Journal of
Electronic Imaging

JElectronicImaging.org

Binary descriptor-based dense line-scan stereo matching

Kristián Valentín
Reinhold Huber-Mörk
Svorad Štolc

SPIE.

IS&T
imaging.org

Kristián Valentín, Reinhold Huber-Mörk, Svorad Štolc, “Binary descriptor-based dense line-scan stereo matching,” *J. Electron. Imaging* **26**(1), 013004 (2017), doi: 10.1117/1.JEI.26.1.013004.

Binary descriptor-based dense line-scan stereo matching

Kristián Valentín, Reinhold Huber-Mörk, and Svorad Štolc*
AIT Austrian Institute of Technology GmbH, Intelligent Vision Systems, Digital Safety and Security Department, Donau-City-Straße 1,
1220 Vienna, Austria

Abstract. We present a line-scan stereo system and descriptor-based dense stereo matching for high-performance vision applications. The stochastic binary local descriptor (STABLE) descriptor is a local binary descriptor that builds upon the principles of compressed sensing theory. The most important properties of STABLE are the independence of the descriptor length from the matching window size and the possibility that more than one pair of pixels contributes to a single-descriptor bit. Individual descriptor bits are computed by comparing image intensities over pairs of balanced random subsets of pixels chosen from the whole described area. On a synthetic as well as real-world examples, we demonstrate that STABLE provides competitive or superior performance than other state-of-the-art local binary descriptors in the task of dense stereo matching. The real-world example is derived from line-scan binocular stereo imaging, i.e., two line-scan cameras are observing the same object line and 2-D images are generated due to relative motion. We show that STABLE performs significantly better than the census transform and local binary patterns (LBP) in all considered geometric and radiometric distortion categories to be expected in practical applications of stereo vision. Moreover, we show as well that STABLE provides comparable or better matching quality than the binary robust-independent elementary features descriptor. The low computational complexity and flexible memory footprint make STABLE well suited for most hardware architectures. We present quantitative results based on the Middlebury stereo dataset as well as illustrative results for road surface reconstruction. [DOI: 10.1117/1.JEI.26.1.013004]

Keywords: stereo vision; image descriptor; line-scan.

Paper 16714P received Aug. 22, 2016; accepted for publication Dec. 19, 2016; published online Jan. 10, 2017.

1 Introduction

Range information from images is typically obtained using time-of-flight sensors,[1] configurations based on pattern projection,[2] illumination variation by photometric stereo,[3] focus variation,[4] multicamera systems,[5] or light field cameras.[6] Line scanning is a popular method for acquiring images of moving objects, especially in machine vision applications. From moving platforms, such as air- or spaceborne scanners, the so-called pushbroom principle is used to acquire sensor lines while moving along a predefined trajectory in space. We utilize this acquisition principle, extended to binocular stereo, for an application in ground reconstruction from a vehicular platform. The application area is the inspection of road surface conditions. Figure 1 shows a few examples of single road images as acquired by the proposed system. We will describe how to obtain depth information from line-scan stereo pairs, e.g., pairs of images taken concurrently from slightly displaced positions.

In stereo imaging, the range for each pixel is obtained from the estimated disparity, i.e., the displacement between corresponding points observed in two (or more) images. The epipolar constraint in a stereo vision system states that a point in one image is found along the corresponding epipolar line in the other image. Epipolar rectification in area-scan stereo pairs aligns epipolar lines to image lines, thus reducing the correspondence estimation to a search over an expected disparity range oriented along image lines. In the presented line-scan stereo system, one adjusts this geometrical constraint mechanically such that epipolar lines are aligned with sensor lines. Estimation of disparities is then performed along sensor lines.

We will discuss the stochastic binary local descriptor (STABLE) for disparity estimation.[7] Since the introduction of the scale invariant feature transform (SIFT), a number of feature detectors and descriptors were suggested over the last decades.[8] Among others, the goal of speeding up SIFT was met in speeded up robust features (SURF).[9] Some representations of local derivatives, e.g., gradient orientation histograms, are commonly used in those descriptors. Higher speed is sometimes also traded against reduced invariance properties, e.g., in binary robust-independent elementary features (BRIEF).[10] Efficient representations and fast matching are obtained by the family of binary descriptors. Oriented BRIEF (ORB) is an alternative to SIFT and SURF that is based on a binary description.[11]

STABLE belongs to a broad class of local binary descriptors, along with the census transform (CT),[12] local binary patterns (LBP),[13] BRIEF,[10] binary robust invariant scalable keypoints (BRISK),[14] or fast retina keypoints (FREAK).[15] The most similar descriptor to STABLE is BRIEF;[10] the main difference lies in the ability of STABLE to have more than one pair of pixels contributing to a single descriptor bit. In general, binary descriptors are known to be robust against intensity variations as relative pixel intensity comparisons are used in descriptor construction followed by bitstring matching, especially when compared to direct intensity comparison using sums of absolute or squared intensity differences. Furthermore, a higher speed could be expected from simple comparison operations.

*Address all correspondence to: Svorad Štolc, E-mail: svorad.stolc@ait.ac.at

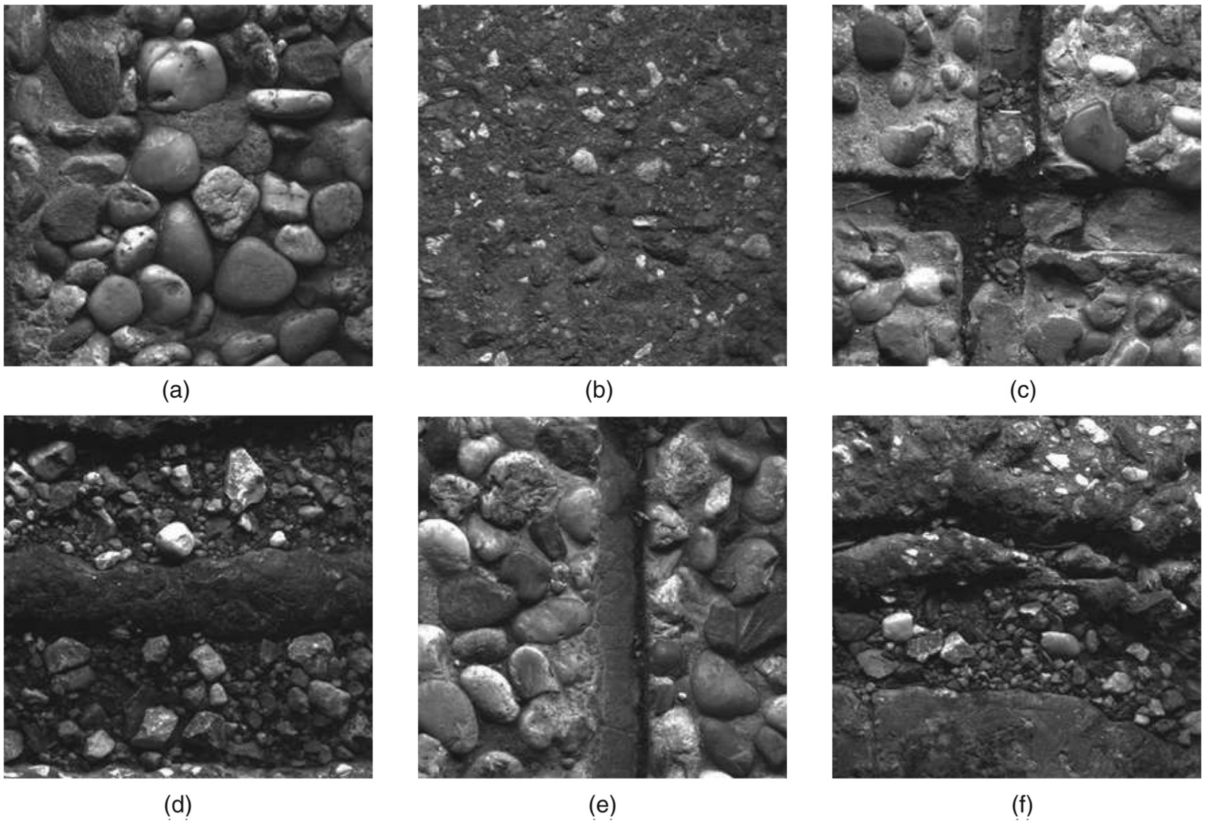


Fig. 1 Samples of ground surface images: (a) washed concrete, (b) coarse asphalt, (c) surface with joints, (d) mixed asphalt and stones, (e) surface with joints, and (f) mixed surface.

