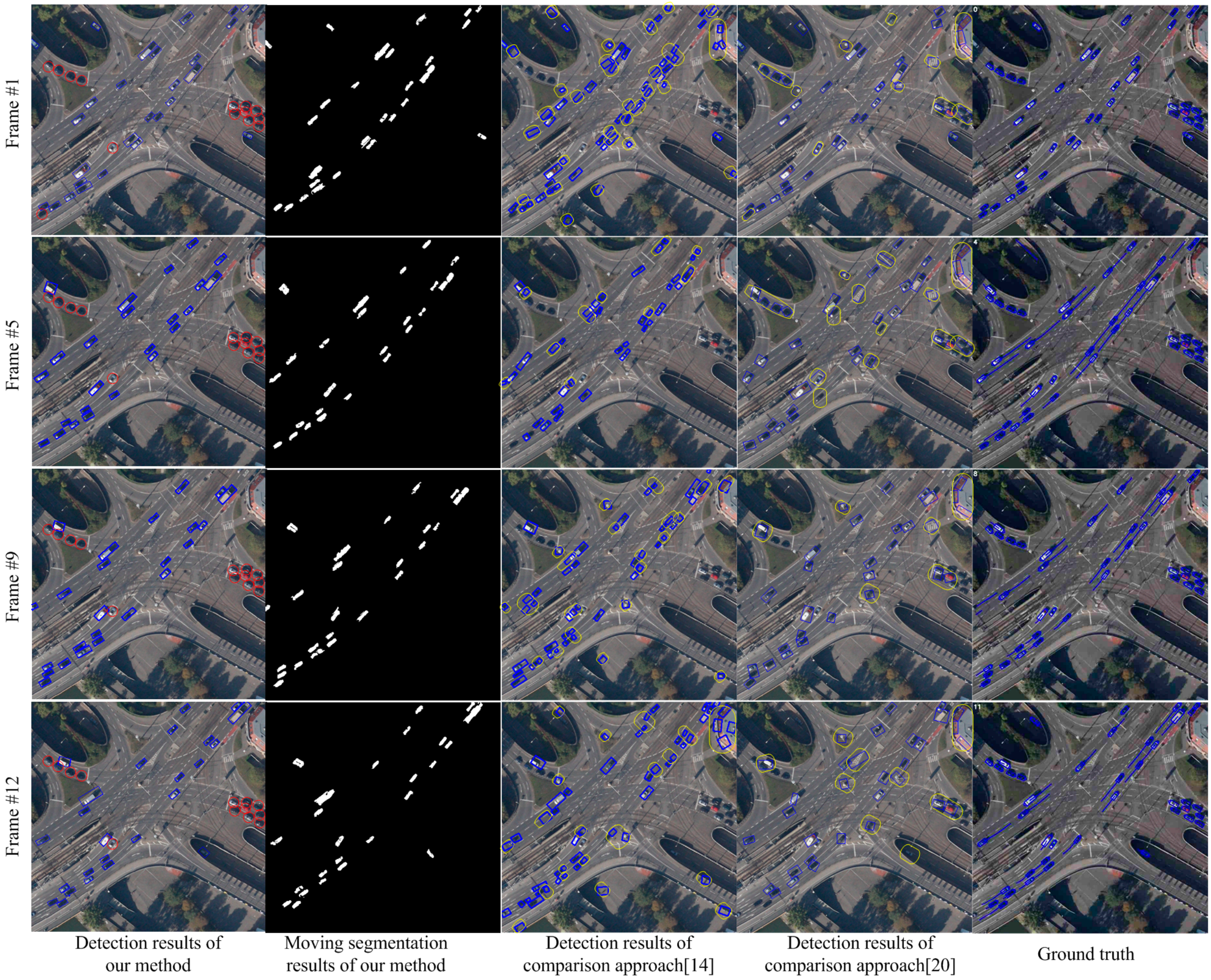
Frame #1
Frame #5
Frame #9
Frame #12
Detection results of our method
Moving segmentation results of our method
Detection results of comparison approach[14]
Detection results of comparison approach[20]
Ground truth

Figure 4. Detection comparisons in complex overpass scenarios.

