Progressive keypoint localization and refinement in image matching

Fabio Bellavia[1][0000-0002-1688-8476], Luca Morelli[2,4][0000-0001-7180-2279], Carlo Colombo[3][0000-0001-9234-537X], and Fabio Remondino[2][0000-0001-6097-5342]

[1] University of Palermo, Palermo, Italy fabio.bellavia@unipa.it
[2] Bruno Kessler Foundation (FBK), Trento, Italy {lmorelli,remondino}@fbk.eu
[3] University of Florence, Florence, Italy carlo.colombo@unifi.it
[4] University of Trento, Italy

Abstract. Image matching is the core of many computer vision applications for cultural heritage. The standard image matching pipeline detects keypoints at the beginning and freezes them until bundle adjustment, by which keypoints are allowed to move in order to improve the overall scene estimation. Recent deep image matching approaches do not follow this scheme, historically imposed by computational limits, and progressively refine the localization of the matches in a coarse-to-fine manner.
This paper investigates the use of traditional computer vision approaches based on template matching to update the keypoint position throughout the whole matching pipeline. In order to improve the accuracy of the template matching, the usage of the coarse-to-fine refinement is explored and a novel normalization strategy for the local keypoint patches is designed. Specifically, the proposed patch normalization assumes a local piece-wise planar approximation of the scene and warps the corresponding patches according to a "middle homography", so that, after normalization, patch distortion is roughly equally distributed within the two original patches. The experimental comparison of the considered approaches, mainly focused on cultural heritage scenes but straightforwardly generalizable to other common scenarios, shows the strengths and limitations of each evaluated method. This analysis indicates promising and interesting results of the investigated approaches, which can effectively be deployed to design better image matching solutions.

Keywords: Image matching · Keypoint refinement · Cross correlation · Middle homography · Patch normalization · Pixel-Perfect SfM · Cultural Heritage.

1 Introduction

1.1 Image Matching Perspectives

Image matching plays a key role in computer vision [26] and photogrammetric applications designed for cultural heritage and archaeology [11]. Among these, Structure-from-Motion (SfM) is generally devised as a downstream task of image matching and its advancements are significantly contributing to document and digitally preserve archaeological artifacts and 3D art works [8]. In order to obtain high-quality digital models replicating the geometry and texture of the original objects with accurate details, image matching needs to register images so that the localization precision of corresponding matches is the highest possible [16].

1.2 Common Ground of Deep and Non-deep Image Matching

Thanks to the ever increasing availability of both data and computational resources, the rise of deep learning has led to impressive advancements in computer vision and its sub-fields, including image matching. State-of-the-Art (SotA) deep image matching includes sparse methods such as SuperGlue [24], Accurate and Lightweight Keypoint Detection and Descriptor Extraction (ALIKE) [28] and Accurate Shape and Localization Features (ASLFeat) [20], or semi-dense methods such as Local Feature Transformer (LoFTR) [25] and the more recent Dense Kernelized Feature Matching (DKM) [7]. The mentioned approaches are end-to-end architectures, whose main advantage with respect to pipelines composed by standalone, separate modules is to allow a global optimization and synchronization of the process. Nonetheless, current end-to-end image matching methods are the final results of the efforts made by the research community on each individual part of the matching pipeline, which can be summarized in terms of multiple reiterations of these steps: keypoint detection, patch normalization, feature description extraction and matching.

Keypoint detection Traditional keypoint detectors combines image derivatives to define corners and Difference-of-Gaussian (DoG) blobs, extracted by the popular handcrafted Harris [13] and Scale Invariant Feature Transform (SIFT) [19] detectors, respectively. Filters designed according to the above functions of the image derivatives are applied to the images and the peaks in the filter response maps obtained by Non-Maximum Suppression (NMS) provides the final keypoints. The Keypoint Network (Key.Net) [2] was the first to introduce the softmax operator, that enables differentiable NMS on the filter response maps, obtained from learned convolutional layers but also by explicitly including first and second order derivatives of the input image. Moreover, differentiable NMS is used to achieve sub-pixel precision in ALIKE [28] or analogously to refine the matches established by correlation by LoFTR [25]. The basic idea for the sub-pixel keypoint estimation is to interpolate the discrete response map around the local neighborhood of the peak so as to obtain the true maximum. Classic approaches use parabolic interpolations [27] or approximate the response map by its derivative as in the case of SIFT [19]. Deep sub-pixel estimation acts instead as Gaussian process regression interpolation, explicitly employed in DKM [7].

