# From Video Matching to Video Grounding

Georgios Evangelidis, Ferran Diego, Radu Horaud

# From Video Matching to Video Grounding

Georgios Evangelidis
INRIA Rhône-Alpes,
38330, Montbonnot, France
georgios.evangelidis@inria.fr

Ferran Diego
University of Heilelberg,
D-69115, Heidelberg, Germany
ferran.diego@iwr.uni-heidelberg.de

Radu Horaud
INRIA Rhône-Alpes,
38330, Montbonnot, France
radu.horaud@inria.fr

## Abstract

*This paper addresses the background estimation problem for videos captured by moving cameras, referred to as video grounding. It essentially aims at reconstructing a video, as if it would be without foreground objects, e.g. cars or people. What differentiates video grounding from known background estimation methods is that the camera follows unconstrained motion so that background undergoes ongoing changes. We build on video matching aspects since more videos contribute to the reconstruction. Without loss of generality, we investigate a challenging case where videos are recorded by in-vehicle cameras that follow the same road. Other than video synchronization and spatiotemporal alignment, we focus on the background reconstruction by exploiting inter- and intra-sequence similarities. In this context, we propose a Markov random field formulation that integrates the temporal coherence of videos while it exploits the decisions of a support vector machine classifier about the backgroundness of regions in video frames. Experiments with real sequences recorded by moving vehicles verify the potential of the video grounding algorithm against state-of-art baselines.*

## 1. Introduction

In this paper we present an algorithm for a relatively new problem referred to as *video grounding* that resembles the background estimation problem [9, 22, 7] or the foreground-background separation [1, 18]. While background estimation tries to estimate the static background based on a set of images captured from the same viewpoint [9, 22, 7] or nearby viewpoints after a *constrained* camera motion (i.e. panning) [9, 16], video grounding tries to extract the background from videos captured from freely moving cameras. In other words, video grounding aims at reconstructing a video by removing all foreground objects, which is called *grounded video*.

Unlike video completion methods that have been mostly applied to videos from static or PTZ cameras [19, 21], we investigate the grounding as a further step of video matching since more videos from moving cameras not only contribute to the background reconstruction but also to the detection of static foreground objects. Without loss of generality, we in this paper focus on the challenging case of videos captured from vehicles following similar trajectories, so that cameras and foreground objects are drastically moving.

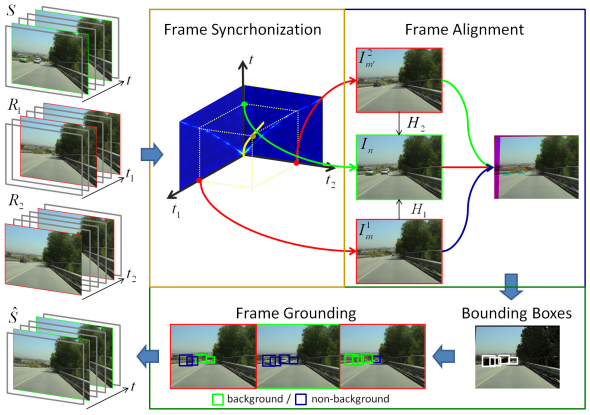



Figure 1. Video Grounding builds on the alignment of video input $S$ with the reference videos $R_1$, $R_2$.

We divide the problem into three stages, i.e. synchronization, alignment, background estimation. Inspired by recent advances in video alignment [15], we enable an efficient retrieval solution for the video synchronization problem instead of a global sequence alignment method [11]. Each newly captured frame referred to as *input frame*, implies a query to the database that contains reference videos. The corresponding reference frame is retrieved through a bag-of-words (BOW) scheme based on SURF descriptors [3] and the temporal mapping is refined through a local temporal coherence assumption. Then, any reference frame is spatially aligned with the input frame based on our modification of [13] that makes the method robust to occlusions. Such a pixel-wise alignment essentially marks the areas that strong changes occur. A support vector machine (SVM) classifier trained from reference examples decides about the "backgroundness" of such areas. This decision is in-

tegrated into a labeling problem through a Markov random field (MRF) formulation that offers the grounded frames. Fig. 1 shows an overview of our approach.

Interactive maps and systems such as Google Street View and Bing Maps Streetside can benefit from video grounding, i.e., the user could navigate with the help of cleaned-up videos, instead of moving among discrete panoramas that contain people or cars. Advanced driver assistance systems (ADAS) [11], video surveillance [12] and various applications that rely on video matching [17] can drastically exploit the grounding since the latter provides an alternative yet meaningful view. Furthermore, any new record contributes to the more groundness of the reference data that may contain unwanted objects and produce false alarms in change detection [11, 12].

Although we do not rely on detectors of specific classes, the main contribution of this work is that it integrates classification results and temporal coherence constraints into an MRF-based background estimation framework.