Figure 5 shows the comparison results on the MunichCrossroad01 dataset. The characteristic of this dataset is that the false objects of parallax error caused by the trees and other elements occupy a large proportion of the image area. In Figure 5, the images from top to bottom show the detection results of the 1st, 7th, 13th and 18th frames. The results in the first column show that our approach can handle the moving object detection in scenarios with many trees and overcome the parallax error caused by trees. In contrast, the traditional detection methods [14,20] based on registration will be influenced by trees and cannot estimate the scene model accurately. Therefore, as shown in the third and fourth columns of Figure 5, the detection rate of traditional method is low and the false alarm is high. Figure 6 shows the detection results on Munich Crossroad02 dataset. This dataset includes many buildings. The transitional methods [14,20] cannot accurately estimate the background parameters and obtain the correct detection and segmentation results in this situation. As shown in the third and fourth columns of Figure 6, many false alarms and missing detections occur. In contrast, the results in the first and second columns show the detection and segmentation results of our paper. The results demonstrate our approach can perceive scenarios and detect moving objects correctly due to multiple background model estimation.

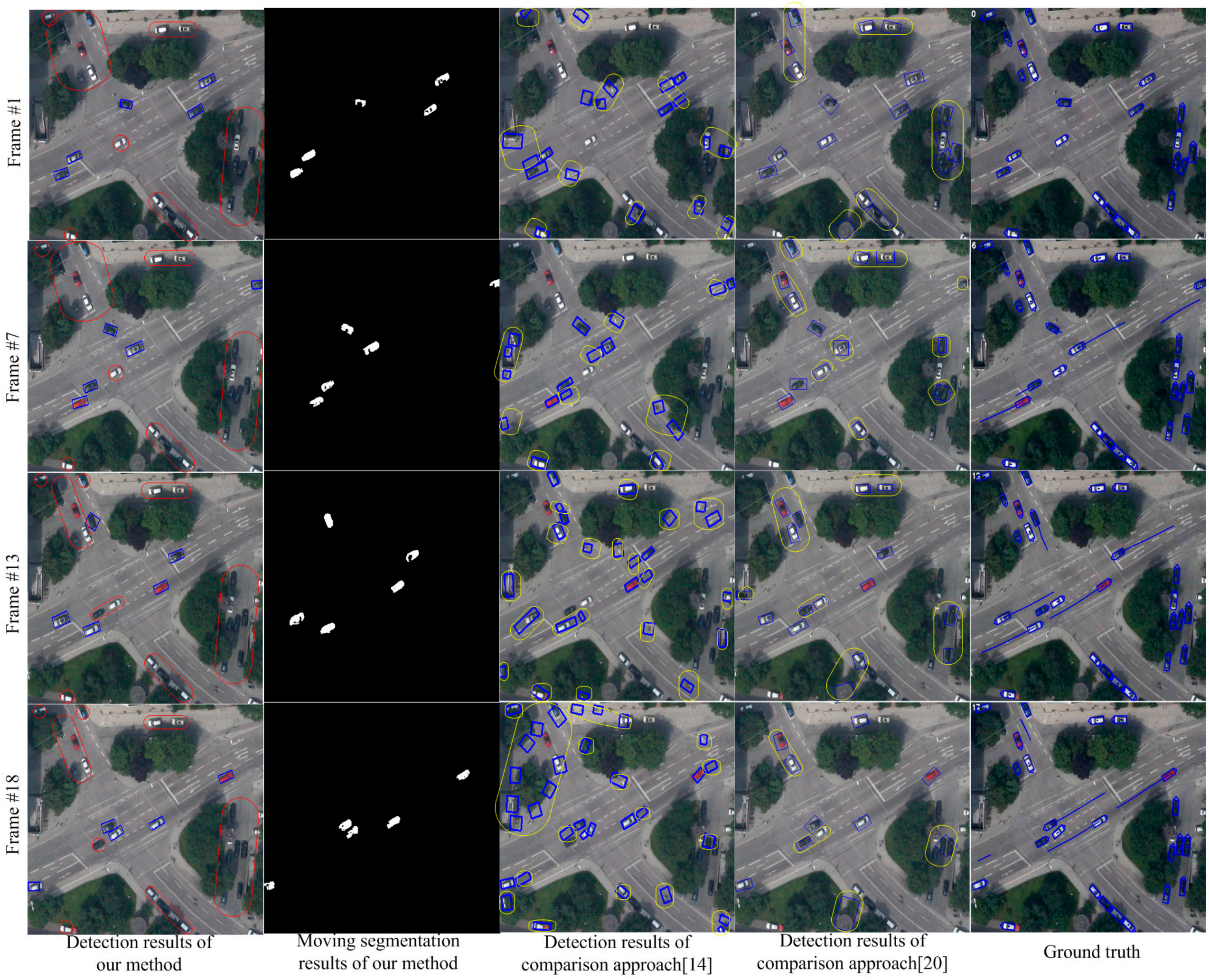
Frame #1
Frame #7
Frame #13
Frame #18
Detection results of our method
Moving segmentation results of our method
Detection results of comparison approach[14]
Detection results of comparison approach[20]
Ground truth

