Robust 3D Tracking with Descriptor Fields

Alberto Crivellaro
Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
alberto.crivellaro@epfl.ch

Vincent Lepetit
Institute for Computer Graphics and Vision
Graz University of Technology
lepetit@icg.tugraz.at

Abstract

We introduce a method that can register challenging images from specular and poorly textured 3D environments, on which previous approaches fail. We assume that a small set of reference images of the environment and a partial 3D model are available. Like previous approaches, we register the input images by aligning them with one of the reference images using the 3D information. However, these approaches typically rely on the pixel intensities for the alignment, which is prone to fail in presence of specularities or in absence of texture. Our main contribution is an efficient novel local descriptor that we use to describe each image location. We show that we can rely on this descriptor in place of the intensities to significantly improve the alignment robustness at a minor increase of the computational cost, and we analyze the reasons behind the success of our descriptor.

1. Introduction

Despite a long history of research in 3D tracking [5, 14], it is still very challenging to reliably register poorly textured, specular objects. This is a clear obstacle to the development of Robotics and Augmented Reality applications in industrial environments, where such objects can typically be found.

In this paper, we introduce an approach that we refer to as "Descriptor Fields" and that resolves this issue. We rely on a dense image alignment framework [15, 2, 7, 1, 18, 19]. Dense alignment is attractive because it globally exploits most of the image information, even when local image features such as interest points or edges are ambiguous. However it typically relies directly on image intensities, which is prone to fail in presence of non-Lambertian effects such as specularities, or when the objects do not exhibit convenient textures. Moreover, a multi-scale approach is typically required for robust alignment, where low-pass filters are applied to the signals to align. When the signals are the image intensities, or a linear combination of them, low-pass filtering rapidly deteriorates information.

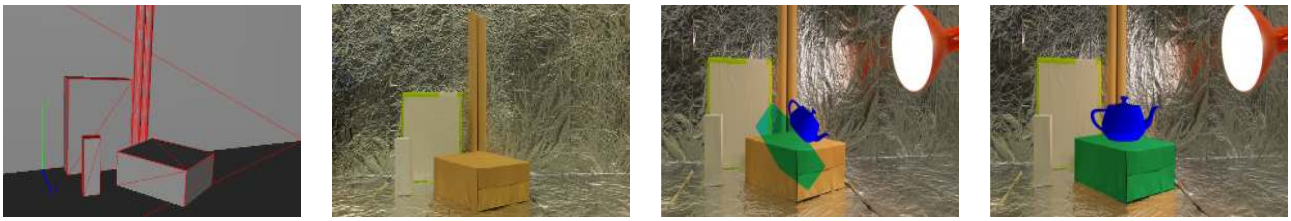
We therefore propose to use a more robust local descriptor in place of the pixel intensities. As shown in Fig. 1, our descriptor allows us to handle challenging imaging artifacts such as a strong lamp moving in a highly specular, poorly textured environment. Our descriptor is computed from a small set of convolutional filters applied to the images, which makes it suitable for real-time applications. However, instead of relying on the simple linear transformation of the intensity signal issued by the convolutions, we apply a non-linear operation that separates the descriptors' positive values from the negative ones. Our experimental results show that this step is crucial for obtaining the best tracking performances.

This can be explained by the fact that, thanks to our non-linear operation, our Descriptor Fields remain discriminant even after low-pass filtering. As a result, large Gaussian kernels can be used to significantly broaden the region of convergence of the alignment optimization algorithms, which is an important factor for robustness. Our approach is somehow related to the recent "Distribution Fields" (DFs) method [23]. However our experiments show that, on our challenging sequences, DFs fail even more often than simple image intensities.

In the remainder of this paper, we first introduce related work. We then describe our Descriptor Fields, and compare them against state-of-the-art methods on challenging sequences.

2. Related work

The literature on 3D tracking is vast and many different approaches have been proposed. The first Computer Vision methods were based on image contours [5, 3]; however, these are relatively fragile in practice. For example, in the environment depicted in Fig. 1, the object contours are perturbed by their reflections on the metallic surface and the contours of the specularities in the background. Then feature point-based methods [26, 28, 8] became popular because they are more robust, but they are suitable only for textured and Lambertian environments. We tested



(a) (b) (c) (d)

Figure 1. Given a partial 3D model of the environment such as the one shown in (a), we register the input images by aligning them with one reference view of the environment. The virtual teapot and the green model for the box in the middle of the image correctly overlaid in the input image show that our approach registered image (d) correctly, despite the strong lamp changing the illumination and partially occluding the scene. By contrast, aligning the images based on the pixel intensities as it is usually done completely fails, as shown in (c).

PTAM [8], a state-of-the-art SLAM system on our datasets without success.

With the growing computational power of modern devices, dense image alignment approaches [15, 2, 7, 1, 18, 16, 19] have become very attractive, because they are not limited to edge or keypoint features and exploit most of the image information. Efficient optimization algorithms have been developed for this purpose, such as the Inverse Compositional Algorithm (ICA) [1] or the Efficient Second-order Method (ESM) [17].

