# A Fast Local Descriptor for Dense Matching

Engin Tola Pascal Fua
Vincent Lepetit
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
{engin.tola,pascal.fua,vincent.lepetit}@epfl.ch
http://cvlab.epfl.ch/ tola/

Technical Report EPFL/CVLAB2007.08

**Abstract**

In this paper, we introduce a local image descriptor that is inspired by earlier detectors such as SIFT and GLOH but can be computed much more efficiently for dense wide-baseline matching purposes. We will show that it retains their robustness to perspective distortion and light changes, can be made to handle occlusions correctly, and runs fast on large images.

Our descriptor yields better wide-baseline performance than the commonly used correlation windows, which are hard to tune. Too small, they do not bring enough information. Too large, they become vulnerable to perspective variations and occlusion. Therefore, recent methods tend to favor small correlation windows, or even individual pixel differencing and rely on global optimization techniques such as graph-cuts to enforce spatial consistency. They are restricted to very textured or high-resolution images, of which they typically need more than three.

Our descriptor overcomes these limitations and is robust to rotation, perspective, scale, illumination changes, blur and sampling errors. We will show that it produces dense wide baseline reconstruction results that are comparable to the best current techniques using fewer lower-resolution images.

## 1 Introduction

Dense short-baseline stereo matching is now well understood [21, 6]. In contrast, larger perspective distortions and increased occluded areas make its wide-baseline counterpart much more challenging. It is nevertheless worth addressing because wide-baseline matching 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 in an image. Thus, most researchers favor simple pixel differencing [20, 4, 13] or correlation over very small windows [23]. They then rely on optimization techniques such as graph-cuts [13] or pde based diffusion operators [24] 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 light changes and repetitive patterns.

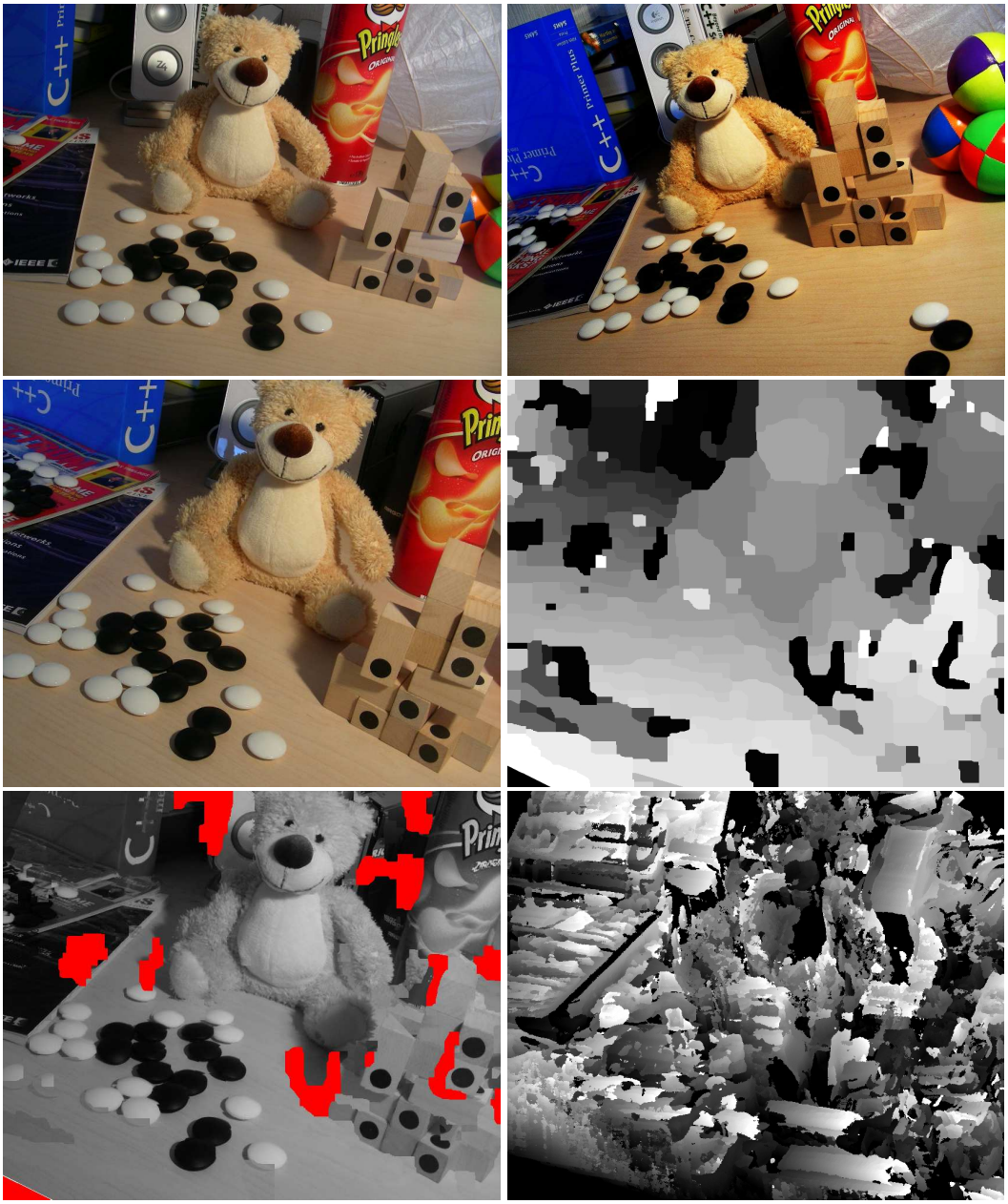


Figure 1: Depth maps for view-based synthesis: **Top row:** Two $800 \times 600$ calibrated images we use as input. **Middle row:** A third image and the depth map computed using only the first two, projected in the referential of the third. **Bottom row:** On the left, image re-synthesized using the depth map and the first two images. It is very similar to the original third image except at places where occlusions were detected. On the right, depth map computed using correlation, which could not handle the large perspective and contrast change between the two input images.

