Bittracker – A Bitmap Tracker for Visual Tracking under Very General Conditions

Ido Leichter, Michael Lindenbaum, and Ehud Rivlin
Abstract

This paper addresses the problem of visual tracking under very general conditions: 1. potentially non-rigid target whose appearance may drastically change over time; 2. general camera motion; 3. 3D scene; 4. no a priori information, but the target’s bitmap as initialization in the first frame. The vast majority of trackers, aiming to gain speed and robustness, work in some limited context (e.g., the target’s appearance or shape is a priori known or restricted to some extent, the scene is planar or a pan-tilt-zoom camera is used.) Therefore, although effective, these trackers may fail in the more general case. To this end, a novel tracker that does not rely on specific assumptions regarding the target, scene or camera motion, is suggested.

The tracking is conducted by estimating in each frame the maximum a posteriori bitmap of a probability distribution function of the target’s bitmap, conditioned on the current and previous frames and on the previous target’s bitmap.

Several experiments demonstrate the capability of tracking, from an arbitrarily moving camera, non-rigid objects whose appearance is drastically changing due to varying pose and configuration, varying illumination and partial occlusions.

Index Terms

Tracking, Motion, Pixel classification.

I. INTRODUCTION

This paper is concerned with the visual tracking under very general conditions. Specifically, the target may be non-rigid and of appearance drastically changing over time, the camera undergoes general motion, the scene is 3D, and no information regarding the target or the scene is a priori given except initialization by the target’s bitmap in the first frame.

Much research has been done in the field of visual tracking, bearing fruit to an abundance of visual trackers, but very few trackers were aimed for such a general context. The vast majority of existing trackers, in order to reduce computational load and enhance robustness, are restricted to some a priori known context by using some (possibly updatable [9]) appearance model or shape model for the tracked object in the images and tracking in a low-dimensional state-space of the target’s parameters (e.g., via CONDENSATION [7].) For example, in [23] it is both assumed that the color histogram of the target does not change very much over time and that the target’s 2D shape in the image may change only in scale. These trackers are effective only in the context for
which they were designated. That is, as long as the target obeys the tracker’s appearance model and its shape modeling, the tracking may be robust. However, once the target does not obey the tracker’s appearance or shape model (due to un-modeled factors such as object deformations, change in viewing direction, partial occlusions, spatially or temporally varying lighting, etc.), the tracking is likely to fail without recovery. Thus, such context-specific trackers are not suitable under the aforementioned general conditions.

The seminal results by Julesz have shown that humans are able to visually track objects merely by clustering regions of similar motion [10]. As an extreme case, consider images (a) and (b) in Figure 1. These two images constitute a consecutive pair of images in a video displaying a random-dot object moving in front of a random-dot background undergoing a different motion. Since the patterns on the object, on the background and around the object’s contour are all alike, the object is indistinguishable from the background for the observer who is revealed to these images at nonconsecutive times. However, given these two images one just after the other in the same place, like in a video, the observer is able to extract the object in the two images, shown in images (c) and (d), respectively.

![Figure 1](image)

**Fig. 1.** (a) and (b) constitute a consecutive pair of images in a video of a random-dot object moving in front of a random-dot background undergoing a different motion. The object in the two images is shown in (c) and (d), respectively. See text for details.

Founded on this observation, this paper suggests an algorithm (Bittracker) for visual tracking, based merely on three conservative assumptions:

**Short-time Constancy of Color** – The color projected to the camera from a point on a surface is approximately similar in consecutive frames;

**Spatial Motion Continuity** – The optical flow in the image region corresponding to an object is spatially piecewise continuous. That is, the optical flow of the vast majority of the pixels in this area is spatially continuous;
Spatial Coherence – Adjacent pixels of similar color belong to the same object with high probability.

The first two assumptions usually hold under sufficiently high frame rate, and the third holds for natural images.

In order to track non-rigid objects of general shape and motion without a priori knowledge of their shape, the tracker proposed in this paper uses the state-space of bitmaps classifying each pixel in the image whether it belongs to the tracked object or not (hence the tracker’s name.) Note that this state-space is even more general than the state-space of non-parametric contours, since the former may also accommodate for holes in the tracked object. As no specific target-related, scene-related or camera motion-related assumptions are made, the resulting tracker is suitable for tracking under the aforementioned very general conditions.

The tracking is conducted by estimating in each frame the MAP (maximum a posteriori) estimate of a PDF (probability distribution function) of the target’s bitmap in the current frame, conditioned on the current and previous frames and on the bitmap in the previous frame. A lossless decomposition of the information in the image into color information and pixel-location information enables to separately treat colors and motion in a systematic way for the construction of the PDF.

One important advantage of the proposed tracker is that instead of clinging to a sole optical flow hypothesis, the target’s bitmap PDF is marginalized over all possible motions per pixel. In contrast, other general-context trackers perform optical flow estimation, which is prone to error and is actually a harder, more general problem than the mere object tracking, or do not use the motion cue at all.

Another advantage of the proposed algorithm over other general-context trackers is that the target’s bitmap PDF is formulated directly in the pixel-level (unlike image segments). Thus, the precursory confinement of the final solution to objects composed of preliminarily-computed image segments is avoided.

Related work is reviewed in a comparable view to this one in Section II. Section III outlines the proposed algorithm. Experimental results are given in Section IV, and Section V concludes the paper.
II. PREVIOUS WORK

As this paper is concerned in tracking general objects whose appearance may drastically change over time, where no information about them is given (except the initializing bitmap), the relevant previous work is mainly in the area of video segmentation. However, very few video segmentation algorithms are aimed for the very general context discussed here.

Many video segmentation algorithms were developed in the context of a stationary camera (e.g., [13], [20], [31]) or under the assumption that the background has a global, parametric motion (e.g., affine [24] or projective [28], [30].) Recently, the last restriction was relaxed to a planar scene with parallax [11]. Other algorithms were constrained to track video objects modeled well by parametric shapes (e.g., active blobs [25]) or motion (e.g., translation [2], 2D rigid motion [28], affine [3], [21], projective [6], small 3D rigid motion [19] and Normally distributed optical flow [12], [29].) These algorithms are suitable only for tracking rigid objects or specific preset types of deformations. The algorithm proposed in this paper, however, addresses the tracking of potentially non-rigid objects in 3D scenes from an arbitrarily moving camera, without a priori knowledge except from the object’s bitmap in the first frame.

There are papers that address video segmentation and achieve object tracking under general conditions as an aftereffect. That is, they do not perform explicit tracking in the sense of estimating the current state conditioned on the previous one and on the previous frame(s). For example, in [26] each set of a few (five) consecutive frames is spatiotemporally segmented without consideration of the previous results (except for saving calculations.) In [14] each frame is segmented into object/background without considering previous frames or classifications. (Furthermore, the classification requires a training phase, upon which the classification is performed, prohibiting the targets’ appearances from undergoing major changes.) In the contour tracking performed in [8], an active contour is run in each frame separately, while the only information taken from previous time is the previously estimated contour for initialization in the current frame. In this paper, the state (target’s bitmap) is explicitly tracked by approximating a PDF of the current state, conditioned on the pervious state and on the current and previous frames, and estimating the MAP state.

Optical flow is an important cue for visually tracking objects, especially under general conditions. Most video segmentation algorithms make a point-estimate of the optical flow, usually
prior to the segmentation (e.g., [2], [3], [6], [12], [16], [17], [19], [22], [29], [30]) and seldom in conjunction with the segmentation (e.g., [21].) An exception is [18], where each pixel may be assigned multiple flow vectors of equal priority. However, the segmentation there is only applied to consecutive image pairs, and in all three experiments the objects were rigid and either the camera or the entire scene was static. Since optical flow estimation is prone to error, other algorithms avoid its use altogether (e.g., [8], [14], [15], [27].) The latter kind of algorithms tend to fail when the target is in proximity to areas of similar texture, and may erroneously classify newly appearing regions with different textures. This is shown in an example in [15], where occlusions and newly appearing areas are prohibited due to the modeling of image domain relations as bijections. An exception in the way optical flow is used is [26], where a motion profile vector that captures the probability distribution of image velocity is computed per pixel, and motion similarity of neighboring pixels is approximated based on the resemblance of their motion profiles. In the work here, the optical flow is not estimated as a single hypothesis nor discarded, but a marginalization over all possible pixel motions (under a maximal flow assumption) is performed in the bitmap’s PDF construction.

There is a class of video segmentation and tracking algorithms in the context of an arbitrarily moving camera that aim at coping with general object shapes and motions by tracking a non-parametric contour influenced by intensity/color edges (e.g., [27]) and motion edges (e.g., [17].) However, this kind of algorithms does not deal well with cluttered objects and partial occlusions, and may be distracted by color edges or additional moving edges in proximity to the tracked contour.