These methods look for the pose of an input image by aligning the pixels of this image with those of a registered template. The quality of the alignment is typically asserted by the sum of squared differences of the location intensities. This assumes that the visible surfaces are Lambertian, and is not robust to specularities that appear, for example, on metallic surfaces. To handle specularities better, [24] proposes to split the tracked surface, such as a CD cover, into patches and normalize the patches independently. Although this significantly improves the robustness, it is not clear how to split an arbitrary surface, especially in 3D. [9] proposed a method to exploit specularities lying on such surfaces to improve the accuracy of the registration, but this works only in controlled environments.

Another family of approaches learn a distance function that can be optimized to track the target [20, 21]. Such approaches can be robust to challenging artifacts, but they are mostly suited for some specific targets such as a human face, and less to a general 3D scene, as they require a cumbersome learning stage.

Recent works have focused on detecting and tracking poorly textured objects [6]; however, they have not been demonstrated on specular objects. As tracking in [6] is done using a 3D reconstruction of the scene obtained with a depth camera, it is unlikely to perform well on specular objects as a 3D reconstruction of such objects is also difficult to obtain, even with a depth camera.

We show that we can rely on a simple local descriptor, computed for each image location, in place of the location intensity to significantly improve the image alignment in case of challenging illumination effects. Moreover, using our descriptor also widens the basin of attraction in a way related to Distribution Fields (DFs) [23]. DFs represent the image using histograms, but they have been demonstrated only with pixel intensities, and can handle only limited illumination changes.

Our method is also related to dense descriptors [25, 27], as we compute a descriptor at each pixel location. We show here what really matters in a local descriptor for robust alignment, and how it can be obtained with simple operations. As a result, our descriptor is much better suited to real-time applications as it is much lighter while sufficient for robustness.

3. Dense Alignment for Camera Tracking

Our general framework for camera registration is very similar to the ones of previous works. We describe it in this section for completeness. The next section will describe our main contribution, the Descriptor Fields.

We assume that we have a partial 3D model of the scene such as the one shown in Fig. 1(a), and a small set of registered images $\mathcal{T} = \{T_i\}$ of this scene that we refer to as templates. [1] Given an input image J, we estimate the camera pose $\widehat{\mathbf{p}}$ for this image by aligning J with one of the templates T. To find an appropriate T given J, we use the same method as in [8] and pick the template that maximizes the normalized cross-correlation with J.

As shown in Fig. 2, alignment is done based on a transfer function $W(\mathbf{x}, \mathbf{p}_T, \mathbf{p})$. This function backprojects image location $\mathbf{x}$ on the scene 3D model using $\mathbf{p}_T$, the pose for template T, to find its corresponding 3D location, and returns the 2D projection of this 3D location under pose $\mathbf{p}$. We look for the pose $\widehat{\mathbf{p}}$ that transfers the image locations in T to locations in J that are similar according to some criterion. More exactly, we consider the following objective

[1] We use the semi-automatic ImageModeler software to quickly register the templates in $\mathcal{T}$ and simultaneously obtain the 3D model.

function:

$$F(\mathbf{p}) = \sum_{\mathbf{x}} \|d(J, W(\mathbf{x}, \mathbf{p}_T, \mathbf{p})) - d(T, \mathbf{x})\|^2 , \quad (1)$$

where $\mathbf{p}_T$ is the camera pose for template T and $d(I, \mathbf{x})$ is a function that returns a descriptor for location $\mathbf{x}$ in a generic image I, and take:

$$\widehat{\mathbf{p}} = \underset{\mathbf{p}}{\operatorname{argmin}} F(\mathbf{p}) . \quad (2)$$

In previous dense alignment works, $d(I, \mathbf{x})$ is almost always taken as $I(\mathbf{x})$, the intensity in image I at location $\mathbf{x}$. The Distribution Fields method considers a function that returns a vector where all values are 0 but one, and which depends on the interval $I(\mathbf{x})$ belongs to. In this paper, we show how to compute $d(I, \mathbf{x})$ for much better performances when complex illumination changes occur.

Several algorithms have been proposed to efficiently optimize functions in the form of Eq. (1), including the Lucas-Kanade (LK) algorithm [15, 1], the Inverse Compositional Algorithm (ICA) [1], and the Efficient Second Order Method (ESM) [17].

In practice, a multi-scale approach is used to optimize Eq. (1) by taking and considering the intermediate objective function:

$$F(\mathbf{p}; \sigma) = \sum_{\mathbf{x}} \|D_\sigma(J, W(\mathbf{x}, \mathbf{p}_T, \mathbf{p})) - D_\sigma(T, \mathbf{x})\|^2 , \quad (3)$$

where $D_\sigma(\mathbf{x})$ is a low-pass version of $d(\mathbf{x})$:

$$D_\sigma(\mathbf{x}) = (G^\sigma * d)(\mathbf{x}) \quad (4)$$

with G^σ a Gaussian kernel of standard deviation σ. The optimization scheme starts with a large value for σ, optimizes $F(\mathbf{p}; \sigma)$ to obtain a first estimate $\widehat{\mathbf{p}}$ of the actual pose, decreases σ, optimizes $F(\mathbf{p}; \sigma)$ again starting from $\widehat{\mathbf{p}}$, and iterates for a fixed number of iterations.

