Dense Correspondences Across Scenes and Scales

Moria Tau and Tal Hassner

Abstract—We seek a practical method for establishing dense correspondences between two images with similar content, but possibly different 3D scenes. One of the challenges in designing such a system is the local scale differences of objects appearing in the two images. Previous methods often considered only small subsets of image pixels; matching only pixels for which stable scales may be reliably estimated. More recently, others have considered dense correspondences, but with substantial costs associated with generating, storing and matching scale invariant descriptors. Our work here is motivated by the observation that pixels in the image have contexts – the pixels around them – which may be exploited in order to estimate local scales reliably and repeatably. Specifically, we make the following contributions. (i) We show that scales estimated in sparse interest points may be propagated to neighboring pixels where this information cannot be reliably determined. Doing so allows scale invariant descriptors to be extracted anywhere in the image, not just in detected interest points. (ii) We present three different means for propagating this information: using only the scales at detected interest points, using the underlying image information to guide the propagation of this information across each image, separately, and using both images simultaneously. Finally, (iii), we provide extensive results, both qualitative and quantitative, demonstrating that accurate dense correspondences can be obtained even between very different images, with little computational costs beyond those required by existing methods.

✦

1 INTRODUCTION

Establishing correspondences between pixels in two images is a fundamental step in many computer vision applications. Typically, this is performed by either matching a sparse set of pixels, selected by a repeatable detection method (e.g., the Harris-Laplace [1]), or by matching all pixels in both images. Here we focus on the latter case, seeking a practical means for establishing dense correspondences across images of different scenes in different local scales.

Corresponding pixels are expected to reflect the same visual information. This information, however, may appear at different visual scales in different regions of each image: A car may be close to the camera in one photo, and far away in another; appearing large in the first and small in the second. All the while, buildings in the background remain at the same distance from the camera, appearing the same in both images. Sparse correspondence estimation methods seek stable scales, which can be repeatably detected in different images of the same scene, and which would allow extracting the same visual information regardless of the scales of the objects in the images. This approach, however, is only known to work well for very few pixels – those where stable scales can be reliably detected [2], [3].

Take, for example, the images in Fig. 1 (Top). These present the same semantic content (a "smiley"), ap-

- M. Tau is with the Department of Mathematic and Computer Science, The Open University of Israel, Israel.
- T. Hassner is with the Department of Mathematic and Computer Science, The Open University of Israel, Israel.
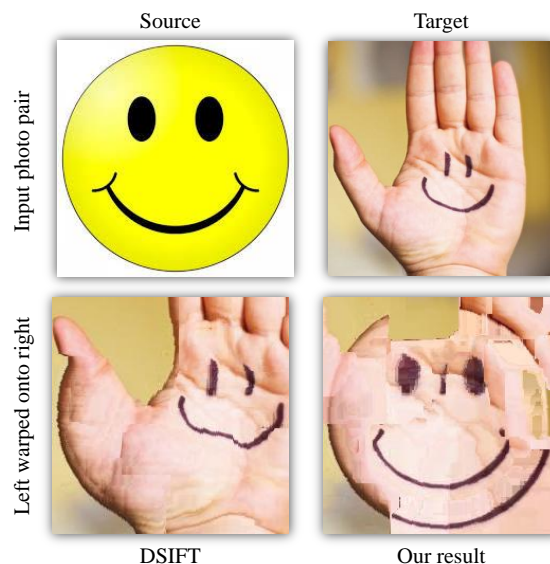 E-mail: hassner@openu.ac.il

Fig. 1: **Dense correspondences between the same semantic content ("smiley") in different scenes and different scales.** Top: Input images. Bottom: Results visualized by warping the colors of the "Target" photo onto the "Source" using the estimated correspondences from Source to Target. A good result has the colors of the Target photo, located in the same position as their matching semantic regions in the Source. Results show the output of the original SIFT-Flow method, using DSIFT without local scale selections (bottom left), and our method (bottom right).

pearing in very different scenes and in different scales. Densely matching the pixels in these two images is a problem made especially challenging due to the wide expanses of homogeneous regions, where stable scales are difficult to determine. In order to estimate correspondences, existing methods therefore make assumptions on the nature of the scenes, the photos, and the desired correspondences themselves.

Stereoscopic systems, for example, generally assume that the images being matched are of the same 3D scene, present objects in mostly the same scales, and were obtained under similar viewing conditions [4]. Recently, the same-scene assumption has been relaxed by the SIFT-Flow method of [5], [6]. Although an important step, SIFT-Flow relies on the Dense-SIFT (DSIFT) descriptor of [7], and therefore implicitly assumes that visual information in both images appears at the same (arbitrarily selected) scale. More importantly, this scale assumption is the same for all pixels in both images; in essence, assuming a single global scale for the two images and so greatly limiting its applicability.

In the past few years, a number of methods have proposed to eliminate this same-scale assumption, thereby allowing for dense correspondences to be obtained under very general settings. These, however, are either designed to match images from the same scenes [8], or require significant computation and storage in order to deal with unknown variations in scale [9], [10].

In this paper we show that dense correspondences can be established reliably, even in challenging settings, such as those exemplified in Fig. 1, with little more computational and storage requirements than needed for the original SIFT-Flow algorithm.

Our work follows the observation that previous attempts to produce robust, dense descriptors, did so by treating each pixel *independently*, without considering the scales of other pixels in the image. We turn to those few pixels where scales have been reliably estimated, and use them in order to estimate scales for all other pixels. Realizing this idea, however, requires that we answer an important question: How should scales be propagated, from the few pixels where they were reliably determined to all others, in a way which would ensure repeatable scale assignments, and consequent accurate dense correspondence estimation, regardless of local scale changes?

We answer this question by examining three means of propagating scale information across images, from detected key-points where scale is available to pixels where scales are not. Each of these methods considers progressively more information in order to more reliably propagate scales:

1) **Geometric.** We propagated scale information from detected interest points by considering only the spatial locations where scales were detected (Sec. 3.1).

