Occlusion Detection by Temporal Integration of Optical Flow

Virginia Estellers, Stefano Soatto

Abstract

Occlusions become apparent as failing regions in optical-flow models when integrated over time because violations of the brightness-constancy constraint accumulate and grow in occluded areas. Based on this observation, we propose a new variational model for joint occlusion and flow estimation that emphasizes violations of the brightness constraint in order to detect occlusions by temporal integration of both flow and occlusions. To this purpose, we estimate the flow with respect to a single reference frame that accumulates the errors in the flow model over a short-time interval; this formulation allows us to distinguish occlusions from noise and non-Lambertian phenomena by means of spatio-temporal regularizers over the occlusion set. In terms of minimization, we approximate the resulting variational problem by a sequence of convex optimizations and develop an efficient primal-dual algorithm to solve them. Our experiments show the benefits of the proposed formulation, both the single-frame formulation and the occlusion regularizers, in comparison to the state of the art.

Index Terms

Occlusion Detection, Optical Flow.

I. Introduction

Given two consecutive frames of a video sequence, optical-flow techniques estimate the apparent motion of the scene by matching pixel intensities under smoothness assumptions of the estimated flow [1]. This intensity matching fails in regions that are only visible in one frame because an object in the scene occludes another one. Occlusions appear because no single two-dimensional image can fully capture the content of a three-dimensional scene; therefore, when we match the content of two images by optical flow, occlusions are not a residual to be neglected but a source of information of the geometry of the scene.

Unfortunately, given only two video frames, it is not possible to know if the violation of the brightness constraint of optical flow is due to occlusions, noise, changes of illumination, or non-Lambertian phenomena. Our key idea originates from the hypothesis that it is possible to detect the cause of brightness violations by an extended temporal observation of the scene. Occlusions then become apparent by integrating the flow over time as consistent failing regions of the model. The first novelty of our approach lies in the temporal integration of optical flow with respect to a single reference frame, which accumulates the violations of the brightness constraint over time in order to emphasize occlusions. As a result, our formulation trades-off flow accuracy for occlusion accuracy, as occlusions are easier to detect for the large displacements –that result from estimating the flow from a single frame–, while the flow is easier to estimate for pairs of consecutive frames.

Previous occlusion models [2], [3], [4], [5], [6], [7] neglect the temporal dimension of the occlusion problem and primarily detect areas of brightness violation. Our method, instead, models the spatial and temporal regularity of occlusions to distinguish them from noise and non-Lambertian phenoma. This constitutes the second novelty of our method, giving us a competitive advantage at the prize of a more complex model.

We formalize this idea as a joint minimization problem to estimate both optical flow and occlusions from a video. The objective functional of the minimization has a data term that penalizes violations of the optical-flow model in co-visible regions and spatio-temporal regularizers for both flow and occlusions. The functional is minimized with respect to each variable independently, and the problem is reduced to a sequence of convex minimizations that are efficiently solved with a new primal-dual algorithm–to be release upon publication– adapted to the characteristics of the problem.

Our contributions are thus threefold: first, a new brightness-constancy constraint, Equation (2), that is integrated in an optical flow model to emphasize occlusions; second, the introduction of a temporal model for occlusions; third, the development of efficient and easy-to-code algorithms to solve the resulting minimization problem.

The rest of the paper is organized as follows: Section II reviews related methods and puts the paper in context; Sections III and IV present the proposed model and its variational formulation, Section V develops a numerical algorithm for its minimization, and Sections VI and VII present experimental results and conclusions.

II. Literature Review

Variational methods [8], [9] are the state-of-the-art in optical-flow because they provide accurate dense estimates and result in minimization problems that can be efficiently solved [10], [11]. Two key issues, however, remain open: the robustness of the method to large displacements, and the computation of the flow in occlusion areas.

Large displacements are difficult to detect because the brightness constraint of optical-flow models cannot be linearized, the resulting minimization problem is not convex, and numerical algorithms converge to a local minimum close to the initialization point. To alleviate the effects of initialization, classical techniques adopt a multi-resolution strategy that finds a minimum of the model present at large scales but ignores small structures not present at the coarsest-resolution level. Large-displacement methods [9], [12], [13] solve this issue by introducing a descriptor-matching step – previous to the variational model – that guides the multi-resolution to a local minimum relevant for small structures. Variational methods are thus still the core of optical flow, but few of them include occlusions explicitly in the model. This is the goal of this paper.

Flow errors appear in occlusion areas because the brightness constraint forces intensity matching in areas where no correspondence is possible. Occlusion-aware techniques avoid these errors by taking into account occlusions in the model. This can be done implicitly by ignoring the brightness constraint in areas where the flow model breaks down [2], [5], [6], or explicitly by introducing an occlusion variable in the model [14], [3], [7], [15], [16]. A second criterion that differenciates these techniques is how occlusions are incorporated into the model: multiple-step procedures first estimate the flow ignoring occlusions, use the unreliable flow to detect occlusions, and then correct the flow in occlusion areas [14], [15], [16]; whereas joint methods [3], [7] explicitly introduce occlusions in the model and formulate a single minimization where flow and occlusion variables interact. The optimization of joint methods is more difficult, but the models are more robust because flow and occlusions jointly explain the data. For this reason, we propose a joint model but design it to emphasize occlusion detection rather than flow estimation.

Independently of the level of interaction between flow and occlusions, there are two criteria to detect occlusions from optical flow: the first one detects occlusions as unexplained pixels by the flow [3], [5], and the second detects occlusions as pixels where there is no correspondence between forward and backward flows [2], [6], [17]. We adopt the first criterion and introduce a temporal model for the occlusion set; this differentiates our model from existing techniques that only handle two images and neglect the temporal dimension of occlusions in video.

Although related, techniques to detect occlusion boundaries [18], [19], [20], [21], [22], [23] and layered models [24], [25], [26], [27], [20], [28], [18] solve a different problem. They detect occlusion boundaries from a segmentation of the image in order to find the objects in the scene and their relative order; as a result, they are closer to image and motion segmentation than to our method. Recently, machine-learning classifiers have also been used for occlusion detection in [15], [16]. The learning-from-data nature of these approaches is far from our model, which is designed from physical constraints instead of data analysis and does not require a training stage.

In summary, we propose a variational method for joint occlusion detection and flow estimation, but focus on occlusions in the design of our model. As a variational model, our method has the flexibility to incorporate large-displacement techniques [9], [12], [13] and more robust data terms [29] to improve flow estimation.

III. A Flow Model to Emphasize Occlusions

A. *Optical-flow Correspondence*

Optical-flow techniques estimate the apparent motion of a scene by solving a correspondence problem between the pixels of consecutive images [30], [1]. This correspondence, however, is only valid under three assumptions: (i) the same points are visible in both images, (ii) the scene is Lambertian, and (iii) scene illumination is constant. In such conditions, there exists a local differentiable mapping between the domains of both images that describes their pixel correspondence. In general, given two images I_1 and I_2, this is formulated with the following brightness-constancy constraint:

$$I_2(x) = I_1(x + \boldsymbol{u}_1(x)) \qquad \forall x \in D \setminus \Omega, \tag{1}$$

where $\Omega \subset D$ is the subset of the image domain where conditions (i)-(iii) are violated. Unfortunately, it is not possible to determine which of conditions (i)-(iii) fail from two images, and occlusions cannot be distinguished from other phenomena.

B. *Occlusion Detection and Temporal Integration*

Our main hypothesis is that it is possible to detect the cause of the brightness violation by an extended temporal observation of the scene. If the observation interval is short, the illumination can be assumed constant, and violations of the brightness constraint are only due to occlusions or non-Lambertian phenomena. That is, Ω is partitioned into occlusions Θ and non-Lambertian effects $\Omega \setminus \Theta$.

Both sets are multiply connected and change as a function of time, but their temporal behavior is different. While occlusions accumulate and grow from initial occluded points; non-Lambertian phenoma depend on the shape of the underlying surface and, without additional information, are better modeled by independent random noise in the image domain. As a result, the temporal behavior of occlusions can be exploited to differentiate them from non-Lambertian phenoma in flow models that integrate multiple frames.

To this purpose, we must choose a reference frame to define flow and occlusions, as pixels in one image are occluded or visible only with respect to a reference image. Rather than computing flow and occlusions between pairs of consecutive frames [8], [31], [32], [33], [34], we compute the flow with respect to a single frame for the purpose of occlusion detection. For instance, the flow between frames I_1 and I_3 can be decomposed into intermediate flows between intermediate images, as suggested by [35] and [36] for non-rigid objects:

$$I_3(x) = I_2(x+\boldsymbol{u}_2(x)) = I_1(x + \boldsymbol{u}_1(x+\boldsymbol{u_2}(x)) + \boldsymbol{u}_2(x)).$$

Given frames $I_1,\ldots, I_T$ from a video sequence $\{I_i\}$, we propose estimate the flow with respect to the central frame I_c with the following brightness constraint:

$$I_c(x) = \begin{cases} I_i(x + \sum_{j=i}^{c-1} \boldsymbol{u}_j(x+\sum_{l=j+1}^{c-1} \boldsymbol{u}_l(x))) & i < c \\ I_i(x - \sum_{j=c}^{i} \boldsymbol{u}_j(x+\sum_{l=c+1}^{i} \boldsymbol{u}_l(x))) & i \geq c \end{cases} \quad x \in D \setminus \Omega_i, \tag{2}$$

where we use forward flows for frames previous to I_c and backward flows for posterior ones. For large sequences, we use a sliding window as suggested in [35]. This temporal integration is less accurate than pairwise models [37], [34] for the purpose of flow estimation, but it emphasizes occlusions because the regions where the flow model is violated, Ω_i, are all defined in the domain of the reference image. As a result, flow errors accumulate in a single frame and the model can differentiate occlusions from other phenoma by temporal integration.

To this purpose, we describe the occlusion regions by their characteristic functions – for each Θ_i, its characteristic function $\chi_i\colon D \to \{0, 1\}$ satisfies $\chi_i(x) = 1 \Leftrightarrow x \in \Theta_i$– and propose an occlusion models with the following terms:

Size regularization: The size of the occlusion region is small in comparison to the image domain; that is, for certain $c_\beta > 0$

$$\int_D \chi_i(x)\,\mathrm{d}x \leq c_\beta. \tag{3}$$

Occlusion models [4], [7] propose a similar penalty for occlusions, but they only compute the flow between two images and neglect the time variable.