STABLE can be related to the principle of "compressed sampling."[16] The compressed sampling theory claims that each signal with a sparse representation in some (potentially unknown) linear basis can be preserved and reconstructed from a small number of random projections. For natural images, this means that, due to the sparsity of image edges and inherent smoothness, it is sufficient to sample the image in a compressive manner without losing any significant information. While the reconstruction is not the main focus in our application, we exploit the principles of compressed solely sampling for deriving an efficient binary representation of any given pattern, i.e., for encoding the pattern into a constant number of bits that is greatly independent from the pattern's size.

2 Image Acquisition

We used two line-scan cameras sensitive in the visible spectrum for stereo acquisition of the road surface while the acquisition device was moving. The surface could be acquired using either one long image sensor line shared by two lenses or two collinearly arranged line-scan image sensors observing the same surface line patch. Figure 2(a) shows the selected setup using two collinearly arranged line-scan sensors observing the surface from two different viewpoints. The optical axes are verged to obtain a larger overlapping region. Some details on the geometrical setup are as follows: a baseline of ∼220 mm, distance to the ground of ∼480 mm, verging of the cameras of $\sim \pm 11$ deg wrt the ground surface normal, and field of view of the camera lens of ∼11.7 deg.

General design principles were that car driving speeds up to 80 km/h should be possible at lateral resolution on the order of magnitude of 0.1 mm/pixel. The used cameras were able to achieve the required line rates of >200,000 lines/s. Regarding the field of view, it was sufficient to cover a small stripe only, i.e., in the center of the region where car tires usually interact with the surface, which made it possible to restrict the line length to 1000 pixels.

Verging of the optical axis has two drawbacks. First, the object resolution decreases from left to right in one view and from right to left in the other view. Second, the limited depth of field might result in sharpness reduction depending on optical parameters and adjustment when compared to a canonical stereo system. Geometric calibration of the sensor lines ensures a constant object pixel size at the regular working distance for planar surfaces.

The depth of field was estimated to be on the order of magnitude of ±6.16 mm for a f-number of 5.6, a magnification of 0.1, and a sensor pixel size of 10 μm. For f-numbers of 1.4 or 2.8, we would obtain a depth of field of ±1.54 mm or ±3.08 mm, respectively. Although these are quite low numbers, it turned out to be sufficient to compensate for the varying distance due to verging and the expected depth variation in road inspection.

The purpose of calibration is the alignment of the sensors lines to ensure that the plane spanned by the left optical axis and left sensor line is coplanar with the plane spanned by the right optical axis and right sensor line. This property is important to fulfill the epipolar constraint at each depth and requires a calibration procedure that ensures collinearity of

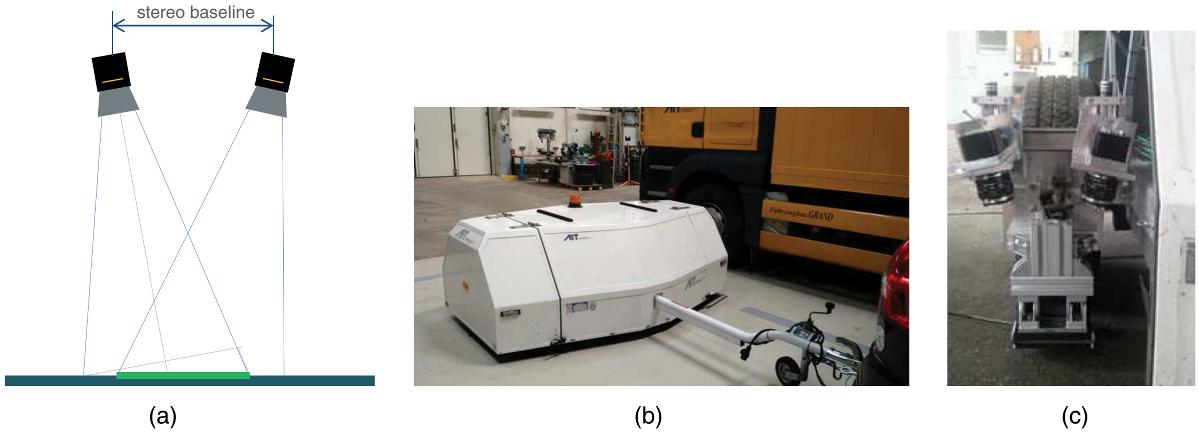


Fig. 2 Binocular stereo image acquisition: (a) sketch of verged stereo geometry for two collinearly arranged line-scan sensors, (b) car trailer carrying the imaging devices, and (c) cameras mounted in trailer.

the sensor lines at a number of distances. To facilitate this requirement, one has to ensure the collinearity of the sensors at least for two different distances. Using a calibration target similar to the one suggested by Luna et al.[17] where target patterns are present at parallel planes at different distances, one is able to determine the camera pose, including the epipolar plane orientation, of a single line-scan camera. A similar calibration target is required for line-scan stereo calibration. In our case, the concurrent mechanical adjustment of both sensor lines ensures the observation of corresponding patterns at different distances and for both cameras, i.e., the epilpolar planes are the same for both sensors. Nevertheless, residual misalignment and vibrations of the system might result in problems during stereo matching. We suggest an additional correspondence search between lines adjacent to the concurrently taken sensor lines.

3 Stereo Image Processing

To obtain depth information from stereo image pairs, corresponding points need to be found. Corresponding points are typically identified via block matching, i.e., comparison of image patches between image pairs. Measures of block similarity include direct comparison of pixel intensities using similarity metrics such as the sum of absolute differences, the sum of squared errors, the normalized cross correlation, and comparison based on measuring some distance between block feature descriptors. While for descriptors, such as SURF or SIFT, vector metrics in high-dimensional spaces are commonly used to quantify descriptor similarity; for binary descriptors, the Hamming distance is applied in most cases.

3.1 *Local Binary Descriptors*

In general, binary descriptors have been used for tasks like texture analysis, recognition, and matching, e.g., LBP[13,18] and the CT.[12] In the context of local descriptors, several fast binary descriptors were also developed recently, e.g., BRIEF,[10] BRISK,[14] FREAK,[15] etc. In our experiments, we considered the center-based descriptors CENSUS and LBP, where center-based refers to the fact that pairwise comparison always involves the central pixel, and the uncentered descriptors BRIEF and STABLE. The main difference in binary descriptors is in the sampling pattern for local intensity comparisons, which results in a binary descriptor vector. The CENSUS-dense descriptor is the only descriptor utilizing exactly all pixels in the considered matching window. We alternatively investigate the CENSUS-sparse descriptor, which uses a subsample of off-center pixels on a regular grid and compares those against the central pixel. The BRIEF descriptor uses a subsample of pixel pairs (typically sparsely) located at arbitrary positions in the matching window. The resulting descriptor lengths equal the number of pixel pair comparisons performed. Finally, with STABLE, we also get pixel pairs at random positions, but we are able to map a larger number of pixel pairs to a smaller number of descriptor bits. Figure 5 shows the compared descriptor masks (the meaning of the numbers in the mask will be explained in the next section).

3.2 *STABLE Descriptor*

We consider an image patch $\mathbf{p}$ of size $X \times Y$ pixels. The operation β derives the ith descriptor bit $d_i \in \mathbf{d}$ from patch $\mathbf{p}$ as follows:

$$\beta(\mathbf{p}, i) = \begin{cases} 1 & \text{if } (\mathbf{p} * f_i) > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where f_i is a filter mask of equal size as the image patch $\mathbf{p}$. We refer to the operation β as the binarized convolution. The filter dictionary $\mathbf{f}$ contains K sparse filter masks f_i. Each entry in f_i is either 0, 1, or -1. The descriptor $\mathbf{d}$ is a K-dimensional bitmask, which is obtained for a given image patch $\mathbf{p}$ using

$$\mathbf{d}(\mathbf{p}) = \sum_{i=1}^{K} 2^{i-1} \beta(\mathbf{p}, i). \tag{2}$$

Figure 3 shows this operation schematically. A set of sparse filter masks from a dictionary are applied to the same image patch and, depending on the number and individual signs of the filter mask entries, a number of pixels is contributing to each descriptor bit.