This multiscale optimization scheme is important in practice as low-pass filtering increases the basin of convergence, but it also degrades the localization of the minimum of the original function in Eq. (1). In our implementation, the optimization is initialized with the camera pose for the template T. We use 4 scales, with σ initialized to a fixed parameter σ_{max} for the coarsest scale, and divided by 2 between each scale level.

The next section discusses how we compute the d function to improve the convergence when the images exhibit challenging artifacts.

4. Descriptor Fields

As mentioned in the previous section, a very common choice for the function $d(I, \mathbf{x})$, which appears in Eq. (1) and on which image alignment is based, is simply

$$d(I, \mathbf{x}) = I(\mathbf{x}) , \quad (5)$$

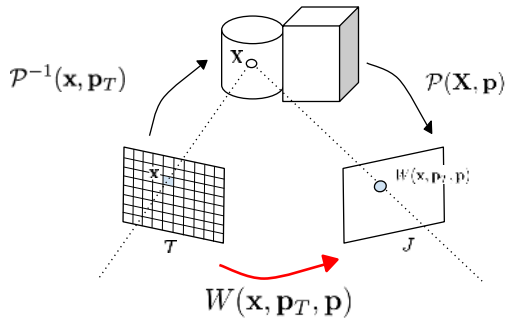


Figure 2. The transfer function $W(\mathbf{x}, \mathbf{p}_T, \mathbf{p})$ backprojects image location $\mathbf{x}$ on the scene 3D model using pose $\mathbf{p}_T$, the pose for template T, to find its corresponding 3D location, and returns the 2D projection of this 3D location under pose $\mathbf{p}$.

that is, the pixel intensity in image I at location $\mathbf{x}$. However, this option is very sensitive to complex light changes, especially in the absence of texture, as our evaluations presented in the next section will show.

To improve robustness, [23] proposed to use instead a vector of the form:

$$d(I, \mathbf{x}) = [\delta_{0 \le I(\mathbf{x}) < I_1}, \delta_{I_1 \le I(\mathbf{x}) < I_2}, \ldots, \delta_{I_{n-1} \le I(\mathbf{x}) < I_n}]^\top \quad (6)$$

where

$$\delta_{I_i \le I(\mathbf{x}) < I_{i+1}} = \begin{cases} 1 & \text{if } I_i \le I(\mathbf{x}) < I_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

for a fixed number of quantization bins n. Thanks to this "explosion" of the image intensities, large Gaussian kernels can be applied as in Eq. (4) in a multiscale approach to broaden the basin of attraction of the objective function, without blending the intensities together and loosing the image information. Unfortunately, this approach can only handle moderate changes in illumination, and failed on our test sequences.

While they have never been used for direct image alignment —to the best of our knowledge— it seems interesting to use "local jets" for the d function [4, 22, 10, 11]. Local jets are vectors often used as local descriptors and efficiently computed by convolving an image with a series of filters:

$$d(I, \mathbf{x}) = [(\mathbf{f}_1 * I)(\mathbf{x}), \ldots, (\mathbf{f}_n * I)(\mathbf{x})]^\top , \quad (8)$$

where the $\mathbf{f}_i$ filters are typically Gaussian derivatives kernels. We experimented with this approach and the results will be detailed in the next section. We also considered the following function, which is at the core of our Descriptor Fields:

$$d(I, \mathbf{x}) = \big[\quad [(\mathbf{f}_1 * I)(\mathbf{x})]^+, [(\mathbf{f}_1 * I)(\mathbf{x})]^-, \ldots, \\ [(\mathbf{f}_n * I)(\mathbf{x})]^+, [(\mathbf{f}_n * I)(\mathbf{x})]^- \quad \big]^\top , \quad (9)$$