An alternative would be to use local region descriptors such as SIFT [16] or GLOH [17], which have been designed for robustness to perspective and lighting changes and have proved successful for sparse wide-baseline matching. They can be used to match larger image regions, even under severe perspective distortion, and are less prone to errors in the presence of weak textures or repetitive patterns because chances are better that at least part of the region can provide a reliable match. However, they are also much more computationally demanding than simple correlation. Thus, for dense wide-baseline matching purposes, they have so far only been used to match a few seed points [26] or to provide constraints on the reconstruction [24].

In this paper, we introduce a new descriptor that is inspired by SIFT and GLOH and retains their robustness but can be effectively computed at every single image pixel. We then use it to perform dense matching and view-based synthesis using stereo-pairs whose baseline is too large 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 computation using our descriptor over a $800\times600$ image, whereas it takes over 250 seconds using SIFT. Furthermore, it gives visually very similar results as one of the best current techniques [23] we know of on difficult examples using fewer lower-resolution images as will be discussed in the result section.

To be more specific, SIFT and GLOH owe much of their strength to their use of gradient orientation histograms, which are relatively robust to distortions. The key insight of this paper is that computing the bin values of the histograms can be achieved by convolving orientation maps, which can be done very effectively in the dense case. This lets us match relatively large patches –usually $31\times31$ and sometimes $73\times73$ patches for very high resolution images– at an acceptable computational cost. This improves robustness over techniques that use smaller patches in unoccluded areas, but could bring its own set of problems if occlusion boundaries were not handled properly. We address this issue by considering several different masks at each pixel location and chose the best. This is inspired by the earlier works of [10, 12, 11] where multiple or adaptive correlation windows are used. However, we formulate the problem in a more formal EM framework and achieve more refined occlusion estimates compared to the case where the full descriptor is used without any EM treatment.

After discussing related work in Section 2, we introduce our new local descriptor in and present a very efficient way to compute it in Section 3. In Section 3.4, we test its behavior under various transformations and compare it to that of SIFT [16] and of correlation windows of different sizes. Finally, we present our dense reconstruction results and compare them with those of [23] in Section 5.

# 2 Related Work

Even though multi-view 3–D surface reconstruction has been investigated for many decades [21, 6], it is still far from being completely solved because 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 [8], space carving [14], graph-cuts [20, 5, 13], PDE [1, 24], or EM [23]. In this paper, we do not focus on the method used to impose the global constraints and use a standard one [5]. Instead, we concentrate on the similarity measure all these algorithms rely on.

In a short baseline setup, the reconstructed surfaces are often assumed to be nearly fronto-parallel, so the similarity between pixels can be measured by cross-correlating square windows. This is less prone to errors than using 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 [13, 1, 24, 23] tend to rely on very

small correlation windows or revert to point-wise similarity measures, which looses the discriminative power larger windows could provide. This loss can be compensated by using multiple [2] or high-resolution [24] images. The latter is particularly effective because areas that may 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, [24] shows results either for wide baseline without light changes, or with light changes but under a shorter baseline.

As we will see, our feature descriptor reduces the need for higher-resolution images and achieve comparable results using less 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 [7]. 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 rely on an initial reconstruction.

Local image descriptors have already been used in dense matching but to match only sparse pixels that are feature points, in a more traditional manner [25, 16]. In [24, 26], these matched points are used as anchors for computing the full reconstruction. [26] propagates the disparities of the matched feature points to their neighbors, while, in a much safer way, [24] uses them to initialize an iterative estimation of the depth maps.

Local descriptors are therefore proved their usefulness in dense matching. The first obstacle to extending their use over all the pixels is the important computation time. We solve most of this problem by convolving orientation maps, which can be computed very effectively in the dense case, to compute the bin values of our local descriptor histograms.