Patch normalization Patch normalization warps the local neighborhood of the keypoints so that patches become roughly aligned in order to compare them. The main assumption in the non-deep approaches is that any general spatial or radiometric transformation can be locally approximated by a simpler one with less degrees of freedom. Normalization by the mean and standard deviation of the intensity values of the patch is generally sufficient to achieve good radiometric invariance [19]. Robust spatial patch normalization is instead more complex to obtain. In the case of SIFT, normalization assumes to work with patches related only by a similarity (scale and orientation) transformation [19], and experiences decreasing performances in the presence of more severe perspective distortions. Local affine normalization [21] better tolerates these scene configurations. SotA affine patch normalization is achieved by the deep Affine Network (AffNet) [23] so that similar patches are clustered together in the transformed space of the normalized patches. This is not achieved explicitly according to some patch characteristic, e.g. edge shapes, but implicitly using hard negative mining triplet loss introduced in the Hard Network (HardNet) [22] descriptor. ASLFeat extends deep patch normalization on dense maps by employing Deformable Convolutional Networks (DCNs) [15] which basically act in two steps: in the first one DCNs look locally inside the patch, then according to the gathered information decide the shape of the convolution filter to use. While a better shape adaptability is guarantee, this is still dependent from the local data. This dependency has been surpassed by transformers, successfully employed by SuperGlue and LoFTR, which extract relations between distant image areas.

Feature description extraction and matching Keypoint description extracts features able to compare the keypoint local patches. Ideally, in the case of perfectly registered patches and in absence of noise, the cross correlation of the normalized image patches would be the optimal choice. For real scenarios, robust handcrafted feature descriptors are generally based on histograms of the orientations of the image gradient, as for SIFT [19], candidate matches are established to Nearest Neighbor (NN) strategies [3], and final matches are obtained by robust correspondence filtering based on spatial constraints through RANdom SAmple Consensus (RANSAC) [9]. Spatial constraints include strong ones such as planarity and stereo epipolar geometry [14], or loose constraints such as the spatial neighborhood consistency used in Adaptive Locally-Affine Matching (AdaLAM) [6] and Delaunay Triangulation Matching (DTM) [3]. Since the original aim of deep architectures is to extract features, feature descriptors were the first components of the image matching pipeline to be successfully implemented by deep networks. HardNet is a deep SotA standalone feature descriptor which extracts features by processing the patch through successive convolutional layers. Conversely, effective keypoint matching was accomplished by deep learning only later. The first architecture to succeed was SuperGlue, which employs the Sinkhorn algorithm behaving as a differentiable NN matching and graph neural networks (of which the transformer can be thought as a later and lightweight version) to infer and apply spatial constraints to the matches. Match similarity is measured by the correlation in the feature space of the corresponding patches.

Image matching pipeline evolution A last, essential characteristic that has contributed to the success of end-to-end deep image architectures is to be sought in the deep structure of the networks composed by a sequence of stacked layers. Even without explicitly designing the network to have a coarse-to-fine architecture as for LoFTR [25] and DKM [7], the deep structure allows to progressively and successively refine the matching process. On the one hand, this can be associated to multiple successive passes of a base matching pipeline. On the other hand, patch normalization and the effective keypoint matching can be thought of the same image matching process at micro and macro levels, respectively:

inside a patch, point-like features are extracted and matched according to spatial constraints to get their correspondences and to decide if the patches match; inside the whole image, patch-like features are extracted and matched according to spatial constraints to get their correspondences and to decide if the images match. Pixel-Perfect SfM [18] is a deep architecture which extends this idea to the whole SfM pipeline, as it takes keypoint tracks computed by image matching on multiple image pairs and refines them both before SfM and after on the basis of the SfM output together with the 3D coordinates of the keypoints.

1.3 Paper Contribution

The aim of this paper is to investigate how to improve match localization accuracy according to the aforementioned design concepts, yet avoiding the use of deep architectures. The idea is to let every step of the matching pipeline to be explicitly described in an algorithmic way. Such analysis of the process can contribute to implement optimized handcrafted matching pipelines and to understand better and improve deep matching architectures.

The main idea is to re-process image matches already extracted by the matching pipeline. For this objective, template matching approaches [10] which require a robust initial solution can be used since after the first pass raw matches have been roughly detected. Moreover, raw matches define a planar piece-wise approximation of the scene in the local neighborhood of the match, which is more general and adheres better to the actual warping than an affine transformation. Patch normalization is updated according to the knowledge from the previous pass in order to improve the template matching. Worth to note that upgrading patch local transformation was already shown to be effective at improving the estimated scene structure [1]. Sub-pixel registration of the patch is also considered in order to refine the matches. Finally, multiple passes of the match localization refinement are considered too.

The rest of the paper is organized as follows. The different base modules employed for the match refinement are described in detail in Section 2, while the experimental analysis is presented and discussed in Section 3. Conclusions and future work are provided in Section 4.