Spatial regularization: The shape of the occlusion regions is restricted by the geometric regularity of the occluding objects in the scene. We formalize this assumption with a regularizer that measures the perimeter of the occlusion region and penalizes irregular shapes in a similar manner than layered models [24], [25], [26], [27], [20], [28], [18]; it is algebraically described by

$$\sum_{i=1}^{T} \int_D \|\nabla \chi_i\| \,\mathrm{d}x \;<\; c_{\gamma_s}. \tag{4}$$

Temporal regularization: The size of the occlusion regions grows because the relative motion of the occluded objects with respect to the camera is regular in time, not arbitrary. We exploit this idea with the following penalty

$$\int_D |\chi_i(x) - \chi_{i-1}(x)| \,\mathrm{d}x \;<\; c_{\gamma_t}. \tag{5}$$

We consider also consider a constrained model (MC) that requires occlusions to grow from occluded pixels, that is, $\chi_i \geq \chi_{i+1},\; i < c$ and $\chi_i \leq \chi_{i+1},\; i \geq c$. The relaxed constraint that forces the area of occlusions to grow results in additional dual variables in the minimization algorithms, while its experimental performance is similar to the proposed model; for this reason we consider the simpler MC constraint.

IV. Variational Formulation

We formulate occlusion detection as a variational model, i.e., as a minimization problem described by the three components: the minimization variables, the objective function, and the minimization algorithm.

The minimization variables are the flow fields $\{\boldsymbol{u}_i = (u_i, v_i)\}$ and the characteristic functions $\{\chi_i\}$, possibly subject to constraint MC. The objective function is described by (6); it measures the violation of the brightness constraint in co-visible areas with $\mathcal{B}_k$, includes spatio-temporal regularizers $\mathcal{J}$ and $\mathcal{R}$ for the flow and occlusions, and a penalty term $\int_D \chi_k$ for the size-regularization constraint.

$$\sum_{k=1}^{T} \left(\mathcal{B}_k(\chi_k, \{\boldsymbol{u}_i\}) + \mathcal{R}(\chi_k) + \mathcal{J}(\boldsymbol{u}_k) + \beta \int_D \chi_k \right). \tag{6}$$

The term $\mathcal{B}_k$ penalizes violations of the brightness constraint (2) with an ℓ_1 norm to ensure robustness to outliers as follows:

$$\mathcal{B}_k(\chi_k, \{\boldsymbol{u}_i\}) = \int_D w_k (1 - \chi_k) |\varepsilon_k| \tag{7}$$

where

$$\varepsilon_k(x) = \begin{cases} I_c(x) - I_k(x + \sum_{j=k}^{c-1} \boldsymbol{u}_j(x + \sum_{l=j+1}^{c-1} \boldsymbol{u}_l(x))) & k < c \\ I_c(x) - I_k(x - \sum_{j=c}^{k} \boldsymbol{u}_j(x + \sum_{l=c+1}^{k} \boldsymbol{u}_l(x))) & k \geq c \end{cases}, \tag{8}$$

and the scalars $w_k = e^{-|c-k|}$ assign increasing weights to model violations in frames closer to I_c; we have chosen a negative exponential for simplicity.

The flow regularizer

$$\mathcal{J}(\boldsymbol{u}_k) = \int_D \alpha_s g_\mu(\|\nabla u_k\|) + \alpha_s g_\mu(\|\nabla v_k\|) + \alpha_t \|\boldsymbol{u}_k - \boldsymbol{u}_{k-1}\|$$

is introduced to overcome the ill-posed nature of the optical-flow problem, as model (2) does not determine a unique flow in textureless areas or in the direction tangent to the image gradient. We use the Huber norm to regularize the gradients to allow sharp flow discontinuities at the boundaries of the objects and small variations in between. In particular, the Huber penalty

$$g_\mu(\|\nabla u_k\|) = \begin{cases} \frac{\|\nabla u_k\|^2}{2\mu} & \text{if } \|\nabla u_k\| < \mu \\ \|\nabla u_k\| - \frac{\mu}{2} & \text{if } \|\nabla u_k\| \geq \mu \end{cases} \tag{9}$$

allows flow discontinuities larger than μ while it acts as Gaussian smoothing in homogeneous flow areas. The positive scalars α_s, α_t, μ are model parameters, where $\alpha_t = 0$ for the first frame of the sequence by convention.

Finally, the occlusion regularizer $\mathcal{R}$ penalizes violations of models M2 and M3. This results in spatial and temporal regularizers with parameters $\gamma_s, \gamma_t \geq 0$

$$\mathcal{R}(\chi_k) = \int_D \gamma_s \|\nabla \chi_k\| + \gamma_t |\chi_k - \chi_{k-1}|. \tag{10}$$

Total variation is a better regularizer than the Huber penalty for χ_k because the binary variables $\{\chi_i\}$ shall be relaxed to real-valued functions in $[0, 1]$ in Section V. Again, we assume $\gamma_t = 0$ for the first frame of the sequence.

In summary, detecting occlusions requires solving the minimization problem

$$\min_{\substack{\{\boldsymbol{u}_i, \chi_i\} \\ \chi_i \in \{0,1\}}} \sum_{k=1}^{T} \mathcal{B}_k(\chi_k, \{\boldsymbol{u}_i\}) + \mathcal{R}(\chi_k) + \mathcal{J}(\boldsymbol{u}_k) + \beta \int_D \chi_k, \tag{11}$$

which suffers from two main difficulties. First, variables $\{\chi_i\}$ take binary values and lead to a combinatorial problem computationally too expensive to solve. Second, even if we ignore the discrete nature of $\{\chi_i\}$, the resulting problem is not convex in $\{\boldsymbol{u}_i, \chi_i\}$. This does not mean that we cannot detect occlusions, but that we can only guarantee to find a local minimum of the objective functional, and that out-of-the-box minimization algorithms are slow. The algorithm that we develop in Section V addresses these two issues.

V. Numerical Minimization

We propose a multi-resolution approach to speed up the minimization and be more robust against local minima. At the same time, at each resolution we solve the problem efficiently as a sequence of convex optimizations.

Multi-resolution finds an approximate solution to the problem at a coarse scale and then tracks it through scale as it solves the problem at higher resolutions. At a large scale, the problem will be less prone to suffer from local minima and a first coarse solution is easily found, which is used to initialize the algorithm at smaller scales. As the scale is reduced, local minima appear in the minimization, and tracking of the initial solution guarantees that the solution is kept meaningful. Thus, multi-resolution does not find a global minimum of the non-convex problem, only one that appears at large scales. In terms of efficiency, the algorithm is designed to perform most of its iterations at a large scale; as the scale is reduced, the algorithm is initialized closer to a minimum and requires less iterations to converge. The large-displacement technique [12] can easily be incorporated here.

To speed-up minimization of (11) at each resolution, we alternate the minimization with respect to flow and occlusion variables, as follows:

$$\boldsymbol{u}_k \leftarrow \min_{\boldsymbol{u}_k} \sum_{j=1}^{T} \mathcal{B}_j(\chi_j, \{\boldsymbol{u}_i\}) + \mathcal{J}(\boldsymbol{u}_k) \qquad 1 \leq k \leq T, \tag{12a}$$

$$\{\chi_i\} \leftarrow \min_{\chi_i \in \{0,1\}} \sum_{j=1}^{T} \mathcal{B}_j(\chi_j, \{\boldsymbol{u}_i\}) + \mathcal{R}(\chi_j) + \beta \int_D \chi_j. \tag{12b}$$

We use sequential convex optimization for the minimization in $\boldsymbol{u}_k$, while the minimization with respect to $\{\chi_i\}$ is relaxed to a convex problem by extending the feasibilty set to real-valued functions in $[0, 1]$.

A. Minimization in Flow Variables

Problem (12a) is not convex because the brightness constraint depends on the minimization variable through interpolation. We propose an iterative algorithm that linearizes the brightness constraint around the current flow estimate, solves the resulting convex problem, and uses its solution as the flow estimate for the next iteration. The algorithm stops when a fixed-point is encountered, which happens at a minimimizer of the linearized brightness constraint. The resulting procedure, for pairwise flow, is equivalent to the classic image warping [37].

Without loss of generality, we describe the minimization in $\boldsymbol{u}_k$, $k < C$. Let $\boldsymbol{u}_1^n, \ldots, \boldsymbol{u}_k^n$ be the flow estimates at iteration n, and

$$\tilde{I}_k(x) = I_k(x + \sum_{i=k}^{c-1} \boldsymbol{u}_i^n(x + \sum_{l=i+1}^{c-1} \boldsymbol{u}_l^n(x))) \tag{13}$$

the image warped by these estimates, we linearize the image brightness in $\boldsymbol{u}_k$ by

$$I_k(x+\sum_{i=k}^{c-1} \boldsymbol{u}_i(x+\sum_{l=i+1}^{c-1} \boldsymbol{u}_l(x))) \approx \tilde{I}_k + \nabla\tilde{I}_k\cdot(\boldsymbol{u}_k-\boldsymbol{u}_k^n). \tag{14}$$

At iteration n, the error in the brightness constraint ε_k (8) is approximated by the affine function $e_k^n(\boldsymbol{u}_k)$ in order to obtain a convex minimization problem. To ensure the accuracy of the approximation

$$\varepsilon_k(x) \approx e_k^n(\boldsymbol{u}_k) = I_c - \tilde{I}_k - \nabla\tilde{I}_k \cdot (\boldsymbol{u}_k - \boldsymbol{u}_k^n) \tag{15}$$

we restrict the solution to lie in a ball $\mathcal{B}_k^n$ around $\boldsymbol{u}_k^n$.

