Adaptive Lucas-Kanade tracking

Yassine Ahmine, Guillaume Caron, El Mustapha Mouaddib, Fatima Chouireb

Adaptive Lucas-Kanade tracking

Yassine Ahmine, Guillaume Caron, El Mustapha Mouaddib, Fatima Chouireb

Adaptive Lucas-Kanade tracking

Yassine AHMINE[a,b], Guillaume CARON[a], El Mustapha MOUADDIB[a], Fatima CHOUIREB[b]

[a] *MIS Laboratory, Université de Picardie Jules Verne, 33 rue Saint Leu, Amiens 80000, France*
[b] *LTSS Laboratory, Université Amar Telidji de Laghouat, BP 37G route de Ghardaia, Laghouat 03000, Algeria*

ABSTRACT

Dense image alignment, when the displacement between the frames is large, can be a challenging task. This paper presents a novel dense image alignment algorithm, the *Adaptive Forwards Additive* Lucas-Kanade (AFA-LK) tracking algorithm, which considers the scale-space representation of the images, parametrized by a scale parameter, to estimate the geometric transformation between an input image and the corresponding template. The main result in this framework is the optimization of the scale parameter along with the transformation parameters, which permits to significantly increase the convergence domain of the proposed algorithm while keeping a high estimation precision. The performance of the proposed method was tested in various computer-based experiments, which reveal its interest in comparison with geometric as well as learning-based methods from the litterature, both in terms of precision and convergence rate.

1. Introduction

The estimation of the parametric transformation between two images i.e. image alignment, is a key part of various applications such as: optical flow estimation [9], Visual Odometry (VO) [15], Visual Simultaneous Localization And Mapping (V-SLAM) [14], image mosaicing [10], and image stitching [8]. The ability of the image registration algorithm to be robust to large displacements and low textured scenes, is essential for such applications.

In the literature, image alignment algorithms can be mostly classified into feature-based methods [**?**] and pixel-based (direct) methods [**?**]. In feature-based methods, the image alignment is done by a feature extraction/matching process in the images, followed by an estimation of the transformation parameters from points correspondences. This type of image registration methods has the advantage of being robust to changes in scale, orientation, and lighting because feature descriptors [20, 13] of these methods are relatively invariant to these changes. It is worth noting that the "scale" here refers to a geometric quantity in contrast to the "scale" used in the rest of the paper, which refers to the degree of smoothness of the considered image. Their main drawback is that features have to be evenly distributed and precisely located in each image in order to achieve sub-pixel accuracy, which can be challenging in low-textured scenes.

Direct methods, for their part, use all pixels of the images to estimate the transformation parameters. This type of methods permits to achieve better accuracy than feature-based methods and is more suitable for low-textured scenes. However, these pixel-based methods are more sensitive to large displacements.

One of the mostly used algorithms for direct image alignment is the Lucas-Kanade (LK) algorithm [21], which was developed in the early eighties. Subsequently, numerous extensions and variants to this method were developed and can be found in the literature [5, 24]. [3] proposed a unifying framework for the variants of the LK algorithm. Recently, interest grew for combining feature descriptors with the LK approach [2, 7, 12]. [1] proposed a binary feature descriptor used with the LK algorithm, for dense image alignment under drastic illumination changes. Other approaches considered the use of learning methods combined with the LK algorithm. For instance, [11] proposed the combination of deep learning with the LK algorithm, and achieved subpixel accuracy with large displacements and color variations. Another example is the approach developed by [17], which learns a linear model to predict displacement from image appearance.

As stated before, the LK algorithm suffers from sensitivity

to large displacements, and the image alignment optimization scheme is highly susceptible of finding local minima in such conditions. In order to overcome this drawback a coarse-to-fine approach is generally adopted [6, 25]. This approach permits the first order Taylor's expansion, used by the LK algorithm, to better approximate the cost function and to enlarge the convergence domain (basin of convergence). Alternatively, one can blur the images with an isotropic Gaussian kernel to make the higher order terms of the Taylor's expansion negligible [23]. Such a blurring can be referred to as scale-space smoothing because smoothing an image with an isotropic Gaussian kernel permits to build a scale-space representation of the considered image [19]. [23] in their work, proposed to blur the objective function instead of the input images in order to remove local minima. [25] studied the effect of pre-filtering the input images on the LK optical flow, and concluded that Gaussian filtering provides the best results. However, they only provided empirical results for the choice of the standard deviation of the Gaussian kernel.

In this paper, we present a novel approach to the LK algorithm, where the optimization is done on a scale-space domain. This allows to expand the convergence domain of the algorithm while keeping accuracy. In this scheme, the scale parameter of the scale-space representation of the input image [19] is optimized, unlike [25] where the standard deviation of the Gaussian kernel was set in an empirical manner. By optimizing the scale parameter, the proposed method is able to automatically do the image alignment at coarse levels in a first time, where the input image is strongly blurred (corresponding to important values of the scale parameter); then, refining the result of the estimation at fine levels (corresponding to small values of the scale parameter).
In order to validate our approach, we first compare it to the original LK algorithm (*forwards additive* LK) for translation transformations. After that, we compare our method to the variants of the LK algorithm [3] and state-of-the-art methods [17, 5, 27, 16] for homography transformations. We used the MS-COCO dataset [18] and the Yale face database [4] for the different validations.