### 1.1. Related work

To the best of our knowledge there is no work in literature that exactly deals with the problem in question, since most methods focus on the background estimation from *many* still images captured by the the same viewpoint [22, 7] or when the camera slightly moves [9]. Cohen [9] solves a labelling problem with the assumption that any background point is at least seen in one image. From an algorithmic point of view, a similar approach has been followed by [22] and [7].

When reference data are not available, a purely rotating [2] or a PTZ camera [14, 19] allows for the background-foreground separation, hence the background reconstruction. These methods rely on the fact that the camera undergoes a constrained motion, so that a homography relates the background (static) points in different frames. The above separation has been also achieved by combining classifiers that vote for foreground objects, thus separating the objects from the background [1], or by permitting a weak camera motion so that trajectories of static points are recognized [18].

Image or video inpainting can also be useful for background recovery. In this context, Patwardhan *et al.* [16] permit minor fronto-parallel camera motion and follow a two step video inpainting method. Bertalmio *et al.* [4] adopt fluid dynamics to inpaint images when the background is hidden, while Criminisi *et al.* [10] propose an object removal method using an exemplar-based region filling algorithm. Wexler *et al.* [21] complete video parts using spatiotemporal exemplars in order to recover both background and foreground objects when the camera is static.

As mentioned above, multiple videos contribute to the video grounding. However, when the cameras are drastically moving, the spatio-temporal alignment of videos is required. Diego *et al.* [11] temporally align the videos by fusing different data within a Bayessian network, while synchronized frames are spatially registered with a Lucas-Kanade scheme. Evangelidis and Bauckhage [12] cast synchronization as a retrieval problem using short descriptors and proposed a more accurate spatio-temporal alignment scheme. In this context, Liu *et al.* [15] used a bag-of-words scheme to retrieve similar images from videos and to align them with a flow-based algorithm.

Figure 2. *Top*: An input frame (left) and the corresponding reference frame (right). *Bottom:* The results of inpainting (Criminisi *et al.* [10]) (left) and background estimation (Chen *et al.* [7]) (right).

## 2. Critical issues

### 2.1. Limited number of sources

Given a single image as a scene, the only solution for the background estimation in an occluded area is an inpainting method. However, inpainting deals only with small regions while its performance is highly texture-dependent [10]. Would we be given more images of the same scene captured by the same viewpoint, a labeling solution, e.g. [9, 7, 22], can decide about the appropriate source (label) for each pixel. Such a labeling scheme mainly relies on the median over the source images, so that the case of two sources becomes problematic when large areas are occluded. Fig. 2 shows instances of the inpainting method of [10] and the background estimation method of [7] when one and two sources are used respectively. Apparently, the prior knowledge that the foreground object appears in the second image would simplify the problem. When the background is occluded in both images, which is here referred to as double occlusion, we end up with a degenerate case and a third source image should be given. But even so, the labeling solution might fail and the knowledge about the backgroundness of each image, e.g. which images possibly contain foreground objects, would be very useful. Notice that when it comes to a famous place, new sources can be obtained from web databases [22].

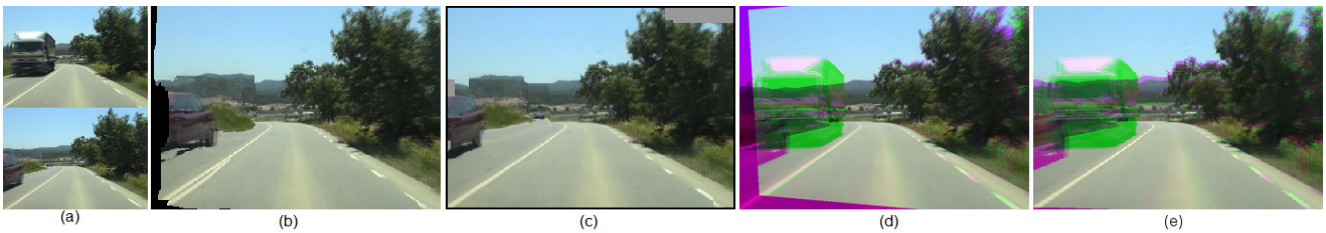



Figure 3. (*Best viewed on-screen*) Registration of (a-top) input and (a-bottom) reference frames: (b) optical flow [20], (c) SIFT-flow [15], (d) RANSAC homography (e) image alignment [13]. In (b,c) the warped frame is shown, while in (d,e) we replace the G channel of the input frame with the one of the warped reference image.

Likewise, previously recorded sequences of the same scene offer to the video grounding, provided that they have been spatio-temporally registered – a difficult task when the cameras are moving. Past videos contribute to the detection of foreground objects as well, regardless of their motion or the class they belong to, e.g. cars, people, etc. Although double occlusion seems to be intractable using two still images, the temporal nature of videos can be exploited. Adjacent frames in the reference videos can be combined in order to provide a more grounded reference frame which can be in turn used for the grounding of the input frame (double grounding). If the foreground object is static and it occludes the background in all frames, inpainting should be invoked as a last resort. However, the temporal coherence can be again exploited as we show in Sec.5.