Problem (12a) is thus solved as the following sequence of convex problems:

$$\begin{aligned} \boldsymbol{u}_k^{n+1} \leftarrow \min_{\boldsymbol{u}_k\in\mathcal{B}_k^n} \int \sum_{j=1}^{k} w_j(1-\chi_j)|e_j^n(\boldsymbol{u}_k)| + \alpha_t\|\boldsymbol{u_k}-\boldsymbol{u_{k-1}}\|+ \\ \alpha_t\|\boldsymbol{u_k}-\boldsymbol{u_{k+1}}\| + \alpha_s[g_\mu(\|\nabla u_k\|)+g_\mu(\|\nabla v_k\|)]. \end{aligned} \tag{16}$$

We solve (16) efficiently with the primal-dual algorithm [38] by providing closed-form solutions for each proximal update. We choose a first-order method because the size of the problem –several million variables– makes second-order methods unfeasible. In the following, we omit iteration superscripts to lighten notation and substitute the constraint $\boldsymbol{u}_k \in \mathcal{B}_k^0$ by an equivalent proximal term

$$G(\boldsymbol{u}_k) = \frac{r}{2}\int_D \|\boldsymbol{u} - \boldsymbol{u}_k^0\|^2. \tag{17}$$

The key observation of our algorithm is the convexity of each of the terms in the objective functional of (16), which allows us to re-formulate it as a saddle-point problem that is separable and easy to solve in each variable. To this purpose, we define the following convex functions:

$$f_j(\boldsymbol{u}) = \int_D w_j(1-\chi_j)|e_j^n(\boldsymbol{u})| \tag{18}$$

$$f_0(\boldsymbol{u}) = \alpha_t\int_D \|\boldsymbol{u} - \boldsymbol{u_{k-1}}\| \tag{19}$$

$$f_{k+1}(\boldsymbol{u}) = \alpha_t\int_D \|\boldsymbol{u} - \boldsymbol{u_{k+1}}\| \tag{20}$$

$$f_d(\nabla u) = \alpha_s\int_D g_\mu(\|\nabla u\|) + \alpha_s g_\mu(\|\nabla v\|). \tag{21}$$

To find the saddle-point formulation, we re-write (16) as the constraint minimization

$$\min_{\substack{\boldsymbol{u}_k,\boldsymbol{d}\\ \{\tilde{\boldsymbol{u}}_j\}}} \sum_{j=0}^{k+1} f_j(\tilde{\boldsymbol{u}}_j) + f_d(\boldsymbol{d}) + G(\boldsymbol{u}_k) \;\; \text{s.t.} \begin{cases} \tilde{\boldsymbol{u}}_j = \boldsymbol{u}_k & j=0\dots k+1 \\ \boldsymbol{d} = \nabla \boldsymbol{u}_k \end{cases} \tag{22}$$

and formulate its Lagrangian $\mathcal{L}(\boldsymbol{\lambda})$ by introducing a dual variable $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_0, \dots, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\lambda}_d)$ for each constraint in (22),

$$\begin{aligned} \mathcal{L} = \min_{\substack{\boldsymbol{u}_k,\boldsymbol{d}\\ \tilde{\boldsymbol{u}}_0,..,\tilde{\boldsymbol{u}}_k}} & \sum_{j=0}^{k+1} f_j(\tilde{\boldsymbol{u}}_j) + f_d(\boldsymbol{d}) + G(\boldsymbol{u}_k) + \\ & \int_D [\sum_{j=0}^{k+1} \boldsymbol{\lambda}_j \cdot (\boldsymbol{u}_k - \tilde{\boldsymbol{u}}_j) + \boldsymbol{\lambda}_d \cdot (\nabla \boldsymbol{u}_k - \boldsymbol{d})]. \end{aligned} \tag{23}$$

The Lagrangian can be simplified in terms of the conjugates of each of the convex functions $f_0, \dots, f_{k+1}, f_d$ as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\lambda}) = \min_{\boldsymbol{u}_k} & - \sum_{j=0}^{k+1} f_j^*(\boldsymbol{\lambda}_j) - f_d^*(\boldsymbol{\lambda}_d) + G(\boldsymbol{u}_k) + \\ & \int_D [\sum_{j=0}^{k+1} \boldsymbol{\lambda}_j \cdot \boldsymbol{u}_k + \boldsymbol{\lambda}_d \cdot \nabla \boldsymbol{u}_k]. \end{aligned} \tag{24}$$

Finally, we make use of convex analysis to write (22) as the saddle-point problem

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \; \min_{\boldsymbol{u}_k} \; & - \sum_{j=0}^{k+1} f_j^*(\boldsymbol{\lambda}_j) - f_d^*(\boldsymbol{\lambda}_d) + G(\boldsymbol{u}_k) + \\ & \int_D [\sum_{j=0}^{k+1} \int_D \boldsymbol{\lambda}_j \cdot \boldsymbol{u}_k + \int_D \boldsymbol{\lambda}_d \cdot \nabla \boldsymbol{u}_k]. \end{aligned} \tag{25}$$

This formulation allows us to apply the primal-dual algorithm of Chambolle and Pock [38], which is designed for problems of the form:

$$\max_{\boldsymbol{\lambda}} \; \min_{\boldsymbol{u}_k} -F^*(\boldsymbol{\lambda}) + G(\boldsymbol{u}_k) + \int_D \boldsymbol{\lambda} \cdot K \boldsymbol{u}_k, \tag{26}$$

where K a continuous linear map, and F^*, G proper, convex, lower- semicontinuous functions. Under such conditions, [38] proposes the following iterative algorithm to solve (26):

$$\boldsymbol{\lambda}^{n+1} \leftarrow \min_{\boldsymbol{\lambda}} \; \sigma F^*(\boldsymbol{\lambda}) + \frac{1}{2} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^n - \sigma K \boldsymbol{z}^n\|^2 \tag{27a}$$

$$\boldsymbol{u}_k^{n+1} \leftarrow \min_{\boldsymbol{u}_k} \; \tau G(\boldsymbol{u}_k) + \frac{1}{2} \|\boldsymbol{u}_k - \boldsymbol{u}_k^n + \tau K^* \boldsymbol{\lambda}^{n+1}\|^2 \tag{27b}$$

$$\boldsymbol{z}^{n+1} = \; \boldsymbol{u}_k^{n+1} + \theta(\boldsymbol{u}_k^{n+1} - \boldsymbol{u}_k^n), \tag{27c}$$

where τ, σ, θ are algorithm parameters and $\boldsymbol{z}^0 = \boldsymbol{u}_k^0$.

In our case, $K^* = [I, \overset{(k+2)}{\cdots}, I, \nabla^*]$, with I the identity operator, and minimization (27a) is separable in $\boldsymbol{\lambda}_j, \boldsymbol{\lambda}_d$ as

$$F^*(\boldsymbol{\lambda}) = \sum_{j=0}^{k+1} f_j^*(\boldsymbol{\lambda}_j) + f_d^*(\boldsymbol{\lambda}_d). \tag{28}$$

As a result, each dual variable $\boldsymbol{\lambda}_0, \dots, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\lambda}_d$ can be updated separately, and the algorithm is reduced to

$$\begin{aligned}
\boldsymbol{\lambda}_j^{n+1} &\leftarrow \min_{\boldsymbol{\lambda}_j} \ \sigma f_j^*(\boldsymbol{\lambda}_j) + \frac{1}{2}\|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_j^n - \sigma \boldsymbol{z}^n\|^2 \quad j = 0 \dots k{+}1 \\
\boldsymbol{\lambda}_d^{n+1} &\leftarrow \min_{\boldsymbol{\lambda}_d} \ \sigma f_d^*(\boldsymbol{\lambda}_d) + \frac{1}{2}\|\boldsymbol{\lambda}_d - \boldsymbol{\lambda}_d^n - \sigma \nabla \boldsymbol{z}^n\|^2 \\
\boldsymbol{u}_k^{n+1} &\leftarrow \min_{\boldsymbol{u}_k} \ \tau r\|\boldsymbol{u}_k - \boldsymbol{u}_k^0\|^2 + \|\boldsymbol{u}_k - \boldsymbol{u}_k^n + \tau[\sum_{j=0}^{k+1} \boldsymbol{\lambda}_j^{n+1} + \operatorname{div} \boldsymbol{\lambda}_d^{n+1}]\|^2 \\
\boldsymbol{z}^{n+1} &= \ \boldsymbol{u}_k^{n+1} + \theta(\boldsymbol{u}_k^{n+1} - \boldsymbol{u}_k^n).
\end{aligned}$$

The efficiency of the proposed algorithm comes from the separability of F^* and from the ability to find closed-form solutions for each of the minimization problems. The derivation of closed-form solutions is detailed next, and the resulting algorithm is summarized in Algorithm 1.

Initialize $\boldsymbol{u}_k = \boldsymbol{0}, \ \boldsymbol{\lambda} = \boldsymbol{0}, \ \boldsymbol{z} = \boldsymbol{u}_k$.

Choose $\tau, \sigma > 0$, $\theta \in [0, 1]$. Let $\tau_r = (\tau r + 1)^{-1}$

while $\|\boldsymbol{u}_k^{n+1} - \boldsymbol{u}_k^n\| > 1^{-4}$ **do**

$\boldsymbol{\lambda}_0^{n+1} = \ \min(\alpha_t, \|\hat{\boldsymbol{\lambda}}_0\|)\frac{\hat{\boldsymbol{\lambda}}_0}{\|\hat{\boldsymbol{\lambda}}_0\|}, \quad \hat{\boldsymbol{\lambda}}_0 = \boldsymbol{\lambda}_0^n + \sigma[\boldsymbol{z}^n - \boldsymbol{u}_{k-1}]$

$\boldsymbol{\lambda}_j^{n+1}$ updated with (32)

$\boldsymbol{\lambda}_0^{n+1} = \ \min(\alpha_t, \|\hat{\boldsymbol{\lambda}}_{k+1}\|)\frac{\hat{\boldsymbol{\lambda}}_{k+1}}{\|\hat{\boldsymbol{\lambda}}_{k+1}\|}, \quad \hat{\boldsymbol{\lambda}}_{k+1} = \boldsymbol{\lambda}_{k+1}^n + \sigma[\boldsymbol{z}^n - \boldsymbol{u}_{k+1}]$

$\boldsymbol{\lambda}_{d_u}^{n+1} = \ \alpha_s \min(1, \frac{\|\hat{\boldsymbol{\lambda}}_{d_u}\|}{\alpha_s + \sigma\mu})\frac{\hat{\boldsymbol{\lambda}}_{d_u}}{\|\hat{\boldsymbol{\lambda}}_{d_u}\|}, \quad \hat{\boldsymbol{\lambda}}_d = \boldsymbol{\lambda}_d^n + \sigma\nabla\boldsymbol{z}^n$

$\boldsymbol{\lambda}_{d_v}^{n+1} = \ \alpha_s \min(1, \frac{\|\hat{\boldsymbol{\lambda}}_{d_v}\|}{\alpha_s + \sigma\mu})\frac{\hat{\boldsymbol{\lambda}}_{d_v}}{\|\hat{\boldsymbol{\lambda}}_{d_v}\|},$

$\boldsymbol{u}_k^{n+1} = \tau_r(\tau r \boldsymbol{u}_k^0 + \boldsymbol{u}_k^n - \boldsymbol{u}_{k+1} - \tau \sum_{j=0}^{k} \boldsymbol{\lambda}_j^{n+1} + \tau \operatorname{div} \boldsymbol{\lambda}_d^{n+1})$

$\boldsymbol{z}^{n+1} = \ \boldsymbol{u}_k^{n+1} + \theta(\boldsymbol{u}_k^{n+1} - \boldsymbol{u}_k^n)$

end

Algorithm 1: Minimization algorithm in flow variables.