A more efficient implementation of STABLE, avoiding binarized convolutions with K sparse feature filters, uses a single index filter mask $\mathbf{g}$. This mask $\mathbf{g}$ is of the same size as the image patch $\mathbf{p}$ and encodes at nonzero pixel positions

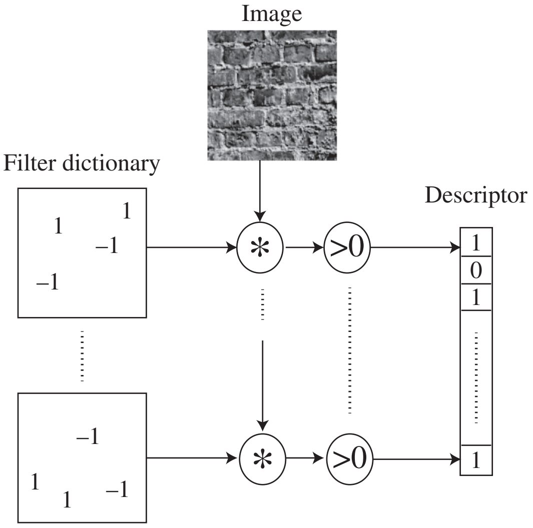


Fig. 3 Operation of the STABLE descriptor: sparse filters from a dictionary where each filter mask mostly consists of entries of 0, other entries {−1,1} are randomly distributed. An image patch is convolved with each filter mask, and the result is thresholded (binarized convolution) and inserted into descriptor bits.

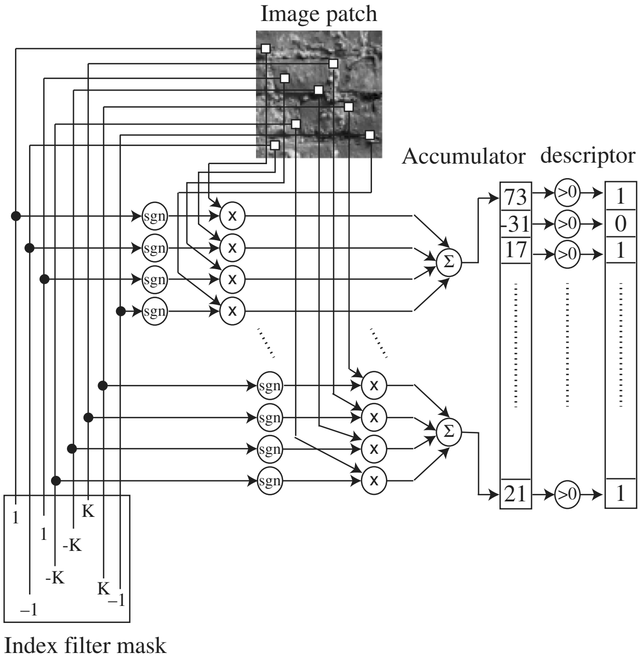


Fig. 4 Efficient implementation of the STABLE descriptor: an index filter mask contains pixel indices and signs. An image patch is accessed using this mask, and a signed sum is inserted into an accumulator array. The descriptor is finally obtained by binarization of the accumulator entries.

the position in the descriptor array $\mathbf{d}$ and a sign. An accumulator array a of size K is used to perform a sign-dependent accumulation in cell i of the pixel values in $\mathbf{p}$ with corresponding filter mask index $|i|, i = 1, \dots, K$. After all accumulators cells are processed, the descriptor $\mathbf{d}$ is derived by thresholding each cell entry of a. The improved operation involving the filter index mask $\mathbf{g}$ instead of the filter dictionary $\mathbf{d}$ is shown in Fig. 4.

The concept of filter masks is also applicable to other binary descriptors, e.g., Fig. 5 shows filter masks corresponding to CENSUS-dense, CENSUS-sparse, LBP, BRIEF, and STABLE. The number in each cell refers to which bit a pixels contributes. The sign indicates whether the pixel value is taken as is $(+1)$ or if it is negated (-1) when using the accumulator-based implementation scheme. The center-based descriptors in Figs. 5(a) to 5(c) utilize the central pixel for each descriptor bit, which is indicated by $-*$.

3.3 *Stereo Matching*

In stereo matching, we consider a discrete range of disparities $[r_1, \dots, r_P]$ for which the descriptors, corresponding to each image pixel position, are compared by the Hamming distance. This results in a cost stack C of dimension $M \times N \times P$, where M and N are the image dimensions and P is the number of evaluated disparities. The cost stack C is searched for by the minimum cost at each pixel, which provides the associated disparity estimation. The cost stack C is then filtered with a 1×3 Gaussian kernel in the cost domain followed by filtering with a 3×3 Gaussian kernel in the image domain. Finally, to efficiently obtain a subpixel accuracy disparity map, a parabola is fitted to three values around the initial integer disparity estimation, a procedure commonly applied in image processing.[19]

4 Results

We will present results on synthetically disturbed data to estimate robustness of STABLE in comparison to other binary descriptors. We also provide detailed comparison to BRIEF, the most similar approach to STABLE, based on stereo matching performance on the Middlebury stereo dataset. Furthermore, we provide illustrative examples on real-world data of freeway road surface. Finally, we provide run-time measurements on GPU platform.

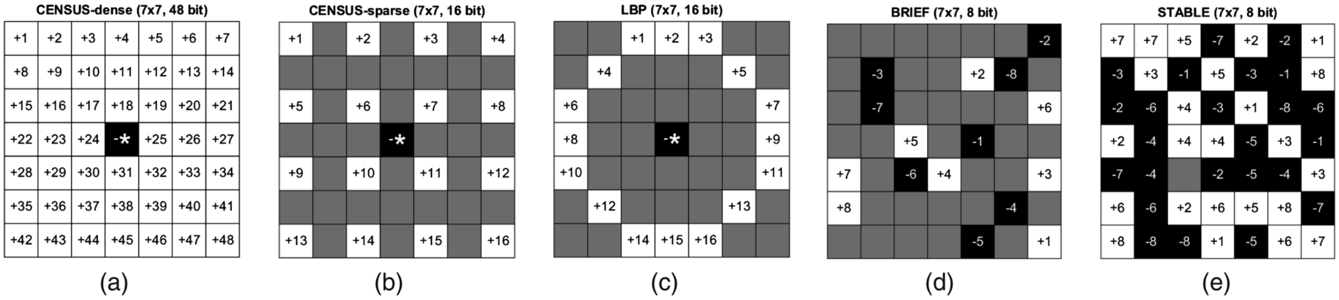
4.1 *Synthetic Data*

To evaluate performance of the STABLE descriptor compared with other state-of-the-art local binary descriptors, we employed a similar evaluation scheme as suggested by Mikolajczyk and Schmid[20] based on the analysis of receiver operator characteristic (ROC) curves. We extracted 1200 grayscale patterns from 48 natural images contained in the data set introduced in Ref. 20, always 25 patterns per image at random locations. Given the perturbation type, for each pattern, we introduced 25 synthetic perturbations, which gave a total number of 30,000 patches. In this study, we considered five different types of perturbations:

- Gaussian additive noise ($\sigma \leq -20$ dB);
- Gaussian blur ($\sigma \leq 4$ px);
- shift in random direction (≤ 3 px);
- scaling ($\leq \pm$ 10%); and
- rotation ($\leq \pm$ 10 deg).

We considered matching windows of size 15×15 pixels.