### 2.2. Image misalignments

Let us consider the example of Fig. 3 where the input and what we obtain as corresponding reference frame are shown (Fig. 3(a)). Apparently, pixel-wise change detection relies on dense correspondences. Since we assume nearby viewpoints, such correspondences can be obtained by either optical flow methods that provide pixel-based displacements or global alignment methods that estimate the homography between the images. Fig. 3(b,c) show the warped frame based on the optical flow methods of [20] and [15] respectively. Except for several local misalignments, we observe artifacts due to occlusions that negatively affect the grounding. Fig. 3(d,e) depicts the result from a feature-based registration and the alignment method of [13]. Again, the occlusions negatively affect the homography estimation, hence the grounding would locally fail because of the misalignments. Note that even if the local misalignments do not cause strong errors in a single grounded image, their spatial randomness in successive frames leads to severe artifacts in the grounded video; the latter is due to the fact that the homography must be re-estimated per frame since the cameras are not jointly moving and the spatial transformation is not fixed over time. As a consequence, a global alignment scheme which is robust to occlusions would be preferable for nearby viewpoints. Therefore, we are going to modify the algorithm of [13], thus providing a self-weighted scheme which is insensitive to large occlusions.

Table 1. Performance of video synchronization algorithms quantified by the percentage of well matching frames using two error thresholds ($e = 0$ and $e = 1$).

| Algorithm | Scenarios | |
|---|---|---|
| | Backroad | Highway |
| | $e = 0/e = 1$ | $e = 0/e = 1$ |
| SURF-based | 71.6/86.6 | 82.5/86.9 |
| SURF-based + LOWESS | 86.7 / 93.2 | 91.5 /97.2 |
| Quad-based [12] | 70.6/84.6 | 74.7/91.3 |
| SIFT-flow [15] | 77.7/85.1 | 76.1/87.2 |
| MAP-Inference [11] | 79.7/82.2 | 71.5/79.5 |
| DTW | 74.0/79.6 | 70.8/76.6 |

## 3. A Video Grounding Algorithm

Let us suppose that $\mathcal{R} = \{I_m\}_{m=1}^{M}$, $\mathcal{S} = \{I_n\}_{n=1}^{N}$ are the reference and input image sequences respectively, that have been recorded at different times with the camera following approximately the same trajectory. In what follows, we describe each step of our algorithm in order to finally obtain a grounded video $\hat{\mathcal{S}}$. Note that if we have two reference sequences at our disposal, $\mathcal{R}_1$ and $\mathcal{R}_2$, the first two steps apply twice.

### 3.1. Frame Synchronization

Our first goal is to extract the corresponding frame $I_m$ that reflects the higher similarity with the input frame $I_n$. To do this, we follow an efficient retrieval approach inspired by [15, 12]. We detect SURF features [3] in all frames of $\mathcal{R}$ and build a visual codebook and an inverted file that stores the occurrences of each visual word. Given an input frame we extract its features and we map each to the closest visual word, thus voting for the assigned frames from the inverted file. The frame with the highest score defines a putative match with the input frame. Next, the putative matches are refined using the robust locally weighted scatter-plot smoothing (robust LOWESS) [8]. The smoothing offers the desired robustness to outliers and provides the corresponding reference frame for each input frame. Note that the temporal mapping is non-linear since the cameras move at different speeds.

Table 1 provides synchronization results of our method and several baselines tested on two road sequences that contain different foreground objects (vehicles) [11]. The synchronization score is the percentage of well matching frames, i.e. the retrieved frames whose temporal distance from the ground truth is up to the error tolerance $e$. MAP-inference [11] and Dynamic Time Warping (DTW) are dynamic programming (global) methods that rely on the temporal continuity assumption, while Sift-flow [15] and Quad-based [12] methods are frame-wise (local) synchronizations that build on successive retrievals. Except for the latter, all methods use the bag-of-words paradigm for the vector representation of frames. While the frame-wise synchronization provides sparse false positives, the global methods suffer from the error propagation. Instead, our synchronization exploits locally the temporal coherence of videos, thus providing synchronization with higher accuracy.

### 3.2. Frame Alignment

Given two corresponding frames, our goal reduces into their spatial registration. Our scenario of nearby viewpoints allows us to approximate the motion with a homography model. The images, however, have been captured at different times. This implies that there may exist appearance variation in corresponding frames. Moreover, moving foreground objects may cause large occlusions as we see in Fig. 3. An image alignment scheme that is robust to photometric distortions is the Enhanced Correlation Coefficient (ECC) algorithm [13]. However, since it is not robust to occlusions as well, we propose a weighted version of ECC, referred here to as Reweighted ECC (R-ECC).