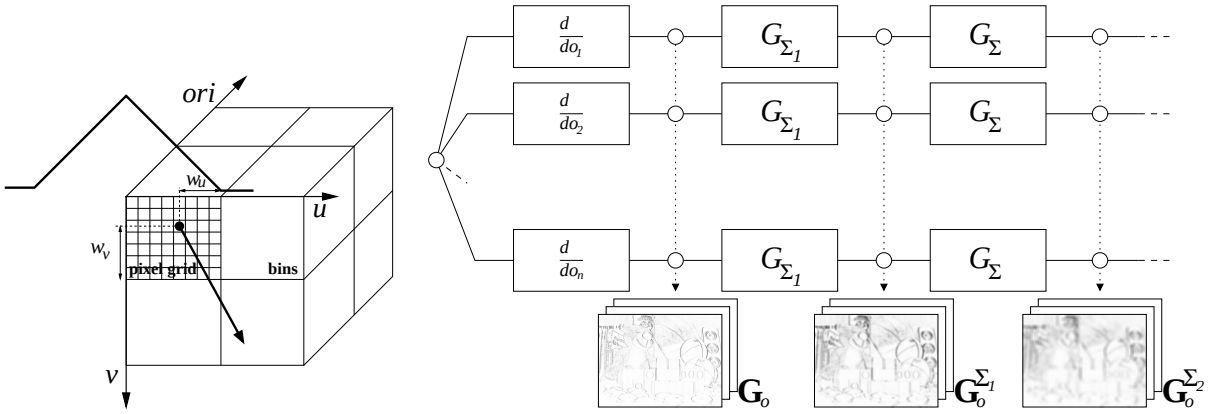
The second obstacle is the weakness to occlusions: Using large image patches gives its discriminative power to our similarity measure but it can fail near occluding boundaries; a well researched problem in the short-baseline case. For example, [12] adapts the window for each pixel location. A first reconstruction is estimated using a very small correlation window, each window is then expanded in the direction that minimizes an appropriate criterion, and the process is iterated. However this method is slow and may not converge towards a satisfying solution. A simpler approach is to compute, at each pixel location, several correlation windows centered around it, so that at least one of the windows does not overlap with both foreground and background when close to an occluding boundary [10, 11]. A unique value is then estimated as the minimum of the corresponding correlation values. We incorporate this idea into our measure in a similar technique in the sense that we consider the distances between local descriptors over different parts. However, we use an EM algorithm to choose the correct parts instead of the minimal correlation value heuristic.

# 3 Our Local Descriptor

In this section, we first briefly describe SIFT [16] and GLOH [18]. We then introduce our own DAISY descriptor and discuss both its relationship with them and its greater effectiveness for dense computations. Finally, we present experiments that demonstrate its reliability when matching images under different varying conditions.

## 3.1 SIFT and GLOH

Before PCA dimensionality reduction, SIFT and GLOH 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 [9, 15].




(a) (b)

Figure 2: 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.

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. 2(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 the convolutions of the original image with several oriented derivatives of Gaussian filters with large standard deviations. 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

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

convolutions where $G_{\Sigma}$ is a Gaussian kernel, and $o$ is the orientation of the derivative. 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 $\mathbf{G}_o = \left(\frac{\partial \mathbf{I}}{\partial o}\right)^+$ 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 a large area of gradient norms. The weights are slightly different: We use a Gaussian kernel where the weighting scheme of SIFT and GLOH corresponds to a kernel with a triangular shape since they weight linearly.

The final values in these descriptors and ours will therefore not be exactly equal; nevertheless, we will capture 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 [19] 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 scales at low cost: Convolution with large Gaussian kernel can indeed be obtained from several consecutive convolutions with smaller kernels: 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 $\mathbf{G}_o^{\Sigma_1}$, since we have:

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

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 orientation maps $\mathbf{G}_o$, 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 $\Sigma$ values to have convolved orientation maps for different scales. As mentioned above, this can be done efficiently by computing these convolutions recursively. Fig. 2(b) summarizes the required computations.

As depicted by Fig. 3, at each pixel location, DAISY consists of a vector made of values in 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 $\Sigma$:

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

where $\mathbf{G}_1^\Sigma$, $\mathbf{G}_2^\Sigma$, and $\mathbf{G}_8^\Sigma$ denote the $\Sigma$-convolved orientation maps. We normalize these vectors so that their norms are 1, 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 will be very different in two images.

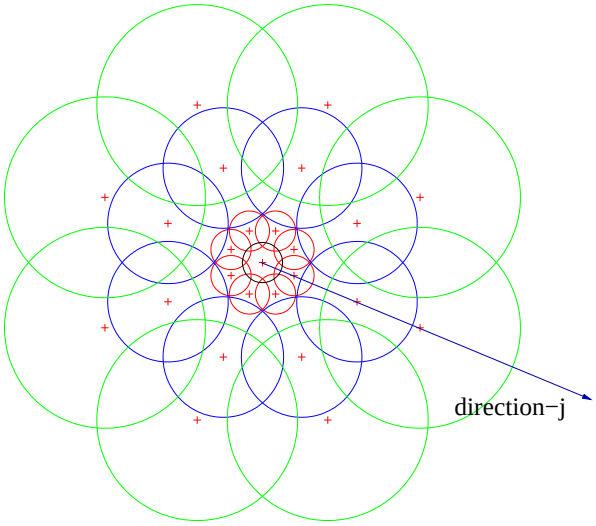



Figure 3: The DAISY descriptor. Each circle represents a region where the radius is proportional to the standard deviations of the Gaussian kernels and the '+' sign represents the locations where we sample the convolved orientation maps center being a pixel location where we compute the descriptor. By overlapping the regions we achieve smooth transitions between the regions and a degree of rotational robustness. The radius of the outer regions are increased to have an equal sampling of the rotational axis which is necessary for robustness against rotation.

