Minimum Delay Moving Object Detection

Dong Lao and Ganesh Sundaramoorthi
King Abdullah University of Science & Technology (KAUST), Saudi Arabia
{dong.lao, ganesh.sundaramoorthi}@kaust.edu.sa

Abstract

We present a general framework and method for detection *of an object in a video based on apparent motion. The object moves relative to background motion at some unknown time in the video, and the goal is to detect and segment the object as soon it moves in an online manner. Due to unreliability of motion between frames, more than two frames are needed to reliably detect the object. Our method is designed to detect the object(s) with* minimum *delay, i.e., frames after the object moves, constraining the false alarms. Experiments on a new extensive dataset for moving object detection show that our method achieves less delay for all false alarm constraints than existing state-of-the-art.*

1. Introduction

Detection and segmentation of object(s) from video are fundamental problems in computer vision. Motion cues play a role in biological visual systems, and may be useful for object segmentation in both biological and computer vision systems [16]. Thus, there have been many works that segment a video by *apparent motion*, i.e., motion induced in the image, in an attempt to segment relevant objects (e.g., [34, 35, 27, 16]). With abuse of nomenclature, we will refer to apparent motion as motion from now on. In these methods, it is assumed that the video is obtained when the objects of interest are already in motion relative to the background. However, that may not be representative of the problem solved by biological systems [5] or that is required in certain computer vision applications such as robotics or surveillance. In such cases, the object may be stationary or out of view of the observer or otherwise have apparent motion indistinguishable from the background when the video starts. Thus, *detection* of the object at the time it moves is needed before segmentation [1].

This paper addresses the problem of *detection* and segmentation of an object by motion in a video. The object moves, at some *unknown* time, differently than the "background", induced from camera motion. Since we eventually aim for real-time closed loop operation (e.g., robotic systems), an *online* algorithm is desired. We define an online system as a system that receives frames sequentially, one at a time, and must make a decision, that is, declare a detection or wait for more data, at each time instant. Observing more frames before declaring a detection may lead to a more accurate detection and segmentation, since more motion may be observed leading to a stronger motion cue. However, this leads to greater delay, which may not be tolerable in closed-loop systems. Thus, our goal is to derive an algorithm with *minimum delay*, defined as the number of frames acquired after the object moves. Of course zero delay can be achieved by always declaring detection at frame 1, irrespective of the data. Thus, we require an algorithm that operates under a constraint on *false alarms*, defined as declarations of detection before the object moves or incorrect or inaccurate segmentation at the detection time.

Quickest Detection (QD) [21, 33, 23, 20] is a theory for reliably detecting changes in an online fashion from a stochastic process with minimum delay. The stochastic process arises from a certain probability distribution before an unknown change time, and a different distribution after the change time. QD theory derives online algorithms to detect the change time with *minimal* delay subject to false alarm constraints. Since our problem of moving object detection resembles the Quickest Detection problem, we build on the techniques in that literature.

In this paper, **1.** We introduce a general framework and the first online algorithm that guarantees *reliable* detection of an object based on motion while *minimizing* the detection delay. To achieve this, we derive statistical models of image sequences, the objects within images, their motions, and occlusion phenomena. We achieve reliable detection by integrating motion over frames, and minimal delay by using the statistical models to formulate the problem as a QD problem. **2.** We derive a new motion segmentation approach by integrating motion from multiple frames, as a sub-problem for our detection scheme. **3**. We provide approximate algo-

[1] We define *detection* as the problem of determining the existence of an object and declaring the first frame when it is in motion. We define *segmentation* as the problem of marking the pixels of the object. For us, an object corresponds to a smooth surface in the scene.

rithms for moving object detection to decrease the computational cost of algorithms from QD. **4**. Finally, we quantitatively evaluate the algorithm on a new extensive benchmark dataset for moving object detection and compare it to existing state-of-the-art in terms of minimizing delay under false alarm constraints.

1.1. Related Work

Our detection method requires motion segmentation, and thus we briefly review that literature. Motion segmentation relies on computing apparent motion, determined from parametric models (e.g., affine) of flow [14] or dense optical flow [3, 4, 39, 10, 25]. Early works (e.g., [34, 35, 17, 7, 32]) on motion segmentation use parametric models of flow and solve a joint problem in segmentation and flow. To deal with deforming objects, non-parametric motion models of flow are used (e.g., [36, 26, 1, 9]), and solved as a joint problem of segmentation and flow estimation. [27, 38] use a similar approach to causally segment videos frame-by-frame. Those approaches typically operate on 2-3 frames. To obtain a stronger motion signal, whole videos are processed in batch [16, 12, 18], rather than online. [16, 12] group trajectories of points across frames in batch to perform segmentation. Dense motion has been computed across many frames in [22], but not for segmentation. Instead of batch processing to integrate motion cues over time, [31] integrates occlusion cues [24, 30, 19] over frames causally. Batch segmentation methods [16, 12] may achieve a stronger motion signal at the expense of processing the whole batch. Our algorithm chooses the fewest number of frames to achieve a strong enough motion signal for reliable detection and segmentation. Existing approaches for motion segmentation typically assume motion from the start and do not address *detection* of a moving object at an unknown time in the video, our main motivation.

The problem of detecting changes (not necessarily moving objects) in a video has a large literature in computer vision [11]. That literature addresses detection and segmentation of objects by background subtraction (e.g., [15, 2]). Those methods do not apply to our problem since we assume moving cameras. While there are methods that deal with dynamic cameras and detect and segment moving objects by motion (e.g., [6]), they do not address the issue of the tradeoff between detection delay and false alarms.