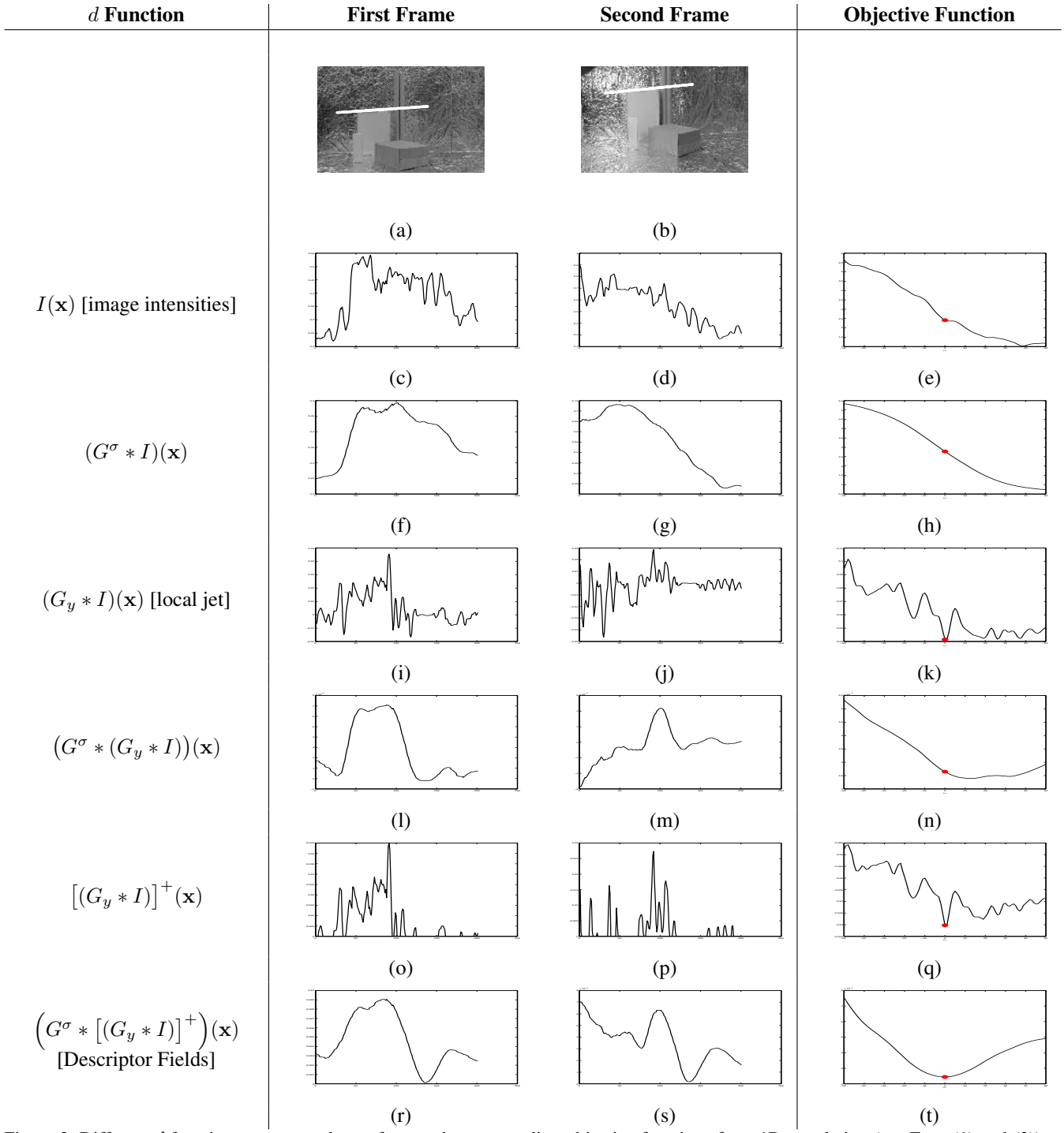


Figure 3. Different d functions on a specular surface, and corresponding objective functions for a 1D translation (see Eqs. (1) and (3)). (a & b) The reprojections of a 3D line lying on the background in two images, and (c & d) the intensity signal sampled on 200 equi-spaced points along the reprojections. (e) Objective function for these signals, which exhibits local minimums at wrong locations. The expected location for the global minimum is marked with a red dot. (f & g) The intensity signals after low-pass filtering. Local minimums disappeared from objective function (h), but there is no minimum at the expected location. (i & j) The intensity signals after convolution with the first derivative of a Gaussian kernel. (k) The objective function computed with local jets exhibits many local minimums. (l & m) The same signals after low-pass filtering. (n) The corresponding objective function becomes smoother, but the global minimum is at the wrong location, and a local minimum appeared. (o & p) The same signals after applying the $[.]^+$ operation and (q) the objective function computed with our Descriptor Fields. (r & s) Smoothing these signals preserves information, leading to the objective function (t), which is much better suited for numerical optimization.

where the $[\cdot]^+$ and $[\cdot]^-$ operations respectively keep the positive and negative values:

$$[x]^+ = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}, \text{ and} \qquad [x]^- = [-x]^+.$$

These operations are simple but fundamental, and we found the last descriptor given by Eq. (9) to be much more effective than the first version of Eq. (8), as exemplified in Fig. 3: when strong Gaussian smoothing is applied by the multiscale optimization described in Section 3, the intensity signal flattens making it difficult to align across two images. The same phenomenon happens to the local jet of Eq. (8), where positive and negative values eliminate each other during the low-pass filtering by a Gaussian kernel. By contrast, the descriptor of Eq. (9) is much more resilient, and stays discriminant after strong Gaussian smoothing. This yields an objective function with a large basin of attraction and a well localized minimum, which is key for robustness of the alignment.

In the next section, we evaluate these different functions, together with different optimization algorithms, on several challenging sequences.

5. Experimental Results

In this section, we first describe the datasets we used to evaluate our approach and the evaluation framework, and then present and discuss the results of the evaluation.