2 Match Refinement Base Modules

2.1 Normalized Cross Correlation (NCC) Matching

Given two images $I_1, I_2 : \mathbb{R}^2 \to \mathbb{R}$ and a coarse match $(\mathbf{x}, \mathbf{x}')$, where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$ are the corresponding keypoint coordinates in the two images, NCC for the patches centered on the given keypoints with radius r is [10]

$$\mathcal{C}^r_{\mathbf{x},\mathbf{x}'} = \sum_{\|\varDelta\|_\infty \le r} \frac{(I_1(\mathbf{x}+\varDelta) - \mu_{I_{1\{\mathbf{x},r\}}})(I_2(\mathbf{x}'+\varDelta) - \mu_{I_{2\{\mathbf{x}',r\}}})}{\sigma_{I_{1\{\mathbf{x},r\}}}\sigma_{I_{2\{\mathbf{x}',r\}}}} \tag{1}$$

with μ_S, σ_S indicating the mean and standard deviation over the set S, respectively, and

$$I_{i\{\mathbf{w},r\}} = \{I_i(\mathbf{w}+\varDelta) : \parallel \varDelta \parallel_\infty \le r\} \tag{2}$$

representing the patch centered in $\mathbf{w}$ as a set. Assuming I_1 as reference, the keypoint on I_2 is updated by cross correlation as $\mathbf{x}' + \varDelta^\star$, where the discrete offset $\varDelta^\star$ maximizes the correlation between the two patches, i.e.

$$\varDelta^\star = \underset{\|\varDelta'\|_\infty \leq r}{\operatorname{argmax}} \, \mathcal{C}^r_{\mathbf{x},\mathbf{x}'+\varDelta'} \tag{3}$$

Notice that NCC is invariant to local affine illumination changes, as the intensity values are normalized by the mean and standard deviation over the local patch windows. Moreover, NNC provides a response map by which to refine the keypoint by sub-pixel interpolation.

2.2 Adaptive Least Square (ALS) Correlation Matching

ALS correlation [12] performs an iterative affine registration between the two patches. The aim of the method is to estimate an affine patch warping $\mathrm{A} \in \mathbb{R}^{2\times3}$ and an affine transformation of the intensity values $\mathrm{L} \in \mathbb{R}^{1\times2}$ to register the two patches. Defining $\tilde{\mathbf{z}} = [\mathbf{z}\ 1]^\mathrm{T}$ as the normalized homogenous vector associated to $\mathbf{z}$, the registration error is given by

$$\mathcal{E}^{\theta,r}_{\mathbf{x},\mathbf{x}'} = \sum_{\|\varDelta\|_\infty \leq r} f_k(\theta) = \sum_{\|\varDelta\|_\infty \leq r} \| \, I_1(\mathbf{x}+\varDelta) - \mathrm{L}\tilde{I_2}(\mathrm{A}\tilde{\mathbf{x}}' + \varDelta) \, \|^2 \tag{4}$$

where $\theta = \{\mathrm{A}, \mathrm{L}\}$ indicates the transformation parameters and $\varDelta = [i\ j]^\mathrm{T}$ such that $k = (i+r) + (2r+1)(j+r)$ is an univocal linear index for each pixel of the patch. Assuming I_1 as reference, the keypoint on I_2 is updated as $\mathrm{A}^\star\tilde{\mathbf{x}}'$, where $\mathrm{A}^\star$ minimized the patch error, i.e.

$$\mathrm{A}^\star = \underset{\mathrm{A}\in\mathbb{R}^{2\times3}}{\operatorname{argmin}} \, \mathcal{E}^{\theta,r}_{\mathbf{x},\mathbf{x}'} \tag{5}$$

The best parameter set $\theta^\star = \{\mathrm{A}^\star, \mathrm{L}^\star\}$ is found by non-linear least square minimization, which is basically the gradient descent employed in deep learning. The initial configuration $\theta' = \{\mathrm{A}' = [\mathbf{I}\ \mathbf{0}], \mathrm{L}' = [1\ 0]\}$ assumes that the original patches are almost registered, and the errors $f_k(\theta)$ are approximated linearly by Taylor expansions as

$$f_k(\theta) = f_k(\theta' + \varDelta_\theta) = f_k(\theta') + \frac{\partial f_k}{\partial \theta}\varDelta_\theta \tag{6}$$

The minimal error solution is then equivalent to

$$\mathrm{F}(\theta') + \mathrm{J}_\theta \varDelta_\theta = \mathbf{0} \tag{7}$$

where $\mathrm{F}(\theta') = \left[f_1(\theta') \ \cdots \ f_{(2r+1)^2}\right]^\mathrm{T}$ and $\mathrm{J}_\theta = \left[\frac{\partial f_1}{\partial \theta} \ \cdots \ \frac{\partial f_{(2r+1)^2}}{\partial \theta}\right]^\mathrm{T}$ is the Jacobian matrix obtained from the discrete derivatives so that

$$\varDelta_\theta = -\mathrm{J}^+_\theta \mathrm{F}(\theta') \tag{8}$$

and the parameters of the transformation get updated iteratively until convergence as $\theta' + \alpha\varDelta_\theta$, where $\alpha = 0.5$ is introduced to prevent that the solution diverges due to the forced linearization of the error function.