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:

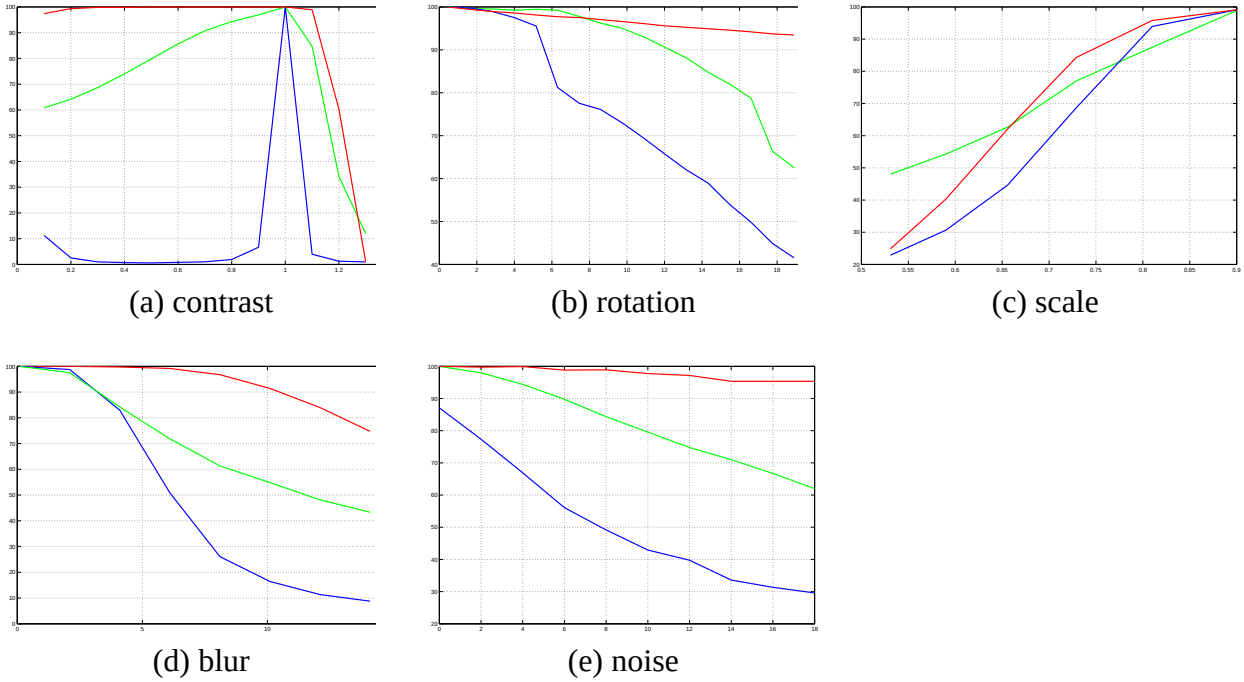
$$\begin{aligned}
&\mathcal{D}(u_0, v_0) = \\
\Big[\;&\widetilde{\mathbf{h}}^\top_{\Sigma_1}(u_0, v_0), \\
&\widetilde{\mathbf{h}}^\top_{\Sigma_1}(\mathbf{l}_1(u_0, v_0, R_1)), \cdots, \widetilde{\mathbf{h}}^\top_{\Sigma_1}(\mathbf{l}_N(u_0, v_0, R_1)), \\
&\widetilde{\mathbf{h}}^\top_{\Sigma_2}(\mathbf{l}_1(u_0, v_0, R_2)), \cdots, \widetilde{\mathbf{h}}^\top_{\Sigma_2}(\mathbf{l}_N(u_0, v_0, R_2)), \\
&\widetilde{\mathbf{h}}^\top_{\Sigma_3}(\mathbf{l}_1(u_0, v_0, R_3)), \cdots, \widetilde{\mathbf{h}}^\top_{\Sigma_3}(\mathbf{l}_N(u_0, v_0, R_3)) \quad \Big]^\top,
\end{aligned}$$

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 in $N$ values. In the experiments presented in this paper, we use $N = 8$ directions with $R_1 = 2.5$, $R_2 = 7.5$, $R_3 = 15$ and $\Sigma_1 = 2.55$, $\Sigma_2 = 7.65$, $\Sigma_3 = 12.7$. Our descriptor is therefore 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 [17]. 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, as we will show in the experiments below.

## 3.4 Empirical Evaluation

In this section, we present some of the tests we performed to compare the DAISY against SIFT and correlation windows. We used 10 real images, applied them respective transformations, and tested the



(a) contrast (b) rotation (c) scale

(d) blur (e) noise

Figure 4: Comparing SIFT(red), Daisy(green), and Correlation(blue). In the plots, the horizontal axis is the sweep range and the vertical axis is the inlier/outlier ratio. (a) Changing the contrast by replacing $I$ by $I^\gamma$, with $\gamma$ ranging from $0.1$ to $1.3$. SIFT and DAISY are unperturbed but correlation fails quickly. (b) Rotating the images from $0$ to $20$ degrees. Since SIFT is computed with rotation invariance it performs well. So does DAISY in the $\pm 15\,^\circ$ range because its circular grid and isotropic Gaussian kernels also gives it rotation invariance. (c) Scaling the image from $90\%$ to $50\%$ of their original sizes. (d) Blurring the images by using Gaussian masks of variance ranging from $0$ to $15$. (e) Adding white noise of variance ranging from $1$ to $18$.

descriptors at 600 point locations. We used a 128-point SIFT descriptor computed at a single scale with rotation invariance enabled and correlation windows of sizes $3 \times 3$, $7 \times 7$, and $15 \times 15$, always picking the one that yields the best result. We ran DAISY with 8-bin histograms, 8-angular orientations and 3-radial levels resulting in 25 regions and 200 vector size. The vertical axis of the graph in Figure 4 show the inlier percent where we tried to match untransformed image pixels with transformed ones over the whole transformed image. A match is assumed to be an inlier if the matched pixel is within $\sqrt{2}$ pixels of the correct one.