2. Models for Object Detection

In this section, we present our statistical models to frame minimum delay object detection as a Quickest Detection problem. We assume that the scene is observed by a possibly moving observer, i.e., the "background" may be moving and at some time Γ, object(s) within the scene begin to move or come into view of the camera. We refer to Γ as the *change time*. Let $\Omega \subset \mathbb{R}^2$ be the domain of the images, and let $I_t : \Omega \to \mathbb{R}^k$, $t \geq 1$ be the image sequence, where t will denote the frame number and $k = 3$ denotes the color channels. We denote n regions $R_t^i \subset \Omega$ for $i = 0, \ldots, n-1$ corresponding to moving objects (from smooth surfaces in the scene) and R_t^0 will correspond to the background, which form a segmentation of I_t. We denote $O_t^i \subset R_t^i$ as the occlusion of region R_t^i induced by a change of viewpoint of the camera or a self-occlusion between time t and $t+1$. See Figure 1 for a schematic.

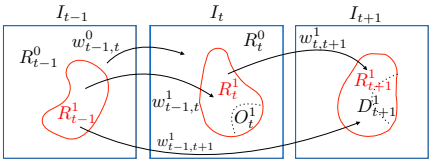

Figure 1: *Schematic of quantities in image / region models.*

Region and displacement models: The *displacement* between adjacent frames for region i is $v_t^i : R_t^i \backslash O_t^i \to \mathbb{R}^2$. Although such displacements are not defined in the occluded parts of the domain, they will be smoothly extended into the occlusion and the entire domain Ω. We denote the *warp* between frames t and $t+1$ as $w_{t,t+1}^i : R_t^i \backslash O_t^i \to R_{t+1}^i$, defined by $w_{t,t+1}^i(x) = x + v_t^i(x)$, which are diffeomorphisms that arise from change of viewpoint or deforming objects. The warp between time 1 and time t will be denoted by w_t^i. This is obtained by composing the warps (see Figure 2), and is determined recursively from

$$w_{t+1}^i(x) = w_t^i(x) + v_t^i(w_t^i(x)), \ t > 1 \quad \text{with} \quad w_0^i(x) = x. \tag{1}$$

Our model for the evolution of the regions across time is that the unoccluded part of the regions is propagated via the warps and concatenated with the disocclusion (part coming into view), $D_{t+1}^i \subset \Omega$, at time $t+1$, as:

$$R_{t+1}^i = w_{t,t+1}^i(R_t^i \backslash O_t^i) \cup D_{t+1}^i \quad R_0^i = R^i, \tag{2}$$

where R^i are the initial regions. Therefore, the region of the object is a smooth warping of an initial region (up to disocclusions), and therefore smoothly varies in time.

Image sequence model: Assuming approximate Lambertian reflectance of the scene, we may relate successive images before the change as

$$I_{t+1}(w_{t,t+1}^0(x)) = I_t(x) + \eta_t(x), \ x \in \Omega \backslash O_t^0, \ t < \Gamma. \tag{3}$$

That is, the sequence is described by one smooth warp. After the change, the image in each region R_t^i is related by

$$I_{t+1}(w_{t,t+1}^i(x)) = I_t(x) + \eta_t(x), \ x \in R_t^i \backslash O_t^i, \ t \geq \Gamma \tag{4}$$

where $\eta_t(x)$ is a Gaussian independent noise process in both t and x, used to model deviations from the Lambertian assumption. We assume $\eta_t(x) \sim \mathcal{N}(0, \sigma_{\eta,i})$ for $x \in R_t^i \backslash O_t^i$.

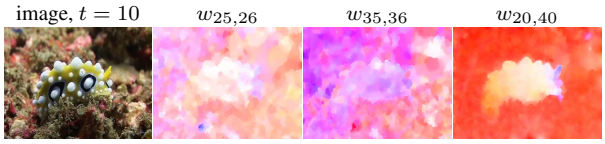

Figure 2: **Composition of warps across frames produces a strong motion signal.** *[Left]: Image. [Middle two]: Optical flows between adjacent frames at two instances show that the object is not clearly visible. [Right]: Composition of warps between frames 10 and 40 shows the object is clearly visible. How many frames does it take to reliably detect the moving object? Our method addresses this question.*

Likelihoods: We specify the pre- and post-change likelihoods based on the models above. These will be necessary for our moving object detection algorithm. Let $\mathbf{I}_{t_1:t_2}$ denote all the images $I_{t_1}, I_{t_1+1}, \ldots, I_{t_2}$, and similarly define $\mathbf{v}^i_{t_1:t_2}$ and $\mathbf{R}^i_{t_1:t_2}$ as displacements and regions across time. Conditional on v^i_t and R_t, the pairs of images I_t, I_{t+1} are independent for all t. Using this, the pre-change probability is $p_0(\mathbf{I}_{i:i+1}|v^0_i) \propto$

$$\exp\left\{-\int_\Omega \rho_0[I_{t+1}(w^0_{t,t+1}(x)) - I_t(x)]\,\mathrm{d}x\right\}, \quad (5)$$

where $\rho_i(y) = \frac{1}{2\sigma^2_{\eta,i}} \min\{|y|^2, \beta\}$ ($\beta > 0$) is a truncated quadratic, which is a robust norm that eliminates the explicit estimation of the occlusion [25]. Similarly, the post-change conditional distribution is $p_1(\mathbf{I}_{t:t+1}|\{v^i_t, R^i_t : 0 \leq i < n\}) \propto$

$$\exp\left\{-\sum_{i=0}^{n-1}\int_{R^i_t} \rho_i[I_{t+1}(w^i_{t,t+1}(x)) - I_t(x)]\,\mathrm{d}x\right\}. \quad (6)$$

3. Multiframe Motion Segmentation

In this section, we present an algorithm for motion segmentation assuming the object(s) are in motion. This is a sub-problem implied by our detection method as shown in Section 4. Unlike other approaches, our segmentation algorithm *composes* motion across multiple frames, which provides a stronger motion signal (Figure 2) than motion computed from adjacent frames, leading to better segmentation.