1) Minimization in dual variables: We solve the minimization in $\boldsymbol{\lambda}_j$ and $\boldsymbol{\lambda}_{\boldsymbol{d}}$ through Moreau's identity [39]:

$$\boldsymbol{\lambda}_j \leftarrow \min_{\boldsymbol{\lambda}_j} \ \sigma f_j^*(\boldsymbol{\lambda}_j) + \frac{1}{2}\|\boldsymbol{\lambda}_j - \hat{\boldsymbol{\lambda}}_j\|^2 \tag{29}$$

$$\Updownarrow$$

$$\boldsymbol{\lambda}_j = \hat{\boldsymbol{\lambda}} - \sigma \boldsymbol{s}_j, \quad \boldsymbol{s}_j \leftarrow \min_{\boldsymbol{s}_j} \ \sigma f_j(\boldsymbol{s}_j) + \frac{\sigma}{2}\|\boldsymbol{s}_j - \frac{\hat{\boldsymbol{\lambda}}_j}{\sigma}\|^2. \tag{30}$$

For $j = 1 \dots k$, the minimization in $\boldsymbol{s}_j$ is a two-dimensional problem that is decoupled for each pixel, as follows:

$$\min_{\boldsymbol{s}_j} \int_D w_j(1-\chi_j)|e_j(\mathbf{0}) - \nabla \tilde{I}_j \cdot \boldsymbol{s}_j| + \frac{\sigma}{2}\|\boldsymbol{s}_j - \frac{\hat{\boldsymbol{\lambda}}_j}{\sigma}\|^2.$$

For each pixel, the first term measures the weighted distance to the line $\{\boldsymbol{s}_j : e_j(\mathbf{0}) - \nabla \tilde{I}_j \cdot \boldsymbol{s}_j = 0\}$, while the second term measures the distance to the point $P = \frac{\hat{\boldsymbol{\lambda}}_j}{\sigma}$. Consequently, the minimizer lies in the perpendicular to the line that passes through P, that is, $\boldsymbol{s}_j = \frac{\hat{\boldsymbol{\lambda}}_j}{\sigma} + l\nabla \tilde{I}_j$ for some scalar l. The problem is now reduced to a 1-dimensional minimization in l.

In terms of l, the minimization is simplified to

$$\min_l \ \frac{\sigma}{2} l^2 \|\nabla \tilde{I}_j\|^2 + w_j(1-\chi_j)|e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j) - l\|\nabla \tilde{I}_j\|^2|.$$

Its optimality condition results in the following equation:

$$l = -w_j \frac{1-\chi_j}{\sigma} \operatorname{sign}(e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j) - l\|\nabla \tilde{I}_j\|^2) \tag{31}$$

By analyzing the possible values of sign function, we find a closed-form solution for l,

$$l = \begin{cases} -\sigma^{-1} w_j(1-\chi_j) & \frac{e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j)}{\|\nabla \tilde{I}_j\|^2} > -\sigma^{-1} w_j(1-\chi_j) \\ \sigma^{-1} w_j(1-\chi_j) & \frac{e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j)}{\|\nabla \tilde{I}_j\|^2} < \sigma^{-1} w_j(1-\chi_j) \\ \frac{e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j)}{\|\nabla \tilde{I}_j\|^2} & \text{otherwise} \end{cases},$$

which results in the following closed-form solution for $\boldsymbol{\lambda}_j$:

$$\boldsymbol{\lambda}_j = \begin{cases} w_j(1-\chi_j)\nabla \tilde{I}_j & \sigma e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j) > -w_j(1-\chi_j)\|\nabla \tilde{I}_j\|^2 \\ -w_j(1-\chi_j)\nabla \tilde{I}_j & \sigma e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j) < w_j(1-\chi_j)\|\nabla \tilde{I}_j\|^2 \\ \sigma e_j(\sigma^{-1}\hat{\boldsymbol{\lambda}}_j)\frac{\nabla \tilde{I}_j}{\|\nabla \tilde{I}_j\|^2} & \text{otherwise} \end{cases} \tag{32}$$

For $j = 0$, the minimization in $\boldsymbol{s}_j$ is the $\ell_2 - \ell_2^2$ problem

$$\min_{\boldsymbol{s}_j} \int_D \alpha_t \|\boldsymbol{s}_j - \boldsymbol{u}_{k-1}\| + \frac{\sigma}{2}\|\boldsymbol{s}_j - \frac{\hat{\boldsymbol{\lambda}}_j}{\sigma}\|^2. \tag{33}$$

Its solution is given pixel-wise by the shrinkage operator:

$$\boldsymbol{s}_j = \boldsymbol{u}_{k-1} + \frac{1}{\sigma}\max(0, \|\hat{\boldsymbol{\lambda}}_j - \sigma\boldsymbol{u}_{k-1}\| - \alpha_t)\frac{\hat{\boldsymbol{\lambda}}_j - \sigma\boldsymbol{u}_{k-1}}{\|\hat{\boldsymbol{\lambda}}_j - \sigma\boldsymbol{u}_{k-1}\|},$$

The dual variable $\boldsymbol{\lambda}_j = \hat{\boldsymbol{\lambda}}_j - \sigma\boldsymbol{s}_j$ is thus updated by

$$\boldsymbol{\lambda}_j = \min(\alpha_t, \|\hat{\boldsymbol{\lambda}}_j\|)\frac{\hat{\boldsymbol{\lambda}}_j}{\|\hat{\boldsymbol{\lambda}}_j\|}. \tag{34}$$

The same minimization and updates apply for $j = k+1$, substituting $\boldsymbol{u}_{k-1}$ for $\boldsymbol{u}_{k+1}$.

Let $\boldsymbol{s}_{d_u}, \boldsymbol{s}_{d_u}$ be the rows of the 2×2 matrix $\boldsymbol{s}_d$, the minimization in $\boldsymbol{s}_d$ is then separable in

$$\begin{aligned}
&\min_{\boldsymbol{s}_{d_u}} \int_D \sigma\alpha_s g_\mu(\boldsymbol{s}_{d_u}) + \frac{\sigma}{2}\|\boldsymbol{s}_{d_u} - \frac{\hat{\boldsymbol{\lambda}}_{d_u}}{\sigma}\|^2 \\
&\min_{\boldsymbol{s}_{d_v}} \int_D \sigma\alpha_s g_\mu(\boldsymbol{s}_{d_v}) + \frac{\sigma}{2}\|\boldsymbol{s}_{d_v} - \frac{\hat{\boldsymbol{\lambda}}_{d_v}}{\sigma}\|^2.
\end{aligned} \tag{35}$$

We observe that the minimization is also decoupled for each pixel and can be solved by minimizing the objective pixel-wise. As the objective is symmetric with respect to $\boldsymbol{s}_{d_u}$ and $\boldsymbol{s}_{d_u}$, we only present the derivation for $\boldsymbol{s}_{d_u}$.

Due to the differentiability of the Hubber norm, with $\nabla g_\mu(\boldsymbol{s}) = \alpha_t \min(1, \frac{\|\boldsymbol{s}\|}{\mu})\frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}$, we obtain the following optimality conditions for (35):

$$\alpha_t \min(1, \frac{\|\boldsymbol{s}_{d_u}\|}{\mu})\frac{\boldsymbol{s}_{d_u}}{\|\boldsymbol{s}_{d_u}\|} + \sigma\boldsymbol{s}_{d_u} - \hat{\boldsymbol{\lambda}}_{d_u} = 0, \tag{36}$$

which are solved by

$$\boldsymbol{s}_{d_u} = \frac{\hat{\boldsymbol{\lambda}}_{d_u}}{\sigma} - \frac{\alpha_t}{\sigma}\min(1, \frac{\|\hat{\boldsymbol{\lambda}}_{d_u}\|}{\alpha_t + \sigma\mu})\frac{\hat{\boldsymbol{\lambda}}_{d_u}}{\|\hat{\boldsymbol{\lambda}}_{d_u}\|}. \tag{37}$$

We then update the dual variable: $\boldsymbol{\lambda}_{d_u}^{n+1} = \hat{\boldsymbol{\lambda}}_{d_u} - \sigma\boldsymbol{s}_{d_u}$ with

$$\boldsymbol{\lambda}_{d_u}^{n+1} = \alpha_t \min(1, \frac{\|\hat{\boldsymbol{\lambda}}_{d_u}\|}{\alpha_t + \sigma\mu})\frac{\hat{\boldsymbol{\lambda}}_{d_u}}{\|\hat{\boldsymbol{\lambda}}_{d_u}\|}. \tag{38}$$

2) Minimization in primal variables : The minimization in $\boldsymbol{u}_k$ can be rewritten as

$$\boldsymbol{u}_k^{n+1} \leftarrow \min_{\boldsymbol{u}_k} \frac{\tau r + 1}{2}\|\boldsymbol{u}_k - \hat{\boldsymbol{u}}\|^2, \tag{39}$$

where

$$\hat{\boldsymbol{u}} = (\tau r + 1)^{-1}(\tau r\boldsymbol{u}_k^0 + \boldsymbol{u}_k^n - \boldsymbol{u}_{k+1} - \tau\sum_{j=0}^{k}\boldsymbol{\lambda}_j^{n+1} + \tau \operatorname{div}\boldsymbol{\lambda}_d^{n+1}).$$

Its solution is thus $\boldsymbol{u}_k^{n+1} = \hat{\boldsymbol{u}}$. Assembling the primal and dual updates, we obtain Algorithm 1.