2) **Image-aware.** Scales are propagated as above, but using image intensities in order to guide scale propagation. This is described in Sec. 3.2.

3) **Match-aware.** Finally, in Sec. 3.3 we consider the two images between which correspondences are to be estimated, propagating only the scales of pixels that were selected as (sparse) key-points in *both* images.

We test these three approaches on a wide variety of qualitative and quantitative experiments, comparing them to the state-of-the-art. Our results show that more contextual information results in better correspondences. More importantly, they demonstrate our proposed approach to not only outperform existing methods, but to do so as efficiently as the original SIFT-Flow.

2 PREVIOUS WORK

Why dense-flow? Matching all the pixels of two images is a basic step in stereoscopic vision, and as such has been the subject of immense research from the early years of computer vision. Surveying the work on stereo correspondences is outside the scope of this paper, and we refer the reader to popular computer vision textbooks for descriptions of previous related work. A comprehensive treatment of this subject is provided in particular in [11].

In recent years, a new thread of work has sought to look beyond the single-scene settings of stereo systems, attempting to provide dense correspondences between images, even if they only share the same semantic content. Here, the motivation rose from the realization that by densely linking the pixels of two images, local, per-pixel information can be transferred from one image to the other. This information can then be used for a wide range of computer vision applications, including single-view depth estimation [12], [13], [14], semantic labels and segmentation [15], [16], image labeling and similarity [17], and even new-view synthesis [18].

In all cases described above, however, the same scale was assumed for the images involved. This, either by enforcing global alignment of the images (e.g., [18]) or by assuming that a large enough collection of images exists such that at least one will portray the same information in the same scales [16]. The method presented here makes neither of these assumptions.

Scale-selection. Objects appear in different scales in different images. Determining the correct scale at which an image portion must be processed has therefore been a long standing challenge in computer vision. Here we only briefly survey the vast literature on this subject, and we refer to [19], [20] for more detailed discussions.

In his pioneering work, Lindeberg [21], [22] was one of the first to suggest seeking image pixels which have well-defined, characteristic scales. He proposed using the Laplacian of Gaussian (LoG) function computed over image scales, which is covariant with the scale changes of the visual information in the image, and so allows extracting scale invariant descriptors.

In a subsequent work, Lowe [2] proposed replacing the computationally expensive LoG function, with its Difference of Gaussians (DoG) approximation, in what has since become one of the standard de facto techniques for scale selection. Specifically, an image is processed by producing a 3D structure of x, y and $scale$, using three sets of sub-octave, DoG filters. This structure is scanned in search of pixels with higher or lower values than their 26 space-scale neighbors (3×3 neighborhood in the current scale and its two adjacent scales). The scale which provides these local extrema is selected as the characteristic scale for the pixel.

These feature detectors, as well as others, select pixels as keypoints if such a characteristic scale can be selected. Some perform scale selection along with elimination of low-contrast pixels to obtain more reliable detections. One popular example is the Harris-Laplace detector [1], which uses a scale-adapted Harris corner detector for spatial point localization and LoG filter extrema for scale selection. The detector performs these two steps iteratively, searching peaks both in space and in scale, and rejecting pixels with responses lower than a given threshold.

Dense-flow with changing scales. A well known limitation of scale selection techniques is that they typically find reliable scales in only very few image pixels. In [3], Mikolajczyk estimated that for a scale change factor of 4.4, as few as 38% of the pixels would be selected by a DoG scale selection criteria, of which only about 10.6% were actually correct. A bit later, Lowe, in [2] estimated that only around 1% of an image's pixels provide stable features which allow for descriptor extraction and matching. If our goal is to obtain dense correspondences between two images, the obvious question becomes: how should scales be selected for the remaining overwhelming majority of the pixels in the two images?

In recent years there have been several solutions proposed to this problem. In [8], image intensities around each pixel were transformed to log-polar coordinate systems. Doing so converted scale and rotation to translation. Translation invariance was then introduced by applying FFT, thus obtaining the Scale Invariant Descriptors (SID). Though SID descriptors were shown to be scale and rotation invariant, even on a dense grid, their use of image intensities directly implies that they are not well suited for matching images of different scenes [9].

The SIFT-Flow method of [5], [6] provided a means for dense correspondence estimation on a dense grid. They represented pixels in the image using Dense-SIFT (DSIFT) descriptors [7], produced at a constant, manually selected scale. This provides some scale invariance – due to the inherent robustness of the SIFT descriptors – but does not address anything beyond small scale changes.

In [9], the DSIFT descriptors used by the SIFT-Flow were replaced by the Scale-Less SIFT (SLS) representation. These are produced by first extracting at each pixel multiple SIFT descriptors, at multiple scales. The set of SIFTs extracted at a particular pixel was used to fit a linear subspace, represented using the subspace-to-point mapping of [23], [24], [25]. The SLS descriptors were shown to be highly robust to scale changes as well as allowing matching between different scenes, but the cost of this was a quadratic inflation in the descriptor size, making them difficult to apply in practice.

A different approach, somewhat related to our own work here, was taken by [26]. They too use SIFT-Flow as the matching engine, and either DSIFT or SID as the underlying representations. In their work, soft segmentation is first performed on images to be matched. When extracting descriptors, pixels contribute to the value of the descriptor in a manner which is inversely proportional to the likelihood of their belonging to the same segment as the keypoint for which the descriptor is produced. Thus, information from the background, or from other scales, has a limited effect on the values of the descriptor. This process requires that all descriptors are produced at the same scale, relying here on the segmentation to introduce scale-dependent information. Scales larger than the one used to extract the descriptors may therefor not be effectively represented. More importantly, it is unclear how segmentation affects scale, and vice versa, and so the limitations of this approach are not clear.

Rather than modify representations, Qiu et al. recently proposed a modified dense-flow estimation procedure [10]. Building on the cost function optimized by SIFT-Flow, they add terms reflecting the smoothness of scales. Specifically, they add a requirement that the relative scale of two neighboring pixels will be the same between their matching pixels in the other image. Though faster than both SID and SLS, their optimization is slower than the original SIFT-Flow. Moreover, their method does not allow computing scale invariant representations a priori, a desirable property when preprocessing is allowed or descriptors are used for applications other than dense correspondence estimation. Here we make a similar smooth scale assumption, yet employ it in preprocessing, rather than when estimating dense correspondences.