Many video segmentation and tracking algorithms perform a spatial segmentation of each frame into segments of homogeneous color/intensity as a preprocessing step in order to use these segments as atomic regions composing objects (e.g., [2], [3], [21].) As a consequence, they also assign a parametric motion per segment. Instead of performing such a confinement of the final solution in a preprocessing and making assumptions regarding the type of motion the segments undergo, the algorithm proposed here uses the aforementioned Spatial Coherence Assumption and works directly in the pixel-level.
III. THE Bittracker ALGORITHM

A. Overview

Every pixel in an image may be classified as belonging to some particular object of interest or not according to the object projected to the pixel’s center point. Bittracker aims at classifying each pixel in the movie frame whether it belongs to the tracked object or not. Thus, the tracker’s state-space is binary images, i.e., bitmaps. The tracking is performed by estimating the tracked object’s bitmap at time $t$, given the movie frames at times $t$ and $t-1$ and the estimate of the previous bitmap at time $t-1$ ($t = 0, 1, 2, \ldots$) Thus, after being initialized by the tracked object’s bitmap in the first frame (for $t = 0$), the tracker causally propagates the bitmap estimate in time in an iterative fashion. At each time $t$, the tracker approximates the PDF of the bitmap $X_t$

$$P(X_t) = \Pr(X_t|I_{t-1}, I_t, X_{t-1}) \quad (1)$$

($I_t$ denotes the frame at time $t$), followed by estimating the MAP bitmap by maximizing its PDF. The estimated MAP bitmap may then be used to estimate the tracked object’s bitmap at time $t + 1$ by the same way, and so forth. Note that the initializing bitmap $X_0$ need not to be exact, as the tracked object’s bitmap may be self-corrected with time based on the assumptions of Spatial Motion Continuity and Spatial Coherence, which are incorporated in (1).

Formulating the bitmap tracking problem as in (1), the solution is targeted directly towards the sought bitmap. Thus, the commonly performed intermediate step of optical flow estimation is avoided. This is an important advantage since computing optical flow is a harder, more general problem, than estimating the bitmap (given the one in the previous time).

B. The Bitmap’s PDF

Modeling the bitmap’s PDF (1) is very complex. In order to simplify the modeling, this PDF is factored into a product of two simpler PDFs. To this end, instead of regularly considering a discrete image $I$ as a matrix of color values representing the pixel colors at the corresponding coordinates, we consider it here as a set of pixels with indices $p = 1, 2, \ldots, |I|$ ($|I|$ denotes the number of pixels in image $I$), each one having a particular color $c_p$ and location $l_p$ (coordinates). (The pixel indexing is arbitrary and unrelated to its location in the image. To remove any doubt, there is no connection between the indexing of the pixels in $I_t$ and the indexing in $I_{t-1}$. Specifically, if a pixel of index $p$ in $I_t$ and a pixel of index $p'$ in $I_{t-1}$ are such that $p = p'$, it
does not imply that the two pixels are related by their colors or locations.) Taking this alternative view, a discrete image \( \mathcal{I} \) may be decomposed into the pair \( \mathcal{I} = (C, L) \), where \( C = \{c^p\}_{p=1}^{\left|\mathcal{I}\right|} \) and \( L = \{l^p\}_{p=1}^{\left|\mathcal{I}\right|} \). Note that no information is lost because the image may be fully reconstructed from its decomposition. This enables us to decompose \( \mathcal{I}_t \) into \( \mathcal{I}_t = (C_t, L_t) \). Therefore, the bitmap’s PDF (1) may be written as

\[
P (X_t) = \Pr (X_t | I_{t-1}, C_t, L_t, X_{t-1}) .
\]

(2)

Applying Bayes’ rule to (2), the bitmap’s PDF may be factored into

\[
P (X_t) \propto \Pr \left( X_t | I_{t-1}, C_t, X_{t-1} \right) \cdot \Pr \left( L_t | X_t, I_{t-1}, C_t, X_{t-1} \right) .
\]

(3)

As will be seen in the sequel, these two components are easier to model due to the separation of the color information from the location information.

We denote the boolean random variable representing the bitmap’s value at pixel \( p \) in \( \mathcal{I}_t \) by \( x_t^p \) that may receive one of the following values:

\[
x_t^p = \begin{cases} 
1 & \text{Pixel } p \text{ in } \mathcal{I}_t \text{ belongs to the tracked object}, \\
0 & \text{otherwise}.
\end{cases}
\]

(4)

Note that the notation \( X_t \) is the abbreviation of \( \{x_t^p\}_{p=1}^{\left|\mathcal{I}_t\right|} \).

1) Modeling \( F_1(X_t) \): The first factor in (3), \( \Pr \left( X_t | I_{t-1}, C_t, X_{t-1} \right) \), is the PDF of the tracked object’s bitmap in time \( t \) when considering in the \( t \)-th frame only the pixels’ colors and disregarding their coordinates. Not given \( L_t \), there is zero information on the motion from frame \( t-1 \) to frame \( t \), as well as on the relative position between the pixels in \( \mathcal{I}_t \). Under these circumstances, the dependency of the bitmap’s bits on the pixels’ colors is much stronger than the dependency between the bits themselves. That is, a pixel may be decided as object/non-object mostly by examining its color with respect to the colors of already-classified pixels in the previous frame. Therefore, it is reasonable to approximate the bitmap’s bits as independent, conditioned on the pixels’ colors:

\[
F_1(X_t) = \prod_{p=1}^{\left|\mathcal{I}_t\right|} \Pr \left( x_t^p | I_{t-1}, C_t, X_{t-1} \right) .
\]

(5)

In practice the optical flow may be such that a pixel in \( \mathcal{I}_t \) does not exactly correspond to a single pixel in \( \mathcal{I}_{t-1} \). Yet in our model, the correspondence of a pixel \( p \) in \( \mathcal{I}_t \) is either to some

March 30, 2006
pixel $p'$ in $\mathcal{I}_{t-1}$ (denoted by $p \rightarrow p'$) or to a surface that was not visible in time $t-1$ (denoted $p \rightarrow \text{none}$.) Approximating the optical flow by integer shifts in both axes is common in visual tracking and segmentation applications (e.g., [26].) Now, the PDF of a single bit $x^p_t$ inside the product in (5) may be marginalized over all the potential correspondences of $\mathcal{I}_t$’s pixel $p$ to pixels $p'$ in $\mathcal{I}_{t-1}$, including the event of it corresponding to none:

$$f_1(x^p_t) = \sum_{p' \in \mathcal{N}_{t-1} \cup \{\text{none}\}} \Pr(x^p_t \mid p \rightarrow p', \mathbb{C}_t, X_{t-1})$$

where $\mathcal{N}_t$ denotes the set $\{1, 2, \ldots, |\mathcal{I}_t|\}$. Note that by this marginalization any hard decision of the optical flow is avoided.

Modeling the color of a pixel $p$ in $\mathcal{I}_t$ as normally distributed with mean equal to the color of the corresponding pixel $p'$ in $\mathcal{I}_{t-1}$ or as uniformly distributed for pixels corresponding to none, yields (after a detailed derivation described in Appendix I)

$$f_1(x^p_t) \propto (1 - P_{\text{none}}) \cdot \sum_{p' \in \mathcal{N}_{t-1}} \Pr(x^p_t \mid p \rightarrow p', x^p_{t-1}) \cdot \frac{1}{|\mathcal{I}_{t-1}|} \cdot N_{\mu, C}^p(c^p_t) + P_{\text{none}} \cdot \Pr(x^p_t \mid p \rightarrow \text{none}) \cdot U(c^p_t),$$

where $N_{\mu, C}$ is the Normal PDF of mean $\mu$ and covariance matrix $C$ ($C$ is set to a diagonal matrix, where the variances reflect the degree of color similarity assumed by the Constancy of Color Assumption), and $U$ is the uniform PDF on the color-space (RGB in our implementation.) $P_{\text{none}}$ is a preset constant, estimating the prior probability of having no corresponding pixel in the previous frame (typically set to 0.1, but as explained in Appendix I, has only minor influence on the tracker).

We see that $f_1(x^p_t)$ may be viewed as a mixture distribution with a component for having a corresponding pixel in the previous frame (with weight $1 - P_{\text{none}}$) and a component for having no corresponding pixel (with weight $P_{\text{none}}$.)

$$\Pr(x^p_t \mid p \rightarrow p', x^p_{t-1})$$

is the probability distribution of the bitmap’s bit at a pixel $p$, when its corresponding pixel in the previous frame, along with its estimated classification bit, are known. Since the MAP bitmap estimated for the previous frame may contain errors, we set this PDF to

$$\Pr(x^p_t \mid p \rightarrow p', x^p_{t-1}) = \begin{cases} P_{\text{correct}} & x^p_t = x^p_{t-1}, \quad p' \in \mathcal{N}_{t-1}, \\ 1 - P_{\text{correct}} & x^p_t \neq x^p_{t-1} \end{cases}$$

March 30, 2006 DRAFT
where \( P_{\text{correct}} \) is a preset constant as well approximating the probability of the estimated bitmap being correct for a pixel (typically set to 0.9.)

\[
\Pr (x^p_t | p \rightarrow \text{none}) \text{ is the prior probability distribution of the bitmap’s bit at a pixel } p \text{ with no corresponding pixel in the previous frame. This probability distribution is set to}
\]

\[
\Pr (x^p_t | p \rightarrow \text{none}) = \begin{cases} 
P_{\text{object}} & x^p_t = 1 \\
1 - P_{\text{object}} & x^p_t = 0 
\end{cases},
\]

where \( P_{\text{object}} \) is another preset constant (typical value of 0.4.)