B. *Minimization with respect to Occlusion Variables*

In the minimization with respect to the binary functions $\{\chi_i\}$, we find an approximate solution by solving a relaxed problem that extends the feasibility set to real-valued functions in $[0,1]$. This results in the following convex problem

$$\min_{\substack{\chi_1,\ldots,\chi_T\\ \chi_i\in[0,1]}} \int_D \sum_{k=1}^{T} \big((1-\chi_k)|\varepsilon_k| + \mathcal{R}(\chi_k)\big)\,. \tag{40}$$

C. *Size Regularization*

If we only consider the assumption that the occlusion set is small ($\gamma_s = \gamma_t = 0$), the solution of the relaxed problem coincides with the solution of the original binary problem. Indeed, the minimization problem (40) is then equivalent to

$$\min_{\substack{\chi_1,\ldots,\chi_T\\ \chi_i\in[0,1]}} \int \sum_{k=1}^{T} (\beta - |\varepsilon_k|)\chi_k \iff \chi_k = \begin{cases} 0 & \beta > |\varepsilon_k| \\ 1 & \beta \le |\varepsilon_k| \end{cases}. \tag{41}$$

The functional is defined point-wise by a linear function, and its minimizers lie in the extremes of the constraint set $[0,1]$ depending on the value $\rho_k = \beta - |\varepsilon_k|$.

D. *Spatial Regularization*

If we introduce spatial regularization ($\gamma_t = 0$), the minimization problem (40) can be solved independently at each frame as the following minimization

$$\min_{\chi_k\in[0,1]} \int_D \rho_k\chi_k + \gamma_s\|\nabla\chi_k\|. \tag{42}$$

The binary problem has the form of the image segmentation model of [40], where the authors show that the thresholded solution to the relaxed problem solves the original binary one. The same proof applies here, showing how to obtain a minimizer of the original problem from the solution of (42).

E. *Spatio-temporal Regularization*

With spatial and temporal regularization, the problem reads

$$\min_{\substack{\chi_1,\ldots,\chi_T\\ \chi_k\in[0,1]}} \int \sum_{k=1}^{T} \big(\rho_k\chi_k + \gamma_s\|\nabla\chi_k\|\big) + \gamma_t \sum_{k=2}^{T} |\chi_k - \chi_{k-1}|,$$

and is equivalent to the constrained minimization in $\chi = [\chi_1, \cdots, \chi_T]$

$$\min_{\substack{\chi, \boldsymbol{d}, t \\ \chi \in [0,1]^T}} \int \sum_{k=1}^{T} \rho_k \chi_k + \gamma_s \|\boldsymbol{d}_k\| + \gamma_t |t_k| \quad \text{s.t.} \begin{cases} \boldsymbol{d} = K_d \chi \\ t = K_t \chi, \end{cases} \tag{43}$$

where $\boldsymbol{d} = [\boldsymbol{d}_1 \cdots \boldsymbol{d}_T]$, $t = [t_1 \cdots t_T]$ and

$$K_d = \begin{bmatrix} \nabla & 0 & \cdots & 0 \\ 0 & \nabla & \cdots & 0 \\ & & \ddots & \\ 0 & \cdots & 0 & \nabla \end{bmatrix} \qquad K_t = \begin{bmatrix} -I & I & 0 \cdots & 0 & \\ 0 & -I & I & 0 & \cdots \\ & & & \ddots & \\ 0 & \cdots & 0 & -I & I \end{bmatrix}. \tag{44}$$

The objective functional can now be minimized efficiently with the primal-dual algorithm [38] because the different terms are convex and separable in space and time. To avoid repetition, we simply present here each of the sub-minimization problems:

$$\boldsymbol{d}_k^n \leftarrow \min_{\boldsymbol{d}_k} \int \gamma_s \|\boldsymbol{d}_k\| + \frac{\sigma}{2} \|\boldsymbol{d}_k - \frac{\hat{\boldsymbol{\nu}}_k}{\sigma}\|^2, \;\; \hat{\boldsymbol{\nu}}_k = \boldsymbol{\nu}_k^{n-1} + \sigma K_d \bar{z}^{n-1} \tag{45a}$$

$$t_k^n \leftarrow \min_{t_k} \int \gamma_t |t_k| + \frac{\sigma}{2} \|t_k - \frac{\hat{\eta}_k}{\sigma}\|^2, \quad \hat{\eta}_k = \hat{\eta}_k^{n-1} + \sigma K_t \bar{z}^{n-1} \tag{45b}$$

$$\chi_k^n \leftarrow \min_{\chi_k \in [0,1]} \int \tau \rho_k \chi_k + \frac{1}{2} \|\chi_k - \hat{\chi}_k\|_2^2; \;\; \hat{\chi} = \chi_k^{n-1} - \tau (K_d^* \boldsymbol{\nu}^n + K_t^* \eta^n) \tag{45c}$$

$$\bar{z}^n = \chi^n + \theta(\chi^n - \chi^{n-1}),$$

where we use Moreau's identity in the update of the dual variables $\boldsymbol{\nu}_k^n = \hat{\boldsymbol{\nu}}_k - \sigma \boldsymbol{d}_k^n$ and $\eta_k^n = \hat{\eta}_k - \sigma t_k^n$. As before, the potential of the proposed algorithm lies on the efficient solution of problems (45c)-(45b), as we explain next and summarized in Algorithm 2.

Initialize $\boldsymbol{\nu} = \mathbf{0}, \eta = 0, z = \chi$.

Choose $\tau, \sigma > 0$, $\theta \in [0, 1]$.

while $\|\chi^{n+1} - \chi^n\| > 1^{-4}$ **do**

$\quad \boldsymbol{\nu}_k^{n+1} = \min(\frac{\gamma_s}{\|\hat{\boldsymbol{\nu}}_k\|}, 1)\; \hat{\boldsymbol{\nu}}_k, \quad \hat{\boldsymbol{\nu}}_k = \boldsymbol{\nu}_k^n + \sigma K_d \bar{z}^n$

$\quad \hat{\eta}_k^{n+1} = \min(|\hat{\eta}_k|, \gamma_t)\, \text{sign}(\hat{\eta}_k), \quad \hat{\eta}_k = \hat{\eta}_k^n + \sigma K_t \bar{z}^n$

$\quad \chi_k^{n+1} = \mathcal{P}_{[0,1]}(\chi_k^n - \tau(K_d^* \boldsymbol{\nu}^{n+1} + K_t^* \eta^{n+1} + \rho_k))$

$\quad \bar{z}^{n+1} = \chi^{n+1} + \theta(\chi^{n+1} - \chi_k^n)$

end

Algorithm 2: Minimization algorithm in occlusion variables for the unconstrained model.

The minimization (45a) has the same form than (33) and is solved again with the shrinkage operator:

$$\boldsymbol{d}_k^{n+1} = \frac{1}{\sigma} \max(1 - \frac{\gamma_s}{\|\hat{\boldsymbol{\nu}}_k\|}, 0)\; \hat{\boldsymbol{\nu}}_k, \quad \boldsymbol{\nu}_k^{n+1} = \min(\frac{\gamma_s}{\|\hat{\boldsymbol{\nu}}_k\|}, 1)\; \hat{\boldsymbol{\nu}}_k.$$

The minimization (45b) is a classic $\ell_1 - \ell_2$ problem solved by soft-thresholding:

$$t_k^{n+1} = \frac{1}{\sigma}\max(|\hat{\eta}_k| - \gamma_t, 0)\frac{\hat{\eta}_k}{\hat{\eta}_k}, \quad \hat{\eta}_k^{n+1} = \min(|\hat{\eta}_k|, \gamma_t)\frac{\hat{\eta}_k}{\hat{\eta}_k}.$$

In the case f the constraint model MC, the dual step (45b) is subject to the pixel-wise constraint $t_k \geq 0$, and the minimizer $t_k^{n+1} = \frac{1}{\sigma}\max(|\hat{\eta}_k - \gamma_t, 0)\,\mathrm{sign}(\hat{\eta}_k)$ must be projected into the positive orthant. This results in the following updates:

$$t_k^{n+1} = \frac{1}{\sigma}\max(\hat{\eta}_k - \gamma_t, 0), \quad \hat{\eta}_k^{n+1} = \min(\hat{\eta}_k, \gamma_t).$$

Finally, the minimization (45c) is a quadratic problem decoupled for each pixel in both objective and constraint. Its solution is thus obtained by pixel-wise projection of the minimizer into the unit interval: $\chi_k^{n+1} = \mathcal{P}_{[0,1]}(\hat{\chi_k} - \tau\rho_k)$.

VI. Experimental Results

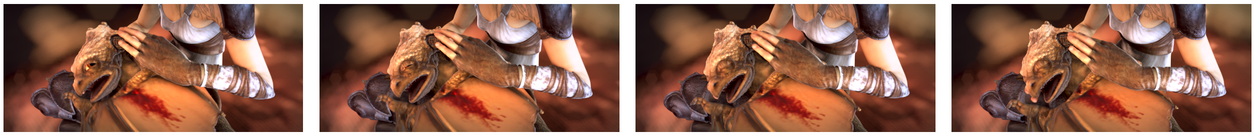
We compare the proposed method to different variational models for joint occlusion detection and flow estimation: a traditional pairwise-flow model with the same spatio-temporal regularizers than our model in order to asses the effects of the proposed single-frame formulation, and two state-of-the-art techniques [2] and [7]. Our code will be available on http://vision.ucla.edu upon publication of the manuscript.