The method described here uses the original SIFT descriptors, varying the scales at which a descriptor is extracted in each pixel. It thus allows for correspondence estimation in the same computation and storage costs as the original SIFT-Flow as well as provides scale-invariant descriptors on a dense grid, usable beyond dense correspondence estimation applications.

3 PROPAGATING SCALES

Scale-invariant correspondences (dense or otherwise) are typically achieved through scale selection. To establish *dense* correspondences, here, we seek *dense scale selection*: selecting scales for all the pixels in the image.

(a) Input images (b) Detections (c) Geometric (d) Image-aware (e) Match-aware
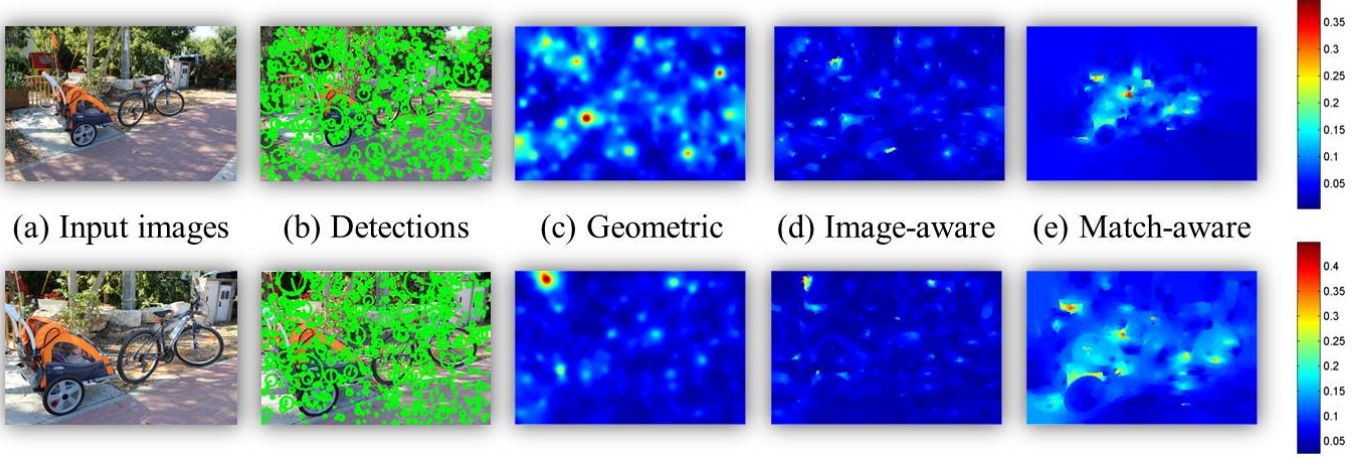
Fig. 2: **Visualizing three means of propagating scales.** (a) Two input images. (b) Sparse interest point detections, using the SIFT, DoG-based feature detector implemented by vlfeat [7]. Each detection is visualized according to its estimated scale. (c-e) Per-pixel scale estimates, $S_I(\mathbf{p})$, color-coded. (c) Geometric scale propagation (Sec. 3.1); (d) Image-aware propagation (Sec. 3.2); finally, (e), match-aware propagation, described in Sec. 3.3. Note how in (e) similar scale distributions are apparent for both images. On the right, color-bars provide legends for the actual scale values.

Formally, the scale space of image $I(x, y)$, denoted by $L(x, y, \sigma)$, is defined by a convolution of $I(x, y)$ with a variable-scale Gaussian $G(x, y, \sigma)$ [27], where

$$L(x, y, \sigma) = G(x, y, \sigma) \star I(x, y) \quad (1)$$

and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \quad (2)$$

The scale space of an image is scanned by multi-scale feature detectors, which seek space-scale locations x, y, σ where stable scales can be determined reliably, typically by seeking extrema in a scale-selection function defined over $L(x, y, \sigma)$.

Most pixel coordinates, however, do not have such extreme values, and are therefore left without scale selection. In the texture rich images of Fig. 2(a), for example, less than 0.1% of the pixels in each image were selected by the SIFT, DoG-based, feature detector, and assigned with scales (Fig. 2(b)). Our goal is to use these few detected pixels and their scale assignments in order to estimate scales for all the remaining image pixels.

We define the *scale-map* $S_I(\mathbf{p})$, for pixel $\mathbf{p} = (x, y)$, of image I as providing the scale $\sigma_\mathbf{p}$ associated with pixel coordinates $\mathbf{p}$ in I. Our goal can be stated as assigning scale values to all pixels in S_I. To this end, our key underlying assumption is stated as follows:

Assumption 0: Similar pixels should have similar scales.

This assumption, of course, leaves the notion of similarity open for interpretation, as well as the means of assigning scales in practice. Formally, we express this general assumption by defining a global cost for a scale assignment, as follows:

$$C(S_I) = \sum_\mathbf{p} \left(S_I(\mathbf{p}) - \sum_{\mathbf{q} \in N(\mathbf{p})} (w_{\mathbf{pq}} S_I(\mathbf{q})) \right)^2. \quad (3)$$

Similar expressions have previously been proposed for image processing tasks ranging from segmentation (e.g. [28], [29], and others) to colorization [30] and depth estimation [31]. Here, we assign scales to all image pixels by minimizing Eq. 3, subject to the constraints expressed by the known scales – the few pixels selected by a multi-scale feature detector, their positions in the image, and their assigned scales.

Intuitively, this cost interprets our assumption by requiring that the scale assigned to pixel $\mathbf{p}$ should be as similar as possible to a weighted average of the scales of its relevant similar pixels, denoted by $\mathbf{q} \in N(\mathbf{p})$. The weight $w_{\mathbf{pq}}$, associated with each of these pixels $\mathbf{p}$ and $\mathbf{q}$, is often referred to as an *affinity function* and takes values which sum to one for all pixels $\mathbf{q}$. It reflects the degree to which the scale of one pixel is assumed to influence another. In the next sections we consider two alternatives for this function, based on different interpretations of pixel similarity.