ECC estimates the geometric transformation (warp) that maximizes the correlation coefficient (zero-mean normalized cross correlation) between $I_n$ and the warped version of $I_m$. At each iteration, ECC warps $I_m$ with the current transformation and updates the latter based on the ECC criterion [13]. Let us assume that the current iteration of the algorithm returns a homography based on which we warp $I_m$ appropriately to register it with $I_n$. We then split both images into $Q$ non-overlapping blocks and we subtract the average from each block. Since the numerator of the ECC score, $C$, consists in the summation of the pixel-wise cross products, it can be written as $C = \sum_{i=1}^{Q} c_i$, where $c_i$ is the cross-correlation that corresponds to the $i^{th}$ block. The value $max(0, c_i/D)$, where $D$ is the denominator of the ECC score (the product of the standard deviations), defines the weight $w_i$ of the $i^{th}$ block. An interpolation scheme on the $w_i$ all provides pixel-wise weights and the latter, after their normalization, apply to both images. Other than the self-weighting, the steps of the algorithm exactly follow the original version [13]. It is important to note that there is no compositional weighting, but the images, i.e., $I_n$ and warped $I_m$, are re-weighted at each iteration. Fig. 4 shows

Figure 4. (*Best viewed on-screen*) (Left) The alignment of the frames in Fig. 3 and (right) the weighted input frame after the convergence of the R-ECC algorithm.

the alignment result for the frames of Fig. 3 by using the R-ECC scheme. Due to the zero-mean blocks, texture-less areas are downweighted. However, this is not an undesired effect, since low-texture areas do not aid the alignment.

Having obtained dense accurate correspondences, we can easily detect changes between the images and estimate bounding boxes by simple morphological processing on image differences.

### 3.3. Frame Grounding

The novelty of this paper mostly counts on this subsection, since we are going to integrate the results of a classifier and the temporal coherence of videos into an MRF formulation, thus solving a labeling problem.

To start with, let us denote as $J_0(\mathbf{p})$ a *multiframe* of a few successive input frames and as $J_u(\mathbf{p})$, $u = 1, ..., U$, the multiframe of the the $u^{th}$ reference sequence that consists in the spatiotemporally aligned reference frames for each of the frames in $J_0(\mathbf{p})$, where $\mathbf{p} = (p, t)$ is a space-time point with $p$ and $t$ denoting the pixel and frame indices respectively. Note that here we consider a couple of sources, e.g. $U = 1$ or $U = 2$, but the algorithm is designed independently of the value $U$. Let us also suppose that a pixel-wise comparison provides $K$ non-overlapping bounding boxes $B_k, k = 1, \ldots, K$ within of which the content of $J_0(\mathbf{p})$ is different than that of any $J_u(\mathbf{p})$, and let $\Omega = \cup_k B_k$ be the total area of changes.

Without loss of generality, we consider multiframes of three successive frames. Our goal is to reconstruct the background of the input multiframe in a temporal coherence framework, e.g., the value $J_0(p, 1)$ can be replaced by $J_u(p, 1)$, $u = 1, ..., U$. To this end, we are looking for the label $f_{\mathbf{p}} \in \mathcal{L}$, with $\mathcal{L} = \{0, 1, ..., U\}$ for all space-time points $\mathbf{p}$ so that the global energy function

$$E(f_{\mathbf{p}}) = E_D(f_{\mathbf{p}}) + \lambda_S E_S(f_{\mathbf{p}}) + \lambda_T E_T(f_{\mathbf{p}}), \quad (1)$$

is minimum. Here, $E_D$ is the data-term, $E_S$, $E_T$ are the spatial and temporal smoothness terms respectively while $\lambda_S$ and $\lambda_T$ balance their contribution to the energy value; the terms are defined below. Note that if $\lambda_T = 0$ and the multiframe consists in a single frame only, the formulation reduces to that of [9, 22, 7].

#### 3.3.1 The data-term

The data-term $E_D(f_\mathbf{p})$ is defined by $E_D(f_\mathbf{p}) = \sum_\mathbf{p} D_\mathbf{p}(f_\mathbf{p})$ where $D_\mathbf{p}(f_\mathbf{p})$ is the cost of assigning the label $f_\mathbf{p}$ to the space-time point $\mathbf{p}$. The proposed data-term takes into account the decisions of a classifier. In specific, the data-cost is defined as

$$D_\mathbf{p}(f_\mathbf{p}) = \begin{cases} \|J_{f_\mathbf{p}}(\mathbf{p}) - J_0(\mathbf{p})\|_2 & \text{if } \mathbf{p} \notin \Omega \\ C^{f_\mathbf{p}}(\mathbf{p}) \left[D^c(f_\mathbf{p}) + D^b(f_\mathbf{p})\right] & \text{if } \mathbf{p} \in \Omega \end{cases}, \tag{2}$$

where $D^c(f_\mathbf{p}) = \|J_{f_\mathbf{p}}(\mathbf{p}) - J_{\bar{M}}(\mathbf{p})\|_2$ is a color stationariness, $D^b(f_\mathbf{p}) = \sum_{l=1}^{L} \|J_{f_\mathbf{p}}(\mathbf{p}) - J_l(\mathbf{p})\|$ is a background stationariness, $J_{\bar{M}}(\mathbf{p})$ is a multiframe whose color at $\mathbf{p}$ is the median (per channel) over all multiframes at this point and $C^{f_\mathbf{p}}(\mathbf{p})$ is a risk term defined in what follows. Note that the first part favours the use of the input multiframe outside the region $\Omega$. As for $\Omega$, not only the color and background stationariness are taken into account, but also the results of a classification task through $C^{f_\mathbf{p}}(\mathbf{p})$.