The rest of the paper is organized as follows: A summary of the contributions of our work is presented in section 2. Section 3 describes the derivation of the *adaptive forwards additive* Lucas-Kanade tracking algorithm. Section 4 presents the results of extended experiments for translation and homography transformations. Lastly, a conclusion about the results is presented in section 5.

2. Contributions Summary

The contributions provided by this work can be summarized in the following points:

1. We consider a scale-space representation of the input image and the template in order to enable the algorithm to automatically tune the degree of smoothness of the input image according to the needs of the image registration task.
2. We propose a framework for the optimization of the scale parameter among the transformation parameters.
3. The conducted evaluations present the effect of the different parameters on the behavior of the proposed algorithm, and show its effectiveness in comparison with geometric and learning-based methods; both in terms of accuracy and convergence domain.

3. Method Description

The proposed AFA-LK method aims at aligning an input image I to a template image T by estimating the parameters of a warping transformation $w(\mathbf{x};\mathbf{p})$ between both images, for all pixel coordinates $\mathbf{x} = (x, y)^T$. $\mathbf{p}$ is the parameters vector for the warping transformation. It is, for instance, a 2-vector $\mathbf{p} = (p_1, p_2)^T$ for a pure translation:

$$w(\mathbf{x};\mathbf{p}) = \begin{pmatrix} x + p_1 \\ y + p_2 \end{pmatrix} \tag{1}$$

or, for another exemple considered farther in this article, a 8-vector $\mathbf{p} = (p_1, p_2, ..., p_8)^T$ for a homography transformation $\mathbf{M}$:

$$\mathbf{M} = \begin{pmatrix} p_1 & p_4 & p_7 \\ p_2 & p_5 & p_8 \\ p_3 & p_6 & 1 \end{pmatrix} \tag{2}$$

leading to:

$$w(\mathbf{x};\mathbf{p}) = \begin{pmatrix} \frac{p_1 x + p_4 y + p_7}{p_3 x + p_6 y + 1} \\ \frac{p_2 x + p_5 y + p_8}{p_3 x + p_6 y + 1} \end{pmatrix}. \tag{3}$$

Classical LK methods optimize parameters $\mathbf{p}$ minimizing the sum of squared differences between target intensities $T(\mathbf{x})$, for each pixel $\mathbf{x}$ of T, and their corresponding intensities $I(w(\mathbf{x};\mathbf{p}))$ in the image. The proposed alignment method considers the scale-space representation [19] $G_I(w(\mathbf{x};\mathbf{p});\lambda)$ and $G_T(\mathbf{x};\lambda_{ref})$ of $I(\mathbf{x})$ and $T(\mathbf{x})$, respectively, according to the scale parameters λ and λ_{ref}:

$$G_I(w(\mathbf{x};\mathbf{p});\lambda) = I(w(\mathbf{x};\mathbf{p})) * g(\mathbf{x};\lambda) \tag{4}$$

and

$$G_T(\mathbf{x};\lambda_{ref}) = T(\mathbf{x}) * g(\mathbf{x};\lambda_{ref}) \tag{5}$$

where $g(\mathbf{x};\lambda)$ is an isotropic Gaussian kernel, and $*$ is the convolution operator. This alignment is done by solving the following problem:

$$\hat{\mathbf{P}} = \underset{\mathbf{P}}{argmin} \frac{1}{2} \sum_{\mathbf{x}} \Big[G_I(w(\mathbf{x};\mathbf{p});\lambda) - G_T(\mathbf{x};\lambda_{ref}) \Big]^2 \tag{6}$$

where $\mathbf{P} = (\mathbf{p}, \lambda)^T$ is a vector containing the parameters vector and the scale parameter. Eq. (6) is optimized based on the *forwards additive* variant of the Lucas-Kanade algorithm, in such a way that the warping parameters are updated as follows:

$$\mathbf{P} \leftarrow \mathbf{P} + \alpha \Delta \mathbf{P} \tag{7}$$

where α is a damping parameter. Eq. (6) is a nonlinear least squares problem that can be solved by considering a Gauss-Newton scheme, similarly to [3]. The expression of the increment is then:

$$\Delta\mathbf{P} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\frac{\partial G_I}{\partial \mathbf{P}}\right]^T \left[G_T(\mathbf{x};\lambda_{ref}) - G_I(w(\mathbf{x};\mathbf{p});\lambda)\right] \quad (8)$$

where $\mathbf{H}$ represents the (Gauss-Newton approximation to the) *Hessian* matrix:

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\frac{\partial G_I}{\partial \mathbf{P}}\right]^T \left[\frac{\partial G_I}{\partial \mathbf{P}}\right]. \quad (9)$$