While the location information \( L_t \) is not used at all for deriving (7) (as the conditioning is on \( C_t \) only), in practice we calculate (7) with two modifications, using pixel location information in a limited way: 1. Note that in (7) pixel correspondences are evaluated through the Gaussian component, comparing pixel colors in \( I_t \) to pixel colors in \( I_{t-1} \). In practice, in order to make the pixel correspondence distributions less equivocal, we compare small image patches (of diameter of 5 pixels) centered around the candidate pixels instead of merely the candidate pixels themselves. This is accounted for by modifying the normal and uniform PDFs in Equation (7) to products of the color PDFs of the pixels in the patches (see Appendix I for details.) 2. We restrict the maximal size of optical flow to \( M \) pixels (in our implementation \( M = 6 \)), and thus compare only between image patches distanced at most by \( M \) and sum over these correspondences only (137 potential correspondences per pixel), which reduces computations.

2) Modeling \( F_2 (X_t) \): The second factor in (3), \( \Pr (L_t | X_t, I_{t-1}, C_t, X_{t-1}) \), is the likelihood function of the pixels’ coordinates (where their colors, as well as the previous frame with its corresponding bitmap are known.) Given \( I_{t-1} \) and \( C_t \), PDFs of pixel correspondences between \( I_t \) and \( I_{t-1} \) are induced (similarly to \( F_1 (X_t) \).) Based on these correspondence PDFs, \( L_t \) induces PDFs of optical flow between these two frames. By the Spatial Motion Continuity Assumption, for an adjacent pair of pixels in a region belonging to a single object (where the optical flow is spatially continuous), the discrete optical flow is very likely to be the same, and for an adjacent pair of pixels belonging to different objects, the optical flow is likely to differ. Thus, the likelihood of an unequal bit-assignment to similarly-moving adjacent pixels should be much lower than an equal bit-assignment, and vice versa for differently-moving adjacent pixels. By the Spatial Coherence Assumption, the likelihood of an equal bit-assignment to similarly-colored adjacent pixels should be much higher than an unequal bit-assignment.
Taking this view and noting that \( \mathcal{L}_t \) determines pixel adjacency in \( \mathcal{I}_t \) and pixel motion from time \( t-1 \) to time \( t \), we model \( F_2(\mathcal{X}_t) \) as a Gibbs distribution with respect to the first-order neighborhood system [4],

\[
F_2 (\mathcal{X}_t) \propto \prod_{\text{unordered pairs}} f_2 (x_{t}^{p_1}, x_{t}^{p_2}),
\]

where \( \Delta_{t}(p_1,p_2) \triangleq l_t^{p_1} - l_t^{p_2} \) and \( \text{adj}(p_1,p_2) \) is the event of pixels \( p_1 \) and \( p_2 \) being adjacent (\( \|l_t^{p_1} - l_t^{p_2}\|_2 = 1 \)).

We shall begin with the first multiplicand in the right-hand-side of (11). By Bayes rule

\[
f_{\text{adj}} (x_{t}^{p_1}, x_{t}^{p_2}) = \frac{\Pr (\text{adj}(p_1,p_2) | x_{t}^{p_1}, x_{t}^{p_2}, c_t)}{f_{\text{adj}} (x_{t}^{p_1}, x_{t}^{p_2})} \cdot \frac{\Pr (\text{adj}(p_1,p_2) | x_{t}^{p_1}, x_{t}^{p_2}, \mathcal{I}_{t-1}, \mathcal{C}_{t-1})}{\Pr (\Delta_{t}(p_1,p_2) | x_{t}^{p_1}, x_{t}^{p_2}, \mathcal{I}_{t-1}, \mathcal{C}_{t-1})},
\]

Not assuming any prior information on the object shape and on the object/non-object color distribution, the influence of the bitmap bits on \( f_{\text{adj}} (x_{t}^{p_1}, x_{t}^{p_2}) \) is dominated by the first multiplicand, thus we approximate

\[
f_{\text{adj}} (x_{t}^{p_1}, x_{t}^{p_2}) \propto p (c_t^{p_1}, c_t^{p_2} | x_{t}^{p_1}, x_{t}^{p_2}, \text{adj}(p_1,p_2)) \cdot \frac{\Pr (\text{adj}(p_1,p_2) | x_{t}^{p_1}, x_{t}^{p_2})}{p (c_t^{p_1}, c_t^{p_2} | x_{t}^{p_1}, x_{t}^{p_2})}.
\]

Applying the chain rule yields

\[
f_{\text{adj}} (x_{t}^{p_1}, x_{t}^{p_2}) \propto p (c_t^{p_1} | x_{t}^{p_1}, x_{t}^{p_2}, \text{adj}(p_1,p_2)) \cdot p (c_t^{p_2} | x_{t}^{p_2}, x_{t}^{p_2}, \text{adj}(p_1,p_2)).
\]

The first multiplicand in the right-hand-side does not depend on the bitmap bits, which leaves only the second multiplicand that we model as

\[
f_{\text{adj}} (x_{t}^{p_1}, x_{t}^{p_2}) \propto \begin{cases} U (c_t^{p_2}) + N_{t,c_t^{p_1},c_{\text{adj}}} (c_t^{p_2}) & \text{if } x_{t}^{p_1} = x_{t}^{p_2} \\ U (c_t^{p_2}) & \text{if } x_{t}^{p_1} \neq x_{t}^{p_2}. \end{cases}
\]
This corresponds to modeling the colors of adjacent pixels as uniformly and independently
distributed in case they belong to different objects. In case these pixels belong to the same
object, their color distribution is a mixture of a uniform distribution (corresponding to the case
of belonging to different color segments) and a Gaussian in their color difference (corresponding
to the case of belonging to the same segment of homogeneous color.) \( C_{adj} \) is assigned very small
variances, reflecting the variance of the color differences between adjacent pixels belonging to
a surface of homogeneous color. (In our implementation it was set to 0.01 for each RGB color
channel, where the range of each color is [0,1].) We see that for differently-colored adjacent pixels
the likelihood is approximately similar for equal and unequal bit-assignments, and for similarly-
colored adjacent pixels the likelihood is much higher for equal bit-assignments, which is in
correspondence with the Spatial Coherence Assumption. Equation (15) may be used to compute
(up to a common, unimportant scaling) the four likelihoods \{ \( f_{adj} (x_{i}^{p}, x_{j}^{q}) \) \}_{b_1, b_2 \in \{0,1\}}.

We turn now the second multiplicand in the right-hand-side of (11), \( f_{\Delta} (x_{i}^{p}, x_{i}^{q}) \). After a
detailed derivation given in Appendix II,

\[
f_{\Delta} (x_{i}^{p}, x_{i}^{q}) = \begin{cases} 
P_{flow_1} \cdot S_1 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) + (1 - P_{flow_1}) \cdot S_2 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) & x_{i}^{p} = x_{i}^{q}, \\
+ 0.25 \cdot S_3 (x_{i}^{p}, x_{i}^{q}; p_1, p_2), & x_{i}^{p} \neq x_{i}^{q}, \\
(1 - P_{flow_2}) \cdot S_1 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) + P_{flow_2} \cdot S_2 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) & x_{i}^{p} = x_{i}^{q}, \\
+ 0.25 \cdot S_3 (x_{i}^{p}, x_{i}^{q}; p_1, p_2), & x_{i}^{p} \neq x_{i}^{q},
\end{cases}
\]  

(16)

where \( S_1 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) \) is the probability that \( I_i \)‘s pixels \( p_1 \) and \( p_2 \) have identical discrete
optical flow from the previous frame, \( S_2 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) \) is the probability of having different
discrete optical flow, and \( S_3 (x_{i}^{p}, x_{i}^{q}; p_1, p_2) \) is the probability that at least one of the two
pixels has no corresponding pixel in the previous frame (and thus has no optical flow.) All these
probabilities are conditional on the two pixel’s classification bits, \( C_i \) and the previous frame along
with its estimated bitmap. (See Appendix II for these probabilities estimation method.) \( P_{flow_1} \)
is a predefined constant approximating the probability that two equally classified, adjacent pixels
have similar discrete optical flows from previous frame (given that the corresponding pixels exist.)
\( P_{flow_2} \) is another predefined constant approximating the probability that two unequally classified,
adjacent pixels have different discrete optical flows from previous frame. Both constants have a
Examining (16), we see that the higher the probability of identical discrete optical flows, the higher the likelihood for \( x_{p1}^{t} = x_{p2}^{t} \), and vice versa for the probability of different discrete optical flows, conforming to the Spatial Motion Continuity Assumption. When at least one of the pixels has no corresponding pixel in the previous frame, there is no preference for any bit assignments, since the optical flow is undefined.

