A Fast Local Descriptor for Dense Matching*

Engin Tola Vincent Lepetit Pascal Fua
Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
{engin.tola,vincent.lepetit,pascal.fua}@epfl.ch

Abstract

We introduce a novel local image descriptor designed for dense wide-baseline matching purposes. We feed our descriptors to a graph-cuts based dense depth map estimation algorithm and this yields better wide-baseline performance than the commonly used correlation windows for which the size is hard to tune. As a result, unlike competing techniques that require many high-resolution images to produce good reconstructions, our descriptor can compute them from pairs of low-quality images such as the ones captured by video streams.

Our descriptor is inspired from earlier ones such as SIFT and GLOH but can be computed much faster for our purposes. Unlike SURF which can also be computed efficiently at every pixel, it does not introduce artifacts that degrade the matching performance.

Our approach was tested with ground truth laser scanned depth maps as well as on a wide variety of image pairs of different resolutions and we show that good reconstructions are achieved even with only two low quality images.

1. Introduction

Though dense shot-baseline stereo matching is well understood [7, 22], its wide baseline counterpart is, by contrast, much more challenging due to large perspective distortions and increased occluded areas. It is nevertheless worth addressing because it can yield more accurate depth estimates while requiring fewer images to reconstruct a complete scene.

Large correlation windows are not appropriate for wide-baseline matching because they are not robust to perspective distortions and tend to straddle areas of different depths or partial occlusions. Thus, most researchers favor simple pixel differencing [21, 5, 14] or correlation over very small windows [24]. They then rely on optimization techniques such as graph-cuts [14] or PDE based diffusion operators [25] to enforce spatial consistency. The drawback of using small image patches is that reliable image information can only be obtained where the image texture is of sufficient quality. Furthermore, the matching becomes very sensitive to illumination changes and repetitive patterns.

Figure 1. Depth maps for view-based synthesis: **Top row:** Two 800×600 calibrated images we use as input. **Bottom row:** On the left, the depth map computed using DAISY and on the right, depth map computed using normalized-cross correlation, which could not handle the large perspective and contrast change between the two input images.

An alternative to performing dense wide-baseline matching is to first match a few feature points, triangulate them, and then locally rectify the images. This approach, however, potentially is not without problems. If some matches are wrong and are not detected as such, gross reconstruction errors will occur. Furthermore, image rectification in the triangles may not be sufficient if the scene within cannot be treated as locally planar.

We, instead, advocate replacing correlation windows with local region descriptors, which lets us take advantage of powerful global optimization schemes such as graph-cuts to force spatial consistency. Existing local region descriptors such as SIFT [17] or GLOH [19] have been designed for robustness to perspective and lighting changes and have

*This work was supported in part by funds of the European Commission under the IST-project 034307 DYVINE (Dynamic Visual Networks).

proved successful for sparse wide-baseline matching. However, they are much more computationally demanding than simple correlation. Thus, for dense wide-baseline matching purposes, local region descriptors have so far only been used to match a few seed points [29] or to provide constraints on the reconstruction [25].

We, therefore, introduce a new descriptor that retains the robustness of SIFT and GLOH and can be computed quickly at every single image pixel. Its shape is closely related to that of [28], which has been shown to be optimal for sparse matching but is not designed for efficiency. We use our descriptor for dense matching and view-based synthesis using stereo-pairs which have too large a baseline for standard correlation-based techniques to work, as shown in Fig. 1. For example, on a standard laptop, it takes about 5 seconds to perform the computations using our descriptor over all the pixels of a 800×600 image, whereas it takes over 250 seconds using SIFT. Furthermore, it gives better results than SIFT, SURF, NCC and pixel differencing as will be shown using laser scanner data as a reference.

To be specific, SIFT and GLOH owe much of their strength to the use of gradient orientation histograms, which are relatively robust to distortions. The more recent SURF descriptor [4] approximates them by using integral images to compute the histograms bins. This method is computationally effective with respect to computing the descriptor's value at every pixel but does away with SIFT's spatial weighting scheme. All gradients contribute equally to their respective bins, which results in damaging artifacts when used for dense computation. The key insight of this paper is that computational efficiency can be achieved without performance loss by convolving orientation maps to compute the bin values. This lets us match relatively large patches — 31×31 — at an acceptable computational cost and improve robustness in unoccluded areas over techniques that use smaller patches. Using large areas requires to handle occlusion boundaries properly though and we address this issue by using different masks at each location and select the best one by using an EM framework. This is inspired by the earlier works of [11, 13, 12] where multiple or adaptive correlation windows are used.

After discussing related work in Sec. 2, we introduce our new local descriptor and present an efficient way to compute it in Sec. 3. In Sec. 4 we detail our EM framework to handle occlusions. Finally, in Sec. 5, we present dense reconstruction results, compare our algorithm's results with those of [24] and give ground truth comparison results with other descriptors including SIFT, SURF, NCC and pixel differencing with increasing baseline.