Given the set of 30,000 patches defined for each perturbation type, there is always a group of 25 associated perturbed versions for each patch in the data set. Making every patch a query, one can assess its Hamming distance to all patches in the data set making use of a particular feature descriptor. Knowing that for each query there are only 25



CENSUS-dense (7x7, 48 bit)

+1	+2	+3	+4	+5	+6	+7
+8	+9	+10	+11	+12	+13	+14
+15	+16	+17	+18	+19	+20	+21
+22	+23	+24	-*	+25	+26	+27
+28	+29	+30	+31	+32	+33	+34
+35	+36	+37	+38	+39	+40	+41
+42	+43	+44	+45	+46	+47	+48

(a)

CENSUS-sparse (7x7, 16 bit)

+1		+2		+3		+4
+5		+6		+7		+8
			-*			
+9		+10		+11		+12
+13		+14		+15		+16

(b)

LBP (7x7, 16 bit)

		+1	+2	+3		
	+4				+5	
+6						+7
+8			-*			+9
+10						+11
	+12				+13	
		+14	+15	+16		

(c)

BRIEF (7x7, 8 bit)

						-2
	-3			+2	-8	
	-7					+6
		+5		-1		
+7		-6	+4			+3
+8					-4	
				-5		+1

(d)

STABLE (7x7, 8 bit)

+7	+7	+5	-7	+2	-2	+1
-3	+3	-1	+5	-3	-1	+8
-2	-6	+4	-3	+1	-8	-6
+2	-4	+4	+4	-5	+3	-1
-7	-4		-2	-5	-4	+3
+6	-6	+2	+6	+5	+8	-7
+8	-8	-8	+1	-5	+6	+7

(e)

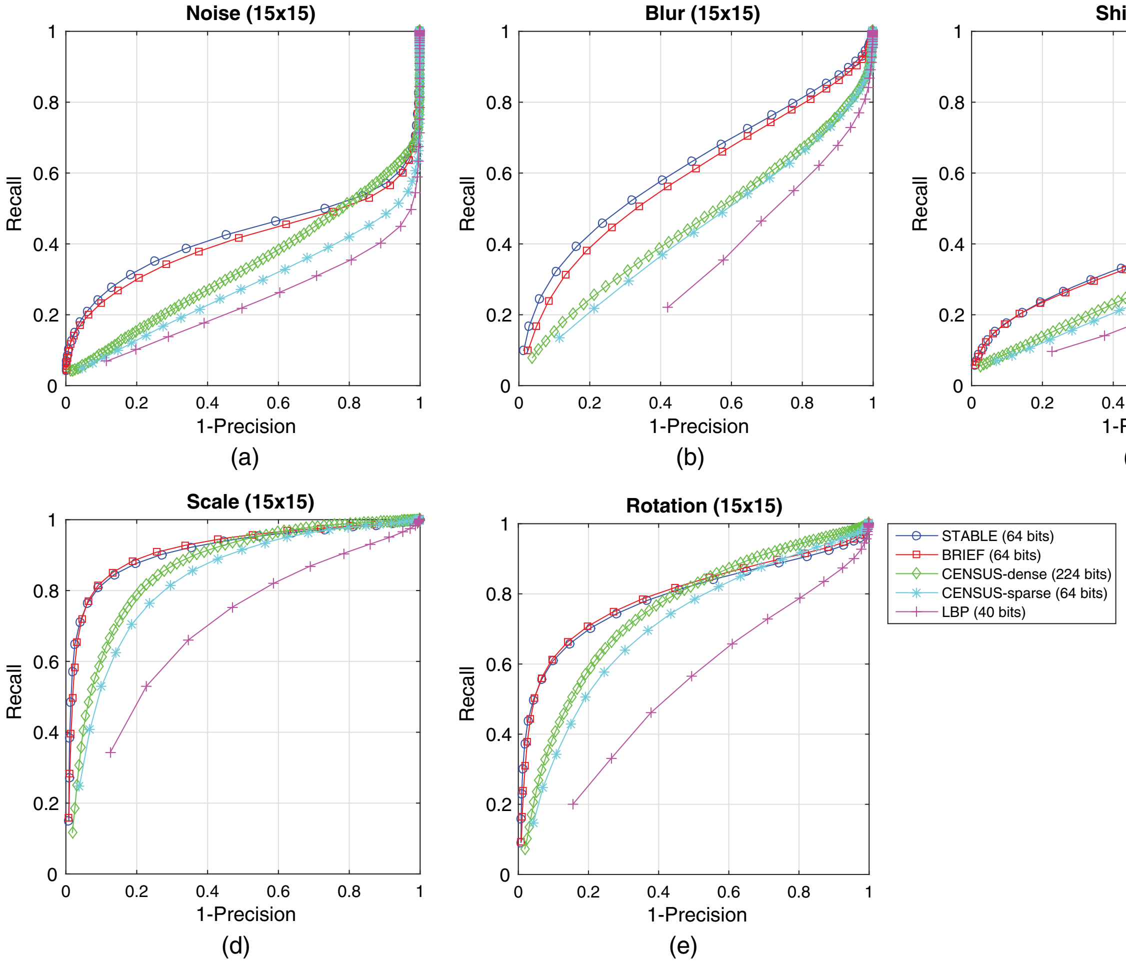
Fig. 5 Examples of index filter masks of different binary feature descriptors defined on the 7 × 7 pixel matching window: (a) CENSUS-dense, (b) CENSUS-sparse, (c) LBP, (d) BRIEF, (e) STABLE.

relevant elements, one can calculate the precision and recall values for all result sets associated with different thresholds put on the Hamming distance. The ROC curve is then defined by the obtained precision and recall values.

In total, we compared five local binary descriptors:

- CENSUS-dense;
- CENSUS-sparse;
- LBP;
- BRIEF; and
- STABLE.

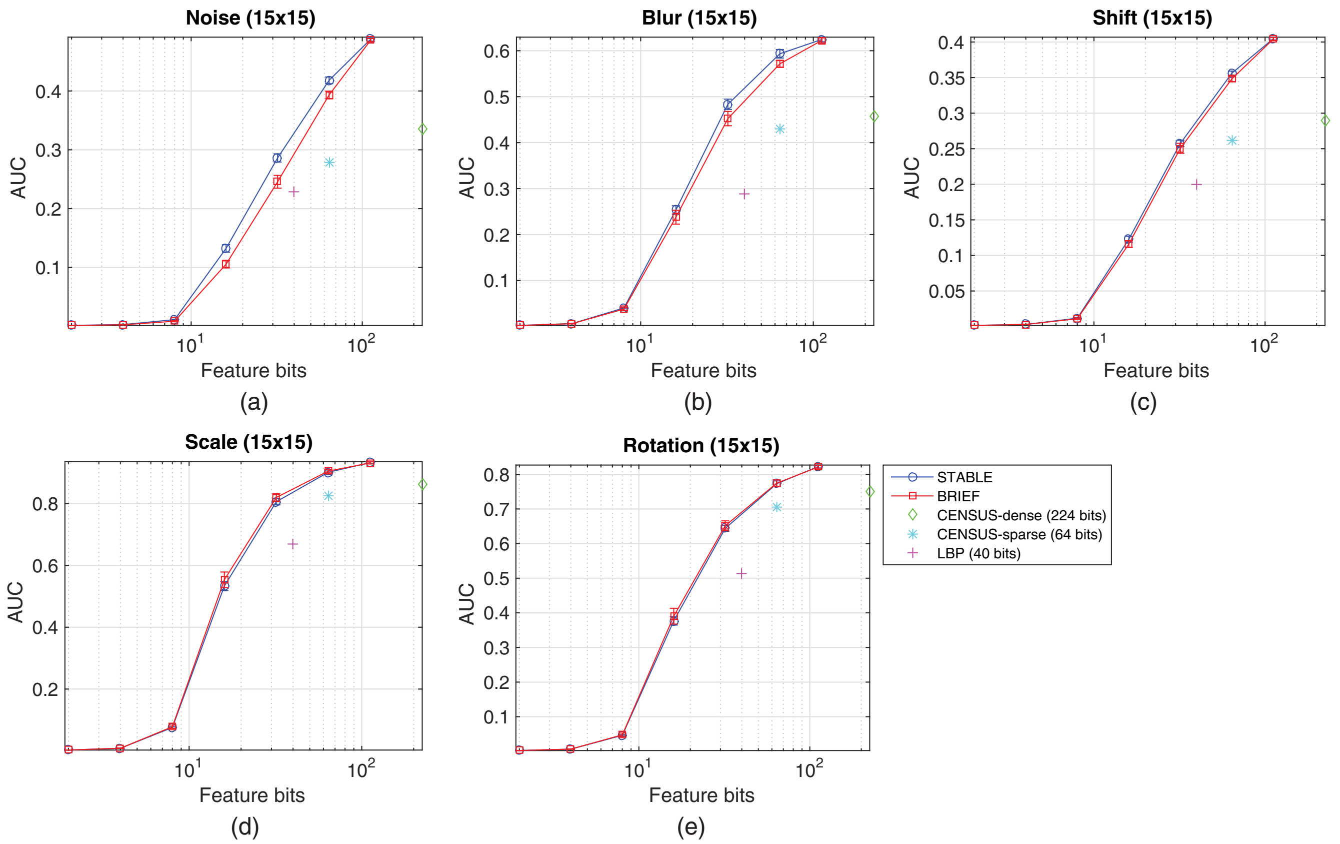
While for CENSUS and LBP, the descriptor size depends on the matching window, in the case of STABLE and BRIEF, the number of feature bits is defined independently from the matching window. Thus, we also looked into the relationship between matching performance, expressed in terms of the area under the ROC curve (AUC), and the descriptor size in bits. Furthermore, as both of these descriptors are generated stochastically, their performance was assessed as the average and standard deviation over 25 trials with different randomly generated filter masks. We believe this should provide a clear picture about the typical performance and stability of those stochastic descriptors.