As shown in Table 1, our descriptor is much faster than SIFT. Even though it is slightly less robust, it is still much better than correlation windows.

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

Table 1: Computation Time Comparison (in seconds)

# 4 Occlusion Handling

To perform dense matching, we use DAISY to measure similarities of locations across images that we then feed to a graph-cut-based reconstruction method of [5]. To properly handle occlusions, we incorporate an occlusion map, which is the counterpart of the visibility maps in other reconstruction algorithms [13]. The reconstruction and occlusion map are estimated by EM and we present a quick formalization below.

We exploit the occlusion map to define *binary masks* over our descriptors. We use them 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 not only for proper handling of occlusions, but also to make the EM converge faster.

## 4.1 Formalization

Given a set of $n$ calibrated images of the scene, we first compute our local descriptor for each image as explained above. We denote the fields of descriptors $\mathcal{D}$ by $\mathbf{D}_{1:n}$. We then estimate the dense depth map $\mathbf{Z}$ for a given viewpoint by maximizing:

$$\zeta = p(\mathbf{Z}, \mathbf{O} \mid \mathbf{D}_{1:n}) \propto p(\mathbf{D}_{1:n} \mid \mathbf{Z}, \mathbf{O}) p(\mathbf{Z}, \mathbf{O}) \; . \tag{2}$$

We introduced an occlusion map $\mathbf{O}$ term that will be exploited below to estimate the similarities between image locations. As in [5], 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:

$$p(\mathbf{D}_{1:n} \mid \mathbf{Z}, \mathbf{O}) = \prod_{\mathbf{x}} p\left(\mathbf{D}_{1:n}(\mathbf{x}) \;\middle|\; \mathbf{Z}, \mathbf{O}\right) \; . \tag{3}$$

Each term $p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}\right)$ of Eq. 3 are estimated thanks to our descriptor. Because the descriptor considers relatively large regions, we introduce binary masks computed from the occlusion map $\mathbf{O}$ as explain below.

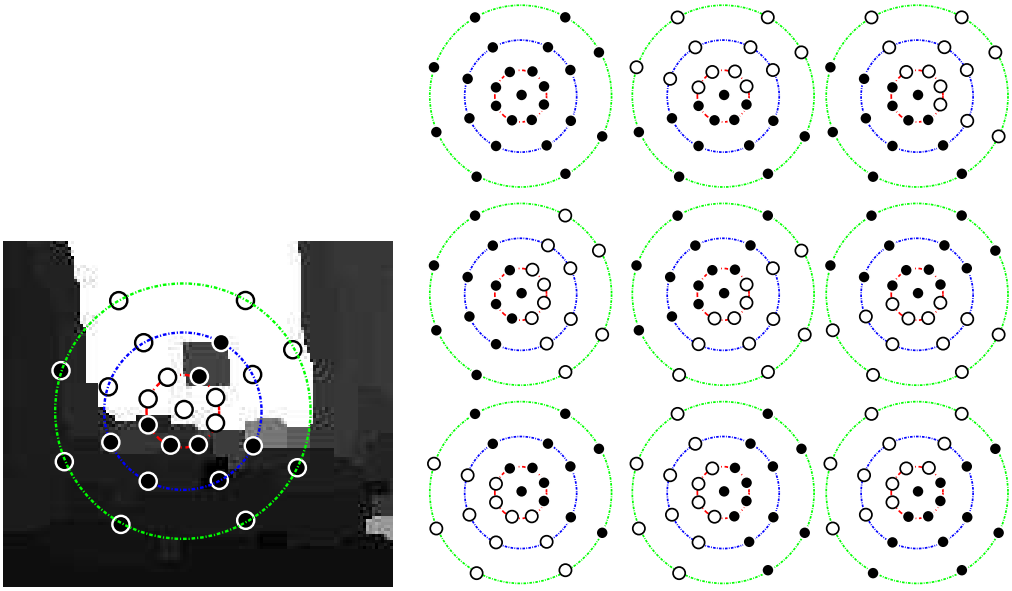
## 4.2 Using Masks over the Descriptor

Without occlusion-handling $p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}\right)$ term of Eq. 3 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 locations obtained by projecting the 3–D point $\mathbf{M}$ defined by the location $\mathbf{x}$ and the depth $\mathbf{Z}(\mathbf{x})$ in the virtual view in 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 $\{\mathcal{M}_m(\mathbf{x})\}$ as the ones depicted in Fig. 5 that allow to take into account only the visible parts when computing the distances between descriptors. Our descriptor being built from 25 locations, these binary masks are defined as 25-binary vectors.