Figure 5. Detection comparisons in scenarios with many trees.

As shown in Figures 4–6, this paper performs much better than traditional detection algorithm basing on registration. Our approach can analyze the multiple background models in scenarios and detect the moving objects accurately. However, since we adopt mean shift color segmentation and pyramid dense optical flow to perceive the multiple background models, the algorithm's efficiency still needs to be improved and more efficient multiple background model estimation algorithms are required. Additionally, this paper focuses on vehicle-sized objects and cannot detect the point objects like humans. We also do not add any special treatment for shadow, so the moving objects after segmenting may contain shadow, which is also the future work.

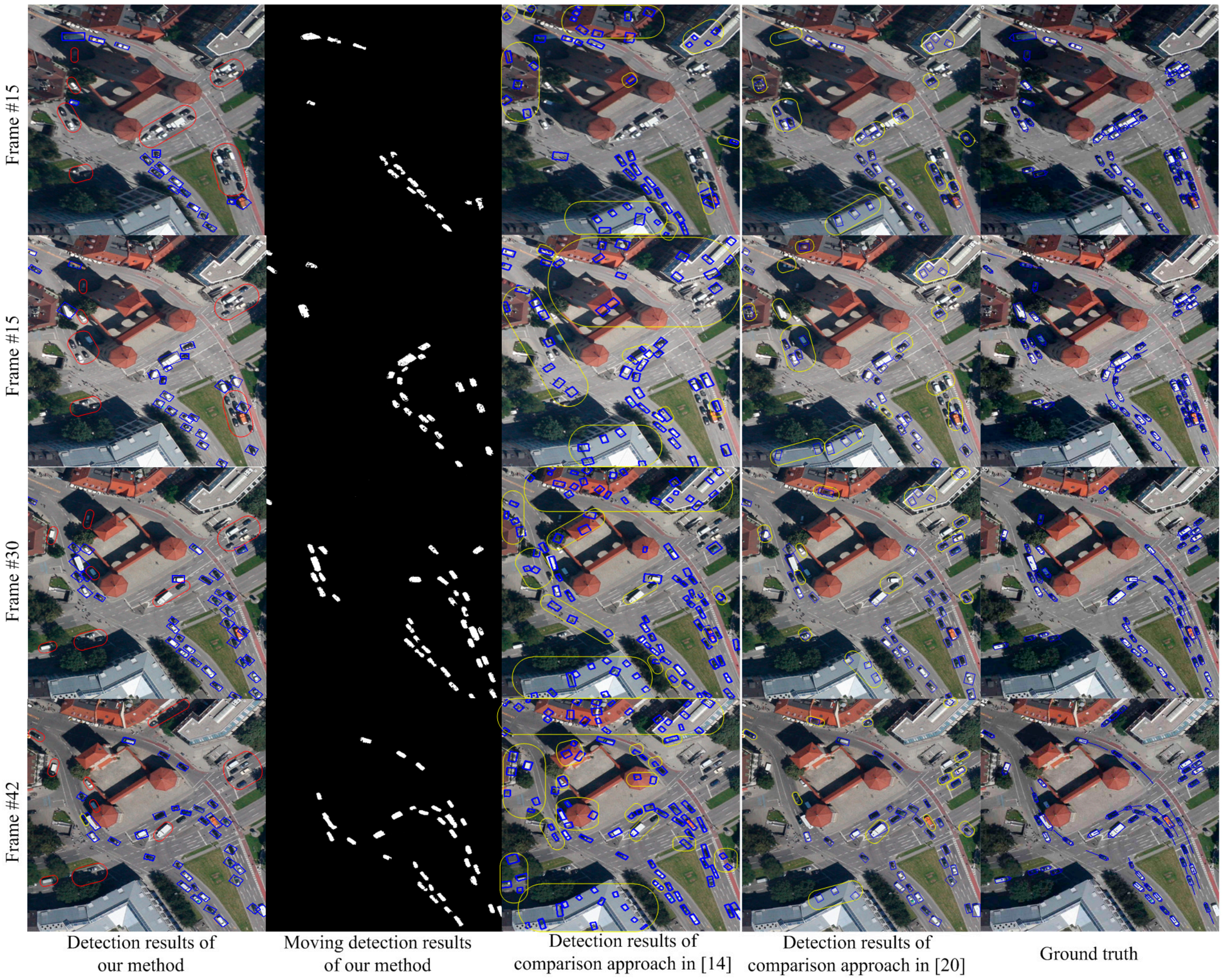
Frame #15
Frame #15
Frame #30
Frame #42
Detection results of our method
Moving detection results of our method
Detection results of comparison approach in [14]
Detection results of comparison approach in [20]
Ground truth

Figure 6. Detection comparisons in scenario with many buildings.

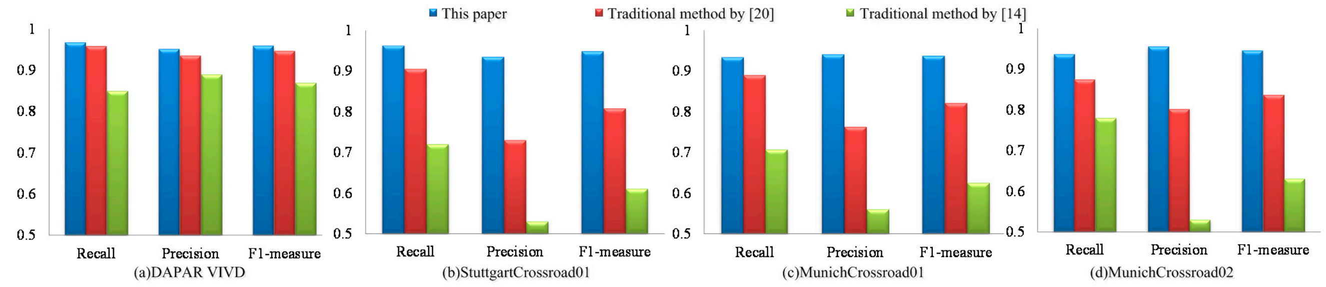

Figure 7. The statistical result of our method and the traditional methods by [14,20].