The term $C^{f_\mathbf{p}}(\mathbf{p})$ reflects the risk of $\mathbf{p}$ belonging to foreground objects. Hence, sources with low risk are preferred towards the labeling inference. In order to estimate this term, we build on decisions from a SVM classifier that decides about the backgroundness of SURF features. For the supervised training, we use features from positive examples (background sub-regions) and negative examples (foreground regions, e.g. vehicles in our scenario), all extracted from the reference sequence. Here, a radial kernel is used and a cross-validation is enabled to estimate the best parameters. With the help of this classifier, we assign a box-risk $c_{k,l}^{box}$ to the $k^{th}$ box of the $l^{th}$ multiframe ($l \in \mathcal{L}$) which is defined as $c_{k,l}^{box} = \frac{\#\text{non-background features of } J_l(\mathbf{p}) \text{ in } B_k}{\#\text{features of } J_l(\mathbf{p}) \text{ in } B_k}$. Since the bounding box does not capture the spatial distribution of the features inside the box, we denote as $\mathcal{A}$ the set of space-time points that belong to the region of a used SURF descriptor (ellipse), and we define a point-risk $c_l^\mathbf{p} = \frac{\#\text{decisions for } \mathbf{p} \in \mathcal{A} \text{ as non–background}}{\#\text{features whose ellipse includes } \mathbf{p}}$. Based on the box-wise and pixel-wise background unreliabilities, we obtain a risk factor for all space-time points in the $l^{th}$ multiframe:

$$C^l(\mathbf{p}) = \begin{cases} c_{k,l}^{box} + c_l^\mathbf{p} & \text{if } \mathbf{p} \in B_k \bigcap \mathcal{A} \\ c_l^\mathbf{p} & \text{if } \mathbf{p} \in \mathcal{A},\ \mathbf{p} \notin B_k \\ c_{k,l}^{box} & \text{if } \mathbf{p} \in B_k,\ \mathbf{p} \notin \mathcal{A} \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Note that $\Omega$ and $\mathcal{A}$ are not necessarily the same areas, as well as that a pixel can participate in the computation of several SURF descriptors. As a result, it is likely to be classified many times. It is important to also note here that we could extend this by using class-dependent detectors (e.g. car or people detector), but this is beyond the scope of this paper.

#### 3.3.2 The smoothness terms

As outlined above, the labeling inference is penalized by the spatial and temporal smoothness terms. In specific, these terms are defined as:

$$E_S(f_\mathbf{p}) = \sum_{\mathbf{p},\mathbf{q} \in \mathcal{N}_t^p} V_{\mathbf{p},\mathbf{q}}(f_\mathbf{p}, f_\mathbf{q}), \tag{4}$$

$$E_T(f_\mathbf{p}) = \sum_{\mathbf{p},\mathbf{r} \in \mathcal{N}_{t+1}^p} V_{\mathbf{p},\mathbf{r}}(f_\mathbf{p}, f_\mathbf{r}) + \sum_{\mathbf{p},\mathbf{z} \in \mathcal{N}_{t-1}^p} V_{\mathbf{p},\mathbf{z}}(f_\mathbf{p}, f_\mathbf{z}), \tag{5}$$

with $\mathcal{N}_t$ being the set of 4 adjacent pixels of $p$ in frame $t$, while $\mathcal{N}_{t+1}^p$ and $\mathcal{N}_{t-1}^p$ are the *corresponding pixels* of $\mathcal{N}_t$ in the next $(t+1)^{th}$ and previous $(t-1)^{th}$ frame, respectively. In order to obtain $\mathcal{N}_{t+1}$ and $\mathcal{N}_{t-1}$, we use an optical flow algorithm that provides the correspondences [6], and we round the coordinates of the corresponding points of $\mathcal{N}_t$. Finally, the pairwise potential is defined as in [9]:

$$V_{\mathbf{p}_1,\mathbf{p}_2}(f_{\mathbf{p}_1}, f_{\mathbf{p}_2}) = \frac{\sum_{i=1}^{2} \|J_{f_{\mathbf{p}_1}}(\mathbf{p}_i) - J_{f_{\mathbf{p}_2}}(\mathbf{p}_i)\|_2}{2}. \tag{6}$$

When the sequences show strong appearance variation, the intensities in the above formula can be replaced by their gradients [22].