Given frames $I_{t_1}, I_{t_1+1} \ldots, I_{t_2}$, we segment each frame based on the post-change models described in Section 2. To do so, we maximize the post-change likelihood over the regions $\mathbf{R}^i_{t_1:t_2}$. Since $\mathbf{v}^i_{t_1:t_2}$ are also unknown, they are estimated jointly by maximizing the same likelihood. By assuming independence of v^i_t for $t_1 \leq t \leq t_2$, maximizing the post-change likelihood $p_1(\mathbf{I}_{\mathbf{c}:\mathbf{t}}|\mathbf{R}^i_{t_c:t}, \mathbf{v}^i_{t_c:t})$ is equivalent to minimizing

$$\sum_{t=t_1}^{t_2}\sum_{i=0}^{n-1}\int_{R^i_t} \rho_i[I_{t+1}(w^i_{t,t+1}(x)) - I_t(x)]\,\mathrm{d}x. \quad (7)$$

The independence assumption on the displacements is for speed in the segmentation. Any noise in any one estimate of the displacement from a pair of images is mitigated by the integration in cumulative warps in (1). Below we describe our joint region and warp estimation algorithm.

Warp estimation: Given the regions, we discuss the optimization in $\mathbf{v}^i_{t_1:t_2}$. Note there are no explicit smoothness priors on the warps, and thus no regularization of the warps appear in (7). Instead, we leverage on the Sobolev framework [37, 29], to impose regularity in the optimization in a coarse-to-fine fashion. This avoids the under/over smoothing problem in global regularization used in optical flow and parameter tuning. Optimizing for v^i_t results in

$$v^t_t = \arg\min_v \int_{R^i_t} \rho_i[I_{t+1}(x + v(x)) - I_t(x)]\,\mathrm{d}x, \quad (8)$$

as other terms are independent of v^i_t. This is extended to form a smooth warp on all of Ω.

Energy for region estimation: Given the warps, we optimize (7) for $\mathbf{R}^i_{t_1:t_2}$. Note that R^i_t are coupled through (2) across frames, imposing regularity over time. In the case of no disocclusions, optimizing in $\mathbf{R}_{t_1:t_2}$ can be replaced by optimization in regions R^i_s at one time with $t_1 \leq s \leq t_2$ subject to the constraint that the other regions are warps of R^i_s. Let $w^i_{s,t}$ denote the warp of R^i_s to R^i_t, determined by composition (1) and the given estimates of $\mathbf{v}^j_{t_1:t_2}$. Define

$$f^i(x) = \sum_{t=t_1}^{t_2} \rho_i[I_{t+1}(w^i_{s,t+1}(x)) - I_t(w^i_{s,t}(x))] \det \nabla w_{s,t}(x), \quad (9)$$

where $\nabla w_{s,t}$ denotes the Jacobian of the warp. To determine the R^i_s, we optimize

$$E_{seg}(\{R^i_s\}_{i=0}^{n-1}) = -\sum_{i=0}^{n-1} \log\,[p(R^i_s)] +$$
$$\sum_{i=0}^{n-1}\int_{R^i_s} [1 - m(x)] f^i(x) - m(x) \log p_{R^i_s}(I_s(x))\,\mathrm{d}x, \quad (10)$$

where $p(R^i_s)$ is the prior probability of the region encoding a smoothness prior (standard boundary length regularization), $p_{R^i_s}$ are local color histograms of I_s within regions, and $m : \Omega \to [0, 1]$ is the *motion ambiguity function*. If m is 0 everywhere and the region prior is excluded, then the energy is (7) (after re-ordering the summations and performing a change of variables to the domain of R^i_s). As motion estimated in textureless parts of regions or in occlusions is unreliable for segmentation, the motion ambiguity function is used to switch between using motion cues and color histograms for grouping. The motion ambiguity function is 1 if pixel $x \in R_s$ is occluded in *all* frames $t = t_1, \ldots, t_2, t \neq s$,

that is, all summands in (9) exceed β, or if x is in a textureless sub-region of I_s defined by small standard deviations within R_s^i local to x.

Joint Region and Warp Estimation: The energy (10) now fits into a form considered in [28]. Thus, we use the optimization specified there, which uses gradient descent due to non-convexity. Even though we assumed no disocclusions, optimization of (10) implicitly computes disocclusion as part of the grouping procedure. Disocclusions at time s are assumed to be parts of the image moving similar to R_s (or similar color intensity in the case of motion ambiguity). Although the optimization problem is in R_s^i, each of R_r^i for $t \in \{t_1, \dots, t_2\} \backslash \{s\}$ can be determined by propagating R_s through the sequence determined by warps $w_{i,i+1}^j$, which when done through [38], includes disocclusion. This yields Algorithm 1.

Initialization: Optimization of E_{seg} requires initialization for R_s^i. We initialize it with a segmentation of the cumulative displacement from frame s to t_2 (and frame s to t_1). Both forward and backward warps are used to address incorrect grouping due to occlusion. Because accurate warp estimation requires a segmentation, which is unknown at initialization, we use Classic-NL [25], robust to motion discontinuities, to approximate the warps frame-to-frame without a segmentation. The cumulative displacement is then computed using (1). The segmentation of the cumulative warps is done by detecting edges [8], then generating the segmentation, which sets the number of regions, n.

Example intermediate results of the algorithm are shown in Figure 3.

Algorithm 1 *Multiframe motion segmentation*

Input: $\mathbf{I}_{t_1:t_2}$ and $s \in [t_1, \dots, t_2]$
// initialize R_s^i for gradient descent of E_{seg}
Compute Classic-NL warp $w_{t,t+1}^{NL} : \Omega \to \Omega$, $\forall t$
Compute $w_{s,t_1}^{NL}, w_{s,t_2}^{NL}$ by composing warps (1)
Use $w_{s,t_1}^{NL}, w_{s,t_2}^{NL}$ as channels to segment using [8]
repeat *// gradient descent of E_{seg} in (10) for R_s^i*
    Propagate R_s^i frame-wise to form R_t^i, $\forall t$ via [38]
    Solve for Sobolev warp $w_{t,t+1}^i$ via (8)
    Compute $w_{s,t}^i$ for $t \in \{t_1, \dots, t_2\} \backslash \{s\}$ using (1)
    Compute f^i and update R_s^i by gradient step of E_{seg}
until R_s^i does not change between iterations
Propagate R_s^i to form $R_{t_2}^i$ via [38]
return $\{R_{t_2}^i\}_{i=0}^{n-1}$ as the segmentation in frame t_2

4. Quickest Moving Object Detection

In this section, we formulate the problem of sequentially *detecting* and segmenting moving objects from a video with minimum delay as a Quickest Detection problem. We briefly summarize the key ideas from that literature first.