3) The Final Bitmap PDF: The multiplicands in (5) and in (10) may be written as

\[
f_1(x_t^p) = c_1(p, t)x_t^p, \quad f_2(x_t^{p1}, x_t^{p2}) = c_3(p_1, p_2, t)x_t^{p1}x_t^{p2} + c_4(p_1, p_2, t)x_t^{p1} + c_5(p_1, p_2, t)x_t^{p2} + c_6(p_1, p_2, t),
\]

where

\[
c_1(p, t) = f_1(x_t^p = 1) - f_1(x_t^p = 0),
\]
\[
c_2(p, t) = f_1(x_t^p = 0),
\]
\[
c_3(p_1, p_2, t) = f_2(x_t^{p1} = 1, x_t^{p2} = 1) - f_2(x_t^{p1} = 1, x_t^{p2} = 0) - f_2(x_t^{p1} = 0, x_t^{p2} = 1) + f_2(x_t^{p1} = 0, x_t^{p2} = 0),
\]
\[
c_4(p_1, p_2, t) = f_2(x_t^{p1} = 1, x_t^{p2} = 0) - f_2(x_t^{p1} = 0, x_t^{p2} = 0),
\]
\[
c_5(p_1, p_2, t) = f_2(x_t^{p1} = 0, x_t^{p2} = 1) - f_2(x_t^{p1} = 0, x_t^{p2} = 0),
\]
\[
c_6(p_1, p_2, t) = f_2(x_t^{p1} = 0, x_t^{p2} = 0).
\]

Substituting (17) into (5) and (10) the bitmap’s PDF (3) is finally

\[
P (\mathcal{X}_t) \propto \prod_{p=1}^{[I_t]} \left[ c_1(p, t)x_t^p + c_2(p, t) \right] \cdot \prod \left[ c_3(p_1, p_2, t)x_t^{p1}x_t^{p2} \right.
\]
\[
\left. \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quan
Since the logarithm is a monotonically increasing function,

\[ \mathcal{X}_t^{\text{MAP}} = \arg \max_{\mathcal{X}_t} \sum_{p=1}^{\vert \mathcal{I}_t \vert} \ln \left( \frac{c_1(p, t) x_t^p + c_2(p, t)}{c_2(p, t)} \right) \]

\[ + \sum_{\text{unordered pairs } p_1, p_2 \in \mathcal{N}_t \text{ of adjacent pixels in } \mathcal{I}_t} \ln \left( \frac{c_3(p_1, p_2, t) x_t^{p_1} x_t^{p_2} + c_4(p_1, p_2, t) x_t^{p_1}}{c_6(p_1, p_2, t)} \right) \]

\[ + \frac{c_5(p_1, p_2, t) x_t^{p_1} x_t^{p_2} + c_6(p_1, p_2, t)}{c_6(p_1, p_2, t)} \]  (21)

Due to the fact that the variables in the objective function are 0-1,

\[ \mathcal{X}_t^{\text{MAP}} = \arg \max_{\mathcal{X}_t} \sum_{p=1}^{\vert \mathcal{I}_t \vert} \ln \left( \frac{c_1(p, t) + c_2(p, t)}{c_2(p, t)} \right) x_t^p + \ln c_2(p, t) \]

\[ + \sum_{\text{unordered pairs } p_1, p_2 \in \mathcal{N}_t \text{ of adjacent pixels in } \mathcal{I}_t} \ln \left( \frac{c_4(p_1, p_2, t) + c_5(p_1, p_2, t) + c_6(p_1, p_2, t)}{c_6(p_1, p_2, t)} \right) \]

\[ \frac{c_3(p_1, p_2, t) + c_5(p_1, p_2, t) + c_6(p_1, p_2, t)}{c_6(p_1, p_2, t)} x_t^{p_1} x_t^{p_2} \]

\[ + \ln \left( \frac{c_3(p_1, p_2, t) + c_6(p_1, p_2, t)}{c_6(p_1, p_2, t)} \right) x_t^{p_1} + \ln \left( \frac{c_5(p_1, p_2, t) + c_6(p_1, p_2, t)}{c_6(p_1, p_2, t)} \right) x_t^{p_2} \]

\[ + \ln \left( \frac{c_4(p_1, p_2, t) + c_6(p_1, p_2, t)}{c_6(p_1, p_2, t)} \right) x_t^{p_1} x_t^{p_2} + \ln c_6(p_1, p_2, t) \]  (21)

\[ = \arg \max_{\mathcal{X}_t} \sum_{\text{unordered pairs } p_1, p_2 \in \mathcal{N}_t \text{ of adjacent pixels in } \mathcal{I}_t} \ln \left( \frac{f_2(x_t^{p_1} = 1, x_t^{p_2} = 1) \cdot f_2(x_t^{p_1} = 0, x_t^{p_2} = 0)}{f_2(x_t^{p_1} = 1, x_t^{p_2} = 0) \cdot f_2(x_t^{p_1} = 0, x_t^{p_2} = 1)} \right) x_t^{p_1} x_t^{p_2} \]

\[ + \ln \left( \frac{f_2(x_t^{p_1} = 1, x_t^{p_2} = 0)}{f_2(x_t^{p_1} = 0, x_t^{p_2} = 0)} \right) x_t^{p_1} + \ln \left( \frac{f_2(x_t^{p_1} = 0, x_t^{p_2} = 1)}{f_2(x_t^{p_1} = 0, x_t^{p_2} = 0)} \right) x_t^{p_2} \]

\[ + \ln \left( \frac{f_1(x_t^p = 1)}{f_1(x_t^p = 0)} \right) x_t^p \]  (22)

After gathering common terms in the resulting polynomial we obtain

\[ \mathcal{X}_t^{\text{MAP}} = \arg \max_{\mathcal{X}_t} \sum_{\text{unordered pairs } p_1, p_2 \in \mathcal{N}_t \text{ of adjacent pixels in } \mathcal{I}_t} \hat{c}_1(p_1, p_2, t) x_t^{p_1} x_t^{p_2} + \sum_{p=1}^{\vert \mathcal{I}_t \vert} \hat{c}_2(p, t) x_t^p. \]  (23)
Unfortunately, maximizing quadratic pseudo-Boolean functions is NP-hard [1]. Although the objective function in (23) is not a general quadratic due to the property that all its quadratic terms are composed of bits corresponding to adjacent pixels, to the best of our knowledge no method has been devised to efficiently find the global maximum of such functions. Thus, instead of maximizing the objective function in (23), we choose to replace each quadratic term $\tilde{c}_1(p_1, p_2, t)x_{t}^{p_1}x_{t}^{p_2}$ with negative coefficient by the term $\frac{\tilde{c}_1(p_1, p_2, t)}{2}x_{t}^{p_1} + \frac{\tilde{c}_1(p_1, p_2, t)}{2}x_{t}^{p_2}$. This discriminates against the two assignments $x_{t}^{p_1} \neq x_{t}^{p_2}$ by $\frac{\tilde{c}_1(p_1, p_2, t)}{2}$, but does not alter the objective function’s value for the assignments $x_{t}^{p_1} = x_{t}^{p_2} = 0$ and $x_{t}^{p_1} = x_{t}^{p_2} = 1$. The resulting objective function has only nonnegative coefficients for the quadratic terms, and therefore its maximization may be reduced into a maximum-flow problem [5]. We specifically chose this method to estimate the maximum of (23) because it discriminates only against unequal bit assignments to adjacent pixel pairs, which typically constitute only a small portion of the bitmap (the object contour.)

Occasionally the estimated MAP bitmap may contain extraneous small connected components. This may happen after a small patch is erroneously attached to the tracked object (due to very similar color or motion) and then disconnected from it as a set of non-object pixels separating between the tracked object and this patch is correctly-classified. (In another scenario, the tracked object may truly split into more than one connected components. Note that the bitmap’s PDF does not assume any a priori topological information.) In this case, only the largest connected component in the estimated bitmap is maintained.

D. Considering Only Object-potential Pixels

Since the optical flow between adjacent frames is assumed to be limited by a maximal size $M$, there is no need to solve (23) for all the pixels in $I_t$. Instead, it is enough to solve only for the set of pixels constituting the object in $I_{t-1}$, dilated with a disc of radius equal to $M$ pixels, and set the bitmap to zero for all other pixels. In other words, $I_t$ is reduced to contain only the pixels that potentially belong to the tracked object. Due to the same reason, $I_{t-1}$ should be reduced to contain only the pixels potentially corresponding to the pixels in the reduced $I_t$, that is, the pixels constituting the object in $I_{t-1}$, dilated twice with the aforementioned disc. See Figure 2 for an illustration. Note that changing the pixel-sets $I_t$ and $I_{t-1}$ affect some normalization constants in the formulae.
Fig. 2. The reduced $I_t$ and the reduced $I_{t-1}$. In practice, the bitmap is estimated in the reduced image, and is set to 0 outside of it.

IV. EXPERIMENTS

Bittracker was tested on several image sequences, the first two synthesized and the rest natural. All the experiments demonstrate the successful tracking of rigid and non-rigid targets moving in 3D scenes and filmed by an arbitrarily moving camera. As no a priori knowledge is assumed regarding the scene or target, and the target’s shape and appearance undergo heavy changes over time (due to deformations, change in viewing direction, partial occlusions and lightning), a tracker of a more restricted context such as [23] would not be suitable here.

As the implementation has been done in MATLAB®, the tracker execution was slow. On a personal computer with a Pentium® IV 3GHz processor, the per-frame execution time was ranging from a few seconds for the small objects, up to two minutes for the large objects.

In all experiments, the parameters were set to the values indicated before, and the tracking was manually initialized in the first frame. Although all the image sequences are colored, they are shown here as intensity images so that the estimated bitmaps, overlaid on top by green, will be clear. Video files of all the presented tracking results are given as supplementary material.

1) Random-dot Sequence: First we tested Bittracker on a random-dot object of gradually time-varying shape and colors moving in front of a random-dot background whose colors are gradually time-varying and is in motion as well. See Figure 1 for the first two frames ((a)-(b)) and the object in each of them ((c)-(d)). Figure 3 shows for a number of frames the estimated bitmap in green, overlaid on top of intensity version images containing only the tracked object. The background was cut from these images to enable the comparison of the estimated bitmap
Fig. 3. Tracking in a random-dot video. The estimated bitmap is shown in green, overlaid on top of intensity version images containing only the tracked object. See text for details.