2.3 Fast Affine Template Matching (FAsT-Match)

FAsT-Match [17] patch error formulation is analogous to ALS correlation but the Sum-of-Absolute-Differences (SAD)

$$\mathcal{A}^{\theta,r}_{\mathbf{x},\mathbf{x}'} = \sum_{\|\varDelta\|_\infty \leq r} |I_1(\mathbf{x}+\varDelta) - \mathrm{L}\tilde{I_2}(\mathrm{A}\tilde{\mathbf{x}}'+\varDelta)| \tag{9}$$

is used instead of the Euclidean distance. Again, assuming I_1 as reference, the keypoint on I_2 is updated by $\mathrm{A}^\star\tilde{\mathbf{x}}'$, where $\mathrm{A}^\star$ minimizes the error between the two patches, i.e.

$$\mathrm{A}^\star = \operatorname*{argmin}_{\mathrm{A}\in\mathbb{R}^{2\times 3}} \mathcal{A}^{\theta,r}_{\mathbf{x},\mathbf{x}'} \tag{10}$$

Unlike ALS correlation, which assumes continuous functions, FAsT-Match exploits discretization by partitioning the space of the allowable transformations and employing a branch-and-bound strategy to efficiently explore the solution spaces and find the best transformation. The vertical sub-pixel offset derivation is similar. FasT-Match is computationally intensive so that in the evaluation to bound the running times default parameters were set to $\epsilon = 0.5$ and $\delta = 0.75$ respectively, and the allowable scale factor to 3 and the orientation range were limited by $\pm\frac{\pi}{3}$. These settings improve the running times with no accuracy loss.

2.4 Parabolic sub-pixel peak interpolation

Parabolic interpolation refines NNC response map $D(u,v) = \mathcal{C}^r_{\mathbf{x},\mathbf{x}'+\varDelta'}$, where $\varDelta' = [u\ v]^\mathrm{T}$, as follows. Assume that $\mathbf{x}'$ has been updated as described in Section 2.1, so that the patch is centered in the peak, i.e. $\varDelta^\star = \mathbf{0}$. Then, the keypoint sub-pixel offset is computed as

$$\varDelta p = \begin{bmatrix} -\dfrac{b}{2a} & -\dfrac{b'}{2a'} \end{bmatrix}^\mathrm{T} \tag{11}$$

The horizontal sub-pixel offset corresponds to the vertex x-coordinate of the parabola $ax^2 + bx + c = y$, interpolated from the 3 points

$$P_d = (d, \mathcal{C}^r_{\mathbf{x},\mathbf{x}'+[0\ d]^\mathrm{T}}) = (d, y_d), \qquad d \in \{-1, 0, 1\} \tag{12}$$

along the horizontal dimension of D, which leads to $a = \dfrac{y_1 - 2y_0 + y_{-1}}{2}$ and $b = \dfrac{y_1 - y_{-1}}{2}$. The vertical sub-pixel offset is computed analogously.

2.5 Taylor approximation sub-pixel peak interpolation

This adapts the SIFT detector sub-pixel precision method [19]. In this case, the second order Taylor expansion of the response map gives around the peak $\varDelta^\star$

$$D(\varDelta') = D(\varDelta^\star + \varDelta_l) = D(\varDelta^\star) + \frac{\partial D^\mathrm{T}}{\partial \varDelta'}\varDelta_l + \frac{1}{2}\varDelta_l^\mathrm{T}\mathrm{H}_{\varDelta'}\varDelta_l \tag{13}$$

where $\mathrm{H}_{\Delta'}$ is the Hessian matrix of D, computed by discrete derivatives. The maximum is achieved when the derivative of $D(\Delta')$ is zero, i.e. when

$$\frac{\partial D^{\mathrm{T}}}{\partial \Delta'} + \mathrm{H}_{\Delta'} \Delta_l = 0 \tag{14}$$

which implies that the requested sub-pixel correction offset is

$$\Delta_l = -\mathrm{H}_{\Delta'}^{-1} \frac{\partial D^{\mathrm{T}}}{\partial \Delta'} \tag{15}$$

Actually, in the original SIFT paper, the offset space is 3D, since the DoG filter operates also on scales.

2.6 Middle Homography (MiHo) Patch Normalization Updating

Patch normalization updating assumes that a set of matches $M = \{(\mathbf{x}, \mathbf{x}')\}$ has been obtained after the first matching pipeline pass. It also assumes that matches for the subset $P \subseteq M$ are related by a planar homography $\tilde{\mathbf{x}}' = \mathrm{H}^{\star}\tilde{\mathbf{x}}$, where $\mathrm{H}^{\star} \in \mathbb{R}^{3\times 3}$ is non-singular, using the same conventions of [14]. The idea of MiHo is to find an associated pair of planar homographies $(\mathrm{H}, \mathrm{H}')$ so that

$$\tilde{\mathbf{m}} = \mathrm{H}\tilde{\mathbf{x}} \quad \wedge \quad \tilde{\mathbf{m}} = \mathrm{H}'\tilde{\mathbf{x}}', \qquad \forall (\mathbf{x}, \mathbf{x}') \in P \tag{16}$$

where $\mathbf{m} = \frac{\mathbf{x}+\mathbf{x}'}{2}$. As shown in Fig. 1a, this heuristic procedure inspired by [4] tends to distribute equally the distortion error over the two patches when these are normalized by H and H′, respectively. Since interpolation degrades with up-sampling, MiHo aims to provide a balance with down-sampling the patch at finer resolution and up-sampling the patch at the coarser resolution. Also, a planar homography is a better local approximation than an affine transformation.