2. Related Work

Even though multi-view 3–D surface reconstruction has been investigated for many decades [22, 7], it is still far from being completely solved due to many sources of errors such as perspective distortion, occlusions, and textureless areas. Most state-of-the-art methods rely on first using local measures to estimate the similarity of pixels across images and then on imposing global shape constraints using dynamic programming [3], level sets [9], space carving [15], graph-cuts [21, 6, 14], PDE [1, 25], or EM [24]. In this paper, we do not focus on the method used to impose the global constraints and use a standard one [6]. Instead, we concentrate on the similarity measure all these algorithms rely on.

In a short baseline setup, reconstructed surfaces are often assumed near fronto-parallel, so the similarity between pixels can be measured by cross-correlating square windows. This is less prone to errors compared to pixel differencing and allows normalization against illumination changes.

In a wide-baseline setup, however, large correlation windows are especially affected by perspective distortions and occlusions. Thus, wide-baseline methods [1, 14, 25, 24] tend to rely on very small correlation windows or revert to point-wise similarity measures, which loose the discriminative power larger windows could provide. This loss can be compensated by using multiple [2, 25] or high-resolution [25] images. The latter is particularly effective because areas that appear uniform at a small scale are often quite textured when imaged at a larger one. However, even then, lighting changes remain difficult to handle. For example, [25] shows results either for wide baseline without light changes, or with light changes but under a shorter baseline.

As we shall see, our feature descriptor reduces the need for higher-resolution images and achieve comparable results using fewer number of images. It does so by considering large image patches while remaining stable under perspective distortions. Earlier approaches to this problem relied on warping the correlation windows [8]. However the warps were estimated from a first reconstruction obtained using classical windows, which is usually not practical in wide baseline situations. By contrast, our method does not require an initial reconstruction.

Local image descriptors have already been used in dense matching, though in a more traditional way, to match only sparse pixels that are feature points [27, 17]. In [25, 29], these matched points are used as anchors for computing the full reconstruction. [29] propagates the disparities of the matched feature points to their neighbors, while [25] uses them to initialize an iterative estimation of the depth maps.

To summarize, local descriptors have already proved their worth for dense wide baseline matching, but only in a limited way. This is due in part to their high computational cost and in past their sensitivity to occlusions. The technique we propose addresses both issues.

3. Our Local Descriptor

In this section, we briefly describe SIFT [17] and GLOH [19] and then introduce our DAISY descriptor. We discuss both its relationship with them and its greater effectiveness for dense computations. The shape of DAISY is similar to that of [28], but it is designed for computational efficiency.

3.1. SIFT and GLOH

SIFT and GLOH before PCA dimensionality reduction, are 3–D histograms in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction. They are computed over local regions, usually centered on feature points but sometimes also densely sampled for object recognition tasks [10, 16].

Each pixel belonging to the local region contributes to the histogram depending on its location in the local region, and on the orientation and the norm of the image gradient at its location: As depicted by Fig. 3(a), when an image gradient vector computed at a pixel location is integrated to the 3D histogram, its contribution is spread over $2 \times 2 \times 2 = 8$ bins to avoid boundary effects. More precisely, each bin is incremented by the value of the gradient norm multiplied by a weight inversely proportional to the distances between the pixel location and the bin boundaries, and also to the distance between the pixel location and the one of the keypoint. As a result, each bin contains a weighted sum of the norms of the image gradients around its center, where the weights roughly depend on the distance to the bin center.

3.2. Replacing Weighted Sums by Convolutions

In our descriptor, we replace the weighted sums of gradient norms by convolutions of the original image with several oriented derivatives of Gaussian filters. We will see that this gives the same kind of invariance as the SIFT and GLOH histogram building, but much faster for dense-matching purposes. More specifically, we compute the

$$\mathbf{G}_o^\Sigma = G_\Sigma * \left(\frac{\partial \mathbf{I}}{\partial o} \right)^+ \quad (1)$$

convolutions where G_Σ is a Gaussian kernel, o is the orientation of the derivative and the operator $(.)^+$ is such that $(a)^+ = max(a, 0)$. We refer to the convolution results $\mathbf{G}_o^\Sigma$ as *convolved orientation maps*. As we will detail below, we will build our descriptor by reading the values in the convolved orientation maps. We will refer to the oriented derivatives of the image $\mathbf{G}_o = (\frac{\partial \mathbf{I}}{\partial o})^+$ as *orientation maps*.

To make the link with SIFT and GLOH, notice that each location of the convolved orientation maps contains a value very similar to what a bin in SIFT or GLOH contains: a weighted sum computed over an area of gradient norms.

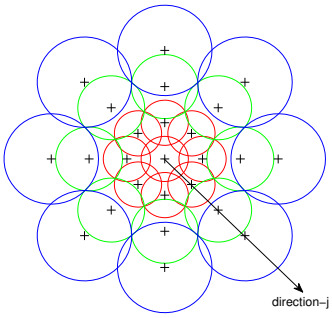


Figure 2. The DAISY descriptor. Each circle represents a region where the radius is proportional to the standard deviations of the Gaussian kernels and the '+' sign represents the locations where we sample the convolved orientation maps center being a pixel location where we compute the descriptor. By overlapping the regions we achieve smooth transitions between the regions and a degree of rotational robustness. The radii of the outer regions are increased to have an equal sampling of the rotational axis which is necessary for robustness against rotation.