Fig. 4. (a) and (b) are two (nonconsecutive) images from a video of a randomly segmented object of gradually time-varying shape, segmentation and colors, moving in front of a randomly segmented background whose colors are gradually time-varying and is in motion as well. The object in the two images is shown in (c) and (d), respectively.

with respect to the tracked object. It is evident that the tracking in this sequence is very accurate. Note that new object pixels and revealed background pixels are correctly classified, which is due to the Spatial Motion Continuity Assumption.

2) Random-segment Sequence: Since the random-dot video contains a lot of texture, the optical flow may be estimated with high precision. To test Bittracker on a less textured video, it was challenged by a randomly segmented object of gradually time-varying shape, segmentation and colors, moving in front of a randomly segmented background whose colors are gradually time-varying and is in motion as well. See Figure 4 for two example images and the corresponding object contained in them. Tracking results are given in Figure 5, where the estimated bitmaps are shown in green, overlaid on top of intensity version images containing only the tracked object. As in the random-dot experiment, the tracking here is accurate too. Note that new object segments and revealed background segments are correctly classified, which is due to the Spatial Motion Continuity and the Spatial Coherence Assumptions.
3) Sellotape Sequence: Here we tracked a rotating and moving sellotape filmed by a moving camera. A few frames with the corresponding tracking results are shown in Figure 6. The sellotape hole, unrevealed in the video beginning, was revealed and marked correctly as the video progressed. Note that this change in object topology could not have been achieved using a state-space of object enclosing contours.

4) Man-in-Mall Sequence: In this experiment we tracked a man walking in a mall, filmed by a moving camera. A few frames with the tracking results overlaid are shown in Figure 7. Although occasionally parts of the tracked object are misclassified, these are corrected with time due to the Spatial Motion Continuity and the Spatial Coherence Assumptions. Note the zoom-in and zoom-out performed near the end of the sequence, and the partial occlusion at the end.

5) Woman-and-Child Sequence: Here Bittracker was tested on a sequence of a woman walking in a mall, taken by a moving camera. See Figure 8 for a few frames and the corresponding tracking results. It may be seen that the tracking overcame lighting changes and long-term...
Fig. 7. Tracking a man walking in a mall filmed by a moving camera. Note the overcoming over the zoom-in and zoom-out performed near the end of the sequence, and the partial occlusion at the end.

6) Flock Sequence: In this experiment Bittracker was tested on a cow running in a flock filmed by a moving camera. A few frames with the tracking results overlaid are shown in Figure 9. It may be seen that the tracking overcame a severe partial occlusion.

7) Boat Sequence: Here we tracked a floating boat, filmed by a moving camera. A few frames with the corresponding tracking results are presented in Figure 10. Note that in this sequence the background motion is caused not only due to the camera motion, but also due to the motion of the water.

8) Lighter Sequence: In this sequence we tracked a lighter undergoing general motion, filmed by moving camera. Figure 11 shows a few frames along with the tracking results. Note that the lighter areas revealed after severe occlusion endings and due to rotations in depth are correctly classified.

9) Ball Sequence: Here we tracked a ball, initially rolling in front of the moving camera, but then partially occluded by a combine. Results are shown in Figure 12. Notice the correct
Fig. 8. Tracking a woman walking in a mall filmed by a moving camera. The tracking overcame lighting changes and long-term partial occlusions. Since the woman and the girl she takes by hand were adjacent and walking in similar velocity through an extended time period (beginning around frame #100), the girl and the woman were joined as the tracking proceeded.

Fig. 9. Tracking a cow in a running flock filmed by a moving camera. Although the tracking object undergoes a severe partial occlusion, the tracking continued.

Fig. 10. Tracking a floating boat filmed by a moving camera. The background is moving both due to the motion of the camera and the motion of the water.
V. Conclusion

A novel algorithm for visual tracking under very general conditions was developed. The algorithm copes with non-rigid targets whose appearance and shape in the image may drastically change, general camera motion and 3D scenes, where no a priori target-related or scene-related information is given (except the target’s bitmap in the first frame for initialization.)

The tracking is conducted by maximizing in each frame a PDF of the target’s bitmap, formulated in the pixel-level through a lossless decomposition of the image information into color information and pixel-location information. This image decomposition enables the separable treatment of color and motion in a systematic way. The tracker relies merely three conservative assumptions: approximate constancy of color in consecutive frames (Short-time Constancy of Color Assumption), spatial piecewise continuity in optical flow of pixels belonging to the same object (Spatial Motion Continuity Assumption), and the belonging of similarly-colored adjacent pixels to the same object (Spatial Coherence Assumption).
Instead of estimating the optical flow by a point-estimate, marginalization over all possible flows per pixel is performed in the bitmap’s PDF construction. This is an important advantage, as optical flow estimation is prone to error, and is actually a harder and more general problem than target tracking.

Another advantage of the proposed algorithm is that the target’s bitmap PDF is formulated directly in the pixel-level. Thus, the precursory confinement of the final solution to objects composed of preliminarily-computed image segments, commonly done in video segmentation algorithms, is avoided.

Experimental results demonstrate Bittracker’s robustness to general camera motion and major changes in object appearance, caused by varying pose and configuration, lighting variations and long-term partial occlusions.

**Appendix I**

**Derivation of \( f_1(x_p^t) \)**

Continuing from Equation (6) by using the chain rule yields

\[
 f_1(x_p^t) = \sum_{p' \in \mathcal{N}_{t-1} \cup \{\text{none}\}} \Pr(p \rightarrow p' | \mathcal{I}_{t-1}, \mathcal{C}_t, \mathcal{X}_{t-1}) \cdot \frac{\Pr(x_p^t | p \rightarrow p', \mathcal{I}_{t-1}, \mathcal{C}_t, \mathcal{X}_{t-1})}{f_1^1(p'; p, t)} \cdot \frac{f_2^2(x_p^t; p')}{f_1^2(x_p^t; p')}. \tag{24}
\]

The first multiplicand inside the sum of (24) is the probability of \( \mathcal{I}_t \)'s pixel \( p \) corresponding to pixel \( p' \) in \( \mathcal{I}_{t-1} \) (or the probability of it corresponding to \( \text{none} \)) when considering only the pixel colors in \( \mathcal{I}_t \) and disregarding their exact placement in the frame. Using Bayes’ rule, we have

\[
 f_1^1(p'; p, t) \propto \Pr(p \rightarrow p' | \mathcal{I}_{t-1}, \mathcal{X}_{t-1}) \cdot p(C_t | p \rightarrow p', \mathcal{I}_{t-1}, \mathcal{X}_{t-1}). \tag{25}
\]

Not given \( \mathcal{L}_t \), the prior on the potentially corresponding pixels \( p' \in \mathcal{N}_{t-1} \) is uniform, and we set the prior probability of having no corresponding pixel in the previous frame to \( P_{\text{none}} \). Subject to this and under the Constancy of Color Assumption, for \( p' \in \mathcal{N}_{t-1} \) we approximate the first multiplicand inside the sum of (24) as

\[
 f_1^1(p'; p, t) \propto \frac{1 - P_{\text{none}}}{|\mathcal{I}_{t-1}|} \cdot N_{p, \mu, C}(c_{p'}^t), \quad p' \in \mathcal{N}_{t-1}, \tag{26}
\]

where \( N_{\mu, C} \) is the Normal PDF of mean \( \mu \) and covariance matrix \( C \) that we set as a constant as well. \( C \) is set to a diagonal matrix, where the variances reflect the degree of color similarity.
assumed in the Constancy of Color Assumption.) For \( p' = \text{none} \) we approximate

\[
f_1^1(p'; p, t) \propto P_{\text{none}} \cdot U(c^p_t),
\]

(27)

where \( U \) is the uniform PDF on the color-space (RGB in our implementation.) Note that the tracker sensitivity to \( P_{\text{none}} \) is minor, because of the very highly-peaked nature of \( f_1^1(p'; p, t) \) (as a function of \( p' \)) due to the multidimensional Gaussian modeling in (26). In our implementation \( P_{\text{none}} \) was set to 0.1.