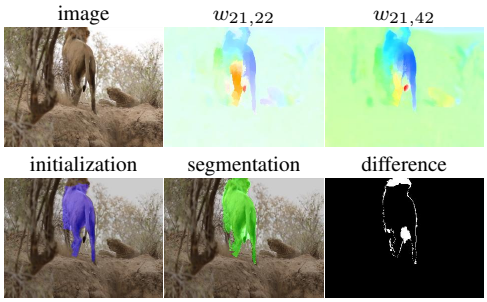

Figure 3: **Demonstration of multiframe motion segmentation**. *[Top row]: an image in the sequence, optical flows between adjacent frames, and composed optical flow. [Bottom]: Initialization to motion segmentation, final segmentation, and the difference between the two.*

4.1. Overview of Quickest Detection

Quickest Detection (QD) [21, 33] considers the problem of detecting changes in distribution of a discrete-time stochastic process $\{X_t\}_{t=1}^{\infty}$ online. X_t is sampled from a distribution p_0 before an unknown *change time* Γ, and X_t is sampled from p_1 at and after Γ. Although the theory is general, the literature focuses on one-dimensional signals, i.e., the range of X_t is $\mathbb{R}$. QD derives algorithms for determining the change with fewest observations X_t after the change subject to constraints on false alarms. A *false alarm* is a declared change by the algorithm before the change time Γ. The motivation is that reliable detection can be achieved by integrating many samples of X_i, reducing stochastic variability. However, this causes delay, i.e., the number of samples considered after the change. Thus, the goal of QD is minimizing the delay with guarantees on reliability of the detections.

Optimization Problem: QD is formulated as an optimization problem. A *stopping time* τ with respect to a stochastic process $\{X_t\}_{t=1}^{\infty}$ is a random variable such that the event $\{\tau = t\}$ is in the sigma-algebra generated by $X_1, \dots, X_t$. Intuitively, τ is a function that may return t if it uses only information determined from $X_1, \dots, X_t$. An example is $\tau = \inf\{t : \sum_{s=1}^{t} X_s \geq b\}$, i.e., τ is the first time t that the sum of X_s up to time t exceeds a threshold b. Let $\mathbb{P}_t$ and $\mathbb{E}_t$ denote the probability measure and expectation, associated with a change time of t. If the true change time is $\Gamma = t$ then the *delay* is the number of samples after the change time, i.e., $\tau - t$ when $\tau \geq t$. The *average detection delay* of a stopping time τ (averaging over randomness arising from random X_s's) is defined as

$$\mathsf{ADD}(\tau) = \sup_{t \geq 1} \mathbb{E}_t[\tau - t | \tau \geq t]. \tag{11}$$

ADD defines the worst case average delay over all change times. The *false alarm rate* of a stopping time is defined as

$\mathsf{FAR}(\tau) = 1/\mathbb{E}_\infty[\tau]$, that is, one over the average stopping time given that there is no change. There are different ways of defining the average detection delay and false alarm rate, but all lead to similar optimal stopping times. The QD optimization problem is to minimize the average delay subject to a constraint on the false alarm rate:

$$\min_{\tau} \mathsf{ADD}(\tau) \text{ subject to } \mathsf{FAR}(\tau) \leq \alpha, \tag{12}$$

where $\alpha \in [0, 1]$ is the maximum tolerable false alarm rate. The constraint on the false alarm rate is needed to avoid a trivial solution, i.e., if the stopping time is always one, the delay is zero, but this leads to many false alarms.

Optimal Stopping Rule: It can be shown that the optimal stopping time is given by the first time the *likelihood ratio* Λ_t exceeds a threshold b, i.e., $\tau^* = \inf\{t : \Lambda_t \geq b\}$. The threshold b is determined explicitly by the false alarm rate α and the distributions p_0 and p_1. The likelihood ratio arises from a test of the null hypothesis that the change occurs before t ($\Gamma < t$) against the alternative hypothesis that the change occurs after time t ($\Gamma \geq t$). That is,

$$\Lambda_t = \frac{\mathbb{P}[\Gamma < t | X_1, \ldots, X_t]}{\mathbb{P}[\Gamma \geq t | X_1, \ldots, X_t]} = \max_{1 \leq t_c < t} \prod_{s=t_c}^{t} \frac{p_1(X_s)}{p_0(X_s)}, \tag{13}$$

where the last equality is made under the assumption that X_s are iid before and after the change. In the moving object detection problem, the post-change distribution is only known conditional on a parameter θ, i.e., the regions and the warps. In this case, the optimal stopping time maximizes the likelihood over θ:

$$\Lambda_t = \max_{1 \leq t_c < t} \max_{\theta} \prod_{s=t_c}^{t} \frac{\mathbb{P}[\Gamma < t | X_1, \ldots, X_t, \theta]}{\mathbb{P}[\Gamma \geq t | X_1, \ldots, X_t]}. \tag{14}$$

The sequential algorithm to detect the change is to acquire data $X_s = x_s$ and, at each new acquisition at time t, one computes Λ_t, by solving a maximization problem over possible change times from $t_c = 1, \ldots, t-1$. At the first t when Λ_t exceeds the threshold b, a detection is declared. Solving the maximization problem directly may be expensive, and in the case that the post-change distribution has an unknown parameter, no general simplifications can be made to avoid direct maximization. We propose a solution for moving object detection in Section 4.3.