Our experiments are performed with videos from MPI-Sintel [41], a computer-generated database with ground-truth occlusions. In our experiments, we use the *final* pass of the dataset, which includes shading, blur, and atmospheric effects in the rendered video sequence. The sequences are 50 frames long, with 24 frames per second for 1024×436-pixel images. To focus on occlusion detection, we select 10 sequences where no displacement larger than 100 pixels occurs, as neither ours nor [2], [7] models integrate large displacement techniques, and the size of the occlusion set does not hinder flow estimation[1]. We use the first 20% of frames of each sequence to tune the parameters of each model and test in the remaining frames; this gives a wide range of flow and occlusion conditions to our experiments. For simplicity, we fix the length of the sliding window to 5 frames in our model.

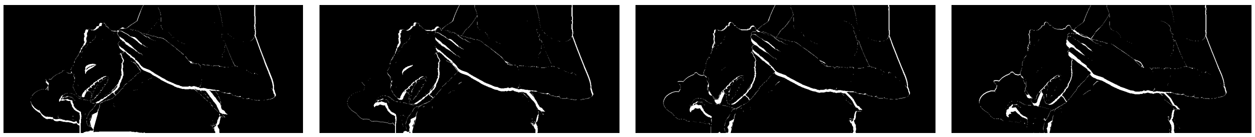
Figures 1 and 2 present the detected occlusions for sequences *bandage_1*, *market_2*. This qualitative evaluation shows the benefits of the single-frame formulation, which allows us to detect the occlusions caused by the movement of the girl's arm, and the smoothing effects of temporal and spatial regularization, which results in consistent occlusion regions in space and time that are more robust to noise. Figure 3 show a faire case in the *bamboo_1* sequence, where the combination of the bamboo oscillations and the

[1]Sequences where less than 10% of the pixels are occluded.

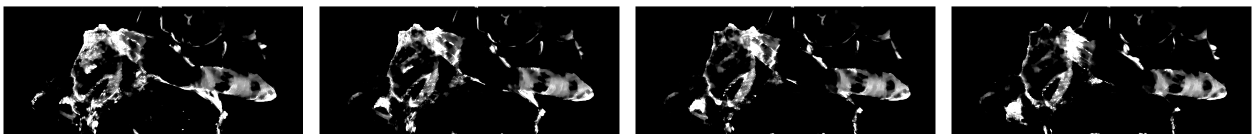
movement of the camera result in non-smooth displacements that do not match our temporal regularizers. More qualitative results are available in the supplementary material.



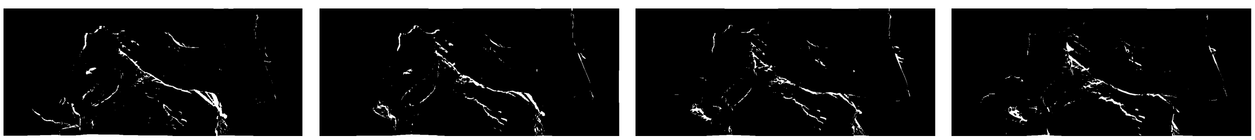
(a) *bandage_1*, frame 21 (b) *bandage_1*, frame 22 (c) *bandage_1*, frame 23 (d) *bandage_1*, frame 24



(e) ground truth, frame 21 (f) ground truth, frame 22 (g) ground truth, frame 23 (h) ground truth, frame 24

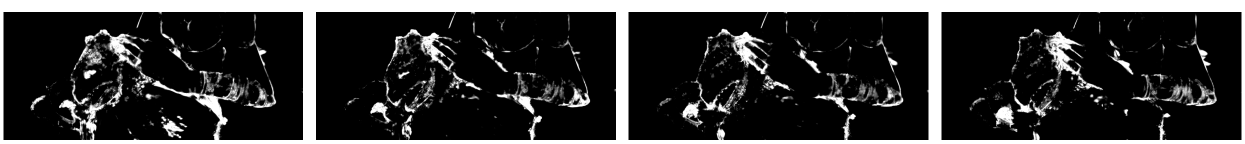


(i) pairwise, frame 21 (j) pairwise, frame 22 (k) pairwise, frame 23 (l) pairwise, frame 24



(m) [2], frame 21 (n) [2], frame 22 (o) [2], frame 23 (p) [2], frame 24

(q) [7], frame 21 (r) [7], frame 22 (s) [7], frame 23 (t) [7], frame 24



(u) our, frame 21 (v) our, frame 22 (w) our, frame 23 (x) our, frame 24

Fig. 1: Comparison of occlusions detection methods for *bandage_1* sequence. We observe how the single-frame formulation is bale to detect the consecutive occlusions caused by the movement of the arm, while models based in pairwise flows missed such an occlusion.

TABLE I: Effects of constraint $\chi_i \geq \chi_{i+1}$ for $i < c$ and $\chi_i \leq \chi_{i+1}$ for $i \geq c$: the lower F1 of the constrained model shows that the assumption that occlusions grow is too restrictive.

	constrained model			unconstrained model		
sequence	ppv	tpr	F1	ppv	tpr	F1
alley_1	0.66	0.15	0.24	0.43	0.25	**0.31**
alley_2	0.49	0.24	0.31	0.43	0.31	**0.34**
bamboo_1	0.46	0.19	0.26	0.45	0.27	**0.33**
bandage_1	0.60	0.20	0.30	0.44	0.29	**0.34**
bandage_2	0.46	0.14	0.21	0.41	0.14	**0.20**
market_2	0.450	0.33	0.39	0.48	0.35	**0.40**
shaman_3	0.31	0.03	0.06	0.15	0.04	**0.08**
sleeping_1	0.03	0.07	0.04	0.03	0.08	**0.04**
sleeping_2	0.40	0.04	0.07	0.14	0.14	**0.14**

A. *Evaluation Criteria*

As a detection problem we can measure the number of true positives and true negatives and define the following metrics: precision or positive predictive value (ppv), recall or true positive rate (tpr), and F1 score. If we consider the pixels independent, precision can be interpreted as the probability that a pixel classified as an occlusion is a true occlusion; while recall is the probability that an occluded pixel is detected by the system. The F1 score considers both precision and recall to compute a metric that reaches its best value at 1 and worst score at 0. The score is in fact a weighted average of precision and recall that generalizes to different metrics by assigning different weights to type I and type II errors. Quantitative results with these metrics are shown in Tables I-III.

Using F1 to summarize the performance of each model into a single score, we can compare and rank them. Table I thus shows that our unconstrained model outperforms the constrained one as the assumption that occlusions grow from a seed point is too restrictive. As a result, we consider our unconstrained formulation for the rest of comparisons.

Table II compares the performance of our model to the standard flow formulation –pairwise brightness constraint with spatio-temporal regularization– and shows the benefits of the proposed single-frame formulation for occlusion detection at the price of flow accuracy, as expected.

Table III compares our model to the variational techniques [2], [7]. In terms of occlusion models, both [2], [7] neglect the temporal nature of occlusions by computing the flow and occlusions between

TABLE II: Comparison of the classic pairwise flow with our single-frame formulation for the same flow and occlusion regularizers. The pairwise model is more accurate on the flow –lower mean angular and end-point flow errors (ae, epe)–, while our formulation is consistently more accurate on occlusions –higher mean precision (ppv), recall (tpr), and F1.

	pairwise flow					single-frame formulation				
sequence	ae	epe	ppv	tpr	F1	ae	epe	ppv	tpr	F1
alley_1	**0.12**	0.78	0.62	0.17	0.26	0.12	0.69	0.43	0.25	**0.31**
alley_2	**0.12**	1.22	0.44	0.08	0.11	0.19	2.21	0.43	0.31	**0.34**
bamboo_1	**0.08**	0.48	0.48	0.18	0.26	0.09	0.48	0.45	0.27	**0.33**
bandage_1	**0.18**	1.09	0.29	0.19	0.22	0.20	1.22	0.44	0.29	**0.34**
bandage_2	**0.19**	0.92	0.48	0.11	0.17	0.20	0.88	0.41	0.14	**0.20**
market_2	**0.18**	1.11	0.42	0.31	0.35	0.19	1.16	0.48	0.35	**0.40**
shaman_3	0.30	1.09	0.11	0.03	0.04	**0.24**	1.08	0.15	0.04	**0.08**
sleeping_1	0.10	0.49	0.50	0.01	0.01	**0.08**	0.46	0.03	0.08	**0.04**
sleeping_2	**0.04**	0.13	0.12	0.08	0.09	0.06	0.32	0.14	0.14	**0.14**

TABLE III: Comparison to existing occlusion-detection methods [2], [7] optmimized for F1 score.

	[2]			[7]			proposed model		
sequence	ppv	tpr	F1	ppv	tpr	F1	ppv	tpr	F1
alley_1	0.24	0.28	0.26	0.02	0.43	0.03	0.43	0.25	**0.31**
alley_2	0.10	0.09	0.09	0.02	0.15	0.03	0.43	0.31	**0.34**
bamboo_1	0.38	0.34	**0.36**	0.02	0.46	0.04	0.45	0.27	0.33
bandage_1	0.17	0.33	0.22	0.01	0.39	0.01	0.44	0.29	**0.34**
bandage_2	0.10	0.28	0.15	0.01	0.21	0.02	0.41	0.14	**0.20**
market_2	0.46	0.34	0.38	0.02	0.47	0.04	0.48	0.35	**0.40**
shaman_3	0.06	0.16	0.08	0.00	0.02	0.00	0.15	0.04	**0.08**
sleeping_1	0.08	0.01	0.02	0.00	0.02	0.01	0.03	0.08	**0.04**
sleeping_2	0.16	0.10	0.12	0.00	0.51	0.01	0.14	0.14	**0.14**

two frames. The main difference between the two models is the criterion used to detect occlusions: [2] detects occlusions as regions where forward and backward flows mismatch, while [7] detects occlusions as pixels non-matched by the forward flow. As a result, [2] solves a larger optimization problem to estimate both forward and backward flows and results in a slower algorithm with more accurate results. Our model compares favorably to both of them because it integrates information from multiple frames

TABLE IV: Comparison of optical-flow results for our method to [2], [7]. Model [2] is more accurate than the proposed model –lower angular and end-point errors (ae, epe)– because it estimates both forward and backward flows at the price of a more complex model, while our method outperforms the additive model of [7] because our multiplicative model does not combine flow and occlusions variables into a single term where their errors can cancel each other.

