Matchability Prediction for Full-Search Template Matching Algorithms

Adrian Penate-Sanchez[1], Lorenzo Porzi[2,3], and Francesc Moreno-Noguer[1]

[1]Institut de Robòtica i Informàtica Industrial, UPC-CSIC, Barcelona, Spain
[2]Fondazione Bruno Kessler, Trento, Italy
[3]University of Perugia, Italy

Abstract

While recent approaches have shown that it is possible to do template matching by exhaustively scanning the parameter space, the resulting algorithms are still quite demanding. In this paper we alleviate the computational load of these algorithms by proposing an efficient approach for predicting the matchability of a template, before it is actually performed. This avoids large amounts of unnecessary computations. We learn the $matchability$ *of templates by using dense convolutional neural network descriptors that do not require ad-hoc criteria to characterize a template. By using deep learning descriptions of patches we are able to predict* $matchability$ *over the whole image quite reliably. We will also show how no specific training data is required to solve problems like panorama stitching in which you usually require data from the scene in question. Due to the highly parallelizable nature of this tasks we offer an efficient technique with a negligible computational cost at test time.*

1. Introduction

Template-based matching has been an important topic in Computer Vision for nearly as long as the field has existed. Recently it has been applied successfully to dense 3D reconstruction [8], image-based rendering [7], image matching [32] and even used with RGBD data for object detection [12].

Template Matching algorithms can consider all possible transformations [18, 31] of the template or just a discrete subset of them [11, 21]. By sampling a subset of all candidate templates we are able to obtain algorithms that can operate at speeds of nearly 30 fps [11]. On the other hand, template matching algorithms that perform a full search on all possible transformations of the template guarantee to find the global maximum of the distance function, yielding more accurate results at the price of increasing the computational cost.

Several approaches have focused on improving specific characteristics of the template matching framework, such as its speed [22, 24], its robustness to partial occlusions [4] or ambiguities [26] and its ability to select reliable templates for rapid visual tracking [2]. Yet, most of these improvements have been designed for the algorithms that do not perform a full search. In this work we will bring these improvements to algorithms that perform full search, and in particular we will consider FAst-Match [15], a recent method that performs near full search at reasonable speed.

Algorithms like FAst-Match, though, may still suffer from false positives. Furthermore, they are only able to estimate affine transformations: this is enough to approximate small projective deformations, but produces bigger errors in the general case (as seen on the third experiment of [15]). We aim to provide a template selection approach to partially overcome these problems. We propose to learn a *matchable template detector* by modeling the probability of a template to be correctly matched. In particular, we combine dense CNN features, due to the promising results obtained on a variety of computer vision tasks [30, 16], and a logistic regression objective. This allows for an extremely efficient GPU implementation of the proposed method, which makes its computational demands negligible when compared to the matching itself.

As finding a close approximation of the correct transformation is not difficult when performing full search, our main aim will be to reduce the overlap error, as well as decreasing the computational effort by preemptively selecting good templates. We use the overlap error as it seems the community agrees on it [15, 19, 20]; this is discussed extensively in [19, 20].

In a similar manner as [10] did for feature points we

Adrian Penate-Sanchez and Lorenzo Porzi have contributed equally to this work.