The parameters $w(\mathbf{x};\mathbf{p})$ and λ of $G_I(w(\mathbf{x};\mathbf{p});\lambda)$ have been omitted for compactness in the writing of the *Jacobian* line $\frac{\partial G_I}{\partial \mathbf{P}} = \left[\nabla_{x,y} G_I \frac{\partial w}{\partial \mathbf{p}}, \nabla_\lambda G_I\right]$ of a pixel $\mathbf{x}$. Gradients of G_I in space and scale are evaluated at $w(\mathbf{x};\mathbf{p})$ using finite differences:

$$\nabla_x G_I(x,y;\lambda) \approx \left.\frac{G_I(x+l_x,y;\lambda) - G_I(x-l_x,y;\lambda)}{2l_x}\right|_{\mathbf{x}=w(\mathbf{x};\mathbf{p})} \quad (10)$$

$$\nabla_y G_I(x,y;\lambda) \approx \left.\frac{G_I(x,y+l_y;\lambda) - G_I(x,y-l_y;\lambda)}{2l_y}\right|_{\mathbf{x}=w(\mathbf{x};\mathbf{p})} \quad (11)$$

$$\nabla_\lambda G_I(x,y;\lambda) \approx \left.\frac{I * (g(\mathbf{x};\lambda+l_\lambda) - g(\mathbf{x};\lambda-l_\lambda))}{2l_\lambda}\right|_{\mathbf{x}=w(\mathbf{x};\mathbf{p})} \quad (12)$$

where $l_x = l_y = 1$ and $l_\lambda = 0.5$. The term $\frac{\partial w}{\partial \mathbf{p}}$ of $\frac{\partial G_I}{\partial \mathbf{p}}$ represents the *Jacobian* of the warping function:

$$\frac{\partial w}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial w_x}{\partial \mathbf{p_1}} & \frac{\partial w_x}{\partial \mathbf{p_2}} & \cdots & \frac{\partial w_x}{\partial \mathbf{p_n}} \\ \frac{\partial w_y}{\partial \mathbf{p_1}} & \frac{\partial w_y}{\partial \mathbf{p_2}} & \cdots & \frac{\partial w_y}{\partial \mathbf{p_n}} \end{pmatrix}. \quad (13)$$

If w is a translation (Eq. (1)), the following expression can be used to express the *Jacobian* of the warp:

$$\frac{\partial w}{\partial \mathbf{p}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (14)$$

If w is a homography (Eq. (3)), the expression of the *Jacobian* of the warp is:

$$\frac{\partial w}{\partial \mathbf{p}} = \frac{1}{c} \begin{pmatrix} x & 0 & -x.a & y & 0 & -y.a & 1 & 0 \\ 0 & x & -x.b & 0 & y & -y.b & 0 & 1 \end{pmatrix} \quad (15)$$

where

$$a = \frac{p_1 x + p_4 y + p_7}{c}$$

$$b = \frac{x[p_2 x + p_5 y + p_8]}{c}$$

$$c = p_3 x + p_6 y + 1.$$

These expressions permit to compute iteratively the parameters $\mathbf{P}$ that minimize the cost function. The integration of the scale parameter to the optimization permits the algorithm to automatically select the most suitable scale to converge. This key behavior increases the performances of the method as shown in the results presented in the next section.

4. Results

4.1. Overview

The proposed method was evaluated for two types of transformations: pure translations and homographies. Concerning the translations, the *Adaptive Forwards Additive* LK (AFA-LK) is compared with the original *Forwards Additive* LK (FA-LK) algorithm, and various initialization settings are presented. This type of transformation is considered because it is usually used in optical flow estimation applications. When homography transformations are considered, the comparison is done between the AFA-LK and state-of-the-art geometric and learning-based methods (Supervised Descent Method, Conditional-LK, ESM, RANSAC+SIFT Homography). The estimation of homography transformations is used in applications such as plane tracking and augmented reality. The MS-COCO dataset [18] and the Yale face database [4] are used for the evaluation process of the different methods. It is worth noting that throughout the validations the intensity values of the images are normalized between 0 and 1. The 110000 images used in the results part are available as the AFAMIS dataset [1]

4.2. Translation Transformation

The algorithms are evaluated using all the images of the validation set of the MS-COCO dataset [18], which consists of 5000 images of complex everyday scenes, by following the subsequent process for the generation of a testing bench (Fig. 1). Every image is used to generate a triplet $\{I, T, \mathbf{p}_{ref}\}$ by first translating the original image according to $\mathbf{p}_{ref}$, which is the reference translation applied in the x and y axes of this original image according to a uniform distribution within the range $[-10, 10]$ pixels. T is set to be the square region (29x29 pixels) around a randomly picked point of interest in the translated image and I is the square region of the same size and at the same location in the image, as shown in Fig. 1. This transformation range ($[-10, 10]$ pixels) is considered because optical flow estimation is done for small displacements and consequently

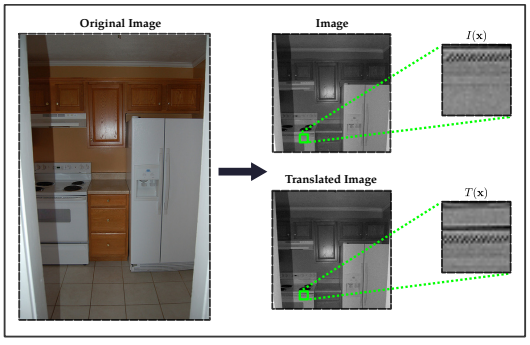


Fig. 1: Generation of $I(\mathbf{x})$ and $T(\mathbf{x})$. The original image is translated according to $\mathbf{p}_{ref}$. $T(\mathbf{x})$ is the square region (29x29) around a randomly picked corner in the translated image and $I(\mathbf{x})$ is the square region of the same size and at the same location in the image.