In order to quantitatively analyze the detection accuracy of this paper, we define recall R, accuracy P and comprehensive evaluation indicators $F1$ as follows:

$$R = DObNum / ObNum \tag{22}$$

$$P = DObNum / DNum \tag{23}$$

$$F1 = 2PR / (P + R) \tag{24}$$

where $ObNum$, $DObNum$ and $DNum$ are the object number of the ground truth, correct detection and detected object number, respectively. If a detected object's overlap rate with a true object is above 0.5, then it is considered as a correct detection. Otherwise, it is a false alarm. In practical applications, higher *R* and *P* are desired, but these two indicators are contradictory in some cases. $F1$ integrates the results of *R* and *P*. Higher $F1$ indicates that the experimental method is more effective. Figure 7 shows the comparison results of our paper, the traditional method by [20] and the traditional method by [14]. The results from left to right are the statistical results of DAPAR VIVID EgTest01, StuttgartCrossroad01, MunichCrossroad01 and MunichCrossroad02 of KIT AIS Data Set. As shown in Figure 7a, these three algorithms can both achieve high detection rate under simple background and their detection precisions are similar. However, $F1$ of our algorithm under complex background is higher than the methods in [14,20], *i.e.*, on StuttgartCrossroad01 dataset, $F1$ of our result is 0.949, which is higher than 0.808 of Shen [20] and 0.611 of the method in [14]. In MunichCrossroad01 dataset, our approach's $F1$ is 0.937, which is higher than 0.821 of Shen [20] and 0.625 of the method in [14]. These results show the significant superiority of our algorithm, as shown in Figure 7b–d.

5. Conclusions

This paper is mainly for the moving object detection problem under complex scenarios for aerial videos. We propose a novel moving object detection algorithm based on multi-model estimation and optimized classification. First, we calculate the dense optical flow of the scene and do color segmentation basing on mean shift to capture the perception of the whole scene. Secondly, we calculate affine transformation models as the multiple background models for each color block with a large area. Through multiple background model cross-validation and merger, accurate multi-model parameters of scene can be obtained. Thirdly, in order to obtain the multiple background segmentation results of the scene, the background points are segmented into multiple background models by using energy optimization method solved via Graph Cuts. Finally, we calculate subordinate degree from foreground regions to multi-background models, remove the false alarm and segment moving object accurately.

Since we break through the single background constraint and adopt multiple background models, our algorithm can handle the moving object detection under complex multiple background scenarios. Moreover, our algorithm can segment the background and foreground regions accurately due to the adoption of Graph Cuts, optical flow information and continuous smooth constraints. The experimental results on many aerial videos indicate that our algorithm can correctly perceive multiple background information of the scene and detect moving object accurately in the complex scenes with multiple backgrounds, buildings and other objects that produce parallax.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61231016, No. 61272288, No. 61303123), the National Natural Science Foundation of China Youth Science Fund

Project (No. 61301193), Foundation of NPU New Ao Xiang Star (No. G2015KY0301), New People and New Directions (No. 13GH014604), the NPU Foundation for Fundamental Research (No. JC201120, No. JC201148, No. JCT20130109), Plan of Soaring Star of Northwestern Polytechnical University (No. 12GH0311), and Foundation of China Scholarship Council (No. 201303070083).