In practice, in order to estimate the pixel correspondences more exactly, we compare small image patches (of diameter of 5 pixels) centered around the candidate pixels instead of merely the candidate pixels themselves. This is accounted for by modifying the Normal and uniform PDFs in Equations (26) and (27), respectively, to products of the color PDFs of the pixels in the patches. In addition, since pixels that are projections of different objects are likely to have different optical flows despite their adjacency in the image plane, we avoid the comparison of an image patch in \( I_t \) to an image patch in \( I_{t-1} \) that contains a mix of pixels \( \hat{p} \) assigned \( x^\hat{p}_{t-1} = 1 \) and pixels assigned \( x^\hat{p}_{t-1} = 0 \). In such cases, we compare only the pixels in the patch that are assigned the same bitmap value as the value assigned to the center pixel, which is the one the correspondence is sought for. We also restrict the maximal size of optical flow to \( M \) pixels (in our implementation \( M = 6 \)), and compare only between image patches distanced at most by \( M \), which reduces computations. Thus, the sum in (24) is computed over a subset of feasible pixels in \( I_{t-1} \) (137 pixels for \( M = 6 \)) and \( \text{none} \), which saves computations. We conclude for the first multiplicand inside the sum of (24):

\[
f_1^1(p'; p, t) \propto \begin{cases} 
(1 - P_{\text{none}}) \cdot N_{\partial_{t-1}^p} (c^p_t) & p' \in \mathcal{D}_{t-1}(p), \\
P_{\text{none}} \cdot U(c^p_t) & p' = \text{none},
\end{cases}
\]

(28)

where \( \mathcal{D}_{t-1}(p) \triangleq \{ p' : \| l^p_{t-1} - l^p_t \|_2 \leq M \} \) is the index-set of pixels in \( I_{t-1} \) within a radius of \( M \) pixels from pixel \( p \), and \( c^p_t \) is the vector of colors of every pixel composing the image patch for pixel \( p \) in \( I_t \), say in raster order. Since all the feasible cases for \( p' \) are covered by \( \mathcal{D}_{t-1}(p) \cup \{ \text{none} \} \), normalizing to a unit sum over \( p' \in \mathcal{D}_{t-1}(p) \cup \{ \text{none} \} \) produces the correct probabilities (although normalizing here is not necessary, as it will only scale \( P(X_t) \), which does not change its maximizing bitmap.)
The second multiplicand inside the sum of (24) is the PDF of the bitmap’s value at pixel $p$ in $\mathcal{I}_t$, conditioned on this pixel correspondence to pixel $p'$ in $\mathcal{I}_{t-1}$, whose bitmap value is given. Since the MAP bitmap estimated for the previous frame may contain errors, we set this PDF to

$$f_1^2(x_t^p; p') = \begin{cases} \text{P}_{\text{correct}} & x_t^p = x_{t-1}^p, \quad p' \in \mathcal{N}_{t-1}, \\
1 - \text{P}_{\text{correct}} & x_t^p \neq x_{t-1}^p \end{cases}$$

(29)

where $\text{P}_{\text{correct}}$ is a preset constant approximating the probability of the estimated bitmap being correct for a pixel (typically set to 0.9.) For $p' = \text{none}$ we set this PDF to

$$f_1^2(x_t^p; \text{none}) = \begin{cases} \text{P}_{\text{object}} & x_t^p = 1 \\
1 - \text{P}_{\text{object}} & x_t^p = 0 \end{cases}$$

(30)

where $\text{P}_{\text{object}}$ a preset constant as well that approximates the probability of a pixel, with no corresponding pixel in the previous frame, to belong to the tracked object (typical value of 0.4.)

To conclude the steps for computing $f_1(x_t^p)$ for pixel $p$ in $\mathcal{I}_t$, we first use Equation (28) to compute the probabilities $f_1^1(p'; p, t)$ for $p' \in \mathcal{D}_{t-1}(p) \cup \{\text{none}\}$, that is, the probabilities for pixel $p$’s different correspondences to pixels in $\mathcal{I}_{t-1}$ (feasible subject to the maximal optical flow assumed), including the probability of having no corresponding pixel. Then, by substituting Equations (29) and (30) into Equation (24), we derive

$$f_1(x_t^p = 1) = \text{P}_{\text{correct}} \cdot \prod_{p' \in \mathcal{D}_{t-1}(p) \cap \{q | x_q^{t-1} = 1\}} f_1^1(p'; p, t)$$

$$+ (1 - \text{P}_{\text{correct}}) \cdot \prod_{p' \in \mathcal{D}_{t-1}(p) \cap \{q | x_q^{t-1} = 0\}} f_1^1(p'; p, t) + \text{P}_{\text{object}} \cdot f_1^1(\text{none}; p, t),$$

(31)

and by complementing

$$f_1(x_t^p = 0) = 1 - f_1(x_t^p = 1).$$

(32)

We remark that there are many computations in (28) that are common for overlapping image patches, which should be taken advantage of in order to reduce computation time.

**Appendix II**

**Derivation of $f_\Delta(x_t^{p_1}, x_t^{p_2})$**

In the following we shall derive and show how we compute $f_\Delta(x_t^{p_1}, x_t^{p_2})$, which equals $\Pr(\Delta_t(p_1, p_2) | adj(p_1, p_2), x_t^{p_1}, x_t^{p_2}, \mathcal{I}_{t-1}, C_t, \mathcal{X}_{t-1})$, where $\Delta_t(p_1, p_2) \triangleq l_t^{p_1} - l_t^{p_2}$ and $adj(p_1, p_2)$
is the event of pixels $p_1$ and $p_2$ being adjacent. This expression is the right-hand-side multiplicand in (11).

Marginalization of $f_\Delta (x_{p_1}^1, x_{p_2}^2)$ over all the potential correspondences of pixels $p_1$ and $p_2$ to pixels in $I_{t-1}$, including the event of corresponding to none, and then applying the chain rule, yields

$$f_\Delta (x_{p_1}^1, x_{p_2}^2) = \sum_{p_1', p_2' \in N_{t-1} \cup \{\text{none}\}} \Pr (p_1 \rightarrow p_1', p_2 \rightarrow p_2' | \text{adj}(p_1, p_2), x_{p_1}^1, x_{p_2}^2, I_{t-1}, C_t, X_{t-1})$$

$$\cdot \Pr (\Delta_t(p_1, p_2) | \text{adj}(p_1, p_2), p_1 \rightarrow p_1', p_2 \rightarrow p_2', x_{p_1}^1, x_{p_2}^2, I_{t-1}, C_t, X_{t-1}) \cdot f_\Delta^C (x_{p_1}^1, x_{p_2}^2, \Delta_t(p_1, p_2), p_1', p_2').$$

(33)

The second multiplicand inside the sum of (33) is the likelihood of the relative position between adjacent pixels $p_1$ and $p_2$ in $I_t$, where the coordinates of their corresponding pixels in $I_{t-1}$, if any, are known (because the likelihood is conditioned on the pixel correspondences and on $I_{t-1}$, which consists of $L_{t-1}$.) In accordance with the Spatial Motion Continuity Assumption, we approximate this likelihood as is summarized in Table I. When $x_{p_1}^1 = x_{p_2}^2$ and both pixels have corresponding pixels in $I_{t-1}$, it is very likely that $\Delta_t(p_1, p_2) = \Delta_{t-1}(p_1', p_2')$. The probability of this event is assigned $P_{flow_1}$, which is a preset constant of typical value 0.99. The complementing event of $\Delta_t(p_1, p_2) \neq \Delta_{t-1}(p_1', p_2')$ is thus assigned $1 - P_{flow_1}$. Equivalently, when $x_{p_1}^1 \neq x_{p_2}^2$ and both pixels have corresponding pixels in $I_{t-1}$, it is very likely that $\Delta_t(p_1, p_2) \neq \Delta_{t-1}(p_1', p_2')$. The probability of this event is assigned $P_{flow_2}$, which is a preset constant as well with a typical value of 0.99. Complementing again, the event of $\Delta_t(p_1, p_2) = \Delta_{t-1}(p_1', p_2')$ is $1 - P_{flow_2}$. When one or both of the pixels have no corresponding pixel in $I_{t-1}$, the Spatial Motion Continuity Assumption is irrelevant and the four different values for $\Delta_t(p_1, p_2)$ are assigned the same chance of 0.25.

Following the partitioning of the possibilities for $p_1'$ and $p_2'$ summarized in Table I, the sum in (33) may be split into three cases:

\[
\{ (p_1', p_2') \in N_{t-1}^2 | \Delta_{t-1}(p_1', p_2') = \Delta_t(p_1, p_2) \},
\{ (p_1', p_2') \in N_{t-1}^2 | \Delta_{t-1}(p_1', p_2') \neq \Delta_t(p_1, p_2) \}\]
\[
\begin{array}{|c|c|c|}
\hline
p'_1 = \text{none or } p'_2 = \text{none} & p'_1, p'_2 \in \mathcal{N}_{t-1} & \\
\hline
x_{p_1} = x_{p_2} & \Delta_{t-1}(p_1, p_2) = \Delta_t(p_1, p_2) & \Delta_{t-1}(p'_1, p'_2) \neq \Delta_t(p_1, p_2) \\
0.25 & P_{f_{low_1}} & 1 - P_{f_{low_2}} \\
\hline
x_{p_1} \neq x_{p_2} & 0.25 & P_{f_{low_2}} \\
\hline
\end{array}
\]