3.1 Geometric scale propagation

Assuming that the only information available to us are the pixel locations and scales returned by a feature detector, we make the following "geometric" assumption, where pixel scales are influenced by the scales of their spatially neighboring pixels:

Assumption 1, Influence of feature geometry on scales: Neighboring pixels (pixels with adjacent coordinates)

should be assigned with the same scales.

This assumption can be interpreted as using a constant value for all affinity functions, or $w_{\mathbf{pq}} = 1/|N|$ ($|N|$ the number of spatial neighbors for each pixel). Our cost function is quadratic and our constraints are linear. This implies large, sparse systems of equations which may be solved using a range of existing solvers [28], [30], [31].

Fig. 2(c) presents the scale-maps produced for each image using geometric scale propagation. As scale assignments are covariant with the underlying scale changes in the images, their ranges are somewhat different. Scale color codes are therefore provided on the right of Fig. 2. Visually, these maps may appear too noisy to be meaningful. In practice, as we show in Sec. 5, scales computed this way can still be beneficial for correspondence estimation.

3.2 Image-aware scale propagation

The use of constant affinity values is convenient whenever recomputing them for each image pair is impractical. Propagating scales using only the geometry of the feature point detections, however, ignores image intensities as valuable cues for scale assignment. We now consider the influence of intensities by revising our previous assumption.

Assumption 2, Influence of intensities on scales: Neighboring pixels with similar intensities, should be assigned with similar scales.

This assumption can be expressed by assigning affinity values using the normalized cross-correlation of the intensities of the two pixels, or:

$$w_{\mathbf{pq}} = 1 + \frac{1}{\sigma_{\mathbf{p}^2}} \left((I(\mathbf{p}) - \mu_{\mathbf{p}})(I(\mathbf{q}) - \mu_{\mathbf{p}}) \right). \tag{4}$$

Here, $\mu_{\mathbf{p}}$ and $\sigma_{\mathbf{p}}$ are the mean and variance of the intensities in the neighborhood of pixel $\mathbf{p}$.

This expression has successfully been used in the past for image colorization in [30]. Earlier, it was shown to reflect a linearity assumption on the relation of color and intensities in [32] and [33]. By using it here, we assume a linear relation between intensities and *scales*, rather than color. That is, that $S_I(\mathbf{p}) = a_{\mathbf{p}} I(\mathbf{p}) + b_{\mathbf{p}}$ with the coefficients $a_{\mathbf{p}}$ and $b_{\mathbf{p}}$ being the same for all the pixels in the immediate neighborhood of $\mathbf{p}$.

Of course, this is a simplifying assumption; the relation between scales and intensities can be more complex than a linear one. For our purposes, however, this assumption need not be strictly true: We only need for similar scales to be selected by considering similar intensity values in corresponding image regions.

Fig. 2(d) visualizes the scale-maps produced by image-aware propagation. These capture more of the underlying image appearance than the ones produced by the simpler geometry based method. In particular, the

distribution of scale assignments for the two images has more regions in common, suggesting better repeatability. Still, quite a lot of both images includes non-matching scale assignments, which we attempt to minimize next.

3.3 Match-aware scale propagation

As evident in Fig. 2(b), the sets of feature point detections in the two images are not identical. In fact, we expect only a small number of features to be correctly detected and common to both images (as discussed in Sec. 2). Here, these few corresponding pixels are used to seed the scale-map assignment process:

Assumption 3, Influence of matching feature points: When two images are being matched, scales should be assigned by considering feature point detections common to both images.

Rather than using all the detected feature points to seed the scale assignments, we first seek correspondences between the scale invariant descriptors, extracted at these sparse locations. This, in the same way that such correspondences are computed and used for parametric image alignment [2]. We take the 20% of the correspondences with the best closest to second-closest SIFT match ratio [2], and use only their scales to seed scale propagation in each image.

The result of this process is visualized in Fig. 2(e), which clearly shows corresponding regions of scale assignments: the same regions are assigned with high (low) scales in the two images.

Comparison with [34]: It is instructional to compare the process described here with the one used for 3D reconstruction from multiple views in [34]. They too begin with feature point extraction and sparse correspondence estimation. Their correspondences are used to build a preliminary 3D point cloud and estimates for the camera matrices of each input image. A continuous 3D surface is then produced by an "expansion" process which uses the initial correspondences to seed a search for neighboring matches in an effort to obtain dense correspondences.

We also use an initial, sparse set of correspondences to seed a search for dense correspondences, by propagating information to neighboring pixels. Here, however, we expand the scale estimates, not the correspondences themselves. This is performed for a single pair of images and without going through the process of 3D reconstruction and camera parameter estimation.

4 DISCUSSION: SCALE ACCURACY VS. FLOW ACCURACY

The assumptions underlying our method guarantee that some scales will be repeatable from one image to the next. In particular, the scales at interest points common to both images, in the match-aware propagation

of Sec. 3.3, will be covariant and would allow extraction of invariant descriptors. We expect that others, however, may still be inconsistent, resulting in descriptors produced at wrong scales with different feature values. It is therefor reasonable to consider: What effect would wrong scale assignments have on the overall quality of the flow?