The weights are slightly different: We use a Gaussian kernel whereas the weighting scheme of SIFT and GLOH corresponds to a triangular shaped kernel since the weights are linear. It is also related with tensor voting in [18] if we think of each location in our orientation maps as a voting component and our aggregation kernel as the voting weights.

The final values in these descriptors and ours will therefore not be exactly equal; nevertheless, our descriptor captures a very similar behavior. Moreover, this gives new insights on what makes SIFT work: The Gaussian convolution simultaneously removes some noise, and gives some invariance to translation to the computed values. This is also better than integral image-like computations of histograms [20] in which all the gradient vectors contribute the same: We can very efficiently reduce the influence of gradient norms from distant locations.

Our primary motivation here is to reduce the computational requirements, since convolutions can be implemented very efficiently especially when using Gaussian filters, which are separable. Moreover, we can compute the orientation maps for different sizes at low cost: Convolution with large Gaussian kernel can indeed be obtained from several consecutive convolutions with smaller kernels. Indeed if we have already computed $\mathbf{G}_o^{\Sigma_1}$ we can efficiently compute $\mathbf{G}_o^{\Sigma_2}$ with $\Sigma_2 > \Sigma_1$ by convolving $\mathbf{G}_o^{\Sigma_1}$:

$$\mathbf{G}_o^{\Sigma_2} = G_{\Sigma_2} * \left(\frac{\partial \mathbf{I}}{\partial o} \right)^+ = G_\Sigma * G_{\Sigma_1} * \left(\frac{\partial \mathbf{I}}{\partial o} \right)^+ = G_\Sigma * \mathbf{G}_o^{\Sigma_1},$$

with $\Sigma = \sqrt{\Sigma_2^2 - \Sigma_1^2}$.

3.3. The DAISY Descriptor

We now give a more formal definition of our *DAISY* descriptor. For a given input image, we first compute eight ori-

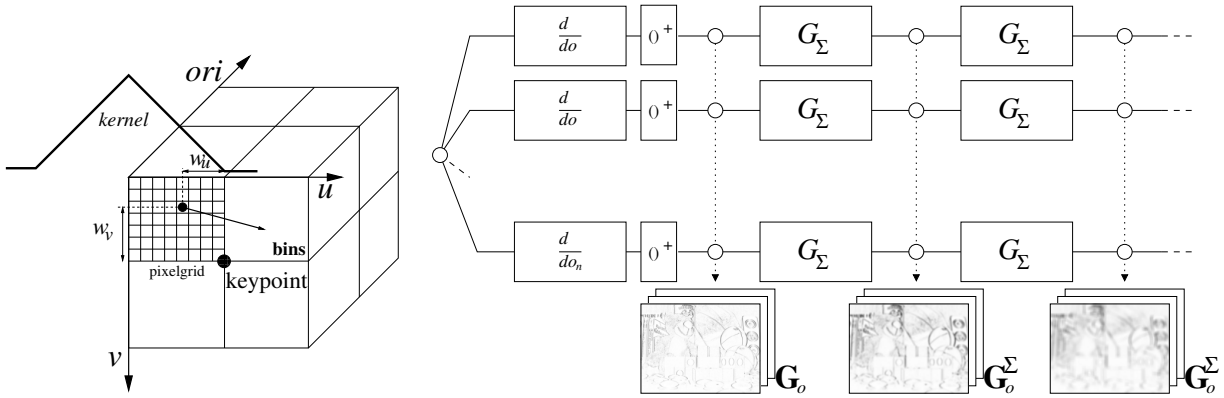


Figure 3. Relationship between SIFT and DAISY. (a) SIFT is a 3–D histogram computed over a local area where each pixel location contributes to bins depending on its location and the orientation of its image gradient, the importance of the contribution being proportional to the norm of the gradient. Each gradient vector is spread over $2 \times 2 \times 2$ bins to avoid boundary effects, and its contribution to each bin is weighted by the distances between the pixel location and the bin boundaries. (b) DAISY computes similar values but in a dense way. Each gradient vector also contributes to several of the elements of the description vector, but the sum of the weighted contributions is computed by convolution for better computation times. We first compute orientation maps from the original images, which are then convolved to obtain the convolved orientation maps $\mathbf{G}_o^{\Sigma_i}$. The values of the $\mathbf{G}_o^{\Sigma_i}$ correspond to the values in the SIFT bins, and will be used to build DAISY. By chaining the convolutions, the $\mathbf{G}_o^{\Sigma_i}$ can be obtained very efficiently.

entation maps, $\mathbf{G}$, one for each quantized direction, where $\mathbf{G}_o(u, v)$ equals the image gradient at location (u, v) for direction o if it is bigger than zero, else it is equal to zero. The reason for this is to preserve the polarity of the intensity change. Each orientation map is then convolved several times with Gaussian kernels of different Σ values to obtain convolved orientation maps for different sized regions. As mentioned in the previous section, this can be done efficiently by computing these convolutions recursively. Fig. 3(b) summarizes the required computations.