(a) (b) (c)

(d) (e)

Fig. 6 ROC curves obtained for different perturbation types. In the case of STABLE and BRIEF, the provided ROC curves represent the best performance case over 25 random trials, i.e., the one with the highest AUC value: (a) Gaussian noise, (b) blurring, (c) shifting, (d) scaling, (e) rotation.



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

Fig. 7 Relationship between matching performance of different feature descriptors and the number of feature bits. Each point on a curve stands for the average AUC value over 25 random trials, while the whiskers show the corresponding standard deviation: (a) Gaussian noise, (b) blurring, (c) shifting, (d) scaling, (e) rotation.

Figure 6 shows the recognition performance obtained by different feature descriptors for a constant configuration of the descriptor size. Going from the worst to the best performing descriptors, it can be seen that the LBP provides the overall worst performance for all perturbation types. It is then followed by CENSUS-sparse and CENSUS-dense, both of which provide comparable performance despite their very different numbers of feature bits. For most perturbation types, it is then followed by BRIEF and finally by STABLE (notice the curve with circles exceeds all the other curves in most cases).

In Fig. 7, the matching performance is analyzed in relationship with the descriptor size. All descriptors with a constant number of bits are marked as points, whereas all the others are represented as curves. In this analysis, it is even more pronounced that the performance of the both CENSUS descriptors and LBP is significantly worse than for STABLE and BRIEF at the respective bit counts. In the case of noise, blur, and shift perturbations, the STABLE descriptor outperforms the BRIEF descriptor, especially for medium numbers of feature bits.

The performance of STABLE versus BRIEF is documented in detail in Fig. 8. The advantage of STABLE over BRIEF is expressed in terms of the recognition performance gain defined as a ratio between AUC values obtained by both descriptors using the same numbers of bits. It follows that AUC ratios above one mark the cases where STABLE outperformed BRIEF and vice versa. It is apparent that the advantage of STABLE is mostly pronounced for medium bit counts, while with the increasing size of the descriptor, the difference is getting smaller as both descriptors become more similar to each other. It should be noted that at the maximum possible number of bits, both descriptors are in fact the same where each bit is generated by just a pair of pixels. There are two cases in which STABLE significantly outperformed BRIEF, namely perturbations by (i) the additive noise and (ii) the blur. In the case of additive noise, a performance gain as large as 30% was obtained with 8-bit descriptors. For blur and shift perturbations, the highest AUC ratios exceeding 5% were obtained for 32-bit and 8-bit descriptors, respectively. For scale and rotation perturbations, STABLE performs generally slightly worse than BRIEF; however, the worst performance loss is still well below 5%.

4.2 *Stereo Matching on Middlebury Stereo Dataset*

We assessed the dense stereo reconstruction performance of STABLE versus BRIEF on real-world data. We used 10 evaluation training sets with disparity ground truth from the Middlebury Stereo Datasets 2014.[21] For both STABLE and BRIEF, we used windows of size of 15 × 15 and descriptor length of 8, 16, 32, and 64 bits. The left view served as the reference view.

As the error metric, we used the percentage of pixels with absolute disparity error greater than 2.0 (dubbed as bad 2.0). We did not include occluded pixels. For each of the 10 datasets, we performed 25 runs (each run with a different index mask for both descriptors) and recorded the best and average values for each metric. Performance gain of STABLE relative to BRIEF averaged over all datasets is shown in Fig. 9.

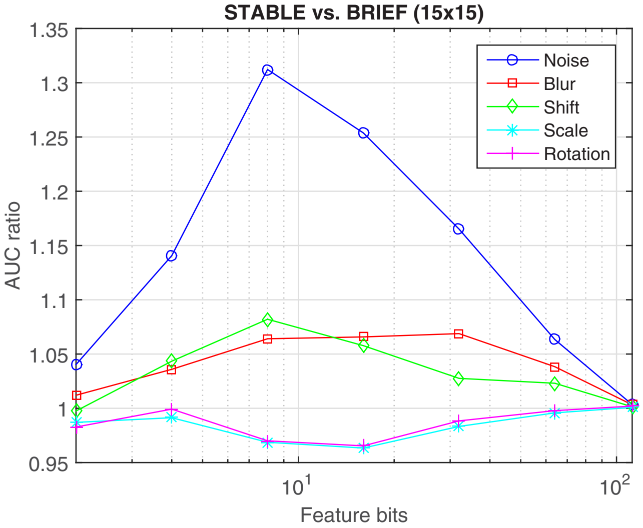


Fig. 8 Recognition performance gain (i.e., the ratio between AUC values) of STABLE over BRIEF.

Results in Fig. 9 show that both for average and best cases and for all tested bit lengths, STABLE outperforms BRIEF. The largest performance gain (4.33% in bad 2.0) was measured for the length of 32 bits. Performance gain for 8 and 64 bits is significantly lower for both metrics. For illustration, Figs. 10(a) to 10(d) show the example where STABLE outperformed BRIEF the most and Figs. 10(e) to 10(h) show the example where STABLE was least superior to BRIEF. The green areas in the difference images in Figs. 10(c) and 10(g) depict areas where STABLE gained a better bad 2.0 score when compared to BRIEF, whereas magenta refers to a better bad 2.0 score for BRIEF. Black to white areas indicate that both methods obtained very similar bad 2.0 errors.

4.3 *Road Surface Data*

In this section, we present results on real world data acquired by driving our system on a freeway. First, we compare STABLE and CENSUS-dense and provide results for STABLE with different descriptor lengths. Subsequently, we provide illustrative examples for selected features found during the road surface survey. The purpose of this survey is to assess 3-D road surface as poor road conditions lead to increased wear and tear on vehicles and has an impact on surface water transport, noise emission, etc.

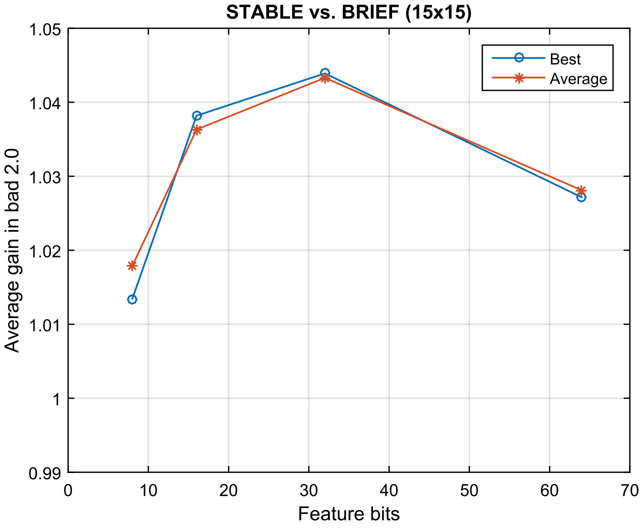


Fig. 9 Stereo matching performance gain of STABLE over BRIEF on Middlebury Stereo Dataset 2014. Displayed are the relative differences between average and best bad 2.0 scores from 25 runs averaged over all datasets.