4.2. Algorithm for Detection and Segmentation

We now consider the image sequence a random process, and apply QD to the models in Section 2 to detect and segment moving objects at the time the objects move. We compute the likelihood ratio, Λ_t, which requires the computation of the pre- and post-change distributions. Since our distributions depend on hidden variables (the regions and displacements), we maximize over such variables as in (14).

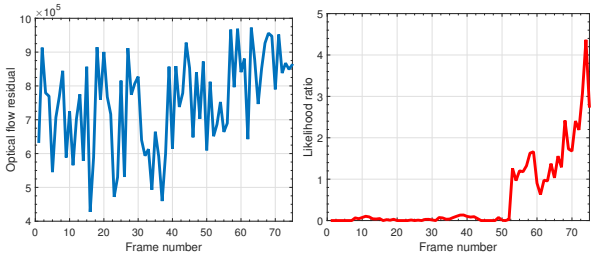

Figure 4: **Robust Likelihood Statistic in Quickest Detection**. *[Left]: The data X_t (in this case the average residual over the image) plotted versus time shows a signal where it is difficult to detect the change time (e.g., when the object moves). [Right]: The statistic Λ_t in QD plotted versus time shows clearly where the change is occurring. The true change time is $\Gamma = 42$.*

Let $\Lambda_{t_c,t}$ denote the likelihood ratio using data between t_c and t maximizing over the conditioned variables, i.e.,

$$\Lambda_{t_c,t} = \frac{\max_{\mathbf{R}^i_{t_c:t}, \mathbf{v}^i_{t_c:t}} p_1[\mathbf{I}_{t_c:t} | \mathbf{R}^i_{t_c:t}, \mathbf{v}^i_{t_c:t}, i = 0, \ldots, n-1]}{\max_{\mathbf{v}^0_{t_c:t}} p_0[\mathbf{I}_{t_c:t} | \mathbf{v}^0_{t_c:t}]}. \tag{15}$$

By using the pairwise independence given the conditioned variables, one can show that

$$-\log \Lambda_{t_c,t} = \min_{\mathbf{v}^0_{t_c:t}} \sum_{s=t_c}^{t} \int_{\Omega} \mathrm{Res}^0_i(x)\, \mathrm{d}x - \min_{\mathbf{R}^i_{t_c:t}, \mathbf{v}^i_{t_c:t}} \sum_{i=0}^{n-1} \sum_{s=t_c}^{t} \int_{R^i_s} \mathrm{Res}^i_s(x)\, \mathrm{d}x \tag{16}$$

$$-\log \Lambda_t = \min_{1 \leq t_c < t} -\log \Lambda_{t_c,t} \tag{17}$$

where $\mathrm{Res}^i_t(x) = \frac{1}{2\sigma^2_{\eta,i}} \rho[I_{t+1}(w^i_{t,t+1}(x)) - I_t(x)]$. Note that the minimization problem in (16) is the problem of segmentation and warp estimation considered in Section 3, which is solved by Algorithm 1.

We now specify our initial algorithm for moving object detection: Algorithm 2. The algorithm re-estimates Λ_t online at each new arrival of I_t. At each new acquisition of I_t, the algorithm finds a change time $t_c \in \{2, \ldots, t-1\}$ by solving a motion segmentation problem using Algorithm 1. By QD theory, the algorithm optimizes the delay.

4.3. Simplifications for Efficiency

Algorithm 2 requires that for each image acquisition, the optimization problem for segmentation be solved by Algorithm 1 for each possible change time. This is computationally expensive. Fortunately, it is possible to reduce the computational cost by estimating the change time from a less expensive problem. Although there may be a loss of optimality, we show in experiments that the loss is minor, while increasing computational efficiency drastically.

Algorithm 2 *Moving Object Detection*

Set $t = 1$
repeat // *compute likelihood ratio* Λ_t
  Increment $t \leftarrow t + 1$, acquire image I_t
  Compute Classic-NL warp $w^{NL}_{t,t+1}$
  for $t_c = 2, \ldots, t-1$ **do** // *find change time* t_c
    Determine $\mathbf{R}^i_{t_c,t}$ calling Algorithm 1 with $\mathbf{I}_{t_c,t}$
    Compute $\Lambda_{t_c,t}$ using (16)
  end for
  Compute $t^*_c = \arg\max_{2 \le t_c < t} \Lambda_{t_c,t}$
  Set $\Lambda_t = \Lambda_{t^*_c}$ and $R^i_t = R^i_{t^*_c,t}$
until $\Lambda_t \ge b$ or end of video
return R^i_t as the detection at time t if $\Lambda_t \ge b$

Fast Change Time Estimation: We propose a simplification to find a probable change time t^*_c without having to explicitly evaluate $\Lambda_{t_c,t}$ for each t_c. This is done by applying QD to simpler distributions than those considered in Section 2. Let the spatial average of the residuals be $r_t = \frac{1}{|\Omega|}\int_\Omega \text{Res}^{NL}_t(x)\,\mathrm{d}x$ determined from Classic-NL optical flow. If they are assumed iid and distributed according to $\mathcal{N}(\mu_0, \sigma)$ pre-change and $\mathcal{N}(\mu_1, \sigma)$ post-change, then the following statistic from the likelihood ratio, which can be computed efficiently, arises

$$F_{t_c,t} = (t - t_c + 1)(\hat{\mu}_{1:t_c-1} - \hat{\mu}_{t_c:t})^2 \qquad (18)$$

where $\hat{\mu}_{1:t_c-1}$ is the average residual before t_c and $\hat{\mu}_{t_c,t}$ is the average residual after t_c up to the current time t. The intuition for this statistic is that if a object starts to move or comes into view, the residual changes due to occlusions. Thus, the maximizer t^*_c of $F_{t_c,t}$ over t_c is proposed as a maximizer of $\Lambda_{t_c,t}$. The first factor mitigates changes close to the current time t, which likely arise from noise.