The Direct Linear Transform (DLT) [14] is used to find H and H′. Actually, MiHo pairs estimation can be repeated on the warped keypoint pairs $(\mathrm{H}\tilde{\mathbf{x}}, \mathrm{H}'\tilde{\mathbf{x}}')$ to refine the solution, by concatenating all the successive homographies. It was experimentally observed that three iterations generally suffice.

Defining an inlier match $(\mathbf{x}, \mathbf{x}')$ for a generic H according to the threshold r by the maximum reprojection error

$$\mathcal{P}^{\mathrm{H},r}_{\mathbf{x},\mathbf{x}'} = \begin{cases} 1 \text{ if } \max(\|\tilde{\mathbf{x}}' - \mathrm{H}\tilde{\mathbf{x}}\|, \|\tilde{\mathbf{x}} - \mathrm{H}^{-1}\tilde{\mathbf{x}}'\|) \leq r \\ 0 \quad \text{otherwise} \end{cases} \tag{17}$$

an inlier for the MiHo pair $(\mathrm{H}, \mathrm{H}')$ is straightforwardly defined as the product

$$\mathcal{P}^{(\mathrm{H},\mathrm{H}'),r}_{\mathbf{x},\mathbf{x}'} = \mathcal{P}^{\mathrm{H},r}_{\mathbf{x},\mathbf{m}} \, \mathcal{P}^{\mathrm{H}',r}_{\mathbf{x}',\mathbf{m}} \tag{18}$$

In order to discover simultaneously both the approximated planes on the scene and their associated MiHo pairs, the whole process is embedded into the RANSAC framework. Starting from $M_0 = M$, RANSAC is used at iteration i to extract the i-th best MiHo pair $(\mathrm{H}_i, \mathrm{H}'_i)$ using threshold r, and strong inliers are removed for the next iteration according to a stricter threshold $\frac{r}{2}$, i.e.

$$M_{i+1} = M_i \setminus \left\{ (\mathbf{x}, \mathbf{x}') : \;\; (\mathbf{x}, \mathbf{x}') \in M_i \;\wedge \mathcal{P}^{(\mathrm{H}_i,\mathrm{H}'_i),\frac{r}{2}}_{\mathbf{x},\mathbf{x}'} \right\} \tag{19}$$

until $M_i = \emptyset$ or last MiHo has only 4 inliers, i.e. the minimum model size. Finally, since more than one MiHo pair can satisfy a match, the MiHo pair $(\mathrm{H}_{\mathbf{x},\mathbf{x}'}, \mathrm{H}'_{\mathbf{x},\mathbf{x}'})$ assigned to a match $(\mathbf{x}, \mathbf{x}')$ is the one with the larger consensus set

$$(\mathrm{H}_{\mathbf{x},\mathbf{x}'}, \mathrm{H}'_{\mathbf{x},\mathbf{x}'}) = \underset{\mathcal{P}_{\mathbf{x},\mathbf{x}'}^{(\mathrm{H}_i,\mathrm{H}'_i),r}=1}{\operatorname{argmax}} \left(\sum_{(\mathbf{x}_j,\mathbf{x}'_j)\in M} \mathcal{P}_{\mathbf{x}_j,\mathbf{x}'_j}^{(\mathrm{H}_i,\mathrm{H}'_j),r} \right) \tag{20}$$

In case no MiHo pair is compatible with a match, the identity matrix will be used for the corresponding patch normalization.

Figure 1b shows an example of rough planes associated to a same MiHo pair. Notice that the input images should not be roughly rotated by 180° in order for MiHo to work. This can be understood considering the case when the global transformation within the images is close to a reflection through a point. In this case, the mid-points $\mathbf{m}$ corresponding to a match $(\mathbf{x}, \mathbf{x}')$ tend to accumulate about the center of reflection, thus providing a configuration close to degeneracy.

3 Evaluation

As shown in Fig. 1c, the evaluation dataset considers 12 image pairs representing scenes of interest for cultural heritage on which 20 matches $(\mathbf{x}, \mathbf{x}')$ have been manually selected by expert users as ground-truth (GT). The images have a resolution of 20 MegaPixel (MP), and the keypoint accuracy for the selected matches at the original resolution is up to 1 px. By down-scaling the images with a factor of 5, images maintain a feasible testing resolution and matches get a sub-pixel accuracy. Bilinear interpolation [10] is used to warp patches for its efficiency. Code and data are freely available[1].