Author Contributions

Yanning Zhang designed the proposed UAV moving object detection algorithm, conducted the analysis of the algorithm and co-wrote the initial manuscript. Xiaomin Tong conducted the analysis of the proposed UAV moving object detection algorithm, designed and performed experiments, and co-wrote the initial manuscript. Tao Yang and Wenguang Ma performed experiments and contributed to the comparison experiment and writing the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Boykov, Y.; Kolmogorov, V. An Experimental Comparison of In-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137.
2. Boykov, Y.; Veksler, O.; Zabih, R. Efficient Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *20*, 1222–1239.
3. Yu, Q.; Medioni, G. A GPU-based implementation of Motion Detection from a Moving Platform. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008; pp. 1–6.
4. Hou, Z.Q.; Han, C.Z. Video Tracking Technology Overview. *Automatica Sinica*. **2006**, *32*, 603–517.
5. Yilmaz, A.; Javed, O.; Shah, M. Object Tracking: A Survey. *ACM Comput. Surv.* **2006**, *38*, doi:10.1145/1177352.1177355.
6. Zhu, S.; Wang, D.; Low, C.B. Ground Target Tracking Using UAV with Input Constraints. *J. Intell. Robot. Syst.* **2013**, *69*, 417–429.
7. Owen, M.; Yu, H.; McLain, T.; Beard, R. Moving Ground Target Tracking in Urban Terrain Using Air/Ground Vehicles. In Proceedings of the 2010 IEEE GLOBECOM Workshops, Miami, FL, USA, 6–10 December 2010; pp. 1816–1820.
8. Radhakrishnan, G.S.; Saripalli, S. Target Tracking with Communication Constraints: An Aerial Perspective. In Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE), Phoenix, AZ, USA, 15–16 October 2010; pp. 1–6.
9. Wang, J.; Zhang, Y.; Lu, J.; Xu, W. A Framework for Moving Target Detection, Recognition and Tracking in UAV Videos. *Adv. Intell. Soft Comput.* **2012**, *137*, 69–76.
10. Fu, X.; Feng, H.; Gao, X. UAV Mobile Ground Target Pursuit Algorithm. *J. Intell. Robot. Syst.* **2012**, *68*, 359–371.
11. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. *Alvey Vision Conf.* **1988**, *15*, 50.

12. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395.
13. Mann, S. Compositing Multiple Pictures of the Same Scene. In Proceedings of the 46th Annual IS&T Conference, Boston, MA, USA, 9–14 May 1993.
14. Ali, S.; Shah, M. COCOA: Tracking in Aerial Imagery. *Proc. SPIE* **2006**, doi:10.1117/12.667266.
15. Cohen, I.; Medioni, G. Detecting and Tracking Moving Objects for Video Surveillance. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June 1999; pp. 319–325.
16. Medioni, G.; Cohen, I.; Bremond, F.; Hong, S.; Nevatia, R. Event Detection and Analysis from Video Streams. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 873–889.
17. Ibrahim, A.W.N.; Pang, C.W.; Seet, G.L.G.; Lau, W.S.M.; Czajewski, W. Moving Objects Detection and Tracking Framework for UAV-based Surveillance. In Proceedings of the 2010 Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT), Singapore, 14–17 November 2010; pp. 456–461.
18. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV), Kerkyra, Greece, 20–25 September 1999; Volume 2, pp. 1150–1157.
19. Aeschliman, C.; Park, J.; Kak, A.C. Tracking Vehicles through Shadows and Occlusions in Wide-Area Aerial Video. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 429–444.
20. Shen, H.; Li, S.X.; Zhu, C.F.; Chang, H.X.; Zhang, J.L. Moving Object Detection in Aerial Video Based on Spatiotemporal Saliency. *Chin. J. Aeronaut.* **2013**, *26*, 1211–1217.
21. Aristodemos, P.; Lazaros, P. 2D Person Tracking Using Kalman Filtering and Adaptive Background Learning in a Feedback Loop. *Lect. Notes Comput. Sci.* **2007**, *4122*, 151–160.
22. Reshma, A.R.; Anooja, S.; George, D.E. Bearing Only Tracking Using Extended Kalman Filter. *Int. J. Adv. Res. Comput. Commun. Eng.* **2013**, *2*, 1140–1144.
23. Smith, K.; Gatica-Perez, D.; Odobez, J.M. Using Particles to Track Varying Numbers of Interacting People. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 962–969.
24. Gunnar, F. Two-Frame Motion Estimation Based on Polynomial Expansion. In Proceedings of the 13th Scandinavian Conference on Image Analysis, Halmstad, Sweden, 25 June 2003; pp. 363–370.
25. Toldo, R.; Fusiello, A. Robust Multiple Structures Estimation with J-Linkage. In Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 12–18 October 2008; pp. 537–547.
26. VIVID Tracking Evaluation Web Site. Available online: http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html (accessed on 8 April 2015).
27. KIT AIS Data Set. Available online: http://www.ipf.kit.edu/downloads.php (accessed on 8 April 2015).