To answer this question, we consider the method used for dense correspondence estimation, here, the SIFT-Flow of [6]. It uses belief propagation to minimize the following cost, defined over the estimated flow field (warp) $\mathbf{w}(\mathbf{p}) = [u(\mathbf{p}), v(\mathbf{p})]^T$ from each pixel in the source image I_A to its corresponding pixel in the target image I_B:

$$
\begin{aligned}
F(\mathbf{w}) = \sum_{\mathbf{p}} &\min \left(|| f(I_A, \mathbf{p}, S_A(\mathbf{p})) \right.\\
&\left. - f(I_B, \mathbf{p} + \mathbf{w}(\mathbf{p}), S_B(\mathbf{p} + \mathbf{w}(\mathbf{p}))) ||_1, k \right)\\
+ \sum_{\mathbf{p}} &\nu \left(|u(\mathbf{p})| + |v(\mathbf{p})| \right) \qquad (5)\\
+ \sum_{(\mathbf{p}_1, \mathbf{p}_2 \in N)} &[\min \left(\alpha |u(\mathbf{p}_1) - u(\mathbf{p}_2)|, d \right) + \\
&\min \left(\alpha |v(\mathbf{p}_1) - v(\mathbf{p}_2)|, d \right)]
\end{aligned}
$$

Here, k and d are constant threshold values and N defines a neighboring pixel relationship (e.g., $\mathbf{p}_1$ and $\mathbf{p}_2$ are nearby). The function f represents the SIFT feature transform, where we make explicit the scales used for computing the descriptors, represented by the scale-maps S_A and S_B.

The second term in Eq. 5 represents a requirement for small displacement. The third term, reflects a requirement for a smooth flow-field. Only the first term is affected by scale estimates, and so presumably, the minimization of Eq. 5 should be at least partially robust to scale estimate errors. In practice, the success of SIFT-Flow using Dense-SIFT (DSIFT) descriptors, implies that this is indeed the case: DSIFT uses a single, arbitrarily selected scale for all image pixels, and so one would expect that at least some pixels would have wrong scale estimates.

We empirically evaluate this tie between scales and accurate flow estimates, in order to obtain a measure of the robustness of SIFT-Flow to scale estimation errors. To this end, we compute the SIFT-Flow between images and themselves using increasing amounts of scale assignment errors.

Initially, the same constant scale is used for all pixels in each image pair. Using the default parameters of the SIFT extraction routine of [7], we take the SIFT bin size to be 8 pixels and the magnification factor to be 3, resulting in a scale value of $8/3 = 2.667$. We then progressively add noise to the scale-map of the target image by randomly selecting increasing numbers of pixels and adding Gaussian noise, with mean zero and STD of 2, to their assigned scales.
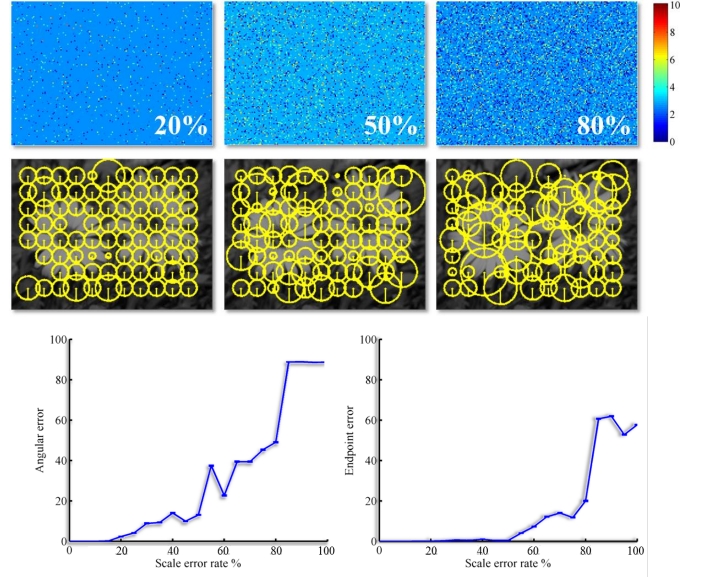

Fig. 3: **Effect of wrong scale estimates on flow accuracy using SIFT-Flow [6].** Top: Scale-maps for 20%, 50%, and 80% scale assignment errors, visualized by color coding scales (color-bar on the right). The correct scale is the default value of 2.667 for all pixels. Mid: Visualizing the assigned scales, for every 15th pixel. Bottom: Angular errors (left) and endpoint errors (right), ± SE, for increasing errors in scale estimates. Evidently, flow remains accurate up until about 20% errors rates.

Fig. 3(top) shows scale-maps with noise added to 20%, 50%, and 80% of the pixels. These synthetically modified scale-maps were used to extract SIFT descriptors (visualized in Fig. 3(mid)), which were then matched using SIFT-Flow. The quality of the resulting flow is evaluated by considering the angular and endpoint errors [35].

Fig. 3(bottom) plots the effect of wrong scale estimates vs. these two errors measures (± SE not shown as it was very small). Evidently, the endpoint errors reported in Fig. 3(bottom) remain almost zero, up until a rate of half the image pixels being assigned with wrong scales. Angular errors appear more sensitive to the noise, beginning to grow at 20% scale assignment errors.

These experiments are synthetic: In a practical scenario, simply resizing one of the images would result in *all* its pixels being assigned wrong scales. Fig. 3 suggests that in such cases dense correspondence estimation would fail completely, which was indeed shown to be true for SIFT-Flow in [9]. The figure also suggests, however, that it may be sufficient to bring scale assignment errors down to only 20% in order for accurate dense correspondences to be obtained.

5 EXPERIMENTS

We tested our proposed methods on tasks involving images from different scenes (Sec. 5.2), and stereo with

scale changes (Sec. 5.3). These tests demonstrate the application of our method for the tasks of stereo matching and image hallucination[1].

Our experiments all use SIFT-Flow [6] to compute the dense correspondences, varying the representations used in order to compare the following alternatives: Dense SIFT (DSIFT) [7]; Scale Invariant Descriptors (SID) [8]; Scale-Less SIFTs (SLS) [9]; and the two segmentation aware descriptors of [26], the segmentation aware SID (Seg. SID) and the segmentation aware SIFT (Seg. SIFT). In all cases, we used the code published by the respective authors of each method, with their recommended parameters unchanged. These methods were compared against our own geometric scale propagation (Geo.), image-aware propagation (Image) and match-aware propagation (Match).