As depicted by Fig. 2, at each pixel location, DAISY consists of a vector made of values from the convolved orientation maps located on concentric circles centered on the location, and where the amount of Gaussian smoothing is proportional to the radius of the circles.

Let $\mathbf{h}_\Sigma(u, v)$ be the vector made of the values at location (u, v) in the orientation maps after convolution by a Gaussian kernel of standard deviation Σ:

$$\mathbf{h}_\Sigma(u, v) = \left[\mathbf{G}_1^\Sigma(u, v), \ldots, \mathbf{G}_8^\Sigma(u, v)\right]^\top , \qquad (2)$$

where $\mathbf{G}_1^\Sigma$, $\mathbf{G}_2^\Sigma$, and $\mathbf{G}_8^\Sigma$ denote the Σ-convolved orientation maps. We normalize these vectors to unit norm, and denote the normalized vectors by $\widetilde{\mathbf{h}}_\Sigma(u, v)$. The normalization is performed in each histogram independently to be able to represent the pixels near occlusions as correct as possible. If we were to normalize the descriptor as a whole, then the descriptors of the same point that is close to an occlusion would be very different in two images.

The full DAISY descriptor $\mathcal{D}(u_0, v_0)$ for location (u_0, v_0) is then defined as a concatenation of $\widetilde{\mathbf{h}}$ vectors, and can be written with a slight abuse of notation as:

Image Size	DAISY	SIFT
800x600	5	252
1024x768	10	432
1290x960	13	651

Table 1. Computation Time Comparison (in seconds)

$$\begin{bmatrix} \mathcal{D}(u_0, v_0) = \\ \widetilde{\mathbf{h}}_{\Sigma_1}^\top(u_0, v_0), \\ \widetilde{\mathbf{h}}_{\Sigma_1}^\top(\mathbf{l}_1(u_0, v_0, R_1)), \cdots, \widetilde{\mathbf{h}}_{\Sigma_1}^\top(\mathbf{l}_N(u_0, v_0, R_1)), \\ \widetilde{\mathbf{h}}_{\Sigma_2}^\top(\mathbf{l}_1(u_0, v_0, R_2)), \cdots, \widetilde{\mathbf{h}}_{\Sigma_2}^\top(\mathbf{l}_N(u_0, v_0, R_2)), \\ \widetilde{\mathbf{h}}_{\Sigma_3}^\top(\mathbf{l}_1(u_0, v_0, R_3)), \cdots, \widetilde{\mathbf{h}}_{\Sigma_3}^\top(\mathbf{l}_N(u_0, v_0, R_3)) \end{bmatrix}^\top ,$$

where $\mathbf{l}_j(u, v, R)$ is the location with distance R from (u, v) in the direction given by j when the directions are quantized into N values. In the experiments presented in this paper, we use $N = 8$ directions with $R_1 = 2.5$, $R_2 = 3R_1$, $R_3 = 6R_1$ and $\Sigma_1 = 2.55$, $\Sigma_2 = 3\Sigma_1$, $\Sigma_3 = 5\Sigma_1$. Thus, our descriptor is made of $8 + 8 \times 3 \times 8 = 200$ values, extracted from 25 locations and 8 orientations.

We use a circular grid instead of SIFT's regular one since it has been shown to have better localization properties [19]. In that sense, our descriptor is closer to GLOH before PCA than to SIFT. Also, the descriptor is naturally resistant to rotational perturbations as well by the use of isotropic Gaussian kernels with a circular grid. The overlapping regions ensure a smooth changing descriptor along the rotation axis and by increasing the overlap, we can further increase the robustness up to a certain point.

One advantage of the circular design and using symmetric kernels is that the descriptor can be computed in any orientation simply by rotating the sampling grid without the

need to recompute convolved orientation maps. The histograms will then also need to be shifted circularly but the total operation can be implemented very efficiently and the overhead is insignificant.

4. From Descriptor to Depth Map

We assume that we are given at least 2 calibrated gray-scale images and we compute the dense depth map of the scene with respect to a particular viewpoint which can either be equal to one of the input view points or it can be a completely different virtual position.

To perform dense matching, we use DAISY to measure similarities across images which we feed to the graph-cut-based reconstruction method of [6]. To properly handle occlusions, we incorporate an occlusion map, which is the counterpart of the visibility maps in other reconstruction algorithms [14]. The reconstruction and occlusion map are estimated by EM and a quick formalization is given below.

We exploit the occlusion map to define *binary masks* over our descriptors, which we use to avoid integrating occluded parts in the similarity estimation. We introduce predefined masks that enforce the spatial coherence of the occlusion map, and show they allow for proper handling of occlusions.