5.1. Datasets

There is no benchmark for the evaluation of 3D tracking algorithms on the challenging environments we consider, therefore we created our own datasets, in two different environments:

- **Experimental Setup Dataset**: Fig. 1 illustrates this first dataset. It is made from an experimental setup, where the background is covered with aluminum foil, and the foreground is made of non-textured boxes. A strong lamp can be moved freely in the scene. Since the aluminum foil is very reflective, the images contain many specularities that move with the lamp. The lamp can also partially occlude the scene. We used only one template and the 3D model made of 168 triangles shown in Fig. 1(a)-(b). We captured two video sequences. The first one is made of 394 frames, the camera remains still, and the lamp is moved around. The second video is made of 365 frames, and both the camera and the light source move.

- **ATLAS Dataset**: Fig. 4 shows images from this second dataset. This dataset was captured in the LHC particle detector of ATLAS experiment at CERN. We captured a first video made of 209 frames with a fixed camera and a strong moving light source. The second video is longer, with 683 frames, and is much more challenging. The camera moves sometimes very fast, which results in motion blur. The light source generates very bright specularities in an extremely dark environment. This mimics the conditions of images captured by a camera mounted on the helmet of a worker in the LHC. We used the very simple 3D model showed in Fig. 4(a) made of just 12 triangles, and 24 templates.

Moreover, in order to give an example of how our approach behaves in a Lambertian environment, we report the results of the tests performed on a video sequence of 414 frames showing the popular STOP sign of the METAIO Dataset ([13]) printed on a sheet of paper, with limited motion blur and stable illumination conditions. For this sequence we employed the same workflow described above, retrieving the full 3D pose with a 3D model (made of 2 triangles) and 11 templates.

We tested PTAM, the state-of-the-art SLAM method of [8], on the two specular scenes. After several attempts, we managed to initialize the 3D tracking under ambient light; however, tracking was lost as soon as a lamp was switched on. This attests the difficulty of our datasets, and shows that a feature point-based approach, such as PTAM, is not adapted.

To obtain the ground truth camera poses for our datasets, we had to register the images by manually matching 3D points on the scene models with their 2D reprojections in the images, and use these correspondences to estimate the camera poses with a PnP algorithm [12]. Our testing datasets, as well as some supplementary material, are available on the project page at `http://cvlab.epfl.ch/research`.

5.2. Evaluation Framework

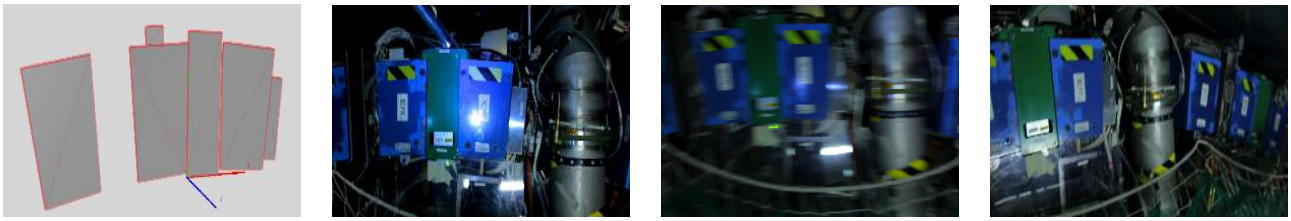
We evaluated different possibilities for the d function in Eq. (1) including the ones discussed in Section 4. In the following, we will refer to them as:

- Intensity: the simple case when $d(I, \mathbf{x}) = I(\mathbf{x})$;

- Magnitude of image gradient: in this case, d is taken as $d(I, \mathbf{x}) = \sqrt{(G_x * I)(\mathbf{x})^2 + (G_y * I)(\mathbf{x})^2}$, the magnitude of the image gradient at location $\mathbf{x}$. Like our descriptors, it is a non-linear function of the image intensities;

- 1st-order local jet: the simple local jet $d(I, \mathbf{x}) = [(G_x * I)(\mathbf{x}), (G_y * I)(\mathbf{x})]^\top$ as given in Eq. (8), where G_x and G_y are the first derivatives of the Gaussian kernel of standard deviation 1.0;

- 1st- and 2nd-order local jet: the simple local jet as given in Eq. (8), with $\mathbf{f}_1 = G_x$, $\mathbf{f}_2 = G_y$, $\mathbf{f}_3 = G_{xx}$, $\mathbf{f}_4 =$



(a) (b) (c) (d)

Figure 4. (a) The 3D model we use for the ATLAS dataset. (b,c,d) Some images from the second video sequence of this dataset. The images exhibit large and bright specular spots and strong motion blur.

G_{xy}, $\mathbf{f}_5 = G_{yy}$, the first and second derivatives of the Gaussian kernel of standard deviation 1.0.

- 1^{st}-order Descriptor Fields: in this case, d returns our descriptor as given in Eq. (9), with $\mathbf{f}_1 = G_x$ and $\mathbf{f}_2 = G_y$, the first derivatives of the Gaussian kernel of standard deviation 1.0;

- 1^{st}- and 2^{nd}-order Descriptor Fields: in this case, d returns our descriptor as given in Eq. (9), with $\mathbf{f}_1 = G_x$, $\mathbf{f}_2 = G_y$, $\mathbf{f}_3 = G_{xx}$, $\mathbf{f}_4 = G_{xy}$, $\mathbf{f}_5 = G_{yy}$, the first and second derivatives of the Gaussian kernel of standard deviation 1.0.