\{ (p'_1, p'_2) \in (\mathcal{N}_{t-1} \cup \{ \text{none} \})^2 \mid p'_1 = \text{none or } p'_2 = \text{none} \}. \text{ For } x_{p_1} = x_{p_2} \text{ Equation (33) becomes}

\[
f_\Delta (x_{p_1}^{p_1}, x_{p_2}^{p_2}) = P_{f_{low_1}} \cdot \sum_{p'_1, p'_2 \in \mathcal{N}_{t-1} \text{ such that } \Delta_{t-1}(p'_1, p'_2) = \Delta_t(p_1, p_2)} f_\Delta^1 (p'_1, p'_2, x_{p_1}^{p_1}, x_{p_2}^{p_2}, p_1, p_2) + (1 - P_{f_{low_1}}) \cdot \sum_{p'_1, p'_2 \in \mathcal{N}_{t-1} \text{ such that } \Delta_{t-1}(p'_1, p'_2) \neq \Delta_t(p_1, p_2)} f_\Delta^1 (p'_1, p'_2, x_{p_1}^{p_1}, x_{p_2}^{p_2}, p_1, p_2)
\]

\[
+ 0.25 \cdot \sum_{p'_1, p'_2 \in \mathcal{N}_{t-1} \cup \{ \text{none} \} \text{ such that } p'_1 = \text{none or } p'_2 = \text{none}} f_\Delta^1 (p'_1, p'_2, x_{p_1}^{p_1}, x_{p_2}^{p_2}, p_1, p_2), \quad x_{p_1}^{p_1} = x_{p_2}^{p_2},
\]

and for \( x_{p_1}^{p_1} \neq x_{p_2}^{p_2} \)

\[
f_\Delta (x_{p_1}^{p_1}, x_{p_2}^{p_2}) = (1 - P_{f_{low_2}}) \cdot S_1 (x_{p_1}^{p_1}, x_{p_2}^{p_2}, p_1, p_2) + P_{f_{low_2}} \cdot S_2 (x_{p_1}^{p_1}, x_{p_2}^{p_2}, p_1, p_2) + 0.25 \cdot S_3 (x_{p_1}^{p_1}, x_{p_2}^{p_2}, p_1, p_2), \quad x_{p_1}^{p_1} \neq x_{p_2}^{p_2}. \quad (35)
\]

The term inside the summations (which is the first multiplicand inside the sum of (33)) is the joint probability that \( I_t \)'s pixels \( p_1 \) and \( p_2 \) correspond to pixels \( p'_1 \) and \( p'_2 \) in \( I_{t-1} \), respectively (or corresponding to \( \text{none} \).) This term is similar to \( f_\Delta^1 (p'; p, t) \), which is the correspondence distribution for a single pixel, although in the former the conditioning is also on the boolean variables of \( I_t \)'s pixels. Calculating the sums in (34) and (35) for a single pair of pixels under one
(x_i^{p_1}, x_i^{p_2})-hypothesis (out of four different hypotheses) would require estimating this term for a number of cases that is quadratic in the size of the image region for searching corresponding pixels, which we find to be too computationally demanding. (For M = 6 pixels, this number of cases is (137 + 1)^2 = 19, 044.) To reduce the computational cost we replace the exact calculation of the three sums by an estimate based on the marginal PDFs

$$f^\Delta_{\text{marginal}}(p', x_i^p; p) \overset{\Delta}{=} \Pr(p \rightarrow p'|X_t, I_{t-1}, C_t, X_{t-1}), \ p \in \mathcal{N}_t, \ p' \in \mathcal{N}_{t-1} \cup \{\text{none}\},$$  

(36)

and obtain estimates for the four likelihoods $f^\Delta(x_i^{p_1} = b_1, x_i^{p_2} = b_2)$ ($b_1, b_2 \in \{0, 1\}$.)

In the sequel we first show the calculation of $f^\Delta_{\text{marginal}}(p', x_i^p; p)$, and then present how we use it to obtain an estimate for $f^\Delta(x_i^{p_1}, x_i^{p_2})$.

1) Calculating $f^\Delta_{\text{marginal}}(p', x_i^p; p)$: For $p' \in \mathcal{N}_{t-1}$, marginalizing $f^\Delta_{\text{marginal}}(p', x_i^p; p)$ over the correctness of $x_i^{p'}$, followed by applying the chain rule yields

$$f^\Delta_{\text{marginal}}(p', x_i^p; p) = \Pr(x_i^{p'} - 1 \text{ is correct}) \cdot \Pr(p \rightarrow p'|x_i^{p'} - 1 \text{ is correct}, X_t, I_{t-1}, C_t, X_{t-1})$$

$$+ \Pr(x_i^{p'} - 1 \text{ is incorrect}) \cdot \Pr(p \rightarrow p'|x_i^{p'} - 1 \text{ is incorrect}, X_t, I_{t-1}, C_t, X_{t-1}),$$

and using the predefined constant $P_{\text{correct}}$ leads to

$$f^\Delta_{\text{marginal}}(p', x_i^p; p) = P_{\text{correct}} \cdot \Pr(p \rightarrow p'|x_i^{p'} = x_i^{p'} - 1 \text{ is correct}, X_t, I_{t-1}, C_t, X_{t-1})$$

$$+ (1 - P_{\text{correct}}) \cdot \Pr(p \rightarrow p'|x_i^{p'} - 1 \text{ is incorrect}, X_t, I_{t-1}, C_t, X_{t-1}).$$

This equals to one of two expressions, depending on the value of $x_i^p$ for which $f^\Delta_{\text{marginal}}$ is calculated:

$$f^\Delta_{\text{marginal}}(p', x_i^p; p) = \begin{cases} 
P_{\text{correct}} \cdot \Pr(p \rightarrow p'|x_i^{p'} = x_i^{p'} - 1 \text{ is correct}, X_t, I_{t-1}, C_t, X_{t-1}) & x_i^p = x_i^{p'} - 1, \ p' \in \mathcal{N}_{t-1}, \\
(1 - P_{\text{correct}}) \cdot \Pr(p \rightarrow p'|x_i^{p'} - 1 \text{ is incorrect}, X_t, I_{t-1}, C_t, X_{t-1}) & x_i^p \neq x_i^{p'} - 1,
\end{cases}$$

(37)

By Bayes’ rule

$$\Pr(p \rightarrow p'|x_i^{p'} - 1 \text{ is (in)correct}, X_t, I_{t-1}, C_t, X_{t-1}) \propto \Pr(p \rightarrow p'|x_i^{p'} - 1 \text{ is (in)correct}, X_t, I_{t-1}, X_{t-1})$$

$$\cdot \Pr(C_t|p \rightarrow p', x_i^{p'} - 1 \text{ is (in)correct}, X_t, I_{t-1}, X_{t-1}), \ p' \in \mathcal{N}_{t-1}. \quad (38)$$

March 30, 2006
DRAFT
As in (26), the prior on the potentially corresponding pixels $p' \in \mathcal{N}_{l-1}$ is uniform, but here it is over $\mathcal{I}_{l-1}$'s pixels of a similar bitmap bit as of pixel $p$ ((non)-object pixels may only correspond to (non)-object pixels.) Based on this and using the Gaussian color distribution as in (26), we obtain

$$
\Pr \left( p \rightarrow p' \mid x_{l-1}^{p'} \text{ is correct}, \mathcal{X}_l, \mathcal{I}_{l-1}, C_l, \mathcal{X}_{l-1} \right) \propto \frac{1 - P_{none} \cdot N_{c_{l-1}, C_l}(c_l^p)}{\left| \mathcal{N}_{l-1} \right| P_{correct} + \left| \mathcal{N}_{l-1} \cap \{ q : x_{l-1}^q \neq x_l^q \} \right| (1 - 2P_{correct})} \cdot N_{c_{l-1}, C_l}(c_l^p), \quad x_l^p = x_{l-1}^{p'}, \quad A_=
$$

and

$$
\Pr \left( p \rightarrow p' \mid x_{l-1}^{p'} \text{ is incorrect}, \mathcal{X}_l, \mathcal{I}_{l-1}, C_l, \mathcal{X}_{l-1} \right) \propto \frac{1 - P_{none} \cdot N_{c_{l-1}, C_l}(c_l^p)}{\left| \mathcal{N}_{l-1} \right| + 1 + \left| \mathcal{N}_{l-1} \cap \{ q : x_{l-1}^q \neq x_l^q \} \right| (1 - 2P_{correct})} \cdot N_{c_{l-1}, C_l}(c_l^p), \quad x_l^p \neq x_{l-1}^{p'}. \quad A_\neq
$$

Note that similarly to (26), the denominators of $A_=$ and $A_\neq$ are the (expected) number of pixels in $\mathcal{I}_{l-1}$ that may correspond to pixel $p$. These denominators are almost equal but different from the denominator in (26), because the probabilities are conditioned on $x_l^p$. Substituting these into (37) gives

$$
f_{\Delta, \text{marginal}}^1(p', x_l^p ; p) \propto \begin{cases} 
P_{correct} \cdot A_= \cdot N_{c_{l-1}, C_l}(c_l^p) & \quad x_l^p = x_{l-1}^{p'}, \quad p' \in \mathcal{N}_{l-1}. \\
(1 - P_{correct}) \cdot A_\neq \cdot N_{c_{l-1}, C_l}(c_l^p) & \quad x_l^p \neq x_{l-1}^{p'}. 
\end{cases}
$$

(41)

For $p' = \text{none}$ the conditioning on $\mathcal{X}_l$ has no influence on $f_{\Delta, \text{marginal}}^1(p', x_l^p ; p)$ and using the uniform color distribution as in (27) we obtain

$$
f_{\Delta, \text{marginal}}^1(\text{none}, x_l^p ; p) \propto P_{\text{none}} \cdot U(c_l^p). \quad (42)
$$
As in (28), the size of the optical flow is restricted and image patches are considered instead of single pixels, which concludes for the marginal PDFs of the pixel correspondences in

\[
f_{\Delta_{\text{marginal}}}^1(p', x_t^p; p) \propto \begin{cases} 
  P_{\text{correct}} \cdot A = N_{c_{t-1}, C}(c_t^p) & p' \in D_{t-1}(p) \text{ and } x_t^p = x_t^{p'}, \\
  (1 - P_{\text{correct}}) \cdot A_{\not\in} N_{c_{t-1}, C}(c_t^p) & p' \in D_{t-1}(p) \text{ and } x_t^p \neq x_t^{p'}, \\
  P_{\text{none}} \cdot U(c_t^p) & p' = \text{none},
\end{cases}
\]

where normalizing to a unit sum over the “generalized neighborhood” of \( p, D_{t-1}(p) \cup \{\text{none}\} \), produces the correct probabilities.