The patch radius is set to $r = 15$ px for NCC, ALS correlation and FAsT-Match. GT keypoints $\mathbf{x}$ on I_1 are used as reference, while keypoints $\mathbf{x}'$ on I_2 are perturbed by adding a noise offset of $n = 1, \ldots, 11$ px in one or both directions at testing resolution. Specifically, for a given noise offset n, 4 noisy matches $(\mathbf{x}, \mathbf{y}'_n)$ are obtained where

$$\mathbf{y}'_n \in \begin{cases} \mathbf{x}' + \left\{ n \left[\begin{smallmatrix} \pm 1 \\ 0 \end{smallmatrix} \right], n \left[\begin{smallmatrix} 0 \\ \pm 1 \end{smallmatrix} \right] \right\} \text{ if } n \text{ is odd} \\ \mathbf{x}' + \left\{ n \left[\begin{smallmatrix} \pm 1 \\ \pm 1 \end{smallmatrix} \right], n \left[\begin{smallmatrix} \pm 1 \\ \mp 1 \end{smallmatrix} \right] \right\} \text{ if } n \text{ is even} \end{cases} \tag{21}$$

for a total of $20 \times 11 \times 4 = 880$ tested keypoint matches for each image.

In order to evaluate MiHo patch normalization update, initial matches were estimated using SotA matching pipelines to which noisy matches were added. To make RANSAC plane discovery unrelated from the GT matches, matches within $2r$ of GT matches were removed before including the noisy matches, see Fig. 1b. The employed pipelines are Hz^{+} [5] and Key.Net+AffNet+ HardNet+AdaLAM [2], both with and without upright constraints.

[1] https://drive.google.com/drive/folders/12jPMbU4doWoDRv57unBctjyxXUKhDHRF

[2] https://kornia.github.io/

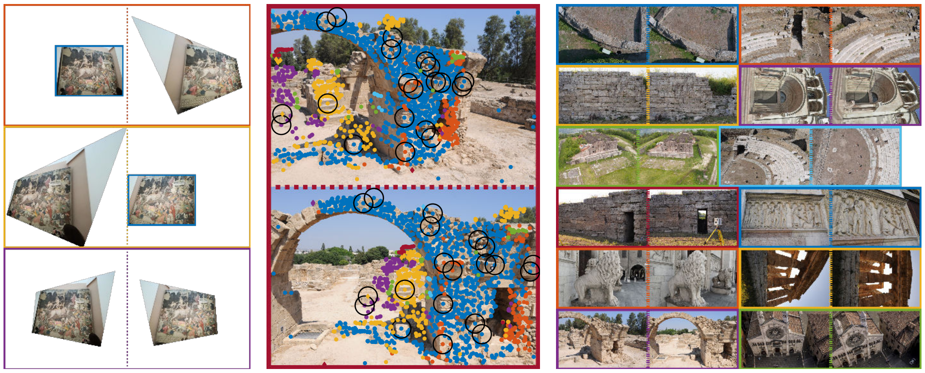
(a) (b) (c)

Fig. 1: (a) Differences between common homography warping and MiHo. I_1 and I_2 original and warped images are respectively on the left and right sides. In the top and middle rows the original image (framed in blue) is used as reference to apply the standard warping to the other image (sided), MiHo warped pairs are shown in the bottow row. (b) Corresponding clusters of raw planes associated to a same MiHo pair for an image pair. Matches are extracted by Hz$^+$, GT matches are highlighted by black circles. (c) Image pairs of the evaluation dataset (best viewed in color, for high resolution images see the additional material[1]).

Table 1 shows the average keypoint shift error on the whole dataset for different noise offsets, ordered by their magnitude. Lighter bars indicate the error percentage with respect to the noise offset magnitude when less than 100%, darker bars when greater than 100%. NCC is used with no sub-pixel refinement, ⊞ indicates the base run with $r = 15$ px and ⊡⊞ a two-step coarse-to-fine run. Specifically, in the latter case in the first step the keypoint is coarsely refined at half testing resolution with $r = 7$ px, and in the next step the updated keypoint is refined again at full resolution with $r = 15$ px. MiHo initial matches have been estimated with Hz$^+$. Detailed results are reported in the additional material[1].

ALS correlation increases the accuracy only when the noise offset magnitude is limited, due to the fact that the image approximation by its derivatives is valid only in a small local neighborhood. NCC and FAst-Match absolute improvements generally do not depend on the noise. For NCC the absolute error is about 4, 2 px respectively without and with MiHo, for FAsT-Match this is 3 px. MiHo patch update remarkably helps NCC, roughly halving the error. On FAsT-Match MiHo improvements are lower, since the method itself uses affine adaptation. Nevertheless, the MiHo solution does better, which implies that planar homography approximation is better than the affine one. When ALS correlation decreases the error, MiHo normalization makes ALS behaves as FAsT-Match, since both approaches perform an affine warping. Coarse-to-fine two-step solutions ⊡⊞ degrade the localization with respect to the base approaches ⊡, besides doubling the running times. Concerning running times, code was implemented in Matlab with no optimizations and was run on a Intel Core I9 10900K. The refinement

Table 1: Average keypoint shift error (px) on the whole dataset.