Avoiding Likelihood Before Change: Once the maximizer t^*_c of $F_{t_c,t}$ is proposed as the change time, the likelihood ratio $\Lambda_{t^*_c,t}$ must be computed and is used to approximate Λ_t. This avoids the full for loop in Algorithm 2. One cannot threshold $F_{t^*_c,t}$ to decide a change has occurred, as that decreases performance as shown in experiments. However, we can avoid the computation of $\Lambda_{t^*_c,t}$, which requires the optimization in Algorithm 1, before the change actually occurs in the video by a simple test. We compute the composed Classic-NL displacement $w^{NL}_{t^*_c,t}$ and then threshold the standard deviation, $\sigma(w^{NL}_{t^*_c,t})$, over all pixels. If the motion is multi-modal, the standard deviation is large, indicating the presence of a moving object. These lead to Algorithm 3.

Algorithm 3 *Faster Moving Object Detection*

Set $t = 1$
repeat // *compute likelihood ratio* Λ_t
  Set $t \leftarrow t + 1$, acquire image I_t, compute $w^{NL}_{t,t+1}$
  for $t_c = 2, \ldots, t-1$ **do** // *find probable change*
    Compute $F_{t_c,t}$ as in (18)
  end for
  Compute $t^*_c = \arg\max_{2 \le t_c < t} F_{t_c,t}$
  if $\sigma(w^{NL}_{t^*_c,t}) \ge d$ **then**
    Determine $\mathbf{R}^i_{t^*_c,t}$ calling Algorithm 1 with $\mathbf{I}_{t^*_c:t}$
    Compute $\Lambda_t = \Lambda_{t^*_c,t}$ using (16)
  else // *change not probable*
    Set $\Lambda_t = 0$
  end if
until $\Lambda_t \ge b$ or end of video
return R^i_t as the object detection at time t if $\Lambda_t \ge b$

5. Experiments

Dataset: There are no datasets that are explicitly designed for *detection* of moving objects. Therefore, we collected a dataset of 78 videos, varying from 100 to 800 frames, called the *Motion Detection Dataset* [2]. The camera moves, and the object moves (differently than the background) at some unknown frame. The videos may consist of a single or multiple objects.

Methods Compared: We compare approaches based on motion segmentation for detection of moving objects. We compare both frame-by-frame approaches [31], which integrates occlusion cues causally across time, and batch approaches [16, 12] integrating motion cues across frames against our method. To apply [16, 12] in an online approach, at each frame t, we segment frames 1 to t in batch, and then choose regions at time t that pass a relative area threshold as the detected regions. If no regions pass the test, there is no detection, and frames 1 to $t+1$ are considered, etc. Similarly, we threshold the result of [31] at each frame t to obtain a detection. We refer to the detection using [31] as **CVOS+det**, [16] as **LVA+det**, and [12] as **Multicuts+det**.

Evaluation: We define the empirical detection delay as the difference in the frame that the object was declared detected and the actual time the object moves, zero if this is negative. The empirical average detection delay (EADD) is the average of delays over all sequences. A false alarm is a declared detection before the actual time Γ or a declared detection after the change time, but with segmentation accuracy less than F_{lim} (we use 0.5 and 0.7) of F-measure compared to ground truth. The empirical false alarm rate EFAR is the number of false alarms over the number of videos. This measures both detection and segmentation accuracy. We evaluate methods in minimizing EADD for various false alarm constraints. Thus, we vary the threshold b in our algorithm, and the area threshold in other approaches.

Parameters: We fix all parameters in our algorithm over the entire dataset. We choose $d = 5$ in all experiments, and

[2] Dataset is available here: http://lao

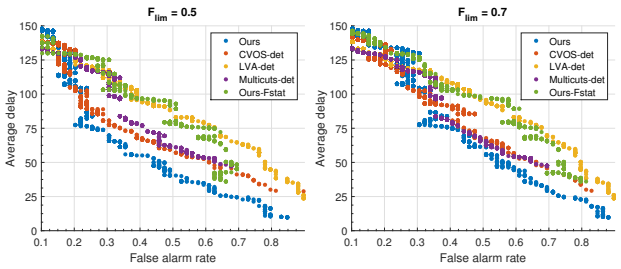

Figure 5: **Delay versus false alarm curves**. *All moving object detectors are compared. [Left]: Threshold for measuring false alarms is $F_{lim} = 0.5$ and [Right]: $F_{lim} = 0.7$. Results show ours method has the least delay.*

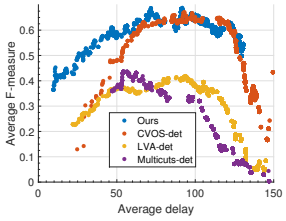

Figure 7: **Segmentation accuracy at detection time**. *All moving object detectors are compared in terms of their average F-measure to ground truth at detection time. Results show our method has highest accuracy at all levels of delay.*

test sensitivity later.

Minimizing Delay Result: The delay versus FAR curves for all algorithms are shown in Figure 5. To additionally test the optimality of the Λ_t statistic, we also compare to thresholding $F_{t_c^*,t}$, which we refer to as "ours-Fstat." Under all false alarm rates, our method has less delay. We also see that using the Λ_t leads to smaller delay than our-Fstat, showing the necessity of computing Λ_t. Further, the results remain consistent under different F_{lim}.

Visual Results: Figure 6 shows representative results operating at a FAR of 0.3. They verify that our method has on average less delay than competing methods.

Accuracy of Segmentation Result: We also display the average F-measure of the segmentation versus delay for detections as we vary thesholds of the detectors (giving various delays). Results are in Figure 7. We see that our method also has greater segmentation accuracy of the detections uniformly over all delays.