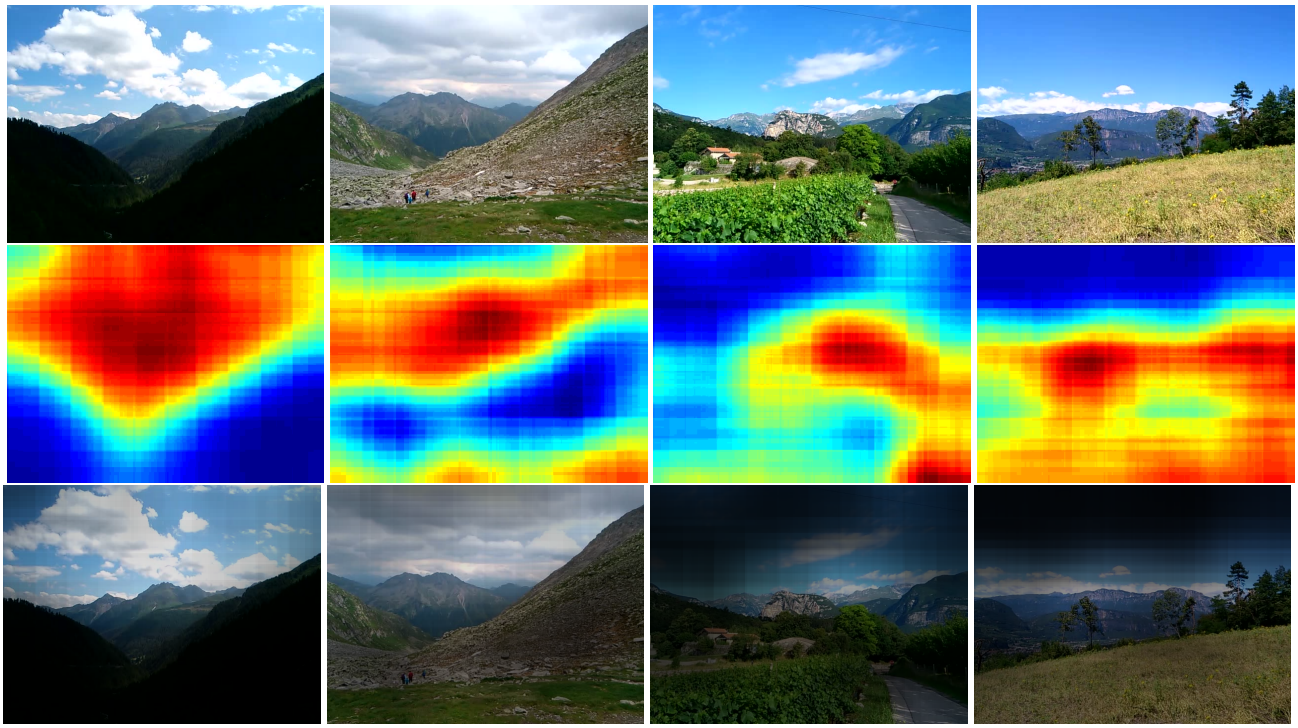

Figure 1. First row: first image from sequences 1,2,4 and 5. Second row: heat maps representing the response of our matchable template detector on the images in the first row (blue: low probability, red: high probability). Third row: original images filtered using the heat maps to highlight the regions most likely to be selected.

will identify which are the areas of a scene that give good $matchability$. The main difference when using templates is that, instead of focusing on a set of points, one has to define the $matchability$ of all the image. An example of how our approach is capable of obtaining the interesting parts from all the image can be seen in Fig. 1. In this work we will show that template selection is able to noticeably improve matching accuracy at the expense of an almost negligible additional computational cost. Furthermore, we will show the proposed approach to have good generalization capabilities.

2. Related Work

As mentioned above, we may roughly classify template matching algorithms into those that perform full search or near full search of the transformation parameters [18, 31], and those that just locally or discretely refine the parameters. The latter allow for fast optimization but do not guarantee a global maximum [11].

In between, there exist a huge body of work that attempts to balance both. In [1] the relation between appearance distance and spatial overlap is exploited to give an upper bound on appearance distance given the spatial overlap of two windows in an image. By doing this, it is then possible to provide a computationally efficient solution to the template matching problem. In order to handle general transformations, in [14] a grayscale template matching algorithm that considered variations of scale and rotation was presented. More recently FAst-Match [15] was proposed as an efficient way to handle general affine transformations; although it was not the first successful approach to try so [9], FAst-Match remains to this day the approach that shows best results in literature. FAst-Match [15] lies between discrete template matching and full search approaches. In particular it considers all the affine transformation space but, using a tree search approach, avoids searching high error subspaces.

2.1. Feature Selection

Feature selection approaches can enhance both speed and robustness of matching techniques. In [13] feature selection is done based on the upper bound of the average error, prioritizing templates that are more robust to small errors in the transformation; this is appropriate to avoid testing huge amounts of candidate templates. In [2], the reliability of a template is defined through a number of characteristics, namely the uniformity of texture, contrast and spatial locality among others. These characteristics are then used to define a scoring scheme using Support Vector Regression [25] that will, at runtime, score the different candi-

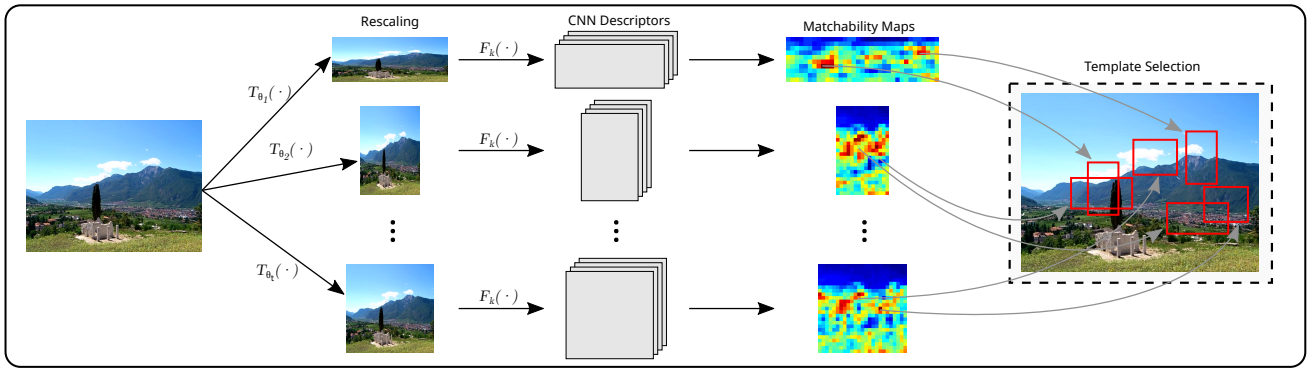

Figure 2. Schematic representation of our matchable template detection method. In a first step we apply a set of transformations $T_{\theta_1}, \ldots, T_{\theta_t}$ to the input image. Then we extract dense CNN features for each of the transformed images and use them to compute the matchability maps. Finally we select the matchable templates as the set of image patches corresponding to local maxima in the matchability maps.

dates. [4] presents a method that selects template pixels that verify the approximation of the tracking algorithm. In [34] method that selects an set of templates by learning a quality measure and by optimizing coverage. The commented work is mainly applied to tracking and builds on the assumption that a full search is not performed on the test image, it just needs the template to be stable to local transformations and does not have to assume free transformations through the whole space.