4.3.1 *Descriptor properties for road surface*

Figures 11(a) and 11(b) show a stereo image pair depicting a top down view of a washed concrete surface. The estimated depth maps shown in Figs. 11(c) and 11(d) are results of 15×15 CENSUS-dense and 15×15 STABLE with 64 bit descriptor length, respectively. The result of STABLE is less noisy (i.e., less "black" pixels) using just 64 bits, while achieving a qualitatively similar, or even slightly better, depth estimation as the $15 \times 15 - 1 = 224$ bit long CENSUS descriptor.

Figure 12 shows the performance of STABLE with a descriptor length ranging from 16 bits to 112 bits, which is the maximum bit count possible for the 15×15 matching window. While the 16-bit long descriptor still provides quite noisy results, using 32- or 64-bit descriptors improves the reconstruction quality significantly. On the other hand, increasing the size of the descriptor to full 112 bits does not seem to improve the result any further.

Finally, Fig. 13 shows the influence of spatial averaging and additive noise on CENSUS-dense and STABLE. In both cases, STABLE outperforms CENSUS-dense descriptor. The images show the estimated disparities, which are linearly related to depth measurements.

4.3.2 *Sample images from road survey*

Due to the lack of ground truth, we refer to a manual annotation of interesting properties visible to human observers and show the derived 3-D reconstruction from which these properties become clearly visible. In most of the results, there is a vertically oriented 3-D structure visible. This stems from diamond grinding, which is a pavement preservation technique used to remove surface irregularities to reduce noise and increase road safety. We applied postprocessing based on total variation (TV) regularization[22] to obtain smoother 3-D renderings, shown in Fig. 14. The brighter the disparity, the closer the observed object point is to the observer.

Figure 14(a) shows an image of a grinded concrete road surface with an expansion joint. The grinding stripes, as well as the expansion joint, are visible in the disparity map in Fig. 14(b). A 3-D rendering of the portion around the expansion joint is provided in Fig. 14(c). Figure 14(d) shows an image of a grinded concrete pavement with a small hole; the corresponding disparity and 3-D rendering of the area of the hole are shown in Figs. 14(e) and 14(f), respectively. A grayscale image, disparity, and 3-D rendering of a larger break out of the surface are shown in Figs. 14(g) to 14(i), respectively. Finally, an image showing two grinding lanes of different depths is provided in Fig. 14(j). Additionally, in the left upper corner, there is some material, which we assume is chewing gum, observed in the area of the deeper grinding. The disparity in Fig. 14(k) shows that the valley of the grinding is not reached at the position of this suspicious object. In the 3-D rendering in Fig. 14(l), the different grinding depths and the object are visible as well.

4.4 *Performance*

For computational complexity analysis, we compared STABLE and BRIEF with K features bits applied to $X \times Y$ image patches implemented using the index filter mask implementation, which was shown in Fig. 4. In general, there are two main operations required for using any of the local binary descriptors—building and matching. The matching operation is typically identical for all binary descriptors, i.e., making use of the Hamming distance applied to binary strings of length K. The difference can thus be only in the computational complexity of the building operation.

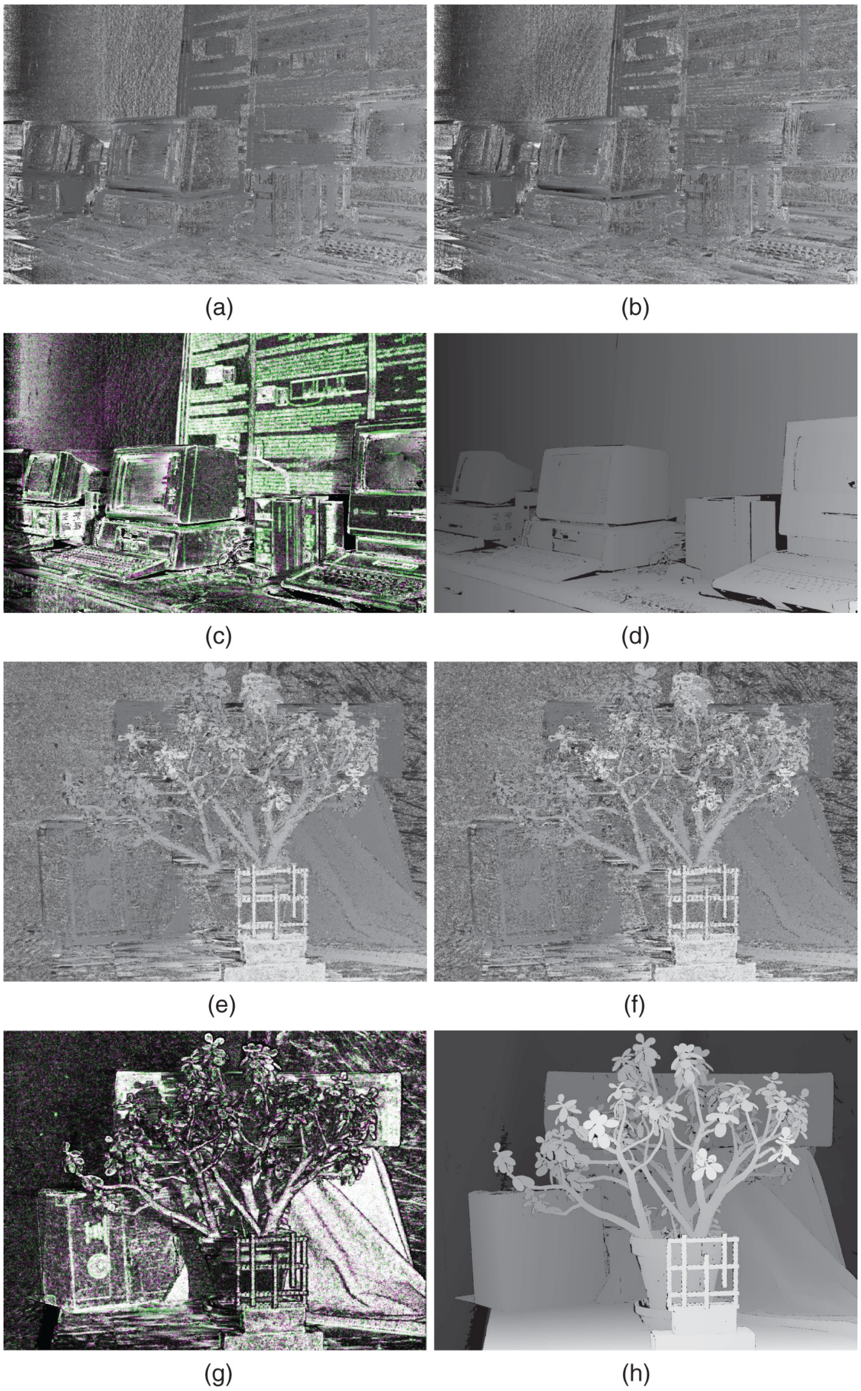


Fig. 10 Stereo matching performance of STABLE and BRIEF on Vintage (a)–(d) and Jadeplant (e)–(h) examples from the Middlebury Stereo Dataset 2014 where STABLE outperformed BRIEF the most (a)–(d) and the least (e)–(h) for the length of 32 bits. (a) and (e) The depth reconstructions when using STABLE, (b) and (f) Depth reconstructions for BRIEF, respectively. (c) and (g) The differences in bad 2.0 error metric wrt the corresponding ground truth depth maps are shown in (d) and (h). The green areas in the difference images depict areas where STABLE gained better bad 2.0 score when compared to BRIEF, whereas magenta refers to a better bad 2.0 score for BRIEF. Black to white areas indicate that both methods obtained very similar bad 2.0 errors.