[1] http://mis.u-picardie.fr/~g-caron/pub/data/AFAMIS_dataset.zip

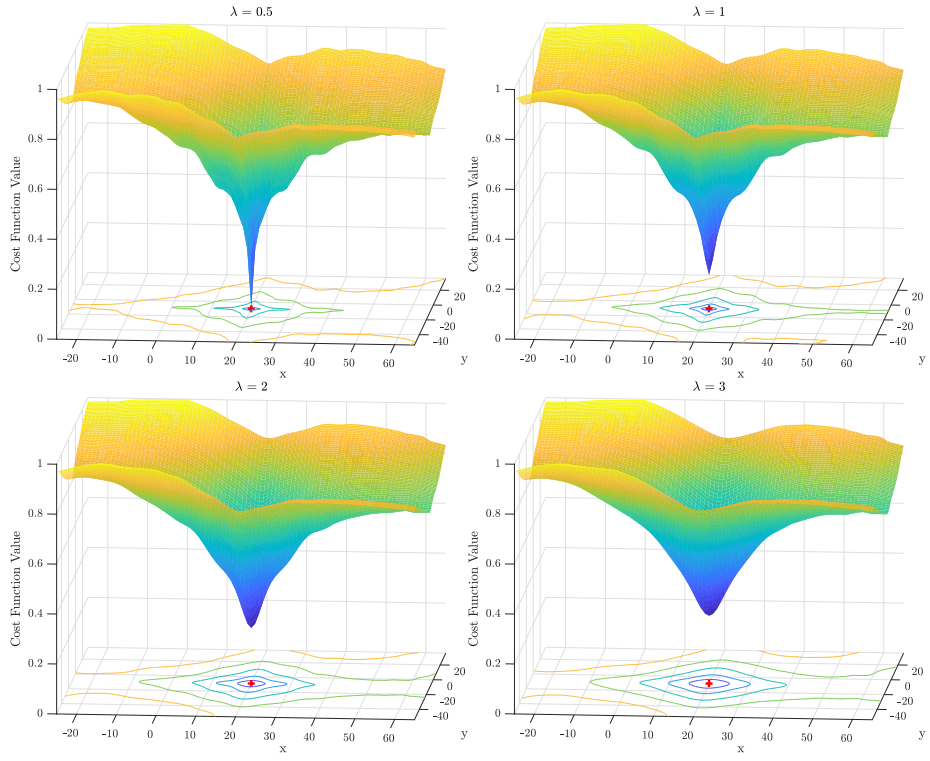


Fig. 2: Cost functions for different values of the scale parameter λ.

the used images sizes are small (29x29 pixels). Each algorithm is provided with the input and the template images I and T, and the initial parameters values ($\mathbf{p}_{init} = [0, 0]^T$).

A 3D visualization of the cost function (Eq. 6), at different scales, where the proposed algorithm was able to converge towards the correct value is presented in Fig. 2. The red marker corresponds to the location of the reference translation ($\mathbf{p}_{ref}$). It shows the effect of the scale parameter on the shape of the cost function. When the value of the scale parameter is high, the cost function is smoothed (typically in the begining of the optimization), which permits to avoid the local minima. Whereas when the value of the scale parameter is small (typically in the end of the optimization) the cost function is sharp, which permits to reach a high estimation precision. This results in a coarse-to-fine approach, where the scale parameter is automatically tuned according to the needs of the optimization process, and not empirically as usually done [25].

The results of the tests on the validation set of the MS-COCO dataset are presented in Fig. 3 and Fig. 4, which represent the Error Cumulative Distribution (ECD) of the AFA-LK and FA-LK algorithms for different values of α and λ_{init} (the initial value of λ for the input image). The metric error used for computing the ECD is:

$$e_c = \|\mathbf{p} - \mathbf{p}_{ref}\|_2 \tag{16}$$

where $\|.\|_2$ represents the L^2 norm of the error vector ($\mathbf{p} - \mathbf{p}_{ref}$). This metric permits to evaluate the ability of each algorithm to converge under important displacements, which gives us insights about the basin of convergence of the AFA-LK and the FA-LK, the effect of the values of α and λ_{init}, and the advantage of considering the optimization of the parameters in the scale-space domain. The maximum number of iterations is fixed to 30 iterations and the value of the scale parameter of the target image $\lambda_{ref} = 0.5$. The value of λ_{ref} is set to a small value in order to keep enough details in the scale-space representation of the target image, and hence allows a high precision of the final estimate. $\lambda_{init} = 4$ for the tests on the value of α (Fig. 3), and $\alpha = 0.3$ for the tests on the value of λ_{init} (Fig. 4).