Another relevant approach [24] enhances the performance and the speed of [11] and [12] by learning what parts of the template are worth analyzing and boosts the best parts of each template only testing those, thus in the process reducing greatly the computational cost. This work focus on texture-less objects and uses edges as features making it specific for such objects. It was also expanded to handle the RGB-D inputs of [12] in a similar manner.

We have seen how learning algorithms have been previously applied to speedup and increase reliability of template matching when using discrete search template matching algorithms [24, 2, 13]. But when dealing with full search [18, 31] or near full search algorithms [15] you are guaranteed to obtain the global maximum of your distance measure so the same criteria does not apply. We will focus our work more on learning how to improve the precision of the yielded matching rather than on the recall that a template can provide.

3. Method

Given a *template* image I_1 and a *target* image I_2, template matching methods aim to estimate a transformation that matches I_1 to I_2 in such a way to minimize some error function $E(I_1, I_2)$. In particular, FAst-Match considers affine plane transformations that minimize the sum of absolute differences (SAD) between the target image and the template. We can express this as an optimization problem:

$$\arg\min_{\mathsf{A},\mathbf{t}} \sum_{\mathbf{p}\in I_1} \|I_1(\mathbf{p}) - I_2(\mathsf{A}\mathbf{p}+\mathbf{t})\|, \qquad (1)$$

where $\mathsf{A} \in \mathbb{R}^{2\times 2}$ and $\mathbf{t} \in \mathbb{R}^2$ are the transformation's parameters, while the vector $\mathbf{p}$ denotes pixel coordinates on the images. FAst-Match describes a branch-and-bound scheme to find a good solution of (1) over the 6-DOF space of affine transformations in a computationally efficient way.

In general, the template I_1 can be a rectangular region of a larger image I, which we call the *source* image. In this case, selecting a good template from I can be crucial for the effectiveness of the template matching algorithm. Consider *e.g.* a natural landscape image: intuitively, a template depicting a large portion of uniformly colored sky could be especially hard to match. Similarly to the well-known approach adopted for feature point descriptors and detectors such as SIFT [17] or SURF [3], we propose a *matchable template detector* which is able to select good candidate templates from an input image. We train our detector with the objective of learning a function that assigns to each candidate template a probability of it being correctly matched.

In this work we only consider FAst-Match, but the method we propose is completely general and can be used with any template matching algorithm.

3.1. Learning a template matchability predictor

Given a template matching algorithm and a set of training source and target images, we sample random templates from the source images and match them to one of the target images using the matching algorithm. Using this procedure we collect a set of observations $(x, y) \in \mathcal{S}$, where $x = (I, \mathbf{p}, w, h)$ denotes the $w \times h$ pixels patch of image $I \in \mathcal{I}$ centered at $\mathbf{p}$, *i.e.* the template, and $y \in \{-1, +1\}$ is a label that indicates whether the template has been correctly matched in a target image ($y = +1$) or not ($y = -1$).

Given a template x, we model the posterior probability of it being matched as

$$\mathrm{P}(y|x,\mathbf{w}) = \frac{e^{y\mathbf{w}^\top \phi(x)}}{1+e^{y\mathbf{w}^\top \phi(x)}}, \quad (2)$$

where $\mathbf{w} \in \mathbb{R}^{\mathsf{m}}$ is a parameter vector to be learned and $\phi(\cdot)$ is a feature function that will be defined in Sec. 3.2. We learn $\mathbf{w}$ by means of a regularized maximum log-likelihood estimation, which yields the well-known L2-regularized logistic regression problem:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{(x,y)\in\mathcal{S}} \log(1+e^{-y\mathbf{w}^\top\phi(x)}), \quad (3)$$

where C is a non-negative regularization parameter. We solve (3) using the trust region Newton method implemented in LIBLINEAR [6].

3.2. CNN Features

A n-layers convolutional neural network can be expressed as a function $F : \mathcal{I} \to \mathbb{R}^{\mathsf{d}_1\times\cdots\times\mathsf{d}_s}$ mapping images to s-dimensional tensors. F can be factorized as the composition of n functions $F = f_{\mathsf{n}} \circ \cdots \circ f_1$, corresponding to the network's layers. We denote by $F_k = f_k \circ \cdots \circ f_1$, $k \leq \mathsf{n}$ the function corresponding to the first k layers of the network. Many common CNN architectures (*e.g.* used in image recognition) can be subdivided in a convolutional and a fully connected part. That is, every F_k for $k \leq \mathsf{k}'$ (the convolutional part) produces a 3-dimensional tensor output, while every F_k for $k > \mathsf{k}'$ (the fully connected part) produces a vector output. In these cases, the functions $F_k(I)$ for $k \leq \mathsf{k}'$ can be understood as dense feature maps over the input image I.