## 4. Complexity

Although the proposed scheme is presented as an offline process, it is worth discussing the complexity and the bottleneck of the pipeline. As mentioned, the synchronization part is very efficient due to the retrieval approach and the use of SURF descriptor. As far as a single frame is concerned, the extraction of SURF features, the query to the codebook and the retrieval takes less than 0.5 sec on average. Smoothing implies a post-processing task. When all corresponding frames are retrieved, LOWESS smoothing that uses here a span of 15 frames requires less than 10 msec for a sequence of 1000 frames.

Once the reference frame indices are known for all input frames, R-ECC scheme spatially aligns every input frame with its match. Our modification of the ECC algorithm does not sensibly add complexity since the only extra steps are the computation of the weight per block, the interpolation of the weights and the weighting of images, which is a linear time task. In practice, the time required for the spatial alignment per frame is less that 0.5 sec.[1]

The morphological processing that detects areas of possible changes requires negligible time. The most demanding task of the proposed pipeline is the grounding step. Despite the low number of states (labels), the MRF solver integrates a temporal smoothness constraint, that builds on

[1]The original ECC algorithm is available in C through the OpenCV library, http://opencv.org

optical flow results, which means that the every space-time point of the input multiframe has to be labeled. Moreover, the SURF features being available from the synchronization process, have to be classified in order to compute the risk factor for each space-time point. A typical implementation of the grounding process requires around 30 sec per multiframe.

Notice that the above times regard a non-optimized combined Matlab-C implementation, while they vary with the parameters such as the resolution of the frames.

## 5. Experiments

We present in this section qualitative results to validate the proposed approach by comparing our algorithm with the state-of-the art. The evaluation counts on experimenting with real data-sets, namely '*Backroad*' and '*Highway*' scenarios [11]. Each scenario consists of two sets of three video sequences captured by in-vehicle cameras that follow the same road at different times. The grounding constitutes a challenging task since the videos show dissimilar content, e.g. different vehicles appear in the sequences, while the speed of vehicles varies irregularly. The resolution of sequences is $720 \times 540$ in space and 2000 frames on average in time. These data-sets are provided with their synchronization ground truth which allows for the evaluation of the proposed synchronization (see Table 1).

We implemented our solver based on the Graph-Cut algorithm (expansion mode) of the GCO library [5], while our method is compared to the most related work [7, 22] using the same aligned corresponding frames. The former [7] aims to estimate the background in an image using images captured by the same viewpoint. The latter [22] instead aims to replace a specific region of an image using images of the same scene retrieved by viewpoint-invariant image search. Both methods formulate a labeling problem [9] and assume that background pixels at least appear in one of the sources. Our grounding algorithm makes use of the optical flow scheme of [6] in order to locate the neighbourhoods in adjacent frames, while we set $\lambda_S = \lambda_T = 1$ in (1).

Fig. 5 shows challenging instances where vehicles may appear in more than one frame. We observe that the proposed method provides promising grounding results. Both the competitors fail more frequently to remove foreground objects. This verifies the contribution of our novel framework that exploits classification results and the temporal coherence of videos. In Fig. 6, the contribution of the various terms in the energy function is shown. It is obvious that the use of the classifier leads to a more smooth labeling with the minimum variation. Since it is difficult to present in a document the performance of a video processing algorithm, we provide additional videos (see supplementary material). Moreover, we discuss below two degenerate cases and give some preliminary results.