	[2]		[7]		proposed model	
	ae	epe	ae	epe	ae	epe
alley_1	**0.12**	**0.55**	0.26	1.27	0.12	0.69
alley_2	0.30	2.48	0.52	3.81	**0.19**	**2.21**
bamboo_1	0.11	0.75	0.18	0.83	**0.09**	**0.48**
bandage_1	0.42	1.63	0.34	1.81	**0.20**	**1.22**
bandage_2	0.32	0.88	0.31	1.31	**0.20**	**0.88**
market_2	0.26	1.64	0.31	1.44	**0.19**	**1.16**
shaman_3	0.72	2.20	0.55	1.97	**0.24**	**1.08**
sleeping_1	0.10	**0.42**	0.30	1.33	**0.08**	0.46
sleeping_2	**0.05**	**0.14**	0.22	0.63	0.06	0.32

in the occlusion detection criterion: the single-frame formulation accumulates occlusions from previous frames and the spatial and temporal regularization eliminates isolated misdetections inconsistent in time or space.

The average processing times for each method in C, in a i7-CPU at 3.4 GHz, are the following: 61 s per frame for our unconstrained model, 65 s for the constrained one, 15 s for the pairwise-flow model with spatio-temporal regularization, 47 s for [7], and 110 s for [2]. As expected, our model is slower than the pairwise flow because the single-frame formulation requires additional interpolation steps, but the occlusions are more accurate. Compared to the literature, our model is slower than [7] because it takes into account the temporal dimension of the problem, but faster than [2] because it uses a more efficient minimization algorithm – convex optimization against the explicit PDE evolution of [2] – and because it does not have to compute forward and backward flows. Consequently, in term of optical flow, the performance of our model is comparable to [2], but considerably better than [7] because the additive model of [7] cannot reliably estimate the flow close to occlusion boundaries. The analysis of flow results is out of the scope of this paper, we summarize it in Table IV.

The penalty assigned to a false alarms or a missed detection depends on the end application: it is

TABLE V: Comparison to exitsing occlusion-detection methods [2], [7] optimized for F0.1 score.

	[2]			[7]			pairwise flow			proposed model		
sequence	ppv	tpr	F.1	ppv	tpr	F.1	ppv	tpr	$F.1$	ppv	tpr	F.1
alley_1	0.06	0.10	0.10	0.01	0.31	0.19	0.10	0.32	0.14	0.18	0.36	**0.35**
alley_2	0.10	0.09	0.09	0.54	0.06	0.06	0.30	0.25	0.20	0.14	0.36	**0.33**
bamboo_1	0.38	0.34	0.34	0.42	0.19	0.19	0.13	0.36	0.18	0.10	0.40	**0.39**
bandage_1	0.17	0.33	0.33	0.00	0.40	0.16	0.02	0.37	0.04	0.12	0.50	**0.47**
bandage_2	0.15	0.35	**0.35**	0.38	0.12	0.12	0.03	0.21	0.06	0.04	0.28	0.26
market_2	0.46	0.34	0.34	0.54	0.23	0.23	0.05	0.26	0.07	0.12	0.48	**0.42**
shaman_3	0.08	0.19	**0.18**	0.00	0.08	0.04	0.11	0.03	0.04	0.15	0.04	0.04
sleeping_1	0.11	0.02	0.02	0.00	0.05	0.01	0.50	0.01	0.01	0.01	0.13	**0.10**
sleeping_2	0.16	0.10	0.10	0.01	0.50	0.24	0.01	0.18	0.02	0.07	0.24	**0.23**

more critical to miss an occlusion for a layer-segmentation model [24], [25], [27], [28] than to incur wrongly detect one, while robotic exploration strategies [42] that aim to discover occluded regions are very sensitive to false alarms. In threshold-based classification, ROC curves measure the performance of a classifier at different conditions by varying the threshold parameter, but in our variational models the curve becomes a 5-dimensional manifold –indexed by $\alpha_s, \alpha_t, \beta, \gamma_s, \gamma_t$– and visualization is impossible. To overcome this issue we also present results for a generalization of the F1 score that assigns different weights to precision and recall. While F1 is designed for a blind detector that penalizes equally a missed detection than a false alarm, and therefore is agnostic to the end application, the proposed F0.1 suits detectors designed to avoid false alarms by assigning a penalty 10 times higher to false positives than false negatives. In this case, our model still outperforms [2], [7], as shown in Table V. The single-frame formulation is still in average beneficial, we outperform [7] in every test, and the comparison to [2] is favorable for 8 out of 10 experiments. For sequences with a high percentage of occlusions, [2] obtains higher F0.1 scores because it is more precise, while for sequences with fewer occlusions our model ranks higher because it is more sensitive.

VII. Conclusions

This paper proposes a variational method for occlusion detection that integrates the brightness constraint of optical flow and detects occlusions as failing regions of the flow model. To emphasize the effects of occlusions, we compute the flow with respect to a single reference frame that collects violations of the brightness constraint by temporal integration; this allows us to distinguish occlusions from noise and non-

Lambertian phenomena by means of spatio-temporal regularizers on the occlusion set. Our experiments show the benefits of a single-frame formulation and the spatio-temporal regularization of occlusions. The potential of our model lies in its variational formulation, which introduces occlusion detection in the core of optical-flow systems and can be used with large displacement techniques. Future work will investigate this integration.

The limitations of our model arise from the assumption that occlusions accumulate and grow smoothly around the occluded object. When occlusions are not temporally consistent in the image domain – due to the combination of the movements of the camera and the scene objects–, as in Figure 3, our formulation under-performs a simple occlusion detector based on pairs of frames. Adapting the sliding window as a function of the estimated displacements would alleviate this problem, but the resulting method would be slower. Future work will investigate such a system.

ACKNOWLEDGEMENT

The authors would like to thank the authors of [2] for sharing their code and D. Davis from UCLA for discussions on optimization. V. Estellers is supported by the Swiss National Science Foundation under grant P2ELP2_148890.

REFERENCES

[1] B. Horn, *Robot Vision*. MIT press, 1986.

[2] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, "Symmetrical Dense Optical Flow Estimation with Occlusions Detection," *Int. J. Comput. Vis.*, vol. 75, no. 3, pp. 371–385, Feb. 2007.

[3] C. Strecha, R. Fransens, and L. V. Gool, "A probabilistic approach to large displacement optical flow and occlusion detection," *Stat. Methods Video Process.*, vol. 3247, pp. 71–82, 2004.

[4] J. Sun and Y. Li, "Symmetric stereo matching for occlusion handling," *IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 399 – 406, 2005.

[5] J. Xiao, H. Cheng, and H. Sawhney, "Bilateral filtering-based optical flow estimation with occlusion detection," *Eur. Conf. Comput. Vis.*, vol. 3951, pp. 211–224, 2006.

[6] S. Ince and J. Konrad, "Occlusion-aware Optical Flow Estimation," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1443–1451, Aug. 2008.

[7] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse Occlusion Detection with Optical Flow," *Int. J. Comput. Vis.*, vol. 97, no. 3, pp. 322–338, Oct. 2011.

[8] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1-3, pp. 185–203, Aug. 1981.

[9] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[10] T. Pock, M. Unger, D. Cremers, and H. Bischof, "Fast and exact solution of Total Variation models on the GPU," *Conf. Comput. Vis. Pattern Recognit. Work.*, no. 1, pp. 1–8, Jun. 2008.

[11] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, “An improved algorithm for TV-L 1 optical flow,” *Stat. Geom. Approaches to Vis. Motion Anal.*, vol. Springer B, pp. 23–45, 2009.

[12] L. Xu, J. Jia, and Y. Matsushita, “Motion Detail Preserving Optical Flow Estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, 2012.

[13] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “DeepFlow : Large displacement optical flow with deep matching,” *Int. Conf. Comput. Vis.*, 2013.

[14] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph cuts,” *IEEE Int. Conf. Comput. Vis.*, vol. 2, pp. 508–515, 2001.

[15] A. Humayun, O. Mac Aodha, and G. J. Brostow, “Learning to find occlusion regions,” *Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 2161–2168, Jun. 2011.

[16] M. Leordeanu, A. Zanfir, and C. Sminchisescu, “Locally Affine Sparse-to-Dense Matching for Motion and Occlusion Estimation,” *Int. Conf. Comput. Vis.*, 2013.

[17] C. Ballester, L. Garrido, V. Lazcano, and V. Caselles, “A TV-L1 Optical Flow Method with Occlusion Detection,” *Pattern Recognition. Lect. Notes Comput. Sci.*, vol. 7476, pp. 31–40, 2012.

[18] M. J. Black and D. J. Fleet, “Probabilistic Detection and Tracking of Motion Boundaries,” *Int. J. Comput. Vis.*, vol. 38, no. 3, pp. 231–245, 2000.

[19] M. E. Sargin, L. Bertelli, B. S. Manjunath, and K. Rose, “Probabilistic occlusion boundary detection on spatio-temporal lattices,” *Int. Conf. Comput. Vis.*, pp. 560–567, Sep. 2009.

[20] A. N. Stein and M. Hebert, “Occlusion Boundaries from Motion: Low-Level Detection and Mid-Level Reasoning,” *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 325–357, Feb. 2009.

[21] A. Ayvaci and S. Soatto, “Motion segmentation with occlusions on the superpixel graph,” in *Int. Conf. Comput. Vis.* Ieee, Sep. 2009, pp. 727–734.

[22] P. Sundberg and T. Brox, “Occlusion boundary detection and figure/ground assignment from optical flow,” in *Int. Conf. Comput. Vis. Pattern Recognit.*, 2011.

[23] X. He and A. Yuille, “Occlusion boundary detection using pseudo-depth,” *Comput. VisionECCV 2010*, pp. 539–552, 2010.

[24] P. Smith, T. Drummond, and R. Cipolla, “Layered motion segmentation and depth ordering by tracking edges.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 479–494, Apr. 2004.