Noise offset mag. (px)			1	$2\sqrt{2}$	3	5	$4\sqrt{2}$	7	$6\sqrt{2}$	9	11	$8\sqrt{2}$	$10\sqrt{2}$	avg.
ALS	no patch norm.	⊞	1.25	1.76	1.82	4.03	3.35	7.45	5.61	10.84	7.87	14.29	10.33	6.24
		⊡⊞	1.87	2.27	2.23	4.15	3.44	7.45	5.53	10.85	7.74	14.17	10.25	6.36
	MiHo	⊞	0.92	1.28	1.39	3.52	2.89	7.28	5.38	10.59	7.78	14.04	10.26	5.94
		⊡⊞	1.38	1.75	1.71	3.71	2.93	7.14	5.27	10.74	7.68	14.12	10.16	6.06
NCC	no patch norm.	⊞	3.62	3.64	3.64	3.57	3.66	3.70	3.69	3.76	3.78	3.81	3.78	3.70
		⊡⊞	4.04	4.00	4.06	4.08	4.09	4.25	4.22	4.37	4.28	4.57	4.38	4.21
	MiHo	⊞	1.75	1.92	1.79	2.06	2.00	2.14	2.07	2.20	2.19	2.44	2.25	2.07
		⊡⊞	1.93	2.09	1.99	2.23	2.12	2.40	2.30	2.40	2.40	2.72	2.50	2.28
FAsT-Match	no patch norm.	⊞	2.01	2.18	2.10	2.40	2.35	2.72	2.66	3.06	2.76	5.76	3.08	2.82
		⊡⊞	2.59	2.66	2.65	2.77	2.80	3.06	2.99	3.40	2.96	5.54	3.58	3.18
	MiHo	⊞	1.93	1.89	1.88	2.18	2.07	2.36	2.23	2.72	2.38	4.87	2.70	2.47
		⊡⊞	2.18	2.21	2.23	2.45	2.29	2.66	2.48	2.87	2.54	4.76	2.79	2.68

of a single match takes 0.05 s for both ALS correlation and NCC, while it is close to 2 s for FAsT-Match. MiHo code is excluded from the current analysis. Clearly, multiple matches can be refined in parallel and code optimization could speed the computation.

According to this comparison, NCC with MiHo provides the best solution. Table 2 adds further experiments with NNC, by including parabolic and Taylor sup-pixel estimation. Moreover, a further two-step approach (indicated by ⊞⊠) is evaluated where the NCC solution is further refined by ALS correlation since the latter can better cope with small noise shifts. According to these results, both parabolic interpolation and ALS refinement provide modest incremental improvements, while Taylor sub-pixel offset generally degrades the base solution. Notice also that ALS refinement doubles the running times.

On average, no methods achieve a sub-pixel refinement, i.e. an error less than 1 px. Nevertheless, as reported by further analyses in the additional material[1], the situation is more articulated. Specifically, considering the sub-pixel accuracy in terms of percentages of keypoints with error less than 1 px after the refinement, ALS correlation achieves for noise offset magnitude less than 2 px values about 55%, 70% without and with MiHo, respectively. FAsT-Match percentage is stable around 40% in any case. For the base NCC, sub-pixel accuracy percentage is the same of FAsT-Match but increases to about 55%, 59% and 60% as MiHo, parabolic fitting and ALS correlation are incrementally included, respectively. The results are in accordance with the previous observations, but also show that it is possible to achieve sub-pixel accuracy with the investigated approaches.

4 Conclusions and Future Works

This paper has presented a thorough comparative analysis of non-deep, conventional approaches to improve the localization accuracy of keypoint matching, focusing in particular on cultural heritage and archaeological scenes. The results suggest that patch normalization is crucial for improving the match localization and that simple NCC paired with parabolic fitting, and optionally ALS correlation, can provide promising results. Future works will focus on further analyses,

Table 2: Average keypoint shift error (px) of NCC sub-pixel on the whole dataset.