We want the masks to depend on the current estimate for the occlusion map $\mathbf{O}$, and we tried three different strategies: The simplest one depicted by Fig. 5(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 $\mathcal{M}_k(\mathbf{x})$.



(a) (b)

Figure 5: 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 1 and a white disks to 0. (a) We use the occlusion map to define the masks; however, considering only predefined masks (b) makes easy it to enforce their spatial coherence and to speed-up the convergence of EM estimation.

The two other strategies use the predefined masks depicted by Fig. 5(b) that have a high spatial coherence. In the second strategy, each mask has a different probability estimated by considering the average visible pixel number, $\overline{v}_m$, and depth variance within the mask region, $\sigma_m(\mathbf{Z})$:

$$p(\mathcal{M}_m(\mathbf{x})|\mathbf{Z},\mathbf{O}) = \frac{1}{Y}\left(\overline{v}_m + \frac{1}{\sigma_m^2(\mathbf{Z})+1}\right) \tag{4}$$

where $Y$ is a normalization factor. The last strategy is a more radical version of the second strategy, where we set the probability of the mask with the highest value according to Eq. 4 to 1 and others to 0.

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

$$\begin{aligned} &p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z},\mathbf{O}\right) = \\ &\textstyle\sum_m p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z},\mathbf{O},\mathcal{M}_m(\mathbf{x})\right) p\left(\mathcal{M}_m(\mathbf{x}) \mid \mathbf{Z},\mathbf{O}\right) \ . \end{aligned} \tag{5}$$

In the first and third strategies, only a single mask has a probability $p\left(\mathcal{M}_m(\mathbf{x}) \mid \mathbf{Z},\mathbf{O}\right)$ equal to 1, all the other masks receive a null probability. In the second strategy, Eq. 5 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 always resulted in a much faster convergence towards a satisfying solution, and therefore, we use this one only. These good performances over the other strategies can be explained by the fact that the chosen masks allow to enforce the spatial consistency when comparing the descriptors.

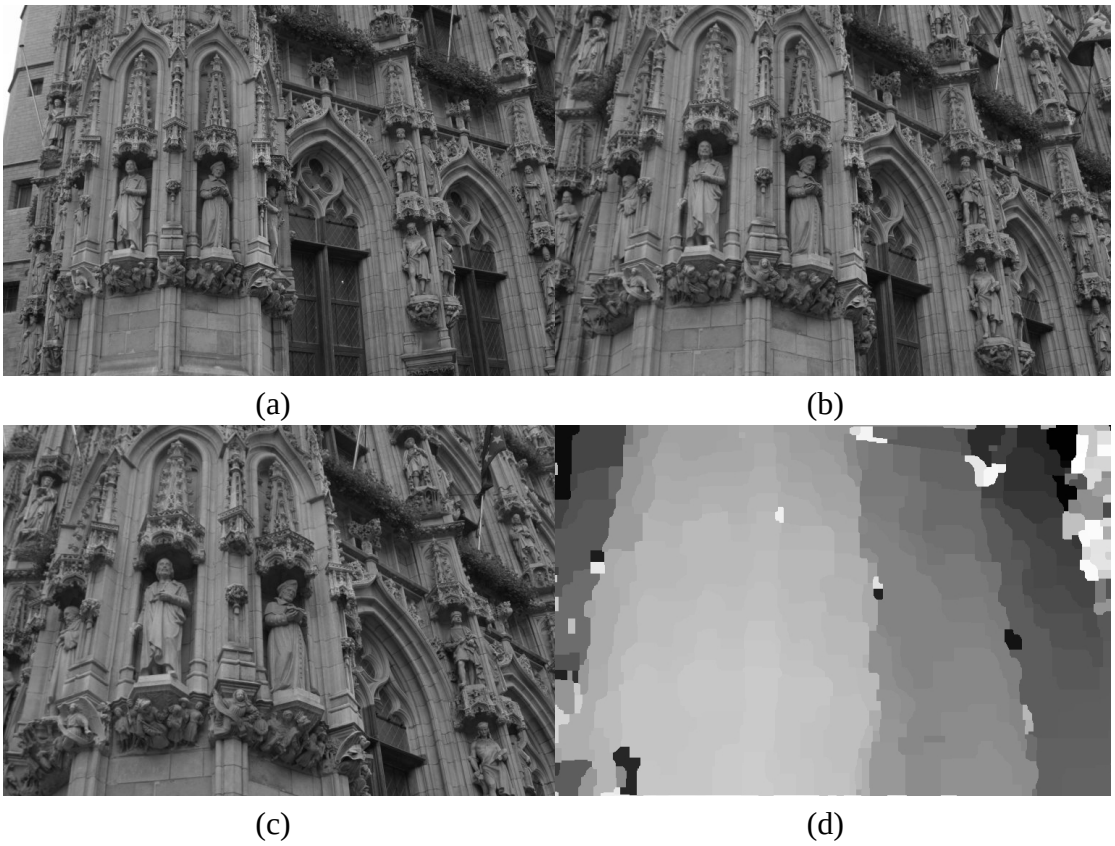


Figure 6: Results on low-resolution versions of the Rathaus images [24]. (a,b,c) Three input images of size $768 \times 512$ instead of $3072 \times 2048$. (d) Depth map computed using all three.