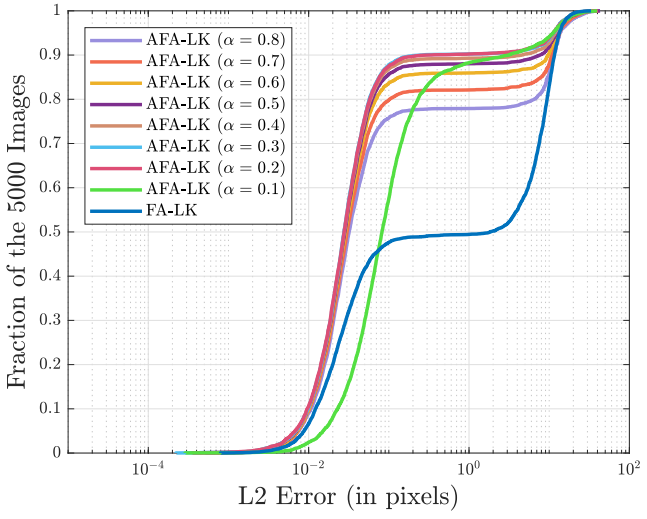
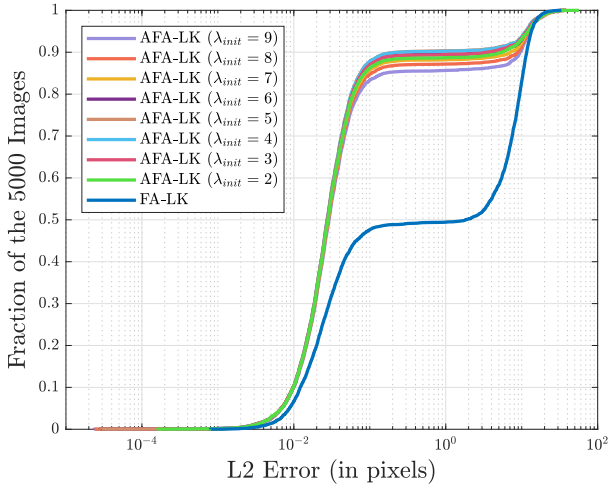
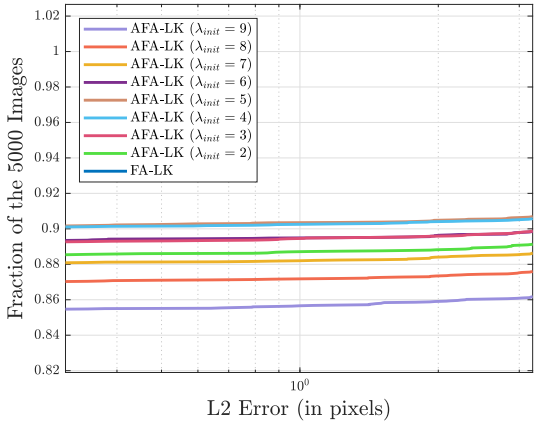


Fig. 3: ECD of the FA-LK and AFA-LK for different values of α.



(a)



(b)

Fig. 4: (a) ECD of the FA-LK and AFA-LK for different values of λ_{init}. (b) is a zoom of (a) around the error of 1 pixel.

The proposed algorithm permits to increase the convergence domain in comparison to the original Lucas-Kanade algorithm without loss in estimation precision. Concerning the effect of the value of α presented in Fig. 3, we can see that values inferior or equal to 0.6 allows to have a convergence rate superior to 85%, and the values of α equal to 0.2 and 0.3 provide the best results. In fact this parameter represents the step size in each iteration. When its value is too large, the algorithm tends to be unstable and diverge. Conversely, when the value of α is too small the speed of convergence becomes too slow. The value $\alpha = 0.3$ will be used in the following validations. According to Fig. 4, we can see that the AFA-LK performs similarly for the values of $\lambda_{init} \in [3; 6]$; because when the value of the scale parameter is too large, the algorithm needs a greater number of iterations to converge, and when the value of the latter is too small, there is not enough smoothing and the algorithm is unable to deal with challenging cases (large motions, local minima).

4.3. Homography Transformation

In this part, the various algorithms are tested and compared for the task of 2D homography estimation.

First, the comparison is done between the AFA-LK and geometric approaches (similarly to [11]): the variants of the LK algorithm (*forwards additive*, *inverse compositional*, *forwards compositional*) [3] and the ESM [5] as pixel-based methods, and SIFT+RANSAC Homography [16] (we used the implementation given in the vlfeat [2] library) as a feature-based method. We used the validation set of the MS-COCO dataset [18] for this comparison.
Second, the AFA-LK is compared to learning methods: the Conditional-LK [17], and the SDM (Supervised Descent Method) [27] in a set of images from the Yale face database [4] (sample images are shown in Fig. 9).

4.3.1. Comparison with Geometric Methods

A similar procedure to the one of [11] for the validation of the algorithms has been used. Every image of the validation set of the MS-COCO dataset was used to generate a template and an input image. The image was first down-sampled so that the shorter side was equal to 240 pixels. Then, a square was randomly cropped in the resized image and set to be the input image I (192x192 pixels). Next, in order to generate the template T, we manually selected a 128x128 square region centered in I and perturbed the four corners of the square using a uniform distribution within the range $[-42, +42]$ pixels (represented by the red quadrilateral in Fig. 5). After that, warp $w(\mathbf{x}; \mathbf{p}_{ref})$ was defined by the homography that maps the template corners to the perturbed ones, and T was generated by applying the reference warp to the input image $I(w(\mathbf{x}; \mathbf{p}_{ref}))$. The initial warp $w(\mathbf{x}; \mathbf{p}_{init})$ is the translation that maps the template domain to the image domain.