Building of the descriptors is comprised of three basic steps:

1. generating the index filter mask;
2. computing the accumulator values; and
3. binarization of the accumulator values.

The index filter mask is generated only once and can be considered an input parameter for the building operation. Therefore, this step can be omitted from our analysis. The binarization step uses the same thresholding algorithm for both analyzed descriptors and can be neglected as well. Hence, the only difference comes from the complexity of computing the accumulator values, as shown in Algorithm 1. While STABLE requires processing of $X \times Y$ elements from the index filter mask as well as from the image patch (or $X \times Y - 1$ for odd number of pixels), BRIEF requires processing only $2K$ such elements. Consequently, for a fixed K, STABLE scales linearly with the number of patch pixels while BRIEF, in principle, requires only a constant time.

In practice, however, the difference between the actual execution time on CPU or GPU platforms and the theoretical one might be more in favor of STABLE due to caching in the on-chip memory. When a memory read for a cell is requested, often nearby cells are fetched and stored in the cache as well (details are hardware-dependent). To enable optimal caching, the data have to be well-organized in the memory, i.e., aligned with the hardware layout, and should be accessed using a predictable memory access patterns, e.g., in the same order as they were stored. This is especially important for GPUs where the global memory latency is higher compared to the CPU memory and thus optimal utilization of the cache memory has a higher impact on the final performance. We believe that such memory caching

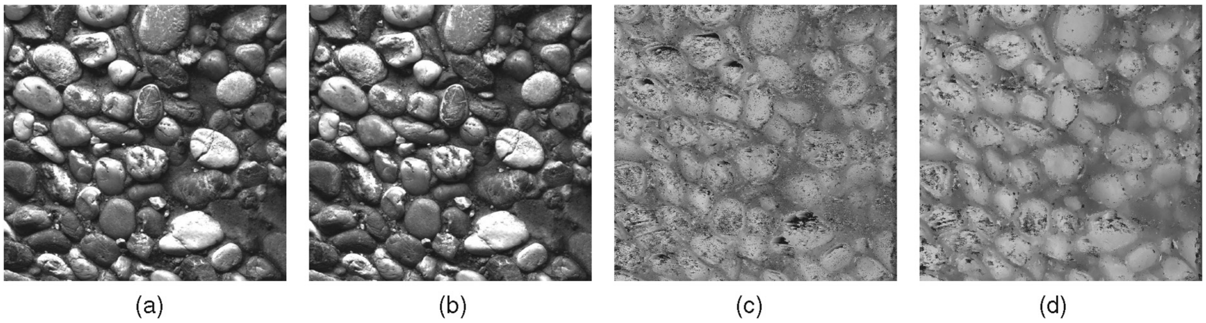


Fig. 11 Depth reconstruction of the road surface from a stereo image pair (a) and (b) using 15 × 15 CENSUS-dense with 224 bits (c), and 15 × 15 STABLE with 64 bits (d).

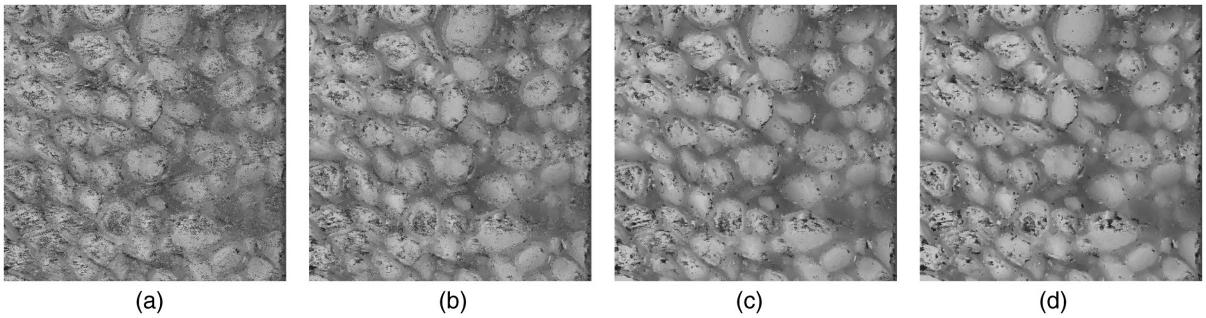


Fig. 12 Depth reconstruction quality obtained by 15 × 15 STABLE with different bit counts: (a) 16, (b) 32, (c) 64, (d) 112.

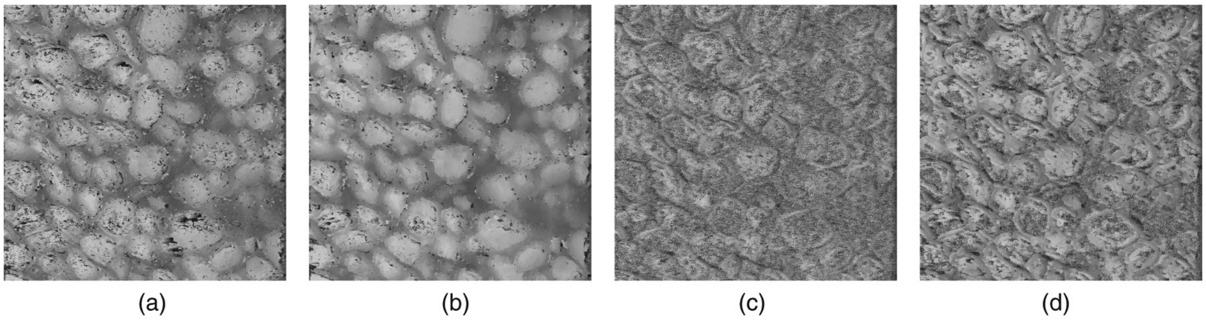


Fig. 13 Depth reconstruction results for 15 × 15 CENSUS-dense with 224 bits and 15 × 15 STABLE with 64 bits: (a) and (b) corrupted by 5 × 5 averaging, (c) and (d) corrupted by Gaussian noise with −20 dB variance.

mechanisms can be better utilized with STABLE as all elements in both index and image patch arrays are always accessed. In particular, they are accessed sequentially. Therefore, the memory access pattern can be fully optimized. On the other hand, as BRIEF uses a random-access sparse memory pattern, prediction algorithms implemented in various memory caching mechanisms are more prone to fail.

To practically measure the difference between execution times of building STABLE and BRIEF descriptors, we implemented the accumulator algorithm, described in Algorithm 1, for a CUDA-enabled GPU in C/C++. Namely, we used the CUDA Toolkit 7.5 and a NVIDIA GTX Titan GPU. As a reference, we also implemented the CENSUS-dense descriptor. The test data were a grayscale image of