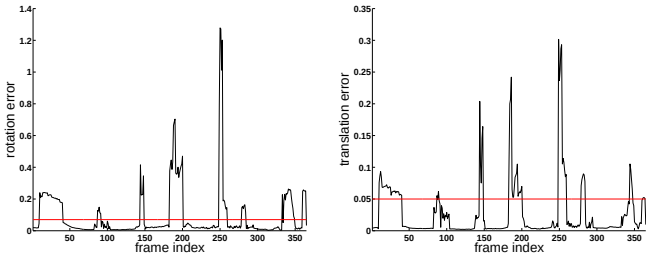
In all our experiments, the Distribution Fields method [23], as summarized in Eq. (6), performed badly whatever the values for n: it successfully registered no more than 10% of the frames. This is due to the fact that local specularities heavily alter the distribution of pixels in the bins of intensity histograms, so that Distribution Fields are totally unsuitable in presence of heavy light changes, even if image intensities are normalized before computing the descriptors.

Before computing these descriptors we first normalized the image intensities by subtracting their mean and dividing them by their standard deviation, as it improved the performances of all the methods.

Each of these descriptors was tested together with the Lucas-Kanade (LK) algorithm [15, 1], the Inverse Compositional Algorithm (ICA) [1], and the Efficient Second Order Method (ESM) [17]. We optimized the parameters of each method to obtain the best performances.

5.3. Evaluation

Table 1 summarizes the results of our experiments. We report the percentage of frames that were correctly registered, together with the average number of iterations required for convergence. To decide if a frame was correctly registered or not, we computed a rotation error R_{err} and a translation error t_{err}. The rotation error was taken as the distance between the exponential maps for the estimated pose and for the ground truth, and the translation error as the distance between the camera centers for these two poses. If



(a) (b)

Figure 5. (a) Rotation and (b) translation errors over the second video sequence of the Experimental Setup dataset, using ESM and our 1^{st}-order Descriptor Fields. The horizontal lines correspond to the thresholds used to detect incorrectly registered frames.

at least one of these errors is larger than a threshold, then we consider that the frame is not correctly registered. We use $\epsilon_{\text{rot}} = 0.07$ for the threshold on the rotation error, and $\epsilon_{\text{transl}} = 0.05$ for the threshold on the translation error. As shown in Fig. 5, the values of these thresholds are not critical: when a frame is not correctly registered, the rotation and translation errors tend to be very large.

As can be seen in the table, the results with our Descriptors Fields are consistently better than the other approaches, for all the videos and the optimization methods.

In all the specular video sequences, our descriptor with first-order Gaussian derivatives outperforms all other approaches based on first-order derivatives. Using both first- and second-order derivatives can further improve performances at a higher computational cost.

Figs. 6 and 7 show some images from our datasets augmented with virtual objects using the poses estimated with our first-order Descriptor Fields. The virtual objects are consistently integrated in the images, which assesses that the camera poses were correctly estimated.

6. Conclusion

We presented a local descriptor that makes dense alignment methods much more robust to various imaging artefacts. It is efficient and very simple to implement. It should therefore be very easy to integrate it into existing imple-

Descriptor	Optimization Method	Lambertian Env. Video	Exp. Setup Video #1	Exp. Setup Video #2	ATLAS Video #1	ATLAS Video #2
Intensity	LK	88.7 (16.2)	25.6 (48.7)	10.7 (76.5)	40.7 (70.3)	21.7 (44.6)
	ICA	16.0 (17.1)	42.1 (72.9)	22.2 (49.2)	88.6 (117.7)	19.3 (32.6)
	ESM	72.7 (43.9)	34.7 (46.8)	21.9 (46.2)	36.8 (62.1)	22.5 (40.8)
Magnitude of image gradient	LK	84.2 (22.)	52.0 (55.6)	81.0 (55.1)	99.5 (45.3)	33.6 (44.1)
	ICA	18.4 (16.7)	83.9 (71.9)	73.9 (45.4)	96.6 (27.2)	29.5 (31.7)
	ESM	67.8 (31.3)	90.8 (30.5)	92.0 (43.1)	89.9 (33.3)	19.7 (28.4)
1^{st}-order local jet	LK	85.2 (29.1)	75.6 (39.0)	52.3 (37.5)	100 (73.6)	31.5 (32.6)
	ICA	28.3 (23.5)	73.0 (33.5)	50.1 (41.7)	100 (50.5)	23.4 (34.2)
	ESM	78.3 (35.0)	75.6 (27.8)	49.5 (25.8)	100 (67.7)	24.7 (35.8)
1^{st}- and 2^{nd}-order local jet	LK	91.2 (27.4)	67.8 (49.0)	46.8 (45.4)	100 (36.1)	31.5 (31.3)
	ICA	57.7 (20.5)	71.0 (40.7)	46.3 (63.6)	98.5 (37.7)	22.9 (27.7)
	ESM	84.9 (26.0)	74.4 (34.2)	50.7 (33.6)	100 (27.9)	24.0 (21.3)
1^{st}-order Descriptor Fields	LK	89.3 (25.4)	**85.0** (**49.0**)	87.9 (86.5)	100 (47.6)	**39.4** (**33.8**)
	ICA	37.0 (22.4)	**91.4** (**51.7**)	82.2 (66.3)	100 (29.9)	32.5 (27.4)
	ESM	77.0 (38.6)	**98.4** (**30.4**)	97.5 (36.9)	100 (51.3)	32.6 (21.3)
1^{st}- and 2^{nd}-order Descriptor Fields	LK	**93.3** (**35.9**)	76.1 (63.3)	**89.3** (**69.9**)	**100** (**24.4**)	39.0 (30.7)
	ICA	**61.9** (**23.6**)	82.7 (47.2)	**85.2** (**62.3**)	**100** (**22.7**)	**32.5** (**24.7**)
	ESM	**87.5** (**33.9**)	92.8 (42.4)	**97.8** (**39.4**)	**100** (**19.0**)	**33.4** (**18.5**)