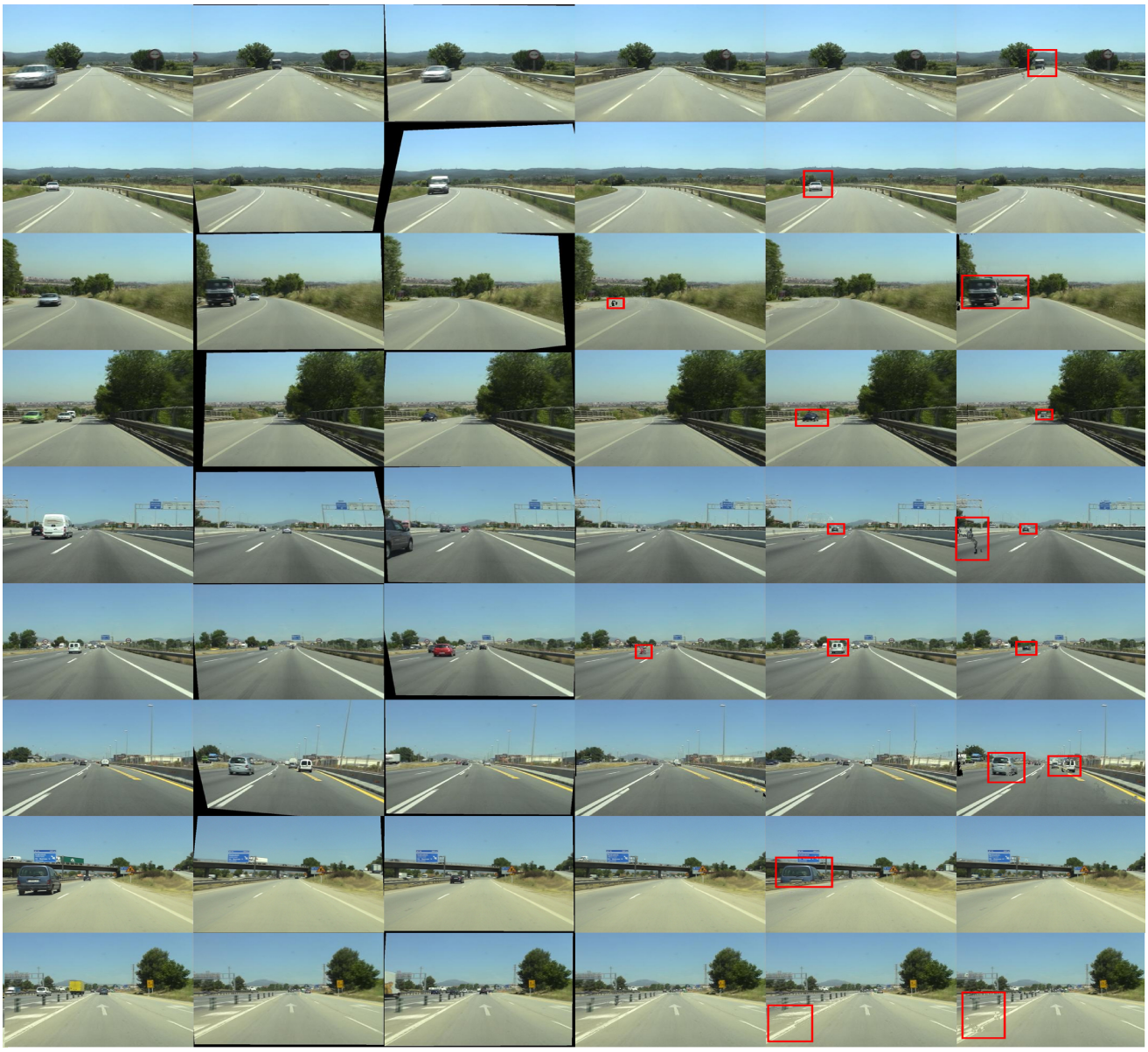
### 5.1. Double grounding

As discussed in Sec.2, when one reference sequence is available, we may end up with double occlusions, i.e. an area $B_k$ is occluded in both frames $I_n$ and $I_m$. Such an area can be detected by checking the condition $min(c^{box}_{k,0}, c^{box}_{k,1}) > T_0$ where $T_0$ is a reasonable threshold. As for the grounding, we need to exploit neighbouring reference frames. In this example, we compute the optical flows [6] between the reference frame $I_m$ and its neighbours $I_{m\pm\lambda m_0}$ for $m_0 = 5$ and $\lambda \in \{1, 2, 3, 4\}$. Note that with a slowly moving camera, a higher step might be needed. Based on the computed flows, we warp each frame with respect to $I_m$, we compute the median and we register it with $I_n$, thus providing a grounded reference frame. We then use the latter as an extra source towards $I_n$'s grounding. Fig. 7 shows a result of double grounding where $T_0 = 0.3$. While the methods fail to reconstruct the background using the two frames, the new grounded reference frame helps the grounding and the reconstruction is better. Alternatively, we could directly warp neighbouring reference frames with respect to $I_n$ and use all of them for the background estimation. We note that double grounding cannot completely reconstruct the background, but works better than using the initial images.

### 5.2. Frame Inpainting

Another degenerate case is when the reference sequence is missing. In other words, there is no source for the background reconstruction and we should use inpainting. We consider the example of Fig. 8 that shows a single frame with two boxes detected by the classifier. To exploit the temporal coherence, we modified the algorithm of Criminisi *et al.* [10] by extending it in the temporal domain. In specific, this method is an exemplar-based inpainting that finds the best exemplar inside the image to fill the missing pixels. Our modification enables both color and motion information to find exemplars, i.e. any exemplar point is represented by a $7D$ vector (3 color channels,4 flows). Again, we use [6] to estimate the flows. As we observe in Fig. 8(b-c) the method of Criminisi *et al.* [10] causes more artifacts in the filled region, while our modification achieves more acceptable results since it favours more the use of exemplars that come from the same object. Note that inpainting cannot ensure the reconstruction and it should be used as a last resort. Moreover, the user intervention may be occasionally required.

**Supplemental material**: Grounded videos obtained by the methods are available in the url `http://team.inria.fr/perception/research/CVVT2013`.



(a) (b) (c) (d) (e) (f)

Figure 5. (a) Input frames, (b,c) aligned corresponding reference frames using our methods described in Sec. 3.1 and Sec. 3.2, and grounded frames of (d) our method, (e) [7] and (f) [22].

## 6. Conclusions

A video grounding approach has been introduced in this paper. The algorithm reconstructs the background of a 3D scene captured from a moving camera, based on reference videos that recorded the same scene in the past. After their spatiotemporal alignment, registered videos are plugged into an MRF solver that solves a labelling problem for the input frames. The data term of the energy function integrates results from a classification task while the smoothness term integrates the temporal coherence of data. Experiments with real videos captured from moving cameras demonstrate the performance of the proposed method. Future research includes the use of class-dependent detectors and segmentation results within the video grounding algorithm.