Finally, following [5], the term $p\left(\mathbf{D}_{1:n}(\mathbf{x}) \mid \mathbf{Z}, \mathbf{O}\right)$ of Eq. 3 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

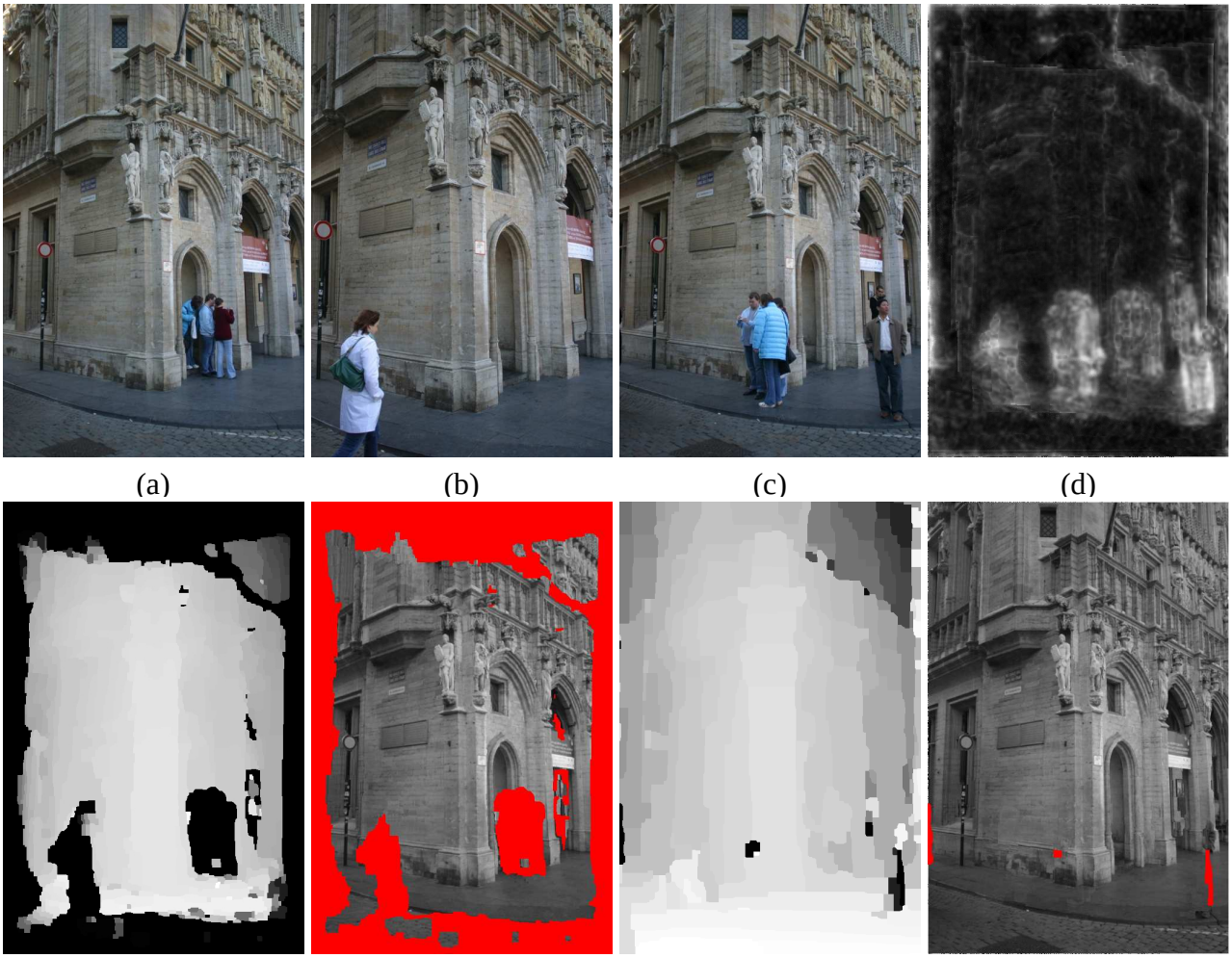
$$\frac{2(n-2)!}{n!} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sqrt{\sum_{k=1}^{25} \frac{\mathcal{M}^{[k]} \left\| \mathbf{D}_i^{[k]}(\mathbf{x}) - \mathbf{D}_j^{[k]}(\mathbf{x}) \right\|^2}{\sum_{q=1}^{25} \mathcal{M}^{[q]}}}, \tag{6}$$

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 our method to Strecha's [22], we ran our algorithm on two sets of his images, the Rathaus sequence of Fig 6 and the Brussels sequence of Fig 7. In his work, Strecha used very high resolution $3072 \times 2048$ images, which helps a lot because, at that resolution, even the apparently blank areas exhibit usable texture. For example, on the walls, the irregularities in the stone provide enough information for matching. Unfortunately, such high-resolution imagery is not always available and we show here that DAISY produces comparable results using much reduced $768 \times 512$ images.

Fig. 7 also highlights our effective occlusion handling. When using only two images, the parts of the church that are hidden by people in one image and not 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.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: Using low-resolution versions of the Brussels images [23]. (a,b,c) Three $768 \times 510$ versions of the original $2048 \times 1360$ images. (e,f) 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 he other are correctly marked as occlusions. (g,h) 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.