In the setting described above, $F_k : \mathcal{I} \to \mathbb{R}^{\mathsf{r}\times\mathsf{c}\times\mathsf{m}}$ associates a feature vector in $\mathbb{R}^{\mathsf{m}}$ to each of $\mathsf{r}\times\mathsf{c}$ local regions of the input image. These regions are given by a sliding window, which is commonly referred to as the *receptive field* of the convolutional neurons of layer k. The size (w_s, h_s) and stride of the window depend on the hyperparameters of the network's bottom k layers. Given these definitions, we define the feature function $\phi(x)$ of (3) as

$$\phi(x) = \begin{bmatrix} \mathsf{Z}_{i',j',1} \\ \vdots \\ \mathsf{Z}_{i',j',\mathsf{m}} \end{bmatrix}, \ \mathsf{Z} = F_k(T_\theta(I)), \quad (4)$$

where we use subscript notation $\mathsf{A}_{a,b,\ldots}$ to denote the element of the tensor A indexed by $(a,b,\ldots)$. $T_\theta(\cdot)$ is an asymmetric image scaling, parameterized by a tuple $\theta = (\sigma_1, \sigma_2)$, such that

$$J = T_\theta(I) \to J(\mathbf{p}) = I(\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}\mathbf{p}). \quad (5)$$

The scaling parameters θ are chosen to exactly match the size of the sliding window to that of the input template x in the transformed image, that is: $\sigma_1 = w_s/w, \sigma_2 = h_s/h$. Similarly, the indices (i', j') are chosen so that the overlap between the corresponding sliding window and the template is maximized.

In practice, instead of taking any template and looking for a transformation $T_\theta(\cdot)$ and window coordinates i', j' that meet the criteria we just described, we follow the reverse approach. Given a network architecture, we select a finite set of transformations and consider only templates that exactly overlap a sliding window in one of the transformed images. This allows a considerable advantage in terms of computational complexity: a single application of the network function F_k to a transformed image $T_\theta(I)$ immediately yields the value of $\phi(\cdot)$ for a large set of templates.

In this paper we consider a pre-trained CNN. In particular, we adopt the AlexNet [16] model, which is trained on an object recognition task with millions of images and 1000 object categories. A common choice is that of using the output of the first fully connected layer of AlexNet as a generic feature representation for an image. Instead, motivated by the recent work of Yosinski *et al.* [33] on CNN feature transferability and by the performance considerations detailed in the previous paragraph, we take the output of the third convolutional layer as our feature representation. This results in a feature vector of length $\mathsf{m} = 384$ and a sliding window with dimensions $w_s = h_s = 99$ pixels.

3.3. Detecting matchable templates

Ideally, given the learned matching probability model in (2) and an input image, we want to apply the model to every template in the set $\mathcal{T}$ of all possible rectangular subregions of I. Since the number of these templates is very high, we chose instead to follow the approach sketched in the last paragraph of Sec. 3.2. We fix a set of scaling transformations $T_{\theta_1}, \ldots, T_{\theta_\mathsf{t}}$ and apply them to the input image I, obtaining the transformed images $J_1 = T_{\theta_1}(I), \ldots, J_\mathsf{t} = T_{\theta_\mathsf{t}}(I)$. Applying $F_k(\cdot)$ to the transformed images yields a set of feature maps $\mathsf{Z}^1 = F_k(J_1), \ldots, \mathsf{Z}^\mathsf{t} = F_k(J_\mathsf{t})$. Each vector $\mathbf{z}^t_{i,j} = [\mathsf{Z}^t_{i,j,1} \ldots \mathsf{Z}^t_{i,j,\mathsf{m}}]^\top$ computed from the feature maps then corresponds to the value of the feature function $\phi(x^t_{i,j})$ for a particular template $x^t_{i,j} \in \mathcal{T}^t \subset \mathcal{T}$. Thus, the templates in each set $\mathcal{T}^t$ have fixed size and are sampled from a rectangular grid over I. The probability of $x^t_{i,j}$ being matched correctly becomes:

$$p^t_{i,j} = \mathrm{P}(y^t_{i,j} = 1|x^t_{i,j}, \mathbf{w}) = \frac{e^{\mathbf{w}^\top\mathbf{z}^t_{i,j}}}{1+e^{\mathbf{w}^\top\mathbf{z}^t_{i,j}}}. \quad (6)$$

By fixing t and varying i and j the probabilities $p^t_{i,j}$ form t "matchability maps" over the sets $\mathcal{T}^t$. As a last step in our algorithm, we apply a simple non-maximal suppression algorithm to these matchability maps to locate their local

Figure 3. Examples of images in each of the 12 outdoor scenes in the Venturi Dataset. We can see that the variability is quite big due to clouds, lighting conditions, etc, we can also see that there are many zones that are clear to be of no interest for matching (shadows, clear skies, etc) making this a good benchmark in which to test our approach.

maxima. The templates of each $\mathcal{T}^t$ corresponding to the local maxima in $p^t_{i,j}$ are finally returned as the detected matchable templates. The proposed method's pipeline is schematized in Fig. 2.

It is interesting to note that the calculation of (6) can be seen as an additional convolutional layer with a $1 \times 1 \times \mathsf{m}$ kernel and sigmoid non-linearity appended to the CNN used for feature computation. Indeed, our whole learning process could be easily cast in a traditional CNN setting, where both the parameters of F_k and $\mathbf{w}$ are learned by optimizing (3) with stochastic gradient descent and back-propagation.

4. Experimental Validation

We evaluate the performance of the proposed matchable templates detector when used in conjunction with the FAst-Match algorithm on a challenging dataset. We adopt a methodology that is similar to that used in [15], where templates from a fixed source image are matched against a set of target images. We also show how our algorithm is able to generalize among several different outdoor scenes.

4.1. Dataset

Korman *et.al.* [15] present results on three datasets: Pascal VOC 2010 [5], the Mikolajczyk [19, 20] dataset and the Zurich building dataset [29]. When evaluated by considering as correct every match showing less than 20% overlap error, the FAst-Match algorithm obtains near perfect results on the first two datasets, suggesting that using a good template selection procedure might not be necessary on that data. Conversely, only a qualitative evaluation of the performance of FAst-Match is presented in [15] for the third dataset, as it does not include any ground truth data. Consequently, we opt to evaluate our template selection algorithm on a different dataset which proves to be challenging for the FAst-Match approach and at the same time provides ground truth. In particular, we chose the Venturi Mountain dataset, previously employed in [23].

The Venturi Mountain dataset consists of 12 outdoors video sequences recorded from several locations in the Alps using a Sony Ericsson XPERIA Arc S smartphone. Each sequence contains between 150 and 500 frames with a resolution of 640×480 pixels, for a total of 3117 images. Camera calibration parameters and absolute orientation are also given for every image, making it trivial to derive a ground-truth homography between any pair of frames. Given that the images are extracted from video sequences, adjacent pairs are in general quite similar to each other. For this reason, we only consider one in fifty frames for evaluation and training. The sequences in the dataset cover several different illumination conditions, weather conditions and landscape characteristics, as well as different kinds of rotation-only camera movements. Fig. 3 shows some samples extracted from the dataset, which is publicly available online[1].

4.2. Experimental Setup

As previously mentioned, in our evaluation we only consider one in fifty frames for each video sequence. For each sequence we fix the first frame as the source image and use all other images from the same sequence in turn as target. Then, using one of the methods detailed below, we select a set of templates from the source and match them to the targets using FAst-Match. As a measure of the template matching error we use the "intersection over union" ratio of the matched template in the target image and its ground truth. Through the rest of this section we will refer to this measure as the "overlap error". As done in [15], only templates that are completely within the boundary of the target image are considered for evaluation.

Each time our approach is evaluated on a new sequence (the *test* sequence), the template matchability detector is trained using data from all sequences except the one under exam (the *training* sequences). In particular, we use the procedure described in the previous paragraph on all training sequences to collect a set of templates with their corresponding overlap errors. For each image pair 128 templates are chosen at random from the source image. Finally, all templates with an overlap error lower than the 25th percentile are labeled as good matches ($y = +1$)

[1] https://venturi.fbk.eu/results/public-datasets/mountain-dataset/

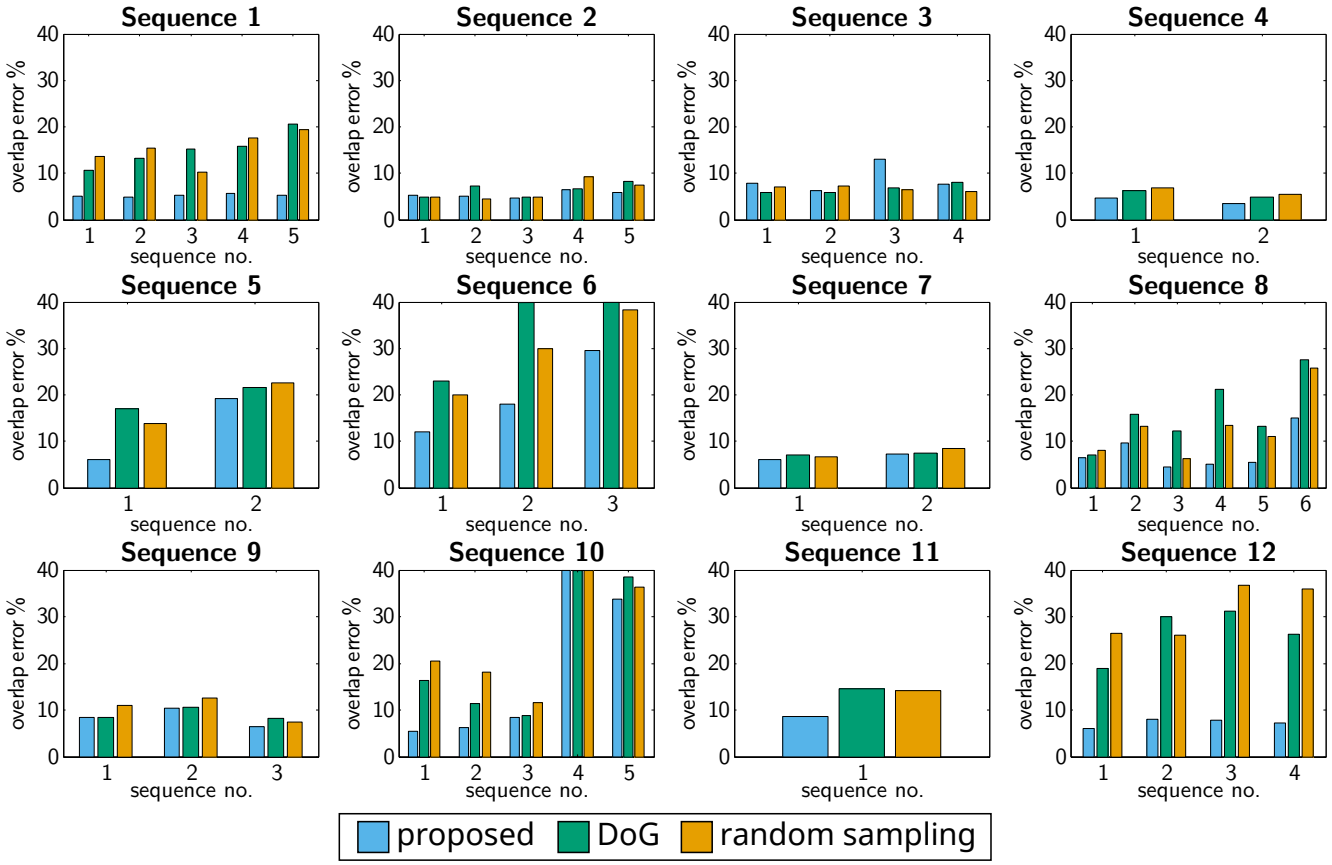

Figure 4. Average overlap errors on 50 patches with random sampling (orange), DoG (green) and our method (blue).

and all templates with an overlap error greater than the 75th percentile are labeled as bad matches ($y = -1$). For our matchable template detector we use a total of 16 scaling transformations (*c.f.r.* Sec. 3.3), given by parameters $\theta_t \in \{1, 0.75, 0.5, 0.25\} \times \{1, 0.75, 0.5, 0.25\}$.

In the evaluation phase, our matchable template detector is applied to the source images, selecting fifty templates corresponding to the local maxima in the matchability maps with the highest predicted matching probability. We refer to this approach as PROPOSED. We compare our method with two additional approaches: RANDOM SAMPLING (or RS) and DoG. In RS we randomly select fifty among all the candidate templates considered by PROPOSED. In DoG we select as templates the image regions that would be used by the SIFT [17] algorithm to compute keypoint descriptors: for each image the fifty regions with highest DoG cornerness score are selected.

It must be noted that, similarly to the experiment performed by Korman *et.al.* on the Zurich bulding dataset, the affine transformation estimated by FAst-Match is only a local approximation of the true homography that relates the images in the dataset. Thus, in general, we expect to measure some amount of unavoidable overlap error in all cases.

	Mean %			Std %		
seq.	**Ours**	**DoG**	**RS**	**Ours**	**DoG**	**RS**
1	**5.3**	15.0	15.2	**2.4**	23.5	21.0
2	**5.5**	6.3	6.2	**2.5**	9.8	8.4
3	8.7	**6.6**	6.7	11.7	8.0	**7.6**
4	**4.0**	5.5	6.2	**2.1**	5.1	5.7
5	**11.3**	18.5	18.2	**18.0**	26.2	24.2
6	**21.6**	35.0	46.4	**27.2**	35.5	39.0
7	**6.7**	7.2	7.5	**2.2**	3.3	5.8
8	**7.7**	16.3	13.0	**11.0**	26.1	20.7
9	**8.4**	8.9	10.3	**3.1**	3.2	11.7
10	**19.5**	25.5	27.0	**17.7**	22.7	25.5
11	**8.6**	14.6	14.1	**3.5**	22.0	16.0
12	**7.1**	24.4	31.3	**2.6**	31.5	34.5

Table 1. Mean and standard deviation of the overlap error for all image pairs tested in each scene. We improve results for all scenes except for scene 3. We can observe that template selection improves not only the mean error but also the standard deviation.

4.3. Results

In Fig. 4 we show the results obtained over the whole Venturi mountain dataset. Due to the difference in length of

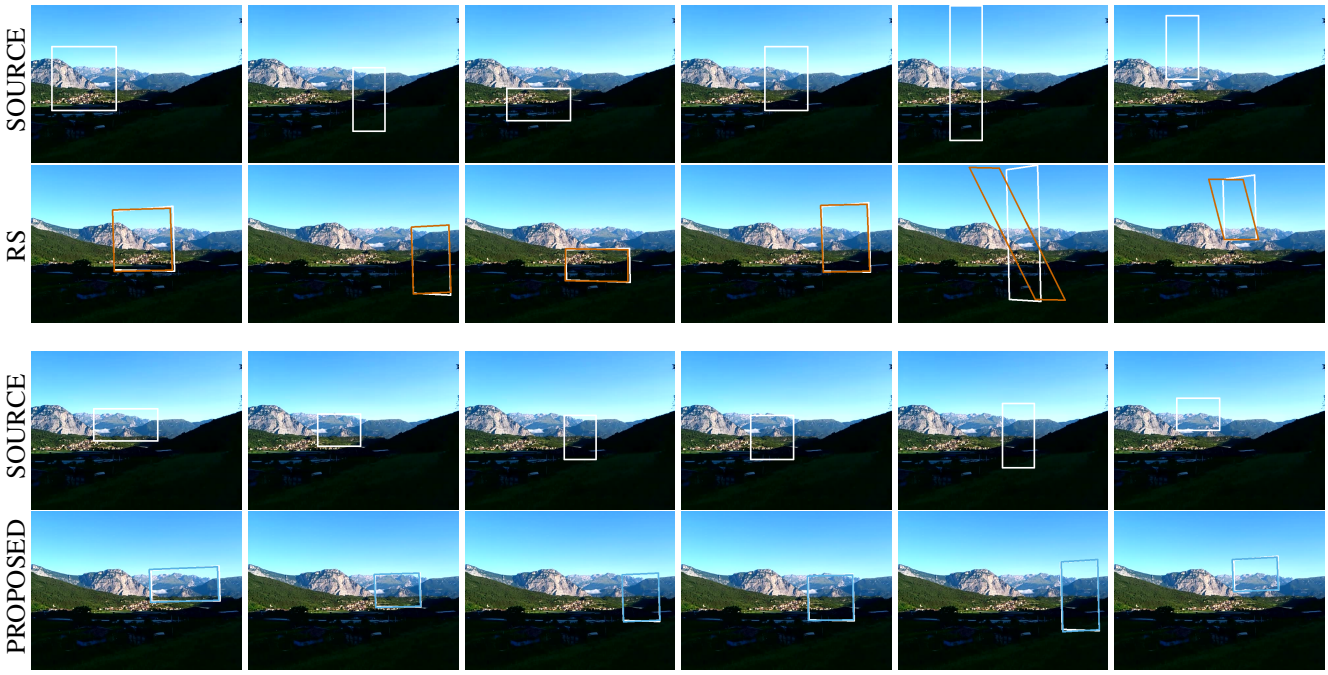

Figure 5. Template selection and matching results for a pair of images out of sequence 10. *First and third row*: templates selected from the source image, respectively using RANDOM SAMPLING and the PROPOSED approach; *Second row*: ground truth template position in the target image (white), matched template position obtained with RANDOM SAMPLING (orange); *Fourth row*: ground truth template position in the target image (white), matched template position obtained with the PROPOSED approach (blue).

the videos each scene contains a different number of test image pairs. We tested on 42 image pairs with 50 templates per pair, totaling 2100 FAst-Match evaluations. We can see that in sequences in which the error is already quite small (*e.g.* sequences $2, 3, 4, 7$), the impact of using template selection is minor. Conversely, in scenes in which FAst-Match does not perform well on itself (*e.g.* sequences $1, 5, 6, 8, 10, 12$), template selection generally provides a much better match, in some cases making the difference between finding a valid match or not at all. Scene 12 is a good example of this last case: with an average overlap error greater than 20%, most RS and DoG matches would have been considered wrong by the criterion proposed in [15]. It is also interesting to comment the results in scene 10. We observe that template selection helps a lot in the first 3 image pairs. When the projective transformation becomes just too big, however, the approximation provided by the affine transform becomes to coarse and all approaches show bad results.

Table. 1 shows the overall results for each scene, reporting the error's standard deviation together with the mean. It can be observed that performing matchable template selection also improves on the variability of the error. It is also interesting to note that DoG's performance is very close to RS, suggesting that the DoG cornerness criterion that works well for selecting keypoints might not be suitable for selecting templates.

To give an intuition of what our matchable template detector considers a "good template", Fig. 1 shows some heatmaps obtained by taking the average of the 16 matchability maps computed over 4 images in the dataset. The third row of the figure shows clearly how dark areas and clear skies are not considered when selecting the template. Similarly, noisy and repetitive patterns, such as the grass in the third scene, are in general given smaller weights. This agrees with the intuition that a matchable template should be distinctive for the image under consideration.

We also present some qualitative comparison between RANDOM SAMPLING and PROPOSED in Fig. 5 and Fig. 6. Fig. 5 in particular shows six template matching results uniformly sampled from those considered in our experiments on sequence 10. One of the typical failure cases for RS is illustrated in the last two images of the first and second row: templates containing large portions of sky are in general quite difficult to correctly match. The proposed algorithm, in contrast, is consistently able to select the most informative parts of the image, which are then correctly matched with high probability. Fig. 6 shows an example of a proof-of-concept application of our method, in the form of panoramic photo stitching. For each image pair we took the matched corners of the 50 templates selected with RS and PROPOSED, and used them to computed homographies using the DLT algorithm. It is quite evident how the accumulation of errors stemming from randomly selecting templates severely degrades the appearance of the final

Figure 6. We show qualitative results of performing panorama stitching using both approaches. We use the corners of the 50 templates matched in both cases to obtain the homography of the panorama. *Top row*: Results of using our approach. *Botton row*: Results of using Random Sampling. It can be seen that panoramas obtained through random sampling are not as accurate when approximating an homography as done in [15]'s third experiment.

panoramic images.