Table 1. Experimental results. We give the percentages of correctly calibrated frames and the average number of iterations in parentheses for each descriptor, each video sequence, and each optimization method we considered. The best results for each video and each optimization methods are in bold. Our Descriptor Fields consistently outperform the other descriptors.

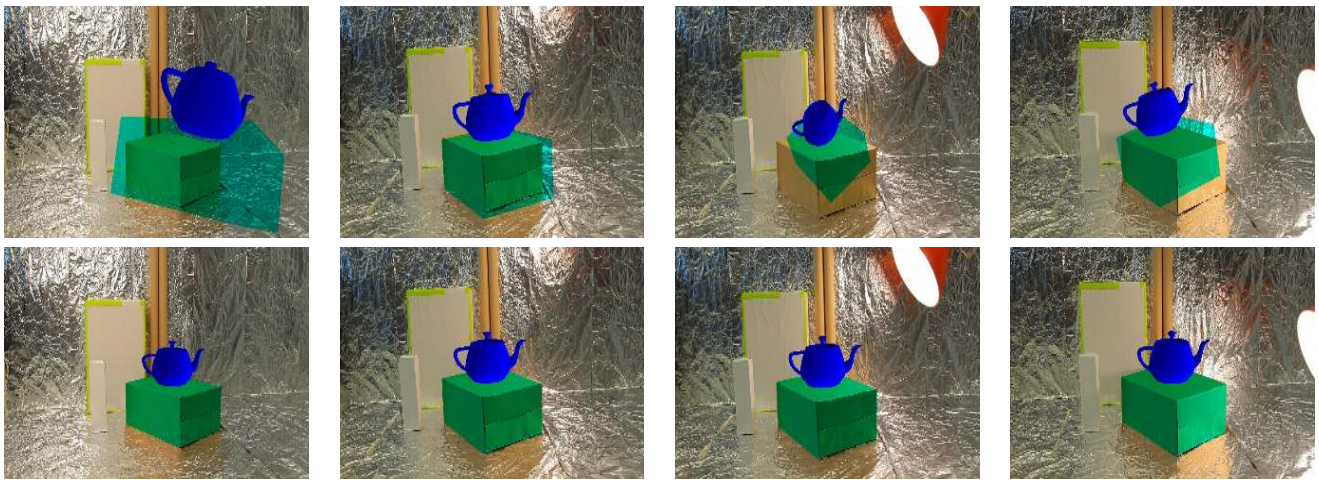


Figure 6. Comparisons on our Experimental Setup dataset. First row: Using the image intensities. Second row: Using our Descriptor Fields. The scene is augmented with the obligatory teapot to visually attest the accuracy of the estimated poses. With our method, the teapot is correctly added to the images, despite the moving lamp that changes the lighting and partially occludes the scene. The full video is available on the project page at `http://cvlab.epfl.ch/research`.

mentations of image alignment algorithms, to improve their robustness.

Aknowledgment

This work was supported in part by the EU project EDUSAFE. The authors thank Tomasz Trzcinski and Roberto Rigamonti for their help and advice.