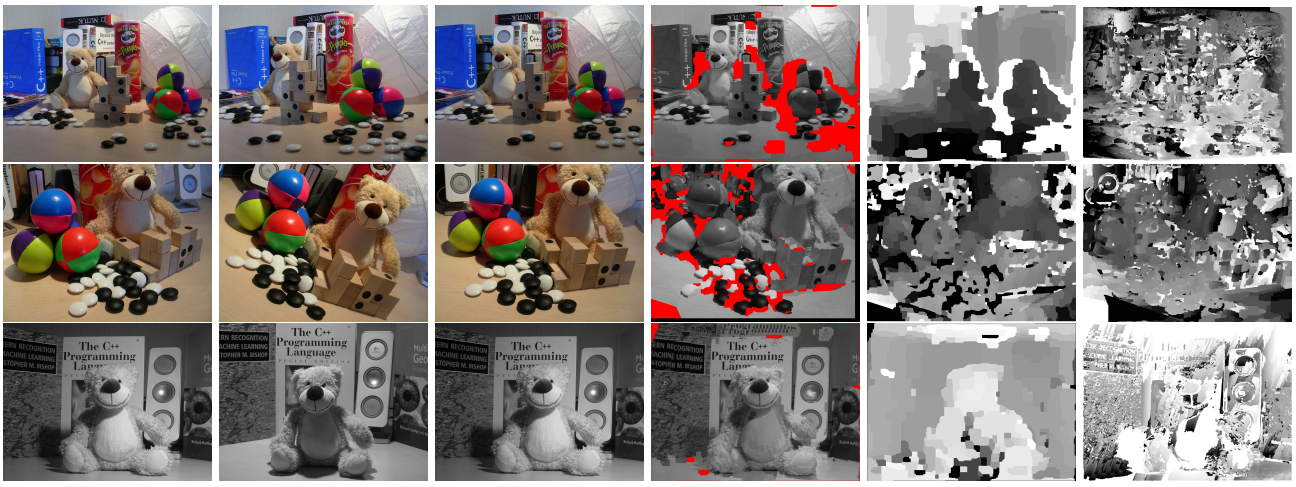


Figure 8: Depth maps and resynthesized images. In each row, the first two images are the inputs to our stereo-matcher. The third one is not used to compute the depth but only to validate the quality of the fourth one, which is synthesized from the first two using the DAISY depth map shown in fifth position. The final image is the depth map computed using correlation. The occluded areas are overlaid in red in the synthesized images and they're black in the depth-maps.

In Fig. 8 we show more disparity maps computed from stereo pairs whose baseline is large enough for standard correlation-based techniques fail and that exhibit substantial occlusions and lighting changes. Our depth maps are correct—except at the detected location of occlusions where there is no depth—as evidenced by the fact that we can use them 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. In the first row, we expected to have a descent result from the correlation approach. However, when we inspected the input images more closely, we noticed that the size of the objects in two input images changes due to the rotation of the camera and this small change plus the amount of low-textured regions becomes enough to disrupt correlation. The most successful result for the correlation window is achieved for image set 2. However, even in this case there are problems on the torso of the teddy bear and constant intensity regions like the lego blocks. In the $3rd$ row, there is a significant light change in the input images and despite this, DAISY manages to find a satisfactory result while correlation fails.

# 6 Conclusion

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

Although we do not explicitly handle scale and rotation invariance, DAISY retains good invariance properties against these transformations, as well as contrast change, blur, and additive noise. It therefore allows matching with baselines significantly greater than correlation-based techniques. In future work, we will address the scale and rotation issue more thoroughly so that we can work with even wider baselines.

# 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. *Journal of Visual Communication and Image Representation*, 13(1/2):3–21, Mar 2002.

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

[3] H. Baker and T. Binford. Depth from edge and intensity based stereo. volume 2, pages 631–636, Aug. 1981.

[4] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. 20(4):401–406, Apr. 1998.

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

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

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

[8] O. Faugeras and R. Keriven. Complete Dense Stereovision using Level Set Methods. Freiburg, Germany, June 1998.

[9] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. 2005.

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

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

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

[13] V. Kolmogorov and R. Zabih. Multi-Camera Scene Reconstruction via Graph Cuts. Copenhagen, Denmark, May 2002.

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

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

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

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

[18] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, volume I, pages 69–81, 2004.

[19] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. volume 1, pages 829–836, 2005.

[20] S. Roy and I. Cox. A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem. pages 492–499, Bombay, India, 1988.

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

[22] C. Strecha, R. Fransens, and L. V. Gool. Wide-baseline stereo from multiple views: a probabilistic account. volume 2, pages 552–559, 2004.

[23] C. Strecha, R. Fransens, and L. V. Gool. Combined Depth and Outlier Estimation in Multi-View Stereo. 2006.

[24] C. Strecha, T. Tuytelaars, and L. V. Gool. Dense Matching of Multiple Wide-Baseline Views. 2003.

[25] T. Tuytelaars and L. VanGool. Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions. pages 412–422, 2000.

[26] J. Yao and W.-K. Cham. 3–D Modeling and Rendering from Multiple Wide-Baseline Images. *Signal Processing: Image Communication*, 21:506–518, 2006.