[25] J. D. Jackson, A. J. Yezzi, and S. Soatto, “Dynamic Shape and Appearance Modeling via Moving and Deforming Layers,” *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 71–84, Dec. 2007.

[26] D. Sun, E. Sudderth, and M. Black, “Layered segmentation and optical flow estimation over time,” *Int. Conf. Comput. Vis. Pattern Recongition*, pp. 1768–1775, Jun. 2012.

[27] J. Xiao and M. Shah, “Motion layer extraction in the presence of occlusion using graph cuts.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1644–1659, Oct. 2005.

[28] A. Ayvaci and S. Soatto, “Detachable object detection with efficient model selection,” in *Energy Minimization Methods Comput. Vis. Pattern Recognit.*, 2011, pp. 191–204.

[29] C. Vogel, S. Roth, and K. Schindler, “An Evaluation of Data Costs for Optical Flow,” in *Pattern Recognit. SE - 37*, ser. Lecture Notes in Computer Science, J. Weickert, M. Hein, and B. Schiele, Eds. Springer Berlin Heidelberg, 2013, vol. 8142, pp. 343–353.

[30] J. J. Gibson, “The Perception of the Visual World.” 1950.

[31] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." in *Imaging Underst. Work.*, vol. 130, 1981, pp. 121–130.

[32] H.-H. Nagel, "Extending the 'Oriented Smoothness Constraint' into the temporal domain and the Estimation of Derivatives of Optical Flow," in *Eur. Conf. Comput. Vis.*, 1990, pp. 139–148.

[33] T. Chin, W. Karl, and A. S. Willsky, "Probabilistic and sequential computation of optical flow using temporal coherence," *IEEE Trans. Image Process.*, vol. 3, no. 6, pp. 773–788, 1994.

[34] A. Salgado and J. Sánchez, "Optical Flow Estimation with Large Displacements : A Temporal Regularizer," Cuadernos del Instituto Universitario de Ciencias y Tecnologías Cibernéticas, Las Palmas, España, Tech. Rep. April, 2006.

[35] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, "Modeling temporal coherence for optical flow," *2011 Int. Conf. Comput. Vis.*, pp. 1116–1123, Nov. 2011.

[36] R. Garg, L. Pizarro, D. Rueckert, and L. Agapito, "Dense multi-frame optic flow for non-rigid objects using subspace constraints," *Asian Conf. Comput. VisionACCV*, pp. 460–473, 2011.

[37] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," *Eur. Conf. Comput. Vis.*, vol. 4, no. May, pp. 25–36, 2004.

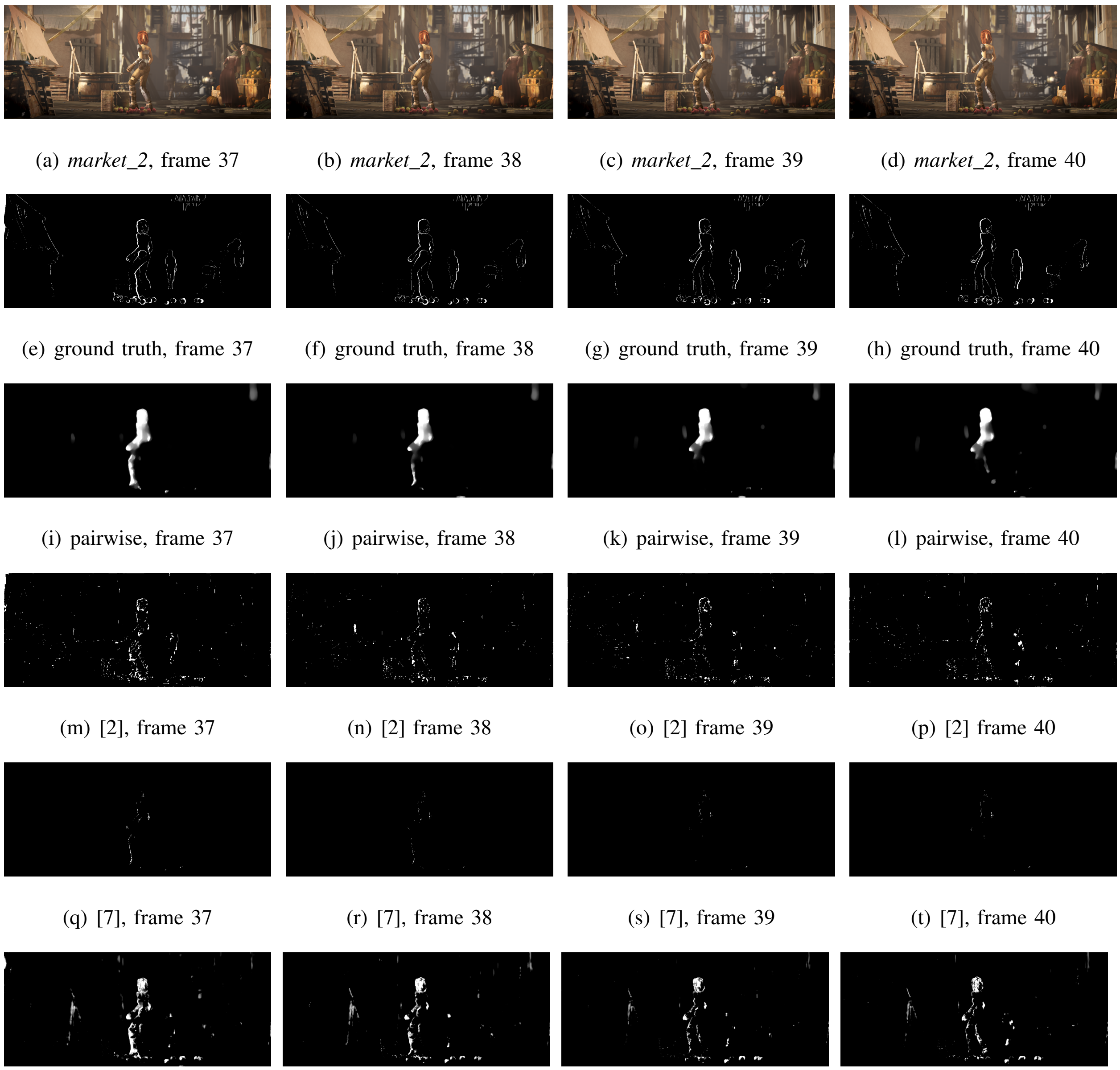
[38] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems withApplications to Imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, Dec. 2010.

[39] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1997, no. 28.

[40] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models," *SIAM J. Appl. Math.*, vol. 66, no. 5, pp. 1632–1648, 2006.

[41] D. Butler, J. Wulff, G. Stanley, and M. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," in *Comput. Vis. ECCV 2012 SE - 44*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Springer Berlin Heidelberg, 2012, vol. 7577, pp. 611–625.

[42] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," pp. 417–433, 1993.



(a) *market_2*, frame 37 (b) *market_2*, frame 38 (c) *market_2*, frame 39 (d) *market_2*, frame 40

(e) ground truth, frame 37 (f) ground truth, frame 38 (g) ground truth, frame 39 (h) ground truth, frame 40

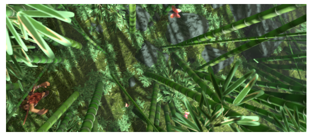
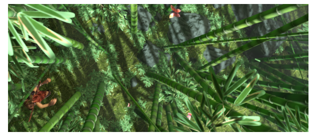
(i) pairwise, frame 37 (j) pairwise, frame 38 (k) pairwise, frame 39 (l) pairwise, frame 40

(m) [2], frame 37 (n) [2] frame 38 (o) [2] frame 39 (p) [2] frame 40

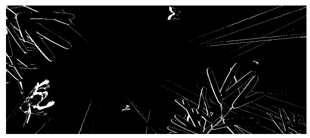
(q) [7], frame 37 (r) [7], frame 38 (s) [7], frame 39 (t) [7], frame 40

(u) our, frame 37 (v) our, frame 38 (w) our, frame 39 (x) our, frame 40

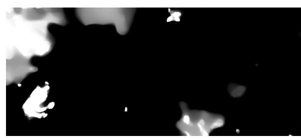
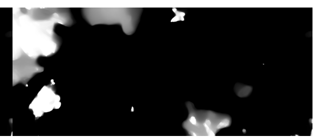
Fig. 2: Comparison of occlusions detection methods for *market_2* sequence. Our single-frame formulation detects more occlusions caused by smooth movements, while models based in pairwise flows miss these occlusions. Our model is more robust to isolated false alarms because of the spatial regularization.

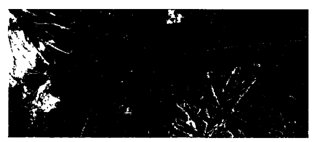
(a) *bamboo_1*, frame 3 (b) *bamboo_1*, frame 4

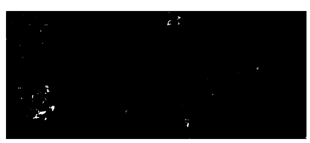
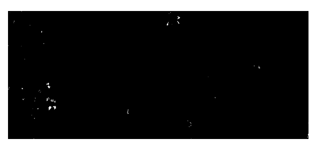
(c) ground truth, frame 3 (d) ground truth, frame 4

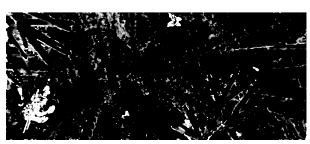
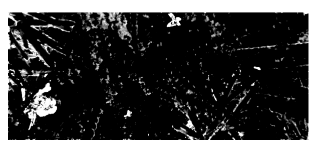
(e) pairwise, frame 3 (f) pairwise, frame 4

(g) [2] frame 3 (h) [2] frame 4

(i) [7], frame 3 (j) [7], frame 4

(k) our, frame 3 (l) our, frame 4

Fig. 3: Failure case, *bamboo_1* sequence. The combination of the bamboo oscillations and the movement of the camera result in non-smooth displacements that do not match our temporal regularizers. Our single-frame formulation over-detects occlusions in comparison to the forward-backward model [2], which neglects the temporal dimension of the problem.