2) Estimating \( f_\Delta(x_t^{p_1}, x_t^{p_2}) \): The third sum in (34) and (35), which is the probability that at least one of the two pixels has no corresponding pixel in the previous frame, is

\[
S_3(x_t^{p_1}, x_t^{p_2}; p_1, p_2) = \sum_{p_2' \in N_{t-1} \cup \{\text{none}\}} f_{\Delta_{\text{marginal}}}^1(\text{none}, p_2', x_t^{p_1}, x_t^{p_2}; p_1, p_2) \\
+ \sum_{p_1' \in N_{t-1} \cup \{\text{none}\}} f_{\Delta_{\text{marginal}}}^1(p_1', \text{none}, x_t^{p_1}, x_t^{p_2}; p_1, p_2) - f_{\Delta_{\text{marginal}}}^1(\text{none}, \text{none}, x_t^{p_1}, x_t^{p_2}; p_1, p_2) \\
= f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_1}; p_1) + f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_2}; p_2) - f_{\Delta_{\text{marginal}}}^1(\text{none}, \text{none}, x_t^{p_1}, x_t^{p_2}; p_1, p_2),
\]

and modeling the events \( p_1' = \text{none} \) and \( p_2' = \text{none} \) as independent, we obtain

\[
S_3(x_t^{p_1}, x_t^{p_2}; p_1, p_2) = f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_1}; p_1) + f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_2}; p_2) \\
- f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_1}; p_1) \cdot f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_2}; p_2).
\]

Turning to \( S_1(x_t^{p_1}, x_t^{p_2}; p_1, p_2) \), which is the probability that the two pixels have identical discrete optical flows from the previous frame, and denoting

\[
k(x_t^{p_1}, x_t^{p_2}; p_1, p_2) \triangleq 1 - f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_1}; p_1) \cdot f_{\Delta_{\text{marginal}}}^1(\text{none}, x_t^{p_2}; p_2),
\]

it is easy to verify the bounds

\[
\sum_{p_1', p_2' \in N_{t-1} \text{ such that } \Delta_{t-1}(p_1', p_2') = \Delta_{t}(p_1, p_2)} \max \left\{ 0, f_{\Delta_{\text{marginal}}}^1(p_1', x_t^{p_1}; p_1) + f_{\Delta_{\text{marginal}}}^1(p_2', x_t^{p_2}; p_2) - k(x_t^{p_1}, x_t^{p_2}; p_1, p_2) \right\}
\]

\[
\leq S_1(x_t^{p_1}, x_t^{p_2}; p_1, p_2)
\]

\[
\leq \sum_{p_1', p_2' \in N_{t-1} \text{ such that } \Delta_{t-1}(p_1', p_2') = \Delta_{t}(p_1, p_2)} \min \left\{ f_{\Delta_{\text{marginal}}}^1(p_1', x_t^{p_1}; p_1), f_{\Delta_{\text{marginal}}}^1(p_2', x_t^{p_2}; p_2) \right\}.
\]
The upper bound is directly obtained from the fact that the joint probability of two events is not larger than the marginal probability of any of the individual events. The lower bound is obtained by bounding the probability of the event that \( p_1 \rightarrow p'_1 \) or \( p_2 \rightarrow p'_2 \) by \( k(x_{t_i}^{p_1}, x_{t_i}^{p_2}; p_1, p_2) \) from above.

By complementing, the second sum in (34) and (35), which is the probability of having different discrete optical flows, is

\[
S_2 (x_{t_i}^{p_1}, x_{t_i}^{p_2}; p_1, p_2) = 1 - S_3 (x_{t_i}^{p_1}, x_{t_i}^{p_2}; p_1, p_2) - S_1 (x_{t_i}^{p_1}, x_{t_i}^{p_2}; p_1, p_2). \tag{48}
\]

Equations (45)-(48) induce immediate bounds on \( f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \)

\[
\text{lower} (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \leq f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \leq \text{upper} (x_{t_i}^{p_1}, x_{t_i}^{p_2}). \tag{49}
\]

Thus, for each unordered pair of adjacent pixels \( p_1 \) and \( p_2 \) in \( I_t \) there are the four intervals

\[
\begin{align*}
\text{lower} (x_{t_i}^{p_1} = 0, x_{t_i}^{p_2} = 0) & \leq f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \leq \text{upper} (x_{t_i}^{p_1} = 0, x_{t_i}^{p_2} = 0), \\
\text{lower} (x_{t_i}^{p_1} = 0, x_{t_i}^{p_2} = 1) & \leq f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \leq \text{upper} (x_{t_i}^{p_1} = 0, x_{t_i}^{p_2} = 1), \\
\text{lower} (x_{t_i}^{p_1} = 1, x_{t_i}^{p_2} = 0) & \leq f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \leq \text{upper} (x_{t_i}^{p_1} = 1, x_{t_i}^{p_2} = 0), \\
\text{lower} (x_{t_i}^{p_1} = 1, x_{t_i}^{p_2} = 1) & \leq f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \leq \text{upper} (x_{t_i}^{p_1} = 1, x_{t_i}^{p_2} = 1). \tag{50}
\end{align*}
\]

Avoiding more computations, we take only these interval restrictions into account and choose the four likelihoods \( f_\Delta (x_{t_i}^{p_1} = b_1, x_{t_i}^{p_2} = b_2) \) \((b_1, b_2 \in \{0, 1\})\), under these interval restrictions, to be closest clustered so that the sum of their differences

\[
\frac{1}{2} \sum_{(b_1, b_2) \in \{0, 1\}^2} \sum_{(b'_1, b'_2) \in \{0, 1\}^2} \left| f_\Delta (x_{t_i}^{p_1} = b_1, x_{t_i}^{p_2} = b_2) - f_\Delta (x_{t_i}^{p_1} = b'_1, x_{t_i}^{p_2} = b'_2) \right| \tag{51}
\]

is minimized. Qualitatively, by this way the effect of \( f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \) on \( F_2 \), and thus also on the bitmap’s PDF (3), is minimized, while obeying the interval restrictions. Therefore, the larger the uncertainty (i.e., interval) in the values of \( f_\Delta (x_{t_i}^{p_1}, x_{t_i}^{p_2}) \), the less the effect of this component on the bitmap’s PDF. It is easily seen through (47) that the more unequivocal the marginal optical flows \( f_{\Delta \text{marginal}} (p'_1, x_{t_i}^{p_1}; p_1) \) and \( f_{\Delta \text{marginal}} (p'_2, x_{t_i}^{p_2}; p_2) \), the smaller these uncertainties. A typical histogram of these interval sizes is presented in Figure 13(b) (the largest interval out of the four is taken per pixel), showing that indeed a large portion of the \( f_\Delta \)’s has small intervals and thus significantly affect the bitmap’s PDF. It can be shown that the minimization of (51) within the
Fig. 13. (b) The histogram of the $f_{\Delta}$’s interval sizes computed for the object marked in (a) (the largest interval out of the four is taken per pixel.) We see that a large portion of the $f_{\Delta}$’s has small intervals and thus affect the bitmap’s PDF.

intervals of (50) may be easily accomplished by sorting the eight bounds

$$\{\text{lower} (x_{p_1}^{p_1} = b_1, x_{p_2}^{p_2} = b_2)\}_{b_1, b_2 \in \{0, 1\}} \cup \{\text{upper} (x_{p_1}^{p_1} = b_1, x_{p_2}^{p_2} = b_2)\}_{b_1, b_2 \in \{0, 1\}}$$

in ascending order, and then measuring for each adjacent pair of bounds $\text{bound}_1$ and $\text{bound}_2$ (seven pairs) the sum of differences (51) obtained by setting each of the four $f_{\Delta} (x_{p_1}^{p_1}, x_{p_2}^{p_2})$’s closest to $\frac{\text{bound}_1 + \text{bound}_2}{2}$ within its interval. The best setting out of the seven settings is an optimal one.

To conclude the steps for computing $f_{\Delta} (x_{p_1}^{p_1}, x_{p_2}^{p_2})$ for a pair of adjacent pixels $p_1$ and $p_2$ in $I_t$, we first use equation (43), followed by normalization, to compute the probabilities $f_{\Delta, \text{marginal}}^1 (p', x_i^t; p)$ for $p' \in D_{t-1}(p) \cup \{\text{none}\}$, that is, the probabilities for pixel $p$’s different correspondences to pixels in $I_{t-1}$ (feasible subject to the maximal optical flow assumed) under each of the two $x_i^t$-hypotheses. Then, using (45)-(48) for the sums in (34) and (35), we obtain the intervals (50), within which we set the four values of $f_{\Delta} (x_{p_1}^{p_1}, x_{p_2}^{p_2})$ so that (51) is minimized by the method explained. As in (28), we remark that there are many computations in (43) that are common for overlapping image patches, as well as computations that are already performed for computing (28). These should be taken advantage of in order to reduce computation time.

REFERENCES