## References


[1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM SIGGRAPH 2009*, 2009. 1, 2

[2] A. Bartoli, N. Dalal, and R. P. Horaud. Motion panoramas. *Computer Animation and Virtual Worlds*, 15:501–517, 2004. 2

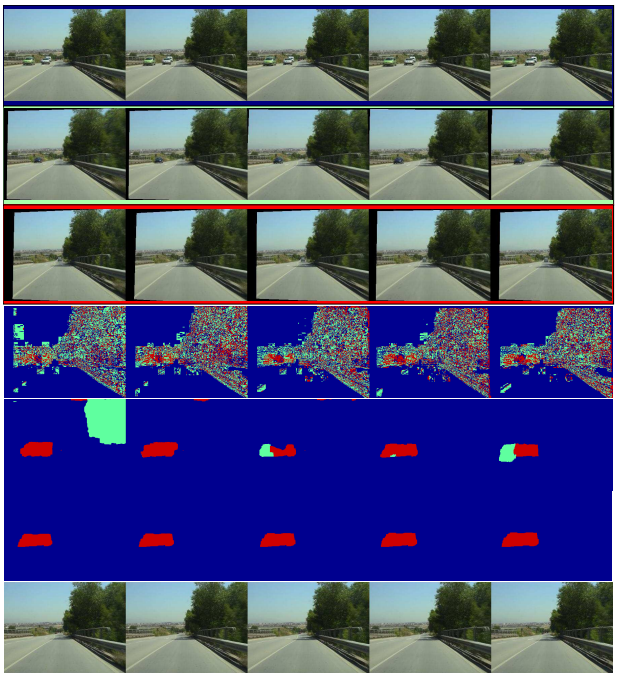


Figure 6. The contribution of different data terms to the final grounding for successive frames. *From top to bottom*: Input frame, two reference frames, the likelihood of the color stationariness ($D^c$), the likelihood of the joint color-background stationariness ($D^c + D^b$), the weighted likelihood ($C(\mathbf{p})[D^c + D^b]$), the final grounded frame.


Figure 7. An example of double grounding. *Top*: (a) Two corresponding frames, (b) our result and (c) result of [7] using frames of (a). *Bottom*: (d) the grounded reference frame and the results of (e) our method, (f) [7] and (g) [22] by considering the grounded reference frame as an extra source.


Figure 8. An example of image inpainting: (a) input frame with the regions to be filled, (b) result by Criminisi et al. method [10], (c) our result.


*PAMI*, 33(3):500–513, 2011. 5, 6

[7] X. Chen, Y. Shen, and Y. H. Yang. Background estimation using graph cuts and inpainting. In *Proc. of Graphics Interface (GI)*, 2010. 1, 2, 4, 6, 7, 8

[8] W. S. Cleveland and S. J. Devlin. Locally-weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988. 3

[9] S. Cohen. Background estimation as a labeling problem. In *ICCV*, pages 1034–1041, 2005. 1, 2, 4, 5, 6

[10] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE TIP*, 13(9):1200–1212, 2004. 2, 6, 8

[11] F. Diego, D. Ponsa, J. Serrat, and A. M. Lopez. Video alignment for change detection. *IEEE TIP*, 20(7):1858–1869, 2011. 1, 2, 3, 4, 6

[12] G. D. Evangelidis and C. Bauckhage. Efficient and robust alignment of unsynchronized video sequences. In *Proc. of 33rd German conference on Pattern recognition*, DAGM, 2011. 2, 3, 4

[13] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE T PAMI*, 30(10):1858–1865, 2008. 1, 3, 4

[14] R. P. Horaud, D. Knossow, and M. Michaelis. Camera cooperation for achieving visual attention. *Machine Vision and Applications*, 16(6):331–342, 2006. 2

[15] C. Liu, J. Yuen, A. Torralba, and W. T. Freeman. Sift flow: dense correspondence across different scenes. In *ECCV*, 2008. 1, 2, 3, 4

[16] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting under constrained camera motion. *IEEE TIP*, 16(2):545–553, 2007. 1, 2

[17] P. Sand and S. Teller. Video matching. *ACM Trans. on Graphics*, 22(3):592–599, 2004. 2

[18] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction from freely moving cameras. In *ICCV*, 2009. 1, 2

[19] Y. Shen, F. Lu, X. Cao, and H. Foroosh. Video completion for perspective camera under constrained motion. In *ICPR*, 2006. 1, 2

[20] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 3

[21] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE T PAMI*, 29:463–476, 2007. 1, 2

[22] O. Whyte, J. Sivic, and A. Zisserman. Get out of my picture! internet-based inpainting. In *BMVC*, 2009. 1, 2, 4, 5, 6, 7, 8