5.1 Implementation and run-time

We implemented all three versions of our scale propagation technique in MATLAB. The multi-scale feature detections used by our proposed methods were obtained using the standard SIFT detector, implemented in the vlfeat library [7]. Minimizing the sparse system of equations resulting from the cost of Eq. (3) was performed using the built-in MATLAB solver, computed on neighborhoods of 3×3 pixels. Finally, scale-varying, dense SIFT descriptors were extracted with vlfeat [7].

Run-time was measured on an Intel Core i5 CPU, 1.8GHz, with 4GB of RAM and running 64Bit Windows 8.1. We use very small images for these tests (78×52 pixels) in order to avoid measuring run-time required for swapping memory, when using the more memory intensive representations (SID and SLS).

Descriptor sizes and flow-estimation run-times are summarized in Table 1. Descriptor dimensions were those measured in practice when running the code provided by the authors of each method. We extract a single, 128D SIFT descriptor per pixel – the same storage required by the standard SIFT-Flow method, and *an order of a magnitude* less storage than required by both the SID and SLS representations. Not surprisingly, the time required for establishing flow using our own method is the same as the time required for the original SIFT-Flow, an order of magnitude less than the SID descriptors, and two orders of magnitude less than SLS.

Finally, we compared the time required for optimizing our cost function of Eq. (3) (propagating the scales) with the time required by SIFT-Flow to estimate correspondences. Here, we varied the size of the images from the original 78×52 pixels to 780×520 pixels. For all image sizes, scale propagation required less than 7% of the time for computing the correspondences themselves, using SIFT-Flow. Consequently, SIFT-Flow performed following scale propagation requires only slightly more time than runing SIFT-Flow once, without scale propagation.

1. More results are reported in a longer version of this paper, submitted for publication. Please see author home-page, for an updated version: www.openu.ac.il/home/hassner

TABLE 1: **Comparison of different descriptor dimensions, and flow-estimation run-time.** Mean run-times were measured using SIFT-Flow, on 78×52 pixel images.

Method	Flow run-time (sec.)	Dim.
DSIFT [6]	0.8	128D
SID [8]	5	3,328D
SLS [9]	13	8,256D
Seg. SID [26]	5	3,328D
Us	0.8	128D

5.2 Qualitative results

Figures 1, 4 and 5 present image hallucination results obtained by computing dense correspondences from source to target images, and then warping the target colors back to the sources using the estimated flows. In all cases, good results would have the target image colors warped to the shapes appearing in the source photos.

The results included in these figures were all selected in an effort to reflect the most challenging instances of the dense correspondence estimation problem. Image pairs exhibit extreme variations in local scales, different scenes, different viewing conditions and more. We additionally emphasize cases where images have large homogenous regions. Existing feature detectors typically cannot estimate local scales in such image regions. By propagating scale estimates, we allow for scale-invariant descriptors to be extracted and dense correspondences to be estimated even in such cases.

Fig. 5 provides a comparison of the three proposed methods of propagating scales: Geometric scale propagation (Sec. 3.1), image-aware propagation (Sec. 3.2), and match-aware propagation (Sec. 3.3). Evidently match-aware propagation provides the most coherent results, though its two simpler alternatives are comparable in the quality of their results.

Though the results obtained with our Match-aware scale propagation (Fig. 4, rightmost column) are sometimes qualitatively similar to those obtained by other representations, ours consistently produces good results. This, despite much lower run-time and storage requirements compared to the scale-invariant descriptors, SID, SLS, and Seg. SID. Unsurprisingly, DSIFT performs worst when applied to image pairs with scale changes.

5.3 Scaled-Middlebury results

We repeat the qualitative experiments reported in [9], measuring the accuracy of stereo correspondences in the presence of extreme scale changes. We use the well-known Middlebury data set [35], containing pairs of images of the same scenes, acquired from different viewpoints. Since these images do not include scale changes these are introduced by re-sizing both images in each pair, one to 0.7 its size and one to 0.2 (the original sizes are not used, due to limitations of memory for the SLS and SID descriptors). Our tests include the image pairs with ground truth dense correspondences, which we use

Fig. 4: **Image hallucination results.** Each row presents dense correspondences established from source images to target images, illustrated by warping the target photos back to the sources using the estimated flows. We compare the following representations, from left to right: DSIFT [7], SID [8], SLS [9], Segmentation aware SID (Seg. SID) [26], and SIFT descriptors extracted using our own Match-aware scale propagation. Good results should have the colors of the target photos, warped to the shapes appearing in the source photos.

to compute Angular Error (AE) and Endpoint Error (EE) rates, along with standard deviations ($\pm$ SD) [35] for each of the representations tested.

Our results are reported in Table 2. These demonstrate that by propagating scales we achieve better accuracy on almost all of the tested pairs, falling in only slightly behind the far more expensive multi-scale representations, when this is not the case.