4.1. Formalization

Given a set of N calibrated images of the scene, we denote their descriptors as by $\mathbf{D}_{1:n}$. We estimate the dense depth map $\mathbf{Z}$ for a given viewpoint by maximizing:

$$\zeta = p(\mathbf{Z}, \mathbf{O} \mid \mathbf{D}_{1:n}) \propto p(\mathbf{D}_{1:n} \mid \mathbf{Z}, \mathbf{O}) p(\mathbf{Z}, \mathbf{O}) \quad (3)$$

where we also introduced an occlusion map term $\mathbf{O}$ that will be exploited below to estimate the similarities between image locations. As in [6], we assume some smoothness on the depth map, and also on our occlusion map using a Laplacian distribution.

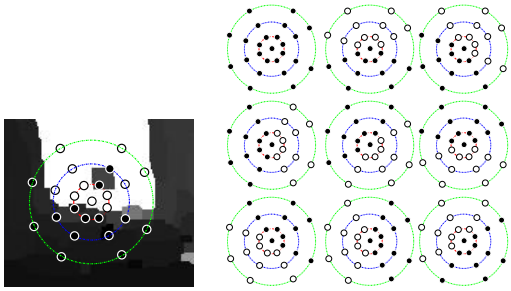
For the data driven posterior, we also assume independence between pixel locations:

$$p(\mathbf{D}_{1:n} \mid \mathbf{Z}, \mathbf{O}) = \prod_{\mathbf{x}} p\left(\mathbf{D}_{1:n}(\mathbf{x}) \;\middle|\; \mathbf{Z}, \mathbf{O}\right) . \quad (4)$$

Each term $p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}\right)$ of Eq. 4 is estimated using our descriptor. Because the descriptor considers relatively large regions, we introduce binary masks computed from the occlusion map $\mathbf{O}$, as explained in the next section, to avoid including occluded parts into our similarity score.

4.2. Using Masks over the Descriptor

Without occlusion-handling the $p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}\right)$ term of Eq. 4 would depend on distances of the form $\|\mathbf{D}_i(\mathbf{M}) - \mathbf{D}_j(\mathbf{M})\|$, where $\mathbf{D}_i(\mathbf{M})$ and $\mathbf{D}_j(\mathbf{M})$ are the descriptors at



(a) (b)

Figure 4. Binary masks for occlusion handling: We use binary masks over the descriptors to estimate location similarities even near occlusion boundaries. In this figure, a black disk with a white circumference corresponds to "on" and a white disks to "off". (a) We use the occlusion map to define the masks; and in (b) predefined masks makes it easy to enforce spatial coherence and to speed-up the convergence of EM estimation.

locations obtained by projecting the 3–D point $\mathbf{M}$ (defined by location $\mathbf{x}$ and its depth $\mathbf{Z}(\mathbf{x})$ in the virtual view) onto image i.

However, simply using the Euclidean distance $\|\mathbf{D}_i(\mathbf{M}) - \mathbf{D}_j(\mathbf{M})\|$ is not robust to partial occlusions: Even for a good match, parts of the two descriptors $\mathbf{D}_i(\mathbf{M})$ and $\mathbf{D}_j(\mathbf{M})$ can be very different when the projection of $\mathbf{M}$ is near an occluding boundary.

We, therefore, introduce binary masks $\{\mathcal{M}_m(\mathbf{x})\}$ such as the ones depicted in Fig. 4, that take into account only the visible parts when computing the distances between descriptors. Since the descriptor is built from 25 locations, these binary masks are also defined as 25 length vectors.

In order for the masks to depend on the current estimate of the occlusion map $\mathbf{O}$, we tried three different strategies: the simplest one depicted by Fig. 4(a) consists in thresholding the current estimate of the occlusion map $\mathbf{O}$ at the locations used by the descriptor to obtain a single binary mask $\mathcal{M}_m(\mathbf{x})$. The two other strategies use the predefined masks depicted by Fig. 4(b) that have a high spatial coherence. In the second strategy, each mask has a different probability which favors masks having large visible areas with similar depth values:

$$p(\mathcal{M}_m(\mathbf{x})|\mathbf{Z}, \mathbf{O}) = \frac{1}{Y}\left(\overline{v}_m + \frac{1}{\sigma_m^2(\mathbf{Z}) + 1}\right) \quad (5)$$

where $\overline{v}_m$ is the average visible pixel number, $\sigma_m(\mathbf{Z})$ is the depth variance within the mask region, and Y is the sum of all mask probabilities. The last strategy is a more radical version of the second strategy, where we only use the most probable mask instead of a mixture.

From a probabilistic point of view, that simply means we consider the following integration to compute $p\left(\mathbf{D}_{1:n}(\mathbf{x})|\mathbf{Z}, \mathbf{O}\right)$:

$$p\left(\mathbf{D}_{1:n}(\mathbf{x})|\mathbf{Z}, \mathbf{O}\right) = \sum_m p\left(\mathbf{D}_{1:n}(\mathbf{x})|\mathbf{Z}, \mathbf{O}, \mathcal{M}_m(\mathbf{x})\right) p\left(\mathcal{M}_m(\mathbf{x})|\mathbf{Z}, \mathbf{O}\right) . \quad (6)$$