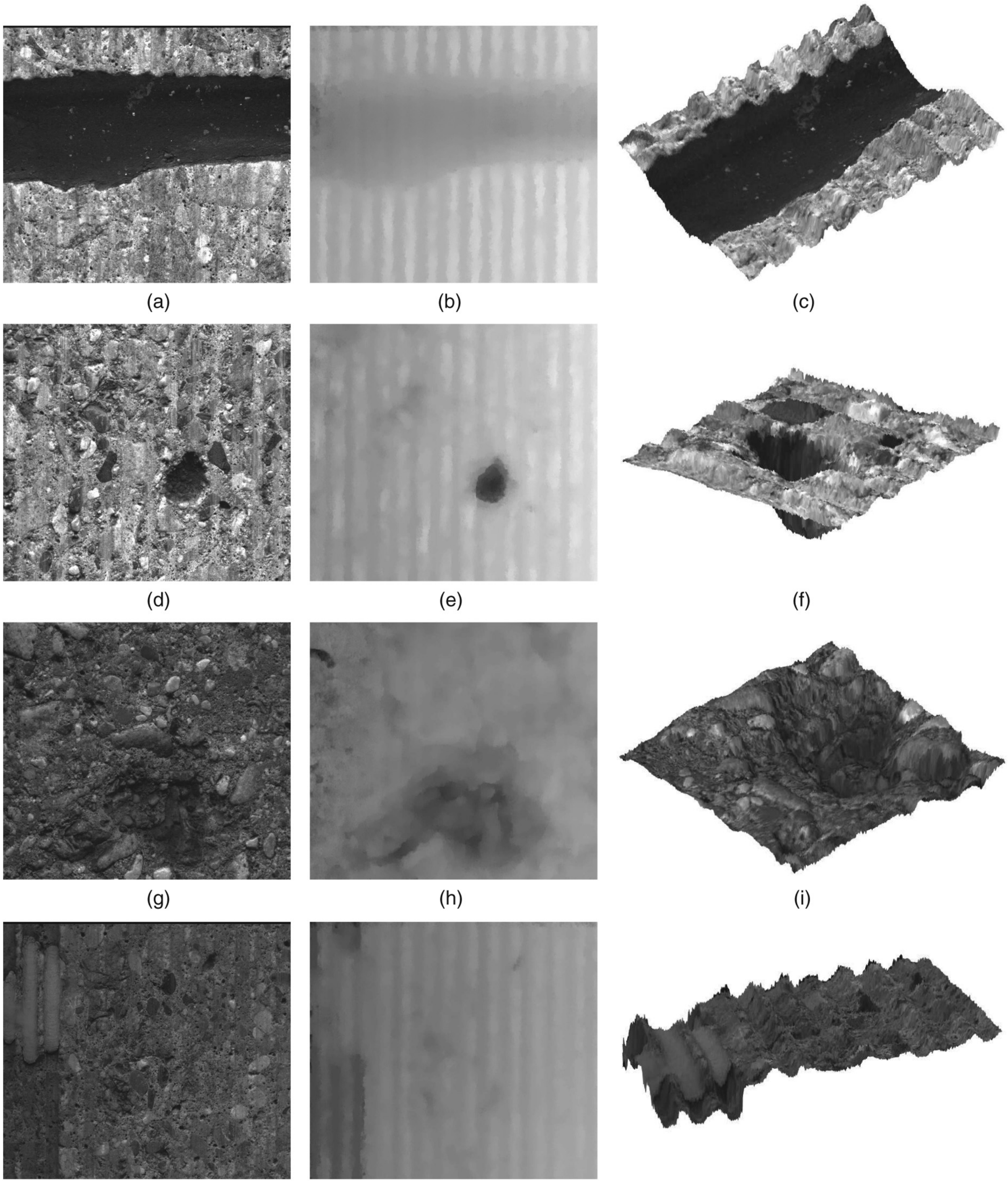


Fig. 14 Illustrative results for road surface data (disparity maps smoothed by TV regularization): grinded concrete surface with expansion joint: (a) image, (b) disparity, and (c) 3-D rendering (cutout); grinded concrete surface with a hole: (d) image, (e) disparity, and (f) 3-D rendering (cutout); ungrinded concrete surface with a larger break out region: (g) image, (h) disparity, and (i) 3-D rendering (cutout); grinding at different depths and a chewing gum like object observed in a portion of the deeper grinding: (j) image, (k) disparity, and (l) 3-D rendering (cutout).

Algorithm 1 Computation of the accumulator values in BRIEF and STABLE using a single index filter mask.

```
Require: image patch $\mathbf{p}$, index filter mask $\mathbf{g}$
  initialize array $a$ to size $K$ with values of 0
  for non-zero $i$ in $\mathbf{g}$ do
    $a[|i|] \leftarrow a[|i|] + \mathrm{sgn}(i) \times \mathbf{p}[\text{position of } i \text{ in } \mathbf{g}]$
  end for
```

Table 1 Average execution time measurements of descriptor generation for BRIEF, STABLE, and CENSUS-dense run on a GPU. A comparison relative to BRIEF is shown on the right side of respective columns.

Parameters	Descriptor	Utilized pixels		Total time (ms)		Time per util. pixel (ns)	
×15,32b	BRIEF	64		8.54		133.48	
	STABLE	224	×3.5	16.83	×1.97	75.12	×0.56
×15,64b	BRIEF	128		16.12		125.92	
	STABLE	224	×3.5	22.04	×1.37	98.40	×0.78
×15,224b	CENSUS-dense	225		5.73		25.58	

1500×1500 pixels. The descriptors of length 32 and 64 bits were represented as packed binary strings using 32- and 64-bit integers, respectively, and were computed from windows of 15×15 pixels. Each CUDA thread computed one descriptor. Threads were arranged into 32×32 thread blocks. The CUDA code was compiled with the preference on L1 cache memory size. We executed the algorithm 500 times, each time with a different randomly generated index filter mask for both descriptors. Average measured execution times are listed in Table 1.

Results in Table 1 show that total execution time of STABLE, in comparison to BRIEF, is lower than expected merely from the number of utilized pixels for both 32- and 64-bit lengths. When considering execution time per utilized pixel, execution time for STABLE is even lower by 44% and 22% for 32- and 64-bit length, respectively. This strongly points to a better utilization of GPU's hardware memory caching.

5 Conclusion

In this paper, we have introduced the STABLE descriptor, suitable for high-performance dense stereo matching, for the application of line-scan stereo matching. STABLE relates to the compressed sensing theory for efficient representation of image patterns. We showed that STABLE provides significantly better matching quality wrt, the efficiency of data representation being preserved in a highly compressed binary form.

Compared with other state-of-the-art binary descriptors, our descriptor achieves the same matching quality with considerably fewer descriptor bits required, or alternatively, significantly better matching quality making use of the same number of descriptor bits. This could be advantageous in storage- and/or memory-limited environments. STABLE offers increased stability and robustness, especially in the cases where data are subject to noise, blur, and/or slight misplacement, which is often observed in practice. In all of the considered data, i.e., synthetically perturbed data from the set introduced in Ref. 20, the Middlebury Stereo Dataset 2014 (Ref. 21) and real-world line-scan stereo data, encouraging results were achieved. Promising illustrative examples from the real-world road survey application were provided.

Unlike some other descriptors, the descriptor size and the matching window are defined independently in STABLE. Moreover, STABLE always utilizes all pixels of the given matching window for producing the required number of feature bits, which makes it suitable for many practical applications where a trade-off between the descriptor size, due to computational performance limitations, and the overall matching performance is necessary. Yet another indication of the same is that STABLE surpasses other analyzed descriptors predominantly in a small-medium range of feature bits.

Despite that STABLE requires more operations to compute than BRIEF for the same window size and bit length, it runs in less time per utilized pixel on GPU as it can take better advantage of the GPUs memory caching mechanisms. Comparable results are expected on different computing platforms implementing similar caching mechanisms.

We have demonstrated that the proposed descriptor works very well for a broad class of natural patterns and that the inherent sparsity of those patterns suffices the assumptions of the compressed sensing theory. Another direction of our future research will go toward ways of mitigating certain matching artifacts originating from a typically rectangular matching window, where each pixel is utilized precisely one time. The calibration of line-scan stereo, which so far has been solved only by a mechanical camera adjustment, will also be considered in more detail in future investigations.

References

1. S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor - system description, issues and solutions," in *Proc. of Computer Vision and Pattern Recognition Workshop* (2004).
2. Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia* **19**, 4–10 (2012).
3. R. Basri, D. Jacobs, and I. Kemelmacher, "Photometric stereo with general, unknown lighting," *Int. J. Comput. Vision* **72**(3), 239–257 (2007).
4. E. Krotkov and J. P. Martin, "Range from focus," in *Proc. of Int. Conf. on Robotics and Automation*, pp. 1093–1098 (1986).
5. M. Okutomi and T. Kanade, "A multiple baseline stereo system," *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(4), 353–363 (1993).
6. R. Ng et al., "Light field photography with a hand-held plenoptic camera," Tech. Rep. CSTR 2005-02, Stanford University (2005).
7. S. Štolc, K. Valentín, and R. Huber-Mörk, "STABLE: stochastic binary local descriptor for highperformance dense stereo matching," in *Proc. of IS&T Int. Symp. on Electronic Imaging: Machine Vision Applications IX* (2016).
8. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**(2), 91–110 (2004).
9. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," *Lect. Not. Comput. Sci.* **3951**, 404–417 (2006).
10. M. Calonder et al., "Brief: binary robust independent elementary features," *Lect. Not. