Boosting Local Matches with Convolutional Co-Segmentation

Erez Farhan
Ben-Gurion University of the Negev
Beer-Sheva, Israel
farhan@post.bgu.ac.il

Abstract

Matching corresponding local patches between images is a fundamental building block in many computer-vision algorithms. Most matching methods are composed of two main stages: feature extraction, typically done independently on each image, and feature matching which is done on processed representations. This strategy tends to create large amounts of matches, typically describing small, highly-textured regions within each image. In many cases, large portions of the corresponding images have a simple geometric relationship. We exploit this fact and reformulate the matching procedure to an estimation stage, where we extract large domains roughly related by local transformations, and a convolutional Co-Segmentation stage, for densely detecting accurate matches in every domain. Consequently, we represent the geometrical relationship between images with a concise list of accurately co-segmented domains, preserving the geometrical flexibility stemmed from local analysis. We show how the proposed co-segmentation improves the matching coverage to accurately include many low-textured domains.

1. Introduction

1.1. Image Matching

Image matching (or registration) is a fundamental problem in computer vision being consistently addressed in research for the last decades. This work focuses on the common case of registering two 2-D RGB images, where we seek for the 2-D correspondence field (with notation abuse):

$$\overrightarrow{F_{x,y}} \sim I_2(u + F_x(u,v), v + F_y(u,v)) \quad \forall (u,v) \in \Omega_{\overrightarrow{F}}$$

Where I_1, I_2 are two RGB images which share different projections of the same 3-D surfaces, and $\Omega_{\overrightarrow{F}}$ is the domain of $\overrightarrow{F_{x,y}}$. By $\sim$, we mean a projection of the same 3-D patch in every point, and not necessarily the same RGB level. We note that in many cases, $\overrightarrow{F_{x,y}}$ is not defined in all of I_1 due to

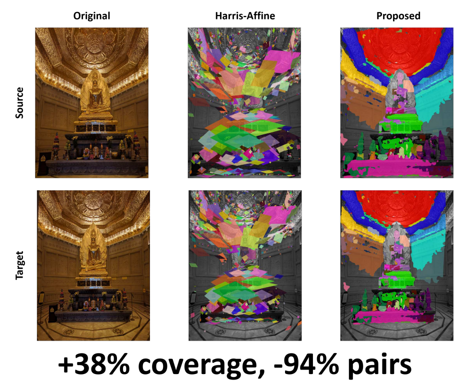


Figure 1: Demonstrating the proposed method vs. its initializer (Harris-Affine in this case): Initial matches are refined and adjusted to *accurately* capture significant parts of $\Omega_{\overrightarrow{F}}$, with a compact representation of the matching field $\overrightarrow{F_{x,y}}$. Corresponding Regions are overlaid with the same color.

common scene or viewpoint variations like occlusions, non-rigid motions or even zoom. We also note that $\overrightarrow{F_{x,y}}$ does not fully represent the geometric relation between the images, as $I_2$is not necessarily contained in the *range* of $\overrightarrow{F_{x,y}}$. For this purpose, we can similarly define the reciprocal:

$$\overrightarrow{F'_{x,y}} \sim I_1(u + F'_x(u,v), v + F'_y(u,v)) \quad \forall (u,v) \in \Omega_{\overrightarrow{F'}}$$

Combining $\overrightarrow{F_{x,y}}$ and $\overrightarrow{F'_{x,y}}$ fully represents the geometric relations between the images with significant redundancy. Since $\overrightarrow{F_{x,y}}$ and $\overrightarrow{F'_{x,y}}$ can be similarly analyzed, we focus our further discussion only on $\overrightarrow{F_{x,y}}$ for simplicity. Local image matching is an attempt to *independently* estimate $\overrightarrow{F_{x,y}}$ in limited domains where its behavior is relatively simple and the radiometrical variations can be easily compensated [23, 2, 6]. In practice, classic local matching approaches still suffer from two main shortcomings: 1) Coverage - repeatable features are typically found only in texture rich locations; 2) False Matches - mostly due to residual inaccuracies of the different extraction methods, and the limited amount of information within small features. These shortcomings introduce a trade-off. Increasing the coverage by considering low-textured features, increases the rate of false matches, while tightening the matching thresholds typically reduces the coverage. We stress that these shortcomings are inherent to two main characteristics of existing local matching approaches: over-localization and image independence. We show that accurately adjusting the shape of local features by considering information from more than one image, drastically reduces the rate of false matches while significantly increasing the coverage.

1.2. Beyond Single Image Feature Detection

For locating appropriate point-matching candidates, state-of-the-art methods had to consider both the *localizability* of the feature and the local *geometric variability* of the image. Localizable features are often related to texture attributes like corners or blobs [16, 23] which are also robust to illumination changes. These texture-based features should be automatically adjusted to capture the same physical patch across different views. In [23] a scale-space is used for locating the characteristic scale of a circular (or square) feature in a given image. The choice of characteristic scale or shape, fundamentally affects the ability to realiably match the features. Choosing a very small scale may introduce a numerical challenge[11], while choosing too-large scales may overshoot or force a complicated geometric model. Thus, the ideal choice of shape would be the *maximal* shape that still preserves a simple model (e.g. affine). Since this choice is depended on the geometrical relations between images (namely in $\overrightarrow{F_{x,y}}$), it cannot be derived from texture analysis of a single image. Thus, single image based feature detectors are not built to systematically capture the ideal shape. In this context, this work has two main contributions: 1) Defining the ideal local feature around a point; 2) Developing a method that manipulates initially detected feature matches, to approximate these ideal features. We show how these improved features both increase the local matching accuracy, and dramatically expand the coverage of $\Omega_{\overrightarrow{F}}$.

1.3. Fully Dense Matching

For many applications, it is very beneficial to have an accurate estimation of $\overrightarrow{F_{x,y}}$ in every pixel[15, 18]. A key challenge here is that isolated patches hold a very small amount of information, while the potential variability of the corresponding 3-D patch is huge. This challenge has been thoroughly addressed in the literature, mainly in the context of global optical-flow[13], stereo[32, 8, 24], or model-free matching methods[15, 18], where specific assumptions are made on the continuity of $\overrightarrow{F_{x,y}}$, or on the extent of the radiometric variations. In the context of local-matching, this motivation was almost irrelevant, as most methods produce a sparse set of matches in highly-textured locations, that cover only a small part of $\Omega_{\overrightarrow{F}}$. In contrast, Super-pixel based methods[4, 10] are used to cover low-textured local regions in a co-segmentation fashion, but lack a tight coupling of geometric constraints to the segmentation task, and often suffer from low repeatability [4, 10]. In this work, we tackle the problem of estimating fully-dense local correspondence fields around local matches by decoupling the problems of registration and co-segmentation. First, we utilize the geometric accuracy stemmed from physically justified local models to achieve accurate geometric hypotheses on local domains of the image. Then, much like [31, 19], we resort to the high representational power of convolutional neural networks (CNN's) to handle the overwhelming verity of radiometric nuisances and densely validate these hypotheses . As a result, we enhance each local match to include an accurate fully-dense correspondence field of its surrounding, including low-textured regions, while preserving the geometric and radiometric robustness of local features.

1.4. Closely Related Work

The idea of performing local analysis on tentative feature matches is not new. [12] introduced an iterative methodology to propagate geometric transformations induced by initially matched regions for locating new matches in neighboring areas. In [29], an optimization problem was formulated for co-expanding each initially detected matching regions, while *jointly* adjusting the local transformation model. In [15], a coarse-to-fine optimization approach was used to capture global non-rigid transformations, while utilizing local constraints in different scales. In [30], parametric models were used as primitives for globally estimating the optical flow. In contrast, we concentrate on analyzing independent local matches based on the affine and perspective models. This enables the decomposition of the registration problem into: 1) Transformation Estimation - where we utilize classic linear estimation techniques; and 2) Co-Segmentation - where we build on the representational power of CNN's, and their proven strengths in segmentation tasks[22]. Utilizing the affine model for expanding the local analysis was explored in [7], while [11] used a similar mechanism to also gradually improve local affine transformation estimations. The rational behind the expansion approach (other than reduction), is that since state-of-the-art feature detection is done from single images, these methods are often driven to a safe choice of small scaled features that is less probable to exceed a simple geometric model. We utilize this idea for accurate co-expansion (or co-reduction) of initially matched features and augment it with a fully dense analysis of local environments. As a result, we represent the geometrical relationship between two images using a concise list of matches (as demonstrated in Fig. 1) that captures a significant part of $\Omega_{\overrightarrow{F}}$.

2. Bigger Local Matches

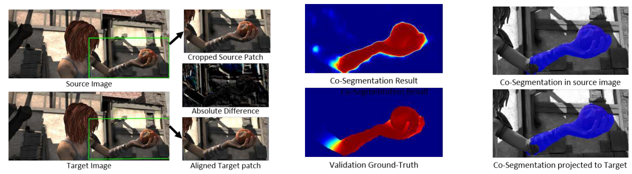
We claim that local features around points indeed have a desired characteristic shape. The choice of shape fundamentally determines the minimal geometrical model that captures the potential transformations of the feature across different views. As the geometry around a point can only get more complicated as the feature expands, the choice of the ideal shape is determined by two fundamental convictions: 1) Bigger features contain more context and allow more accurate geometrical modeling [11]; 2) Features should preserve the simplest model that locally approximates a smooth surface (the affine model). Thus, we define the desired feature shape as the:

Definition 1. Maximal Affine Set (MAS) - The maximal set of points (pixels) around a given point that can be registered by the same affine transformation

We note that a MAS is defined only given more than one image of its surroundings. For example, given two images related by a pure affine transformation, there is a single MAS that contains the entire image. Adversely, in case of highly dynamic scenes, many smaller disjoint MAS exit between different views. In many cases where planar surfaces are very common (e.g. domestic or urban scenes), a maximal affine set can capture a significant local piece of $\Omega_{\overrightarrow{F}}$. Furthermore, adjacent affine sets might have a very similar geometry. In such cases, it is physically justifiable to assume the validity of a 2-D projective model[17] that simultaneously captures several adjacent affine sets, and thus includes more context than each separate set. Detecting such cases will help us produce a more accurate and compact estimation of $\overrightarrow{F_{x,y}}$, while fully preserving the desired geometrical flexibility. In next section we propose a method for detecting MAS between pairs of images in general scenes, while also including a simple scheme for detecting locally planar surfaces, and detecting larger *projective* sets.

3. Maximizing Matches via Aligned Co-Segmentation

Following section (2), we propose a method to adjust initially detected local features for capturing the *MAS* around every match. As the input to our system, we assume that we are given feature matches between two images (*source* and *target*) produced by any standard feature matching pipeline. From every individual match, it is straightforward to extract the underlying affine transformation. The prime observation is that patches around the initial feature that agree on the same affine transformation are necessarily contained in the MAS. Improving the estimation of that transformation



(a) Extracting aligned patches (b) Aligned Co-Seg (c) Re-projection

Figure 2: An example of the co-segmentation process: (2a) - Extracting a rectified rectangle from the source image, and warping the corresponding shape in the target image to the same coordinate system. Absolute difference between the aligned patches is brought for reference. (2b) - The output of the co-segmentation network vs. the ground-truth validation, strong correspondence appears in red. (2c) - Co-segmentation result super-imposed on the original regions.

will allow us to better represent the local geometry, and capture more of the MAS. Thus, at this point, we have two goals: 1- Substantially improve the local transformation estimation. 2- Occupy the maximal affine set around the feature.

3.1. Initial Analysis: Verification, Expansion and Affine Transformation Refinement

Given a set of initial feature matches between two images, we follow the expansion and refinement procedure given in [11], separately for every match, to get a list of affine-consistent point matches around every initial feature match, and their corresponding affine transformation estimation.[1] We then represent each refined match as the minimal rectified rectangle containing all its corresponding affine-consistent points in the source image, and the newly estimated affine estimation. To better approximate the MAS, further expansion steps are performed on the refined match until it cannot be grown anymore.

3.2. Affine Region Unification

Following 3.1, we now have a set of independent local matches attached with an affine transformation for every match. In practice, many of these matches may contain signficant redundancy. For creating a more compact list of accurate matches, we observe that some of these matches may be unified to one larger local match, that will simplify the representation of the matching field $\overrightarrow{F_{x,y}}$. In some common cases, like planar or very distant surfaces, this unification can go very far. The main challenge in unifying these affine matches is in preserving their accuracy. We present a simple scheme for unifying affine matches that doesn't only compact the list of matches, but can also increase the

[1] Full implementation is given in http://www.ipol.im/pub/art/2017/154/

coverage of $\overrightarrow{F_{x,y}}$, while at least preserving the initial accuracy. For this purpose, we make two observations: 1) It may be beneficial to group several affine matches and estimate a projective transformation[3], especially for affine matches lying on the same plane or very distant surface. 2) From an information standpoint, grouping matches together creates a unified match with greater contextual information for higher match accuracy[11].

Given these observations, there are many considerable unification schemes. We provide a simple alternative by scanning the list of matches and performing the following:

Neighbor Extraction - For every scanned match (*current match*), we extract a sub-list of neighbors. These are matches whose rectangles sit close enough in the source image to the rectangle of the current match (a typical threshold would be 10% of the diagonal of the image).

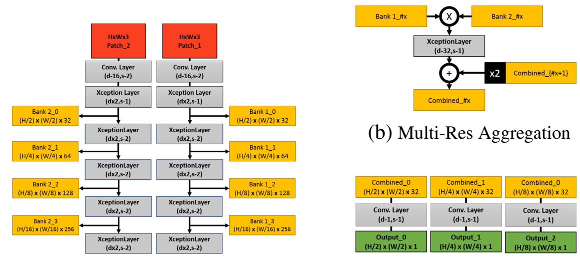
Neighbor Consensus - For each neighbor, we check if the transformation estimation of the current match can predict the transformation of the neighbor. That is, it agrees with the neighbor match up to a desired error threshold (e.g. 2 pixels). For all positive cases, we merge the neighboring matches into the current match and represent it by the rectified rectangle that contains all merged neighbors in the source image.

Model Selection - In this stage, we wish to select the correct geometrical model for the newly unified match. For this, we fit a joint projective transform to the current match and all its positive neighbors [3]. If the re-projection error of the transform on each of the independent matches is smaller than a desired threshold (e.g. 2 pixels), we choose the joint projective model. Otherwise, we preserve the original model of the current match.

List Update - We remove all the positive (merged) neighboring matches from the list.

3.3. Co-Segmenting Aligned Patches

Following 3.2, we now have a set of independent local matches represented by rectified rectangles in the source image and their corresponding local transformation estimations. In most cases, the rectified rectangles will contain outlier pixels that do not agree with the attached local transformation (Fig. 2b). This is especially true in cases of significantly expanded matches. The goal of the next and final step is to accurately select the inlier pixels within each local match, and discard the outliers. To eliminate the known geometrical differences between the patches, we use the local transformation to warp the target patch into the coordinate system of the rectangular source patch to create geometrically aligned candidate frames (Fig. 2a). Thus, our new mission is now to accurately detect the matching pixels between these frames.
At this stage it might be tempting to try and calculate pixel-wise RGB differences between the patches. Locations with RGB differences lower than some threshold will be considered inliers. This approach doesn't only involve setting an arbitrary threshold, but is also not robust to many possible nuisances like radiometric variations, noise and small residual geometric errors. As there are more sophisticated ideas for solving this intuitive comparison problem, it is extremely hard to devise a manual algorithm that will take all challenges into account and optimally combine the information to achieve accurate segmentation of inliers.



(a) Siamese Representation

(b) Multi-Res Aggregation

(c) Shared Output Classifier

Figure 3: The modules of the Co-Segmentation Network:(1) **d-x** - output of x channels (2) **dx2** - output with twice the number of channels as the input (3) **s-x** - output stride of x.

3.4. Fully Convolutional Co-Segmentation of Pairs

For the mission defined in 3.3, we choose to use the massive representational power of CNN's. We combine fundamental ideas from both binary patch comparison methods[14, 31], and fully-convolutional segmentation methods[22] to create a fully-convolutional neural network for co-segmentation of geometrically aligned image patches. The network is relatively shallow and designed for quick deployment with low memory consumption for allowing batch inference of multiple patch pairs. Accordingly, all the convolution operations are based on a 3×3 kernel. The network is composed of three functional modules:

A Siamese Representational Module - this module drives the rich independent representation of each patch in multiple resolutions (Fig. 3a). Following[9], we apply an Xception layer as the basic building block. This operation is economic in memory, quick to train, and still provides strong representational power. The module is built as a common neural encoder, increasing the depth of the representation, while decreasing the spatial dimension and thus increasing the receptive field to include more context. The parameters of the network are shared between the pipelines of the input patches, making it *a Siamese* module. For each of the two pipelines, there are four outputs in different resolutions.

A Multi-Resolution Aggregation Module - in this module, we aggregate the independent representations of both patches to create a representation of the relations between the patches (Fig. 3b). This operation is done on the representations of all resolutions. Inspired by the intuition of normalized cross-correlation, we first normalize the two incoming representation across their depth to a unit vector and then multiply them to one product which is moved through an Xception layer to produce a fixed depth representation (32 in our case). We then combine this result with the output of the Multi-Resolution layer of the lower resolution after a $\times 2$ bilinear up-sampling. Since this operation is recursive in nature, it allows aggregating finer patch-comparison information while preserving contextual information from lower resolution representations.

The Output Classifier - Finally, we wish to produce a 1-channel map that decodes the decision for every pixel of the patch. This output is produced by a point-wise (1×1) convolutional operation, that effectively works as a shared linear classifier for every pixel. The highest resolution output map produced by the network is two times smaller than the original patches, and can be interpolated back to the original resolution if needed. The network also produces outputs in lower resolutions that are discussed in the section concerning network training (4.3).

4. Training the Co-Segmentation Net

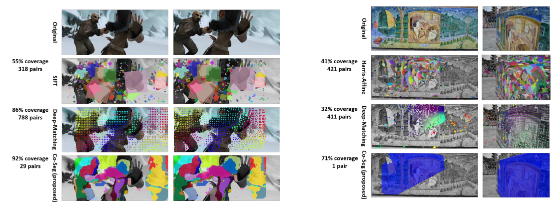
Training was supported by a standard state-of-the-art framework for optimizing neural networks. We used the Adam optimizer[20] for dynamically controlling the learning rate. In the following sub-sections we share a few non-standard notions regarding training.

4.1. Getting Ground Truth

For training the co-segmentation network described in 3.4, we need examples of geometrically aligned patch pairs with a known *agreement* map between them as a label (Fig. 2b). For extracting realistic data, we applied the steps in (3.1-3.3) to extract aligned candidate frames (3.3) from two different datasets containing sub-pixel accurate ground truth for the matching field $\overrightarrow{F_{x,y}}$: the Middlebury stereo dataset[27], and the computer-graphics based FlyingThings dataset [25]. For each pixel in an aligned candidate frame, we measure the euclidean displacement error with respect to the ground truth field.

4.2. Label Generation and Loss Metric

Given the displacement errors, we wish to generate labels appropriate for training. Theoretically, we can set a desired error threshold to create a binary label map. In practice, we map the displacement errors smoothly to the interval $[0, 1]$ to create soft labels that better reflect the ground-truth data. Thus, if we wish an error of τ pixels to be mapped to the border-line decision 0.5, we apply the map: $Z = exp((\delta/\tau)^2 * \log(0.5))$ to every displacement error δ



(a) Sintel example (b) H-Patches example

Figure 4: Result examples on the different tested datasets: (4a) - All methods produce a high rate of roughly correct matches due to relatively small displacements, while the proposed methods produces a more compact and accurate representation. (4b) - Harris-Affine and Deep-Matching produce many accurate matches, but also a considerable amount of false matches, while the proposed Co-Segmentation is more concise and accurate.

to create the ground-truth labels. We notice that this function maps the zero error $\delta = 0$ to $Z = 1$, which is a purely positive label, while Z goes smoothly to a negative label as δ increases. Having $Z \in [0, 1]$, we use the Sigmoid Cross-Entropy loss $L = -x \cdot z + \ln(1 + \exp(x))$ for every output pixel with value x, and ground-truth z.

4.3. Multi-Resolution Training

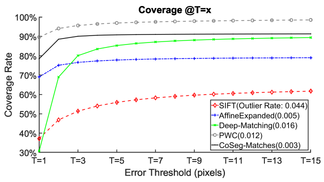
In principle, the network can have a single output map in the desired maximal resolution. As shown in Fig.3c, we have also simultaneously trained against lower resolution versions of the ground-truth labels. While this may seem sub-optimal from an optimization standpoint, we found that it didn't only significantly boost the training process, due to direct gradient propagation, but also effectively boot-strapped the lower-resolution layers to create better representation serving the higher-resolution layer. In practice, we found that avoiding the multi-resolution loss at any stage of the training, didn't noticeably improve the training loss, while deteriorating the validation loss.

5. Matching Experiments

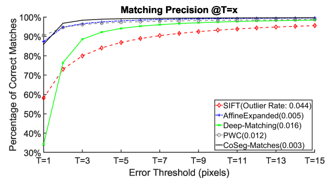
We compare the proposed method to state-of-the-art matching techniques under different scenarios, in two main aspects: scene coverage and matching accuracy. To emphasize the general applicability of the proposed method, we conduct our tests on two conceptually distinct datasets:

The Sintel dataset ("Clean" pass)[5] - containing synthetic images of 23 action clips with moderate motion statistics supplied with pixel-wise ground truth on a total of 1041 image pairs. In the context of *local matching*, the main challenge in this dataset is handling low textured areas.

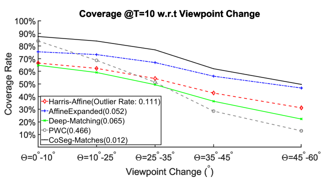
The H-Patches dataset ("Viewpoint" part)[1] - containing images of 59 planar scenes imaged from different



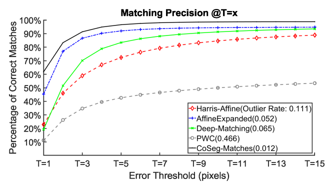
(a) Sintel Coverage



(b) Sintel Precision



(c) H-Patches Coverage



(d) H-Patches Precision

Figure 5: Matching Performance: (5a,5b) Successful coverage and precision of $\Omega_{\overrightarrow{F}}$, achieved by the different methods while varying the error standard (T) on Sintel; (5c) Successful coverage under variable viewpoint change on H-patches; (5d) Precision on H-patches dataset. (errors $>$ 15 pixels considered as outliers)

viewing angles, supplied with the ground truth homographies of a total of 295 image pairs. The main challenge in this dataset is in handling significant viewpoint changes. Image regions outside the main plane of the scene were ignored in all experiments.

We examine the matching performance of the proposed method relative to the matching methods used for initialization (SIFT and Harris-Affine respectively), the results after the affine-expansion[11] we described in 3.1, Deep-Matching[26] which is a state-of-the-art method for model-free combined local and global matching, and PWC [28], which is a fully trainable optical-flow method that specifically excels in wide-baseline scenarios. We compare the different methods under the accurate coverage criterion, where we follow [26] to define $coverage@T$ as the portion of $\Omega_{\overrightarrow{F}}$ covered by "correct" matches. A pixel is defined "correct" when its matching error is below T pixels. Following [26], for all referenced methods, a pixel is considered covered if a correct match exists within a 10 pixel distance. Since the *proposed* method is fully dense, we force a more strict coverage criteria for it, considering a pixel covered only if it explicitly has a correct match.

We note that these criteria present a trade-off between capturing more of $\Omega_{\overrightarrow{F}}$, and avoiding false matches. Since different applications demand different precision, we present the results while varying T. In Fig. 5a, we observe how Deep-Matching achieves significantly more coverage relative to SIFT and affine-expansion on the Sintel dataset for almost every threshold T. The proposed method, initialized by the same SIFT and affine-expansion, outperforms Deep-Matching on all thresholds. We note that the coverage gap between the two methods decreases as T is increased, indicating that a main advantage of the proposed method in this context, lies in precision (5b). For reference, we observe the preferable coverage of PWC on this classic optical-flow scenario, while the results are still comparable to the proposed method, which is purely local.

In fig 5c, we see how Deep-Matching falls slightly behind the affine methods on the geometrically challenging Hpatches dataset (Fig. 4) due to the high applicability of the affine model on planar scenes. Looking at Fig. 5c, we can explain the lower coverage of Deep-Matching on this dataset, by its sensitivity to bigger viewpoint changes. Again, we observe how the proposed method, initialized by the Harris-Affine and affine-expansion, considerably improves upon their coverage on all thresholds T and in all viewpoint conditions, while also improving the overall rate of correct matches (5d) under different thresholds. For reference, we observe how PWC struggles with the growing displacements introduced by the H-Patches data-set, and quickly deteriorates as the viewpoint change increases.

Note on Results - As the proposed method is merely a local boosting method, its results are strongly dependent of the initializing method and areas without nearby local matches might not be covered. We found that the overall coverage of our method can be further increased by using higher quality alternatives for initial matching,[2] such as [21], while preserving the desired matching compactness. These results were omitted from this comparison due to the high computational demand of [21] which is not comparable to the traditional methods used in our tests.

6. Conclusions

We presented a comprehensive method for boosting local image matching methods, that bridges the performance gap between local and global matching techniques in small displacement scenarios, while dramatically outperforming state-of-the art methods in more challenging scenarios. Another important contribution of the presented method, is in packing the geometric relationship between images in significantly bigger atoms that vastly cover the scene while accurately including low-textured regions. This has dramatic implications on point-wise applications like 3-D structure estimation, and combinatorial iterative tasks like parametric motion-segmentation and global transformation estimation, where bigger atoms can be extremely useful to reduce the number of iterations.

References

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and

[2]This is illustrated in the supplementary material attached to this work

learned local descriptors. In *CVPR*, 2017. 5

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006. 1.1

[3] J. Bentolila and J. M. Francos. Homography and fundamental matrix estimation from region matches using an affine error metric. *Journal of mathematical imaging and vision*, 49(2):481–491, 2014. 3.2

[4] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *Proceedings CVPR 2014*, pages 469–476, 2014. 1.3

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 5

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010. 1.1

[7] J. Cech, J. Matas, and M. Perdoch. Efficient sequential correspondence selection by cosegmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1568–1581, 2010. 1.4

[8] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 972–980, 2015. 1.3

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, 2016. 3.4

[10] A. Concha and J. Civera. Using superpixels in monocular slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 365–372. IEEE, 2014. 1.3

[11] E. Farhan and R. Hagege. Geometric expansion for local feature analysis and matching. *SIAM Journal on Imaging Sciences*, 8(4):2771–2813, 2015. 1.2, 1.4, 2, 3.1, 3.2, 5

[12] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *Computer Vision-ECCV 2004*, pages 40–54. Springer, 2004. 1.4

[13] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1–21, 2015. 1.3

[14] D. Gadot and L. Wolf. Patchbatch: a batch augmented loss for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4236–4245, 2016. 3.4

[15] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)*, 30(4):70, 2011. 1.3, 1.4

[16] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988. 1.2

[17] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[18] K. He and J. Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 111–118. IEEE, 2012. 1.3

[19] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 1.3

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[21] W.-Y. D. Lin, M.-M. Cheng, J. Lu, H. Yang, M. N. Do, and P. Torr. Bilateral functions for global motion modeling. In *European Conference on Computer Vision*, pages 341–356. Springer, 2014. 5

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1.4, 3.4

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1.1, 1.2

[24] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016. 1.3

[25] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. 4.1

[26] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):300–323, 2016. 5

[27] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014. 4.1

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 5

[29] A. Vedaldi and S. Soatto. Local features, all grown up. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1753–1760. IEEE, 2006. 1.4

[30] J. Yang and H. Li. Dense, accurate optical flow estimation with piecewise parametric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1027, 2015. 1.4

[31] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1.3, 3.4

[32] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 1.3