Ideal Detectors Result: We now analyze the detectors under the ideal case of perfect detection mechanisms. By detection mechanisms, we mean the test that decides the detection, e.g., the ratio test for ours and the area test for others. We show that under the case of perfect detection mechanisms for all methods, the segmentation procedure from our method leads to the best overall detection schemes compared to other approaches. This shows our segmentation method is better than others for the purpose of detection. To this end, we use ground truth to find the first frame (if it exists) when a method achieves a segmentation accuracy over F_{lim}, which we vary. We then plot the number of correct detections versus F_{lim}, and the average delay versus F_{lim}. Results are in Figure 8. They show that our method out-performs, in the number of correction detections and average delay, competing methods.

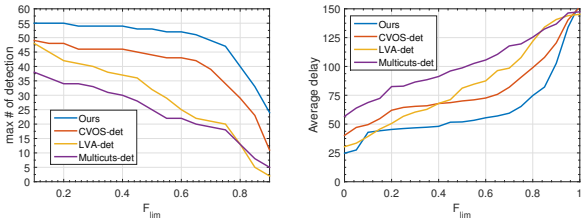

Figure 8: **Correct detections and delay of ideal detection mechanisms**. *[Left]: Correct detections versus F_{lim} the threshold for measuring false alarms. Under any measure of false alarms and ideal detection mechanisms, our method achieves more detections. [Right]: We also achieve less delay.*

Analysis of Algorithm 2 & 3: We now analyze the simplifications made for efficiency in Algorithm 3 and compare to Algorithm 2. Results are shown in Figure 9. First, we vary the threshold d of the standard deviation in Alg. 3 and record the number of times the test in Line 8 failed, saving us from an expensive segmentation operation. This results in an monotone increasing number of segmentations saved (red curve). Now, we plot the increase in average delay (over Algorithm 2) versus the standard deviation threshold d for various different thresholds b of the likelihood test. Results show that the delay does not increase much with increasing standard deviation. Thus, a rather large d decreases computational speed significantly, while leading only to a small increase in delay. We also plot the delay-FAR curve for various standard deviations and compare it to Algorithm 2 (no simplifications). Results show nearly the same curves, indicating little or no performance degradation.

Analysis of Refinement in Segmentation: We have displayed the results (in Figure 10) of our method using only the initialization in Algorithm 1 (called without refinement) without running the gradient descent for motion segmentation and compare against running the gradient descent. The results indicate that delay is reduced at moderate FAR with the gradient descent, but not by much. This indicates large time savings can be achieved by skipping the gradient descent and just using the initialization, without much detection degradation.

Computational Cost: In our unoptimized code, without running the gradient descent in Algorithm 1, using [13] rather than Classic-NL, assuming Line 8 passes and $|t_c - t| = 50$ frames, the cost of Ln 3-8 is 10 secs on a Intel

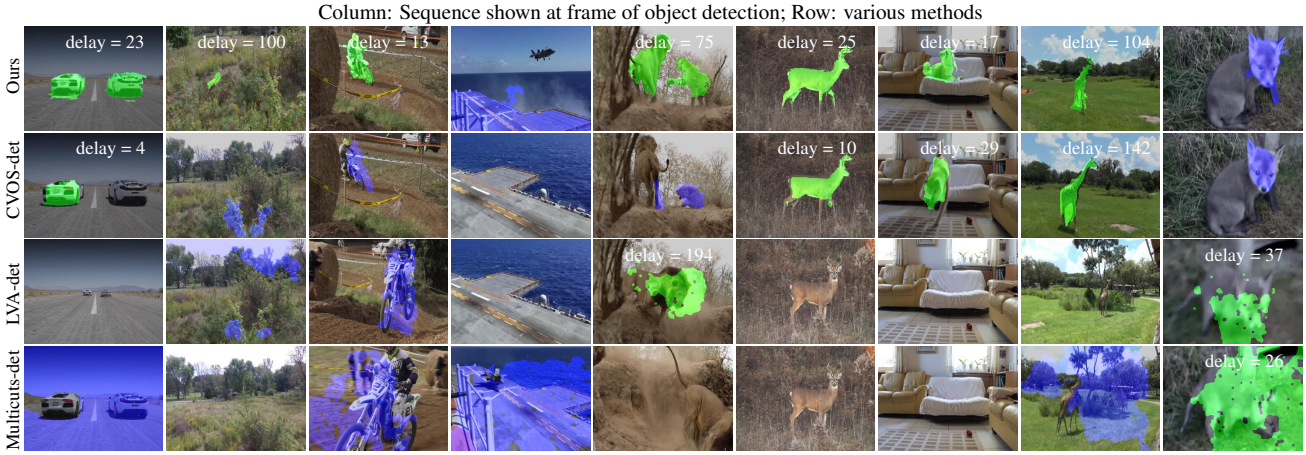

Figure 6: **Representative detections for methods tested**. *We show visualizations of the detections (or non-detections) for each of the competing methods, each operating at a false alarm rate of 0.3. The segmentation result at the detected frame is shown. Green masks indicate a correct detection, while purple masks indicate a false alarm. The delay at the detection is indicated, if the detection is correct. No segmentation result indicates the method did not detect (maximum delay). Results illustrate our method achieves less delay on average.*

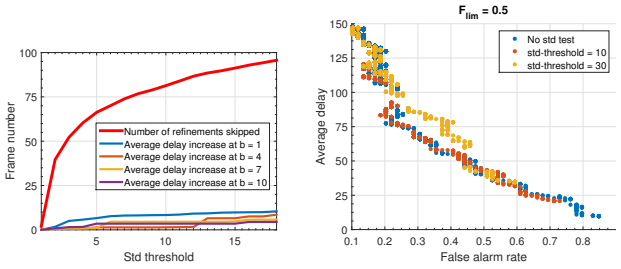

Figure 9: **Analysis of costings savings and degradation of Alg. 2**. *[Left]: Average delay increase versus standard deviation threshold d (in Alg. 2) as thresholds b on likelihood is varied. Curves near zero indicate little delay degradation with high degrees of cost savings. Red curve shows segmentation savings as d varies. [Right]: The delay-FAR curves for various thresholds d show similar performance against Alg. 3.*

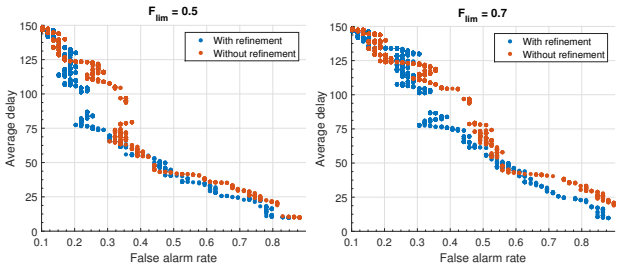

Figure 10: **Refinement analysis in motion segmentation**. *Delay-FAR curves with and without using the gradient descent in motion segmentation to refine the initialization. Modest gains in the detector performance are seen for the refinement.*

I7 processor. Running the gradient descent, for increased segmentation accuracy, increases the cost to 2 minutes.