6 CONCLUSIONS

Modern computer vision systems owe much of their success to the development of effective scale selection techniques, key to the extraction of local, scale-invariant descriptors. These widely used techniques have focused almost entirely on the few image locations where local appearance variations provide sufficient cues for selecting reliable (repeatable) scales. In contrast, we propose a means for determining reliable scales for *all* the pixels
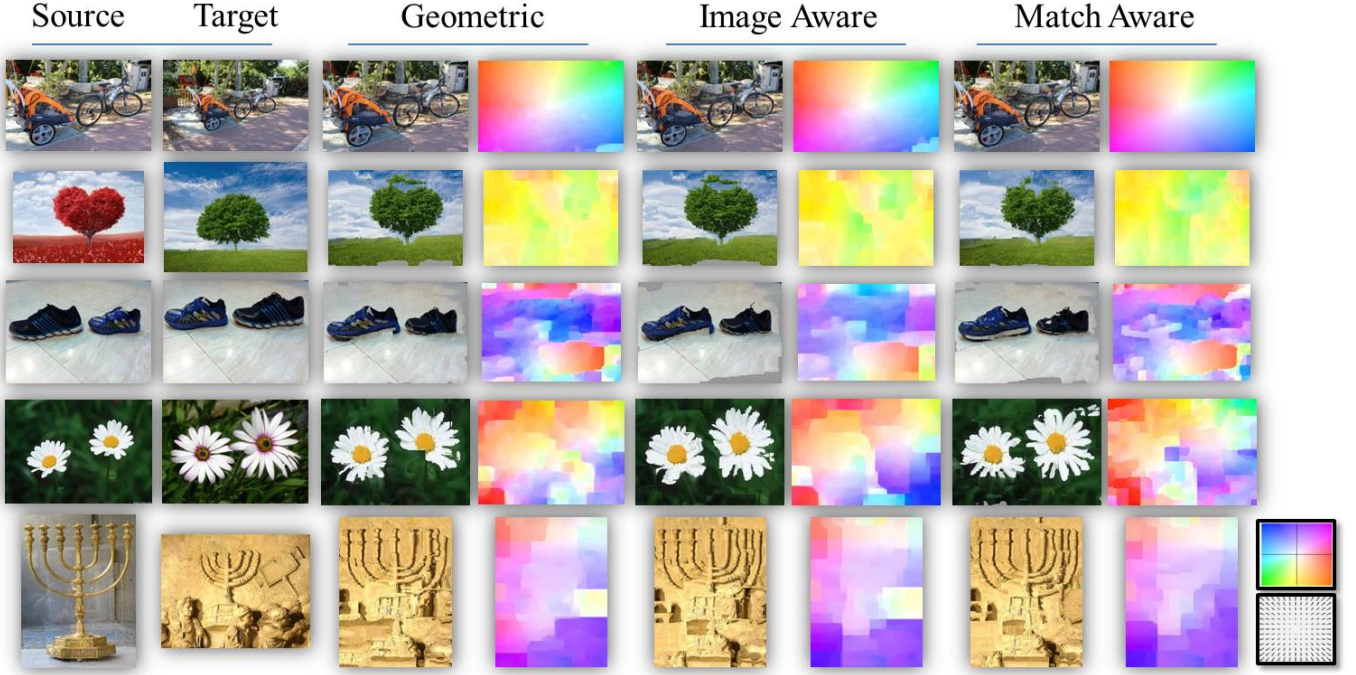
Fig. 5: **Image hallucination results - comparison of proposed methods.** Each row presents dense correspondences established from source images to target images, illustrated by warping the target photos back to the sources using the estimated flows. We compare our three proposed methods for propagating scales, from left to right: Geometric scale propagation (Sec. 3.1), image-aware propagation (Sec. 3.2), and match-aware propagation (Sec. 3.3). Each hallucination result provides also a visualization of the estimated flow field. Flow legend is provided on the bottom right.

in the image, regardless of their local appearances.

We describe three means of propagating scales from pixels selected by a standard, multi-scale, feature detector to all other image pixels. Our approach allows for truly scale-invariant dense SIFT descriptors to be extracted and then matched between images. An important aspect of our method, is that unlike alternatives proposed in the recent past, it makes very little computation and storage requirements beyond those needed for matching standard, non scale-invariant, dense SIFT descriptors. The result is a practical, effective, and efficient method for establishing dense correspondences across scenes, which does not make any assumptions on local scale variations in the images being matched.

Our method was tested qualitatively, by producing image hallucination results for challenging image pairs, as well as quantitatively for its flow accuracy. These have all shown how propagating scales contributes to reliable and robust dense correspondence estimation.

Future work. This paper opens a number of prospective directions for future research. One immediate direction is to explore how well other transformations, chiefly local orientation, may benefit from a similar approach. Our initial experiments conducted by adding an orientation-map, analogous to the scale-maps used here, were inconclusive. We believe this is because rotation may be

a more global phenomenon compared to scale; rotations are often applied to entire images whereas scales frequently change from one portion of the image to another. Further study is required to see if and how orientation can also benefit from a similar approach.

Applications of dense correspondence estimation were surveyed in Sec. 2. An additional line of work would be to explore the impact of our method's improved robustness on existing and new label transfer tasks.

Finally, showing that robust dense correspondences can be established with reasonable computation and storage requirements, raises intriguing questions regarding the possible roles of dense correspondence estimation in biological vision. Correspondence estimation is well known to play a key part in depth perception by stereo vision. The success of label transfer approaches in computer vision suggests that it may be worth while to explore the existence of similar mechanisms in biological visual systems.

REFERENCES

[1] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] K. Mikolajczyk, "Detection of local features invariant to affine transformations," Ph.D. dissertation, Institut National Polytechnique de Grenoble, France, 2002.

TABLE 2: **Results on the scaled-Middlebury benchmark**. Angular errors (AE) and endpoint errors (EE), ± SD, on resized images from the Middlebury benchmark [35]. Lower scores are better; shaded cells are best scoring. Please see text for more details.