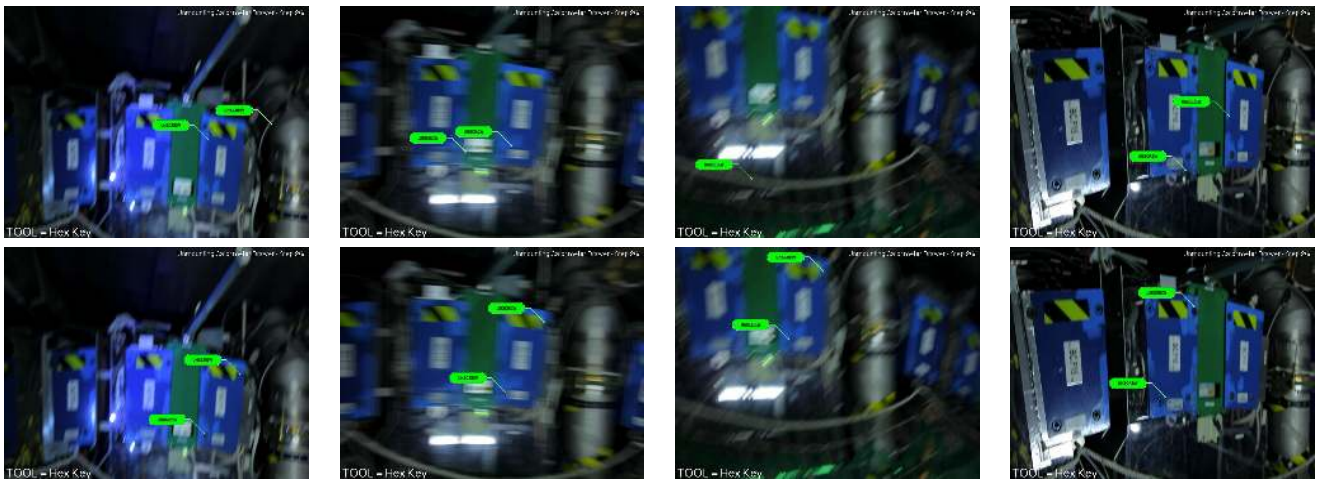


Figure 7. Comparisons on our ATLAS dataset. First row: Using the image intensities. Second row: Using our Descriptor Fields. Despite the bright specularities and the motion blur, we can add virtual labels at the right place in the images with our method. The full video is available on the project page at `http://cvlab.epfl.ch/research`.

References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, March 2004.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. *PAMI*, 23(6), June 2001.

[3] T. Drummond and R. Cipolla. Real-Time Visual Tracking of Complex Structures. *PAMI*, 27(7), July 2002.

[4] L. Florack, B. Romeny, M. Viergever, and J. Koenderink. The Gaussian Scale-Space Paradigm and the Multiscale Local Jet. *IJCV*, 18, 1996.

[5] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Fourth Alvey Vision Conference*, 1988.

[6] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient Response Maps for Real-Time Detection of Textureless Objects. *PAMI*, 34(5), May 2012.

[7] F. Jurie and M. Dhome. Hyperplane Approximation for Template Matching. *PAMI*, 24(7), July 2002.

[8] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, November 2007.

[9] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua. 3D Pose Refinement from Reflections. In *CVPR*, June 2008.

[10] I. Laptev and T. Lindeberg. Local Descriptors for Spatio-Temporal Recognition. In *Spatial Coherence for Visual Motion Analysis, Lecture Notes in Computer Science*, 2006.

[11] A. Larsen, S. Darkner, A. Dahl, and K. Pedersen. Jet-Based Local Image Descriptors. In *ECCV*, 2012.

[12] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate $O(n)$ Solution to the PnP Problem. *IJCV*, 2009.

[13] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. A dataset and evaluation methodology for template-based tracking algorithms. In *ISMAR*, 2009.

[14] D. G. Lowe. Robust Model-Based Motion Tracking through the Integration of Search and Estimation. *IJCV*, 8(2), 1992.

[15] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981.

[16] S. Lucey, Y. Wang, and J. F. Cohn. Non-Rigid Face Tracking with Enforced Convexity and Local Appearance Consistency Constraint. *IJCV*, 28(5), 2010.

[17] E. Malis. Improving Vision-Based Control Using Efficient Second-Order Minimization Techniques. In *ICRA*, 2004.

[18] C. Mei, S. Benhimane, E. Malis, and P. Rives. Efficient Homography-Based Tracking and 3D Reconstruction for Single-Viewpoint Sensors. *IEEE Transactions on Robotics*, 24(6), 2008.

[19] R. Newcombe, S. Lovegrove, and A. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *ICCV*, 2011.

[20] M. Nguyen and F. D. la Torre. Metric Learning for Image Alignment. *IJCV*, 88(1), 2010.

[21] G. Scandaroli, M. Meilland, and R. Richa. Improving NCC-Based Direct Visual Tracking. In *ECCV*, 2012.

[22] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. *PAMI*, 19(5), May 1997.

[23] L. Sevilla-Lara and E. Learned-Miller. Distribution Fields for Tracking. In *CVPR*, 2012.

[24] G. Silveira and E. Malis. Real-Time Visual Tracking Under Arbitrary Illumination Changes. In *CVPR*, 2007.

[25] E. Tola, V. Lepetit, and P. Fua. A Fast Local Descriptor for Dense Matching. In *CVPR*, 2008.

[26] L. Vacchetti, V. Lepetit, and P. Fua. Stable Real-Time 3D Tracking Using Online and Offline Information. *PAMI*, 26(10), October 2004.

[27] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008.

[28] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose Tracking from Natural Features on Mobile Phones. In *ISMAR*, September 2008.