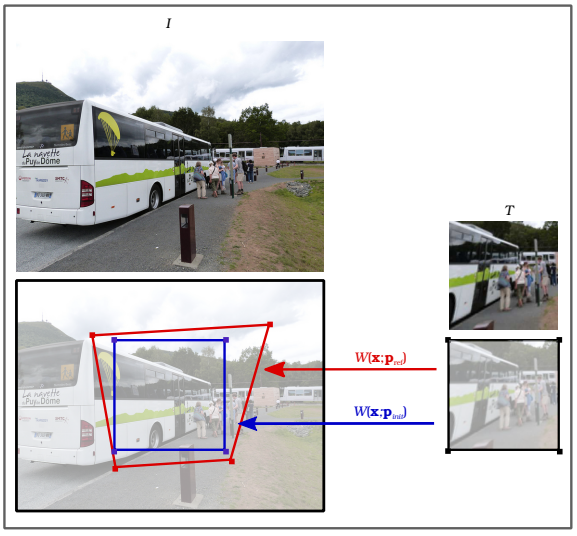


Fig. 5: Generation of the template image T from the input image I. $w(\mathbf{x}; \mathbf{p}_{init})$ represents the translation that maps the template domain (the dark square) to the image domain (blue square), $w(\mathbf{x}; \mathbf{p}_{ref})$ represents the homography that maps the template corners (in black) to the perturbed corners (in red).

In order to avoid unrealistic shape distortions resulting from the homography transformation, every angle of the quadrilateral was restricted to be less than $\frac{3}{4}\pi$. Fig. 6 shows examples of

[2] http://www.vlfeat.org/

the pairs $\{I, T\}$. A Gaussian noise of standard deviation 0.02 [3] was added to both the intensity values of $\{I$ and $T\}$ to make the dataset more challenging. Based on [[11]], we used the *corner error* as the metric for the tests, which is defined as follows:

$$e_c(\mathbf{p}, \mathbf{p}_{ref}) = \frac{1}{4}\sum_{j=1}^{4} \|w(\mathbf{e}_j; \mathbf{p}) - w(\mathbf{e}_j; \mathbf{p}_{ref})\|_2 \qquad (17)$$

where $\mathbf{e}_j$ represents the jth corner coordinates. Fig. 7 shows the *corner error* cumulative distribution of the following algorithms:

1. **AFA-LK:** The different parameters are set to the following values $\alpha = 0.3$, $\lambda_{ref} = 0.5$, and $\lambda_{init} = 12$ because we noticed in our experiments that a large value of the initial scale parameter is needed when the considered images are large (128x128 pixels) in comparison with the ones of the validations presented in Sec. 4.2 (29x29 pixels). The number of iterations is equal to 30.
2. **LK variants:** We considered the *forwards additive*, *inverse compositional*, and *forwards compositional* variants of the LK algorithms and used the code provided by [3]. The integration of a pyramid approach to the *forwards additive* LK (3 levels) is also presented. The number of iterations is fixed to 30.
3. **ESM:** We used the code provided by [22] and set the number of iterations to 30.

Fig. 6: Samples of the input images from the MS-COCO dataset and the corresponding templates used for testing the algorithms.

4. **SIFT + RANSAC:** A feature-based homography estimation algorithm implemented in the VLFeat library [26].

[3] corresponding to 2% of pixel intensities since they belong to $[0, 1]$ (see Sec. 4.1)

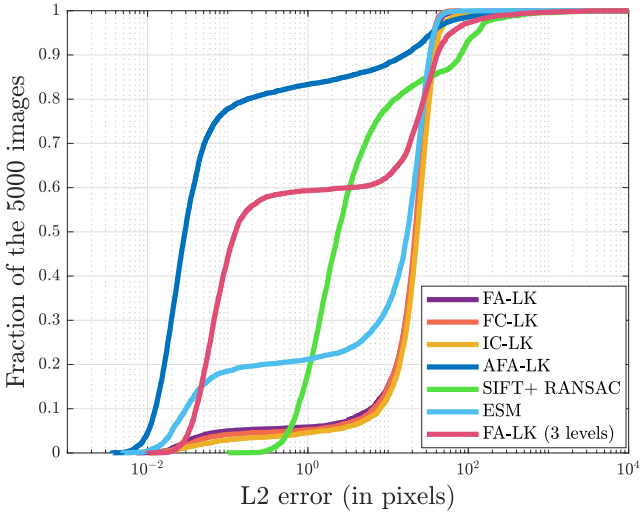


Fig. 7: *Corner error* cumulative distribution for the different algorithms.

Fig. 7 shows that among the various algorithms, the variants of the LK (FA, IC, FC) exhibit the lowest convergence rate because they are unable to cope with huge displacements (up to 42 pixels, as a remembering). The ESM, for its part, provides better results and we can notice its good accuracy when converging. The SIFT + RANSAC algorithm exhibits a comparable convergence domain but is less precise than the pixel-based methods. Because it is a feature-based algorithm, the SIFT + RANSAC fails to deal with low textured images (Fig. 8). Our method outperforms largely the other algorithms in terms of precision and convergence domain, even when comparing it to a pyramidal approach (3 levels pyramid), which appears unable to handle cases where the motion is too large (Fig. 8 shows an example where the algorithm increases the value of λ to remove local minima then converges). We can clearly see that the AFA-LK is able to deal with important displacements and because it is a pixel-based method, it is able to provide accurate results even with low textured images.