Data	DSIFT [6]	SID [8]	SLS [9]	Seg. SIFT [26]	Seg. SID [26]	Geo.	Image	Match
				Angular Errors ± SD				
Dimetrodon	3.13 ± 4.0	0.16 ±0.3	0.17 ±0.5	2.45 ± 2.8	0.23 ± 0.7	0.61 ± 0.7	2.95 ± 4.2	0.15 ± 0.3
Grove2	3.89 ± 11.9	0.66 ±4.4	0.15 ±0.3	4.77 ± 15.3	0.22 ± 0.6	2.30 ± 2.3	1.78 ± 2.1	0.13 ± 0.3
Grove3	2.67 ± 2.8	1.62 ±6.9	0.15 ±0.4	8.93 ± 15.6	0.22 ± 0.6	6.26 ± 19.3	1.72 ± 2.1	0.18 ± 0.4
Hydrangea	9.76 ± 18.0	0.32 ±0.8	0.22 ±0.8	7.10 ± 10.6	0.23 ± 0.7	1.72 ± 2.3	6.25 ± 11.6	0.19 ± 0.4
RubberWhale	5.27 ± 8.6	0.16 ±0.3	0.15 ±0.3	6.13 ± 17.2	0.16 ± 0.3	1.56 ± 2.1	3.31 ± 5.4	0.13 ± 0.2
Urban2	3.65 ± 10.7	0.37 ±2.7	0.32 ±1.3	2.82 ± 4.1	0.25 ± 1.1	0.53 ± 0.8	4.28 ± 6.8	0.21 ± 0.5
Urban3	3.87 ± 5.1	0.27 ±0.6	0.35 ±0.9	3.53 ± 4.4	0.31 ± 1.0	1.43 ± 1.96	3.79 ± 7.9	0.24 ± 0.5
Venus	2.66 ± 2.9	0.24 ±0.6	0.23 ±0.5	2.77 ± 6.7	0.23 ± 0.5	1.32 ± 1.2	2.43 ± 2.3	0.30 ± 0.6
				Endpoint Errors ± SD				
Dimetrodon	10.97 ± 8.7	0.7 ±0.3	0.8 ±0.4	10.34 ± 7.5	0.97 ± 1.1	2.72 ± 1.5	11.21 ± 10.2	0.81 ± 0.3
Grove2	14.38 ± 11.5	1.5 ±5.0	0.77 ±0.4	15.50 ± 11.0	1.05 ± 1.9	12.8 ± 10.2	9.06 ± 9.4	0.71 ± 0.3
Grove3	13.83 ± 9.7	4.48 ±10.5	0.87 ±0.4	24.33 ± 20.0	1.37 ± 3.3	14.4 ± 14.7	9.22 ± 7.7	1.15 ± 2.5
Hydrangea	25.32 ± 17.1	1.59 ±2.8	0.91 ±1.1	24.21 ± 17.3	0.88 ± 0.6	10.2 ± 8.9	15.69 ± 19.2	0.79 ± 0.5
RubberWhale	22.59 ± 15.8	0.73 ±1.1	0.8 ±0.4	17.33 ± 14.8	0.73 ± 0.4	7.63 ± 8.5	11.27 ± 15.6	0.68 ± 0.3
Urban2	18.96 ± 17.5	1.33 ±3.8	1.51 ±5.4	13.36 ± 10.3	1.21 ± 3.7	2.73 ± 1.7	15.51 ± 15.2	0.96 ± 1.4
Urban3	19.83 ± 17.1	1.55 ±3.7	9.41 ±24.6	15.44 ± 11.5	1.47 ± 4.1	6.10 ± 4.9	14.91 ± 15.0	1.15 ± 1.8
Venus	9.86 ± 8.7	1.16 ±3.8	0.74 ±0.3	11.86 ± 11.4	0.74 ± 0.5	4.25 ± 2.0	10.92 ± 11.5	0.82 ± 0.4

[4] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *Int. J. Comput. Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[5] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman, "SIFT flow: dense correspondence across different scenes," in *European Conf. Comput. Vision*, 2008, pp. 28–42, people.csail.mit.edu/celiu/ECCV2008/.

[6] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[7] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. int. conf. on Multimedia*, 2010, pp. 1469–1472, available: www.vlfeat.org/.

[8] I. Kokkinos and A. Yuille, "Scale invariance without scale selection," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2008, pp. 1–8, available: vision.mas.ecp.fr/Personnel/iasonas/code/distribution.zip.

[9] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On SIFTs and their scales," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2012, pp. 1522–1528.

[10] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu, "Scale-space sift flow," in *Proc. Winter Conf. on Applications of Comput. Vision*. IEEE, 2014.

[11] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[12] T. Hassner and R. Basri, "Example based 3D reconstruction from single 2D images," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2006.

[13] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *European Conf. Comput. Vision*. Springer, 2012, pp. 775–788.

[14] T. Hassner and R. Basri, "Single view depth estimation from examples," *arXiv preprint arXiv:1304.3915*, 2013.

[15] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, 2011.

[16] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2013, pp. 1939–1946.

[17] M. Rubinstein, C. Liu, and W. T. Freeman, "Annotation propagation in large image databases via dense image correspondence," in *European Conf. Comput. Vision*. Springer, 2012, pp. 85–99.

[18] T. Hassner, "Viewing real-world faces in 3D," in *Proc. Int. Conf. Comput. Vision*, 2013.

[19] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[20] H. Aanæs, A. L. Dahl, and K. S. Pedersen, "Interesting interest points," *Int. J. Comput. Vision*, vol. 97, no. 1, pp. 18–35, 2012.

[21] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[22] ——, "Principles for automatic scale selection," *Handbook on Computer Vision and Applications*, vol. 2, pp. 239–274, 1999.

[23] R. Basri, T. Hassner, and L. Zelnik-Manor, "Approximate nearest subspace search with applications to pattern recognition," in *Proc. Conf. Comput. Vision Pattern Recognition*, June 2007.

[24] ——, "A general framework for approximate nearest subspace search," in *Proc. Int. Conf. Comput. Vision Workshop*. IEEE, 2009, pp. 109–116.

[25] ——, "Approximate nearest subspace search," *Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 266–278, 2010.

[26] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, "Dense segmentation-aware descriptors," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2013, pp. 2890–2897.

[27] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales,"," *J. of App. stat.*, vol. 21, no. 2, pp. 225–270, 1994.

[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[29] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *Proc. Int. Conf. Comput. Vision*, vol. 2. IEEE, 1999, pp. 975–982.

[30] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. on Graphics.*, vol. 23, no. 3, pp. 689–694, 2004.

[31] M. Guttmann, L. Wolf, and D. Cohen-Or, "Semi-automatic stereo extraction from video footage," in *Proc. Int. Conf. Comput. Vision*. IEEE, 2009, pp. 136–142.

[32] A. Zomet and S. Peleg, "Multi-sensor super-resolution," in *Proc. workshop on Applications of Computer Vision*. IEEE, 2002, pp. 27–31.

[33] A. Torralba and W. Freeman, "Properties and applications of shape recipes," in *Proc. Conf. Comput. Vision Pattern Recognition*, vol. 2. IEEE, 2003.

[34] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, 2010.

[35] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vision*, vol. 92, no. 1, pp. 1–31, 2001.