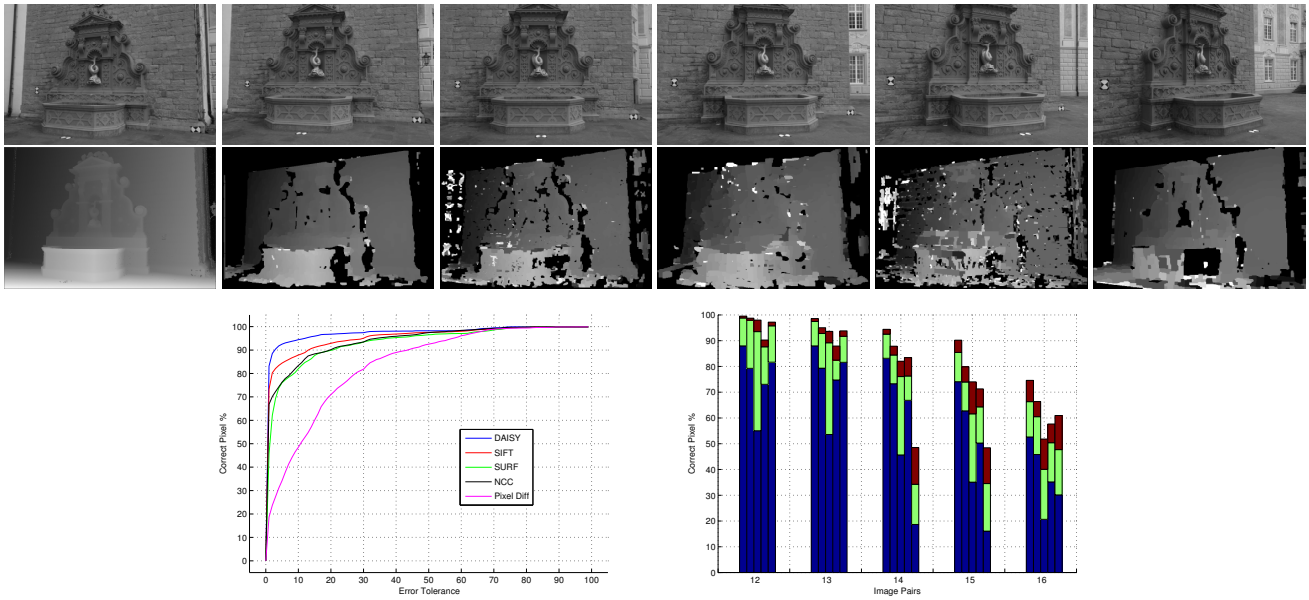


Figure 5. Comparing against ground truth. **First row.** In our tests, we match the left-most image against each one of the other five. **Second row.** The laser-scan depth-map we use as a reference and five depth-maps computed from the first and third images. From left to right, we used DAISY, SIFT, SURF, NCC and Pixel Difference. **Third row.** On the **left**, we plot the corresponding distributions of deviations from the laser-scan data, expressed as a fraction of the scene's depth-range. On the **right**, we summarize these distributions for the five stereo pairs of increasing baseline. Each group of bars represents a pair where the baseline increases gradually from left to right. Within groups, individual bars correspond to DAISY, SIFT, SURF, NCC, and Pixel Difference in that order. Within the bars, the bottom block denotes the percentage of correctly computed depths where the error threshold is set to be 1% of the scene's depth range, the middle block 5%, and the top one 10%. In all cases DAISY does better than the others and the wider the baseline, the most significant the difference.

In the first and third strategy, we use only one mask whereas in the second strategy, Eq. 6 is a mixture computed from several masks.

The mask probabilities are re-estimated at each step of the EM algorithm. In our experiments, using predefined masks resulted in more acceptable reconstructions and the last strategy, which always resulted in a much faster convergence towards a satisfying solution, was selected. These good performances over the other strategies can be explained by the fact that the chosen masks enforce the spatial consistency when comparing the descriptors.

Finally, following [6], the term $p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}\right)$ of Eq. 4 is taken to be $Lap\left(D(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}); 0, \lambda_m\right)$ where D is computed as

$$\frac{2(N-2)!}{N!} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sqrt{\sum_{k=1}^{25} \frac{\mathcal{M}^{[k]} \left\| \mathbf{D}_i^{[k]}(\mathbf{x}) - \mathbf{D}_j^{[k]}(\mathbf{x}) \right\|^2}{\sum_{q=1}^{25} \mathcal{M}^{[q]}}}, \tag{7}$$

where $\mathcal{M}^{[k]}$ is the k^{th} element of $\mathcal{M}$, and $\mathbf{D}_i^{[k]}(\mathbf{M})$ the k^{th} histogram $\widetilde{\mathbf{h}}$ in $\mathbf{D}_i(\mathbf{M})$.

5. Results

To compare DAISY against that of other descriptors, we used the images [26, 23] of Fig. 5 and an associated depth map obtained using a laser scanner, which we treat as a reference. We used DAISY, SIFT, SURF, NCC and pixel differencing to densely compute matching scores. They are then all handled similarly, as described in Section 4, to produce depth maps.[1] The figure's second row shows that DAISY produces fewer artifacts than the other descriptors

Figure 6. Low resolution and slightly blurry images: **Top:** Two input 640×480 images taken by a webcam. **Bottom:** On the left, reconstruction obtained using DAISY and on the right using NCC.