As a final remark, we report on the computational efficiency of the proposed method. We perform our experiments on a server machine equipped with 32 Xeon cores and an Nvidia K40 GPU. Excluding the time needed for template matching itself, we observed that both training the matchable template detector and running it on all testing images take in the order of a few seconds. Applying the proposed method on a single image takes less than one second on average on a Nvidia GTX750M-equipped laptop.

5. Conclusion

In this paper we tackled the problem of automatically selecting *matchable* templates from an image, *i.e.* templates that have high probability of being correctly matched in another target image. To this end we proposed a *matchable template detector*, based on dense CNN features and a logistic regression classifier, which reliably extracts matchable templates at multiple scales and aspect ratios. The effectiveness of our approach on a challenging dataset has been validated in conjunction with the FAst-Match algorithm. The proposed algorithm, however, is in principle "matcher-agnostic" and can be employed with other template matching approaches. In particular, part of our future work consists in exploiting this methodology to build priors for non-rigid template matching algorithms and reduce the inherent ambiguities of problems like those presented in [27, 28].

Acknowledgments

This work has been partially funded by Spanish Ministry of Economy and Competitiveness under project RobInstruct TIN2014-58178-R and ERA-Net Chistera project ViSen PCIN-2013-047; and by the EU project AEROARMS H2020-ICT-2014-1-644271.

References

[1] B. Alexe, V. Petrescu, and V. Ferrari. Exploiting spatial overlap to efficiently compute appearance distances between image windows. In *Advances in Neural Information Processing Systems, NIPS*, 2011.

[2] N. Alt, S. Hinterstoisser, and N. Navab. Rapid selection of reliable templates for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2010.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding, CVIU*, 110(3):346–359, June 2008.

[4] S. Benhimane, A. Ladikos, V. Lepetit, and N. Navab. Linear and quadratic subsets for template-based tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2007.

[5] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision, IJCV*, 88(2):303–338, 2010.

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[7] A. W. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision, IJCV*, 63(2):141–151, 2005.

[8] C. Forster, M. Pizzoli, and D. Scaramuzza. Appearance-based active, monocular, dense reconstruction for micro aerial vehicles. In *Proceedings of Robotics: Science and Systems, RSS*, 2014.

[9] C.-S. Fuh and P. Maragos. Motion displacement estimation using an affine model for image matching. *Optical Engineering*, 30(7):881–887, 1991.

[10] W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2014.

[11] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2010.

[12] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *IEEE International Conference on Computer Vision, ICCV*, 2011.

[13] T. Kaneko and O. Hori. Feature selection for reliable tracking using template matching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2003.

[14] H. Y. Kim and S. A. de Araújo. Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. In *Proceedings of the 2Nd Pacific Rim Conference on Advances in Image and Video Technology, PSIVT*, 2007.

[15] S. Korman, D. Reichman, G. Tsur, and S. Avidan. Fast-match: Fast affine template matching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2013.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, NIPS*. 2012.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, IJCV*, 60(2):91–110, 2004.

[18] S. Mattoccia, F. Tombari, and L. D. Stefano. Fast full-search equivalent template matching by enhanced bounded correlation. *IEEE Transactions on Image Processing, TIP*, 17(4):528–538, 2008.

[19] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 27(10):1615–1630, 2005.

[20] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision, IJCV*, 65(1/2):43–72, 2005.

[21] F. Moreno-Noguer, V. Lepetit, and P. Fua. Pose priors for simultaneously solving alignment and correspondence. volume 5303 of *Lecture Notes in Computer Science*, 2008.

[22] O. Pele and M. Werman. Accelerating pattern matching or how much can you slide?. In *Asian Conference on Computer Vision, ACCV*, 2007.

[23] L. Porzi, S. R. Buló, P. Valigi, O. Lanz, and E. Ricci. Learning contours for automatic annotations of mountains pictures on a smartphone. In *Eighth ACM/IEEE International Conference on Distributed Smart Cameras*, 2014.

[24] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *IEEE International Conference on Computer Vision, ICCV*, 2013.

[25] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, May 2000.

[26] E. Serradell, M. Ozuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer. Combining geometric and appearance priors for robust homography estimation. volume 6313 of *Lecture Notes in Computer Science*, 2010.

[27] E. Serradell, M. Pinheiro, R. Sznitman, J. Kybic, F. Moreno-Noguer, and P. Fua. Non-rigid graph registration using active testing search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(3):625–638, 2015.

[28] E. Serradell, A. Romero, R. Leta, C. Gatta, and F. Moreno-Noguer. Simultaneous correspondence and non-rigid 3d reconstruction of the coronary tree from single x-ray images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[29] H. Shao, T. Svoboda, and L. V. Gool. ZuBuD — Zürich buildings database for image based recognition. Technical Report 260, Computer Vision Laboratory, Swiss Federal Institute of Technology, March 2003.

[30] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems, NIPS*. 2013.

[31] F. Tombari, S. Mattoccia, and L. D. Stefano. Full-search-equivalent pattern matching with incremental dissimilarity approximations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 31(1):129–141, 2009.

[32] Q. Wang and S. You. Real-time image matching based on multiple view kernel projection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2007.

[33] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks?. In *Advances in Neural Information Processing Systems, NIPS*. 2014.

[34] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 31(4):677–692, April 2009.