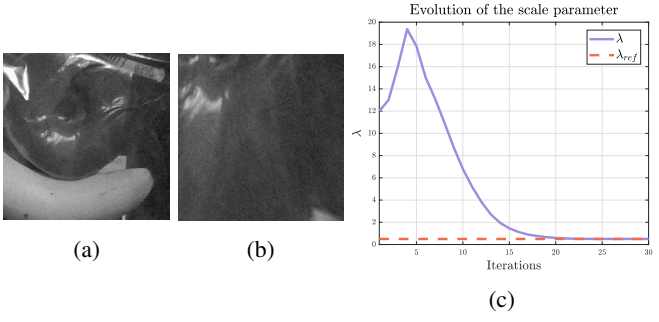


Fig. 8: (a, b) example where the SIFT+RANSAC algorithm was unable to converge ((a) represents I and (b) represents T). (c) example where the AFA-LK was the only algorithm to converge. The value of λ increases, permitting to suppress local minima then converges to λ_{ref}

4.3.2. Comparison with Learning Methods

In this last evaluation, we followed the methodology of [17] in order to compare our algorithm to their method. Every input image I is used to generate a template T in the following manner. A manually selected square region is first defined in the

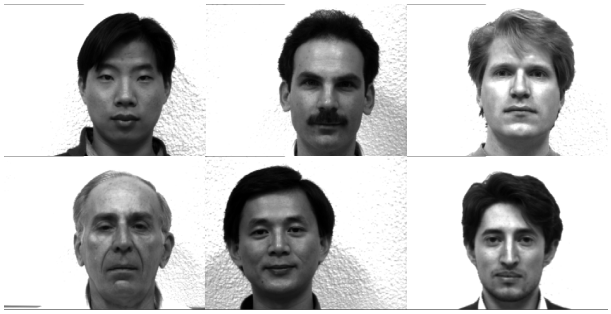


Fig. 9: Samples of the subjects of the Yale face database.

image. The four square corners are then individually perturbed using independent and identically distributed Gaussian noise of standard deviation σ in addition to a single translational noise of the same distribution applied to every corner. Finally, template T is generated from the perturbed corners similarly to Sec. 4.3.1. We followed the procedure described in [17] in order to train the different regressors of the Conditional-LK and the SDM.

We state that the convergence is achieved when the point RMS error, defined in Eq. (18), is less than 1 pixel, and plot the convergence rates of the considered algorithms for different values of σ (the convergence rates of each subject are computed from a total of 1000 tests for each value of σ), as shown in Fig. 10. We used a subset of the Yale face database (Fig. 9). The learning methods were trained using a value of $\sigma = 1.2$ pixel (represented by a vertical dashed line in the figure) in accordance to [17].

$$P_{RMSE} = \sqrt{\frac{1}{4}\sum_{j=1}^{4}\Big(w(e_j;\mathbf{p}) - w(e_j;\mathbf{p}_{ref})\Big)^2} \qquad (18)$$

We can see in Fig 10 that the AFA-LK shows superior convergence properties in comparison to both the Conditional-LK and the SDM. In addition to that, the proposed method has the advantage of requiring no training and consequently permits to considerably save time.

5. Conclusion

In this paper, we proposed the AFA-LK, a novel scale-space image alignment method based on the Lucas-Kanade tracker. It shows its effectiveness in the many evaluations conducted in this article. The *Adaptive Forwards Additive* Lucas-Kanade (AFA-LK) permits to increase the performances of the original FA-LK in terms of estimation precision and the basin of convergence. The presented comparisons show that the proposed algorithm outperforms state-of-the-art algorithms whether they are geometric-based or learning-based. This can be explained by the fact that integrating the scale parameter, directly affecting the smoothness of the cost function, to the optimization enables the method to automatically tune the value of the scale parameter according to its needs, resulting in an increase of the basin of convergence. When the algorithm tends to convergence, the value of the scale parameter decreases resulting in a fine estimation, which explains the precision of the final estimates.

References

[1] H. Alismail, B. Browning, and S. Lucey. Robust tracking in low light and sudden illumination changes. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 2016.

[2] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. P. Zafeiriou. Feature-Based Lucas–Kanade and Active Appearance Models. *IEEE Transactions on Image Processing*, 24(9):2617–2632, sep 2015.

[3] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[5] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 1, pages 943–948. IEEE, 2004.

[6] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[7] H. Bristow and S. Lucey. In Defense of Gradient-Based Alignment on Densely Sampled Sparse Features. In *Dense Image Correspondences for Computer Vision*. Springer, Cham, 2016.

[8] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. In *International Journal of Computer Vision*, 2007.

[9] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 2005.

[10] D. Capel. Image Mosaicing. In *Image Mosaicing and Super-resolution*. Springer, London, 2004.

[11] C.-H. Chang, C.-N. Chou, and E. Y. Chang. CLKN: Cascaded Lucas-Kanade Networks for Image Alignment. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3777–3785. IEEE, jul 2017.

[12] A. Crivellaro and V. Lepetit. Robust 3D Tracking with Descriptor Fields. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3414–3421. IEEE, jun 2014.

[13] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[14] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct monocular SLAM. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8690 LNCS, pages 834–849, 2014.

[15] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, May 2014.