[1] Occlusions are handled in the same way in all cases, as discussed at the beginning of Section 4. The only difference is that we do not use binary masks to modify matching scores for descriptors other than DAISY.

(a) (b) (c) (d) (e) (f) (g)

Figure 7. Using low-resolution versions of the Brussels images [24]. (a,b,c) Three 768×510 versions of the original 2048×1360 images. (d,e) The depth-map computed using images (a) and (b) seen in the perspective of image (c) and the corresponding re-synthesized image. Note that the locations where there are people in one image and not in the other are correctly marked as occlusions. (f,g) The depth-map and synthetic image generated using all three images. Note that the previously occluded areas are now filled and that the people have been erased from the synthetic image.

for a specific pair. The graph on the third row shows that this is in fact true for all pairs we tested and, the wider the baseline, the more significant the difference.

In Fig. 6, we tested DAISY with somewhat blurry, low resolution 640×480 webcam images like the ones that can be obtained from video streams and DAISY again performs much better than correlation. SIFT produces a visually similar result but takes about 50 times longer to compute.

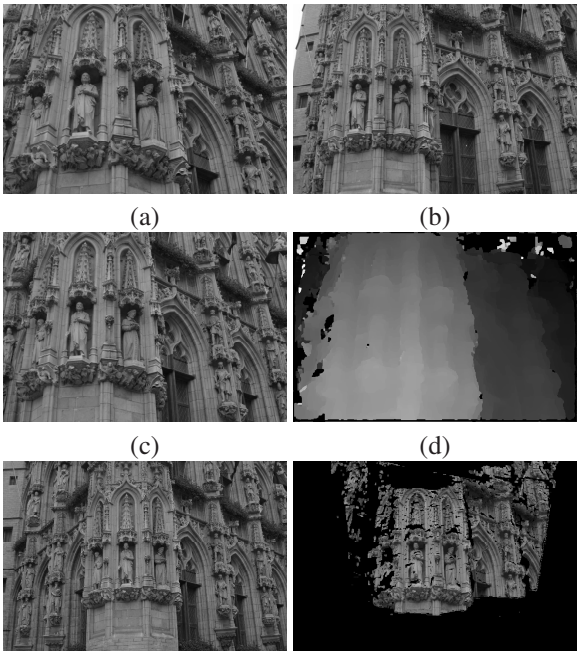
To compare our method to one of the best current techniques [24], we ran our algorithm on two sets of image pairs that were used in that paper, the Rathaus sequence of Fig. 8 and the Brussels sequence of Fig. 7. But instead of using the original 3072×2048 images, whose resolution is high enough for apparently blank areas to exhibit usable texture, we used 768×512 images in which this is not true. DAISY nevertheless achieved visually similar results.

Fig. 7 also highlights the effectiveness of our occlusion handling. When using only two images, the parts of the church that are hidden by people in one image and not in the other are correctly detected as occluded. When using three images, the algorithm returns an almost full depth map that lets us erase the people in the synthetic images we produce.

In Fig. 9 we show more stereo pair results with a baseline large enough for standard correlation-based techniques to fail and with substantial occlusions and lighting changes whereas we can compute reliable depth maps which we can use to synthesize realistic new views, as would be seen from a different perspective. To validate our approach, for each image pair, we use the perspective from a third image and compare that image with the one we synthesize.

6. Conclusion

In this paper, we introduced DAISY a new local descriptor, which is inspired from earlier ones such as SIFT and GLOH but can be computed much more efficiently for dense matching purposes. Speed increase comes from replacing weighted sums used by the earlier descriptors by sums of convolutions, which can be computed very quickly.



(a) (b) (c) (d) (e) (f)

Figure 8. Results on low-resolution versions of the Rathaus images [25]. (a,b,c) Three input images of size 768 × 512 instead of the 3072 × 2048 versions that were used in [24]. (d) Depth map computed using all three images (e) A fourth image not used for reconstruction. (f) Image synthesized using the depth map and the image texture in (a). Note how similar it is to (e). The holes are caused by the fact that a lot of the texture in (e) is not visible in (a)

The experiments suggest that although pixel differencing or correlation is good for short baseline stereo, wide baseline requires a more advanced measure for comparison. We propose to use DAISY for this purpose, as it is very efficient and it produces good reconstructions. Another advantage of our method is that we can use small images for computing reconstructions. This is important as it means that we can use our algorithm to process video streams which are generally at most 640×480 in size.