Noise offset mag. (px)			1	$2\sqrt{2}$	3	5	$4\sqrt{2}$	7	$6\sqrt{2}$	9	11	$8\sqrt{2}$	$10\sqrt{2}$	avg.
none	no patch norm.	⊞	3.62	3.64	3.64	3.57	3.66	3.70	3.69	3.76	3.78	3.81	3.78	3.70
		⊞⊠	3.51	3.51	3.50	3.42	3.50	3.55	3.54	3.61	3.63	3.65	3.64	3.55
	MiHo	⊞	1.75	1.92	1.79	2.06	2.00	2.14	2.07	2.20	2.19	2.44	2.25	2.07
		⊞⊠	1.73	1.85	1.75	1.98	1.93	2.07	1.98	2.13	2.10	2.36	2.17	2.00
parabolic	no patch norm.	⊞	3.69	3.78	3.76	3.66	3.77	3.92	3.90	3.98	4.02	4.44	4.35	3.93
		⊞⊠	3.52	3.59	3.56	3.45	3.60	3.71	3.70	3.77	3.84	4.24	4.19	3.74
	MiHo	⊞	1.67	1.78	1.72	1.94	1.88	2.00	1.96	2.22	2.10	2.66	2.37	2.03
		⊞⊠	1.62	1.75	1.69	1.90	1.85	1.98	1.94	2.17	2.07	2.63	2.36	2.00
Taylor	no patch norm.	⊞	3.81	4.00	4.09	4.02	4.39	4.07	4.39	4.15	4.49	4.13	4.24	4.16
		⊞⊠	3.64	3.75	3.82	3.73	4.03	3.78	4.05	3.86	4.06	3.84	3.89	3.86
	MiHo	⊞	1.99	2.32	2.28	2.60	2.90	2.48	3.07	2.61	3.20	2.82	3.04	2.66
		⊞⊠	1.90	2.17	2.10	2.42	2.55	2.32	2.73	2.40	2.90	2.62	2.70	2.44

incorporating the evaluated modules in practical applications, even between the pipeline steps. Moreover, extension to multi-view patches will be explored and comparisons with deep solutions will be carried out. MiHo results are also quite interesting and will be further investigated, also in the context of its applications to planar matching and benchmarking.

References

1. Barath, D.: On making SIFT features affine covariant. Int. J. Comput. Vis. (2023)
2. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key.Net: Keypoint detection by handcrafted and learned CNN filters. In: Proceedings of the International Conference on Computer Vision (ICCV) (2019)
3. Bellavia, F.: SIFT matching by context exposed. IEEE Trans. Pattern Anal. Mach. Intell. **45**(2), 2445–2457 (2023)
4. Bellavia, F., Colombo, C.: Estimating the best reference homography for planar mosaics from videos. In: Proceedings International Conference on Computer Vision Theory and Applications (VISAPP). pp. 512–519 (2015)
5. Bellavia, F., Mishkin, D.: HarrisZ$^+$: Harris corner selection for next-gen image matching pipelines. Pattern Recognit. Lett. **158**, 141–147 (2022)
6. Cavalli, L., Larsson, V., Oswald, M.R., Sattler, T., Pollefeys, M.: AdaLAM: Revisiting handcrafted outlier detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
7. Edstedt, J., Athanasiadis, I., Wadenbäck, M., Felsberg, M.: DKM: Dense kernelized feature matching for geometry estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
8. Farella, E.M., Morelli, L., Grilli, E., Rigon, S., Remondino, F.: Handling critical aspects in massive photogrammetric digitalization of museum assets. Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. **XLVI-2/W1-2022**, 215–222 (2022)
9. Fischler, M., Bolles, R.: Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)

10. Gonzales, R., Woods, R.E.: Digital Image Processing. Pearson College Division, 4th edn. (2017)
11. Gruen, A., Remondino, F., Zhang, L.: Photogrammetric reconstruction of the Great Buddha of Bamiyan, Afghanistan. Photogramm. Rec. **19**(107), 177–199 (2004)
12. Gruen, A.W.: Adaptive least squares correlation: a powerful image matching technique. South Afr. J. Photogram. Remote Sens. and Cartogr. **14**(3), 175–187 (1985)
13. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference. pp. 147–151 (1988)
14. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, 2st edn. (2000)
15. J. Dai, H.Q., Xiong, Y., Li, Y., Zhang, G., Wei, H.H.Y.: Deformable convolutional networks. In: Proceedings of the International Conference on Computer Vision (ICCV) (2017)
16. Karami, A., Menna, F., Remondino, F.: Combining photogrammetry and photometric stereo to achieve precise and complete 3D reconstruction. Sensors **22**(21), 8172 (2022)
17. Korman, S., Reichman, D., Tsur, G., Avidan, S.: Fast-Match: Fast affine template matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1940–1947 (2013)
18. Lindenberger, P., Sarlin, P., Larsson, V., Pollefeys, M.: Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
20. Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: ASLFeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
21. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
22. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS) (2017)
23. Mishkin, D., Radenovic, F., Matas, J.: Repeatability is not enough: Learning affine regions via discriminability. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
24. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
25. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
26. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer-Verlag, 2nd edn. (2022)
27. Trucco, E., Verri, A.: Introductory Techniques for 3-D Computer Vision. Prentice Hall (1998)
28. Zhao, X., Wu, X., Miao, J., Chen, W., Chen, P.C.Y., Li, Z.: ALIKE: Accurate and lightweight keypoint detection and descriptor extraction. IEEE Trans. Multimed. **early access** (2022)