6. Discussion

We have introduced a framework and derived a method for moving object *detection* and segmentation in a video sequence where the object's apparent motion is distinguishable from the background at some unknown time. The algorithm is designed for closed loop systems and was derived using principles from Quickest Detection. This leads to an online algorithm that minimizes the delay in detection subject to false alarm constraints. Extensive experiments on a new dataset for the moving object detection demonstrated that our method achieves less delay for any false alarm constraint than competing motion-based detection methods, verifying the theory. Analysis of various simplifications of original QD algorithms derived were shown to yield computational savings, while maintaining performance.

Each acquisition of an image in our algorithm, either costs the amount of non-local optical flow computation if the standard deviation test fails or the non-local optical flow computation plus the motion segmentation, if the test passes. By setting an upper limit on the frames used in segmentation, which we do, the cost of motion segmentation is limited. So our method scales at most linearly with frames acquired. Although processing is currently not real-time for real-time closed loop systems, it has the potential. This is because we derived an online algorithm, and the main bottleneck in our method, optical flow computation is a rapidly evolving area and computational gains are expected. Moreover, our optimization methods are based on methods that are also rapidly progressing. Further, speed-ups to our algorithms are possible, in particular, recursive updates of motion segmentation over time may be possible, and we plan to address this in future work.

References

[1] X. Bai, J. Wang, and G. Sapiro. Dynamic color flow: a motion-adaptive color model for object segmentation in video. *ECCV 2010*, pages 617–630, 2010. 2

[2] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3):033003–033003, 2010. 2

[3] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 63(1):75–104, 1996. 2

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36. Springer, 2004. 2

[5] B. Chen and P. Perona. Speed versus accuracy in visual search: Optimal performance and neural architecture. *Journal of vision*, 15(16):9–9, 2015. 1

[6] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999. 2

[7] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, 2005. 2

[8] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1558–1570, 2015. 4

[9] H. Fu, C. Wang, D. Tao, and M. J. Black. Occlusion boundary detection via deep exploration of context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[10] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab. Optical flow estimation with uncertainties through dynamic mrfs. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[11] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. A novel video dataset for change detection benchmarking. *Image Processing, IEEE Transactions on*, 23(11):4663–4679, 2014. 2

[12] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. *IEEE International Conference on Computer Vision (ICCV)*, pages 3271–3279, 2015. 2, 6

[13] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. In *European Conference on Computer Vision*, pages 471–488. Springer, 2016. 7

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981. 2

[15] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–302. IEEE, 2004. 2

[16] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 36(6):1187–1200, 2014. 1, 2, 6

[17] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on pattern analysis and machine intelligence*, 22(3):266–280, 2000. 2

[18] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3227–3234, 2015. 2

[19] J.-M. Perez-Rua, T. Crivelli, P. Bouthemy, and P. Perez. Determining occlusions from space and time image reconstructions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[20] A. S. Polunchenko, G. Sokolov, and W. Du. Quickest change-point detection: a bird's eye view. *arXiv preprint arXiv:1310.3285*, 2013. 1

[21] H. V. Poor and O. Hadjiliadis. *Quickest detection*, volume 40. Cambridge University Press Cambridge, 2009. 1, 4

[22] S. Ricco and C. Tomasi. Dense lagrangian motion estimation with occlusions. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2012. 2

[23] A. Shiryaev. On stochastic models and optimal methods in the quickest detection problems. *Theory of Probability & Its Applications*, 53(3):385–401, 2009. 1

[24] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *International journal of computer vision*, 82(3):325–357, 2009. 2

[25] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, pages 2432–2439. IEEE, 2010. 2, 3, 4

[26] D. Sun, E. B. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2010. 2

[27] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black. A fully-connected layered model of foreground and background flow. In *CVPR*, pages 2451–2458. IEEE, 2013. 1, 2

[28] G. Sundaramoorthi and B.-W. Hong. Fast label: Easy and efficient solution of joint multi-label and estimation problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3126–3133, 2014. 4

[29] G. Sundaramoorthi, A. Yezzi, and A. Mennucci. Coarse-to-fine segmentation and tracking using sobolev active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):851–864, 2008. 3

[30] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, pages 2233–2240. IEEE, 2011. 2

[31] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4268–4276. IEEE, 2015. 2, 6

[32] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[33] V. V. Veeravalli and T. Banerjee. Quickest change detection. *Academic press library in signal processing: Array and statistical signal processing*, 3:209–256, 2013. 1, 4

[34] J. Y. Wang and E. H. Adelson. Representing moving images with layers. *IEEE TIP*, 3(5):625–638, 1994. 1, 2

[35] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 321–326. IEEE, 1996. 1, 2

[36] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European conference on computer vision*, pages 94–106. Springer, 2006. 2

[37] Y. Yang and G. Sundaramoorthi. Shape tracking with occlusions via coarse-to-fine region-based sobolev descent. *IEEE transactions on pattern analysis and machine intelligence*, 37(5):1053–1066, 2015. 3

[38] Y. Yang, G. Sundaramoorthi, and S. Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4408–4416, 2015. 2, 4

[39] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007. 2