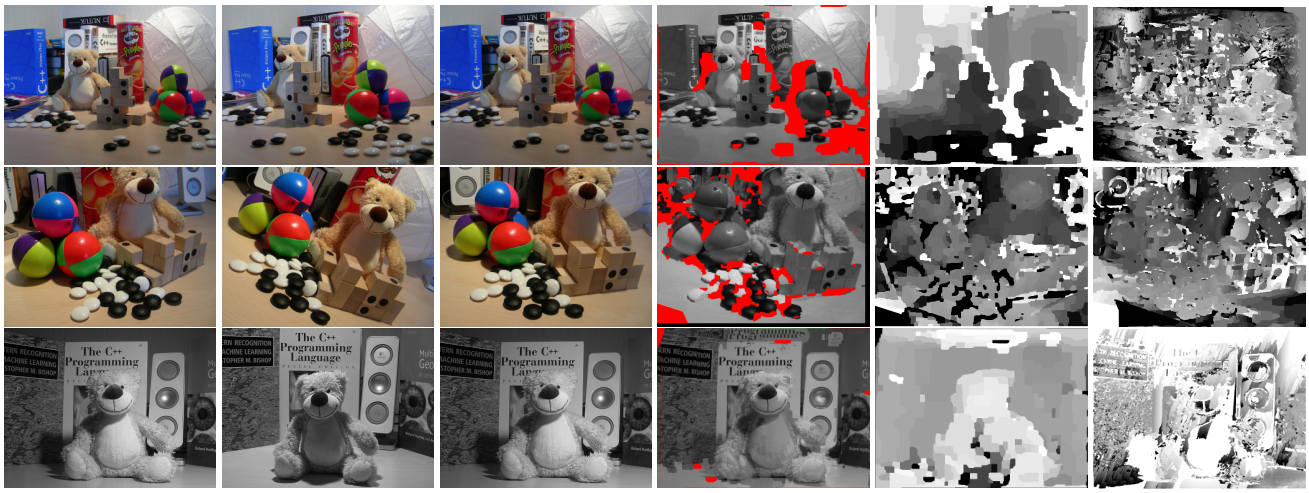


Figure 9. Depth maps and resynthesized images. In each row, the first two images are the inputs to our stereo-matcher. The third one is not used to compute the depth but only to validate the quality of the fourth one, which is synthesized from the first two using the DAISY depth map shown in fifth position. The final image is the depth map computed using normalized cross-correlation. The occluded areas are overlaid in red in the synthetic images. Note that in the last row, the two input images were lit differently which slightly degrades the performance but nevertheless allows credible resynthesis.

References

[1] L. Alvarez, R. Deriche, J. Weickert, J., and Sanchez. Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach. *JVCIR*, 13:3–21, Mar 2002.

[2] N. Ayache and F. Lustman. Fast and Reliable Passive Trinocular Stereovision. In *ICCV*, June 1987.

[3] H. Baker and T. Binford. Depth from edge and intensity based stereo. In *IJCAI*, pages 631–636, Aug. 1981.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[5] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *PAMI*, Apr. 1998.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *PAMI*, 23(11), 2001.

[7] M. Brown, D. Burschka, and G. Hager. Advacnes in computational stereo. *PAMI*, 25(8):993–1008, Aug. 2003.

[8] F. Devernay and O. D. Faugeras. Computing Differential Properties of 3–D Shapes from Stereoscopic Images without 3–D Models. In *CVPR*, pages 208–213, Seattle, June 1994.

[9] O. Faugeras and R. Keriven. Complete Dense Stereovision using Level Set Methods. In *ECCV*, Germany, June 1998.

[10] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *CVPR*, 2005.

[11] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *IJCV*, 14:211–226, 1995.

[12] S. Intille and A. Bobick. Disparity-space images and large occlusion stereo. In *ECCV*, pages 179–186, May 1994.

[13] T. Kanade and M. Okutomi. A Stereo Matching Algorithm with an Adaptative Window: Theory and Experiment. *PAMI*, 16(9):920–932, September 1994.

[14] V. Kolmogorov and R. Zabih. Multi-Camera Scene Reconstruction via Graph Cuts. In *ECCV*, Copenhagen, May 2002.

[15] K. Kutulakos and S. Seitz. A Theory of Shape by Space Carving. *IJCV*, 38(3):197–216, July 2000.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.

[17] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 20(2):91–110, 2004.

[18] G. Medioni, C. Tang, and M. Lee. Tensor voting: Theory and applications. In *RFIA*, 2000.

[19] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *PAMI*, 27(10):1615–1630, 2004.

[20] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *CVPR*, pages 829–836, 2005.

[21] S. Roy and I. Cox. A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem. In *ICCV*, 1998.

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, April-June 2002.

[23] C. Strecha. Multi-view evaluation-http://cvlab.epfl.ch/data, 2008.

[24] C. Strecha, R. Fransens, and L. Van Gool. Combined Depth and Outlier Estimation in Multi-View Stereo. In *CVPR*, 2006.

[25] C. Strecha, T. Tuytelaars, and L. Van Gool. Dense Matching of Multiple Wide-Baseline Views. In *ICCV*, 2003.

[26] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *CVPR*, 2008.

[27] T. Tuytelaars and L. Van Gool. Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions. In *BMVC*, pages 412–422, 2000.

[28] S. Winder and M. Brown. Learning Local Image Descriptors. In *CVPR*, Minneapolis, MI, June 2007.

[29] J. Yao and W.-K. Cham. 3–D Modeling and Rendering from Multiple Wide-Baseline Images. *SPIC*, 21:506–518, 2006.