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, second ed. edition, 2004.

[17] C.-H. Lin, R. Zhu, and S. Lucey. The Conditional Lucas & Kanade Algorithm. pages 793–808. 2016.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV*, pages 740–755. Springer International Publishing, 2014.

[19] T. Lindeberg. Scale selection. In K. Ikeuchi, editor, *Computer Vision: A Reference Guide*, pages 701–713. Springer US, Boston, MA, 2014.

[20] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004.

[21] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[22] C. Mei, S. Benhimane, E. Malis, and P. Rives. Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE Transactions on Robotics*, 24(6):1352–1364, Dec. 2008.

[23] H. Mobahi, C. L. Zitnick, and Y. Yi Ma. Seeing through the blur. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1736–1743. IEEE, jun 2012.

[24] S. Oron, A. Bar-Hille, and S. Avidan. Extended Lucas-Kanade Tracking. pages 142–156. Springer, Cham, 2014.

[25] N. Sharmin and R. Brad. Optimal Filter Estimation for Lucas-Kanade Optical Flow. *Sensors*, 12(9):12694–12709, sep 2012.

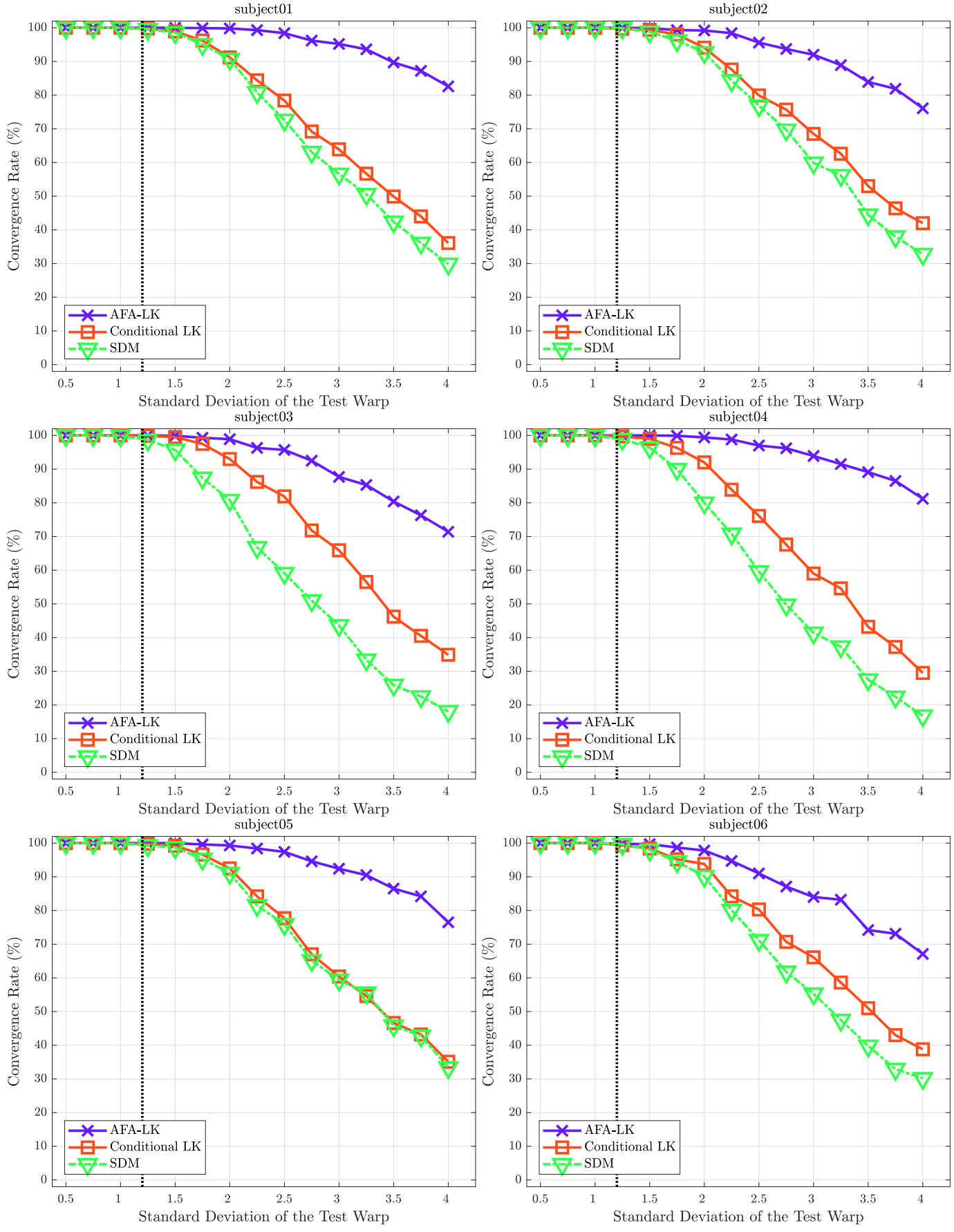


Fig. 10: Convergence rates for different subjects of the Yale database, computed over a total of 15000 image pairs for each subject.

[26] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[27] X. Xiong and F. D. la Torre. Supervised descent method for solving nonlinear least squares problems in computer vision. *CoRR*, abs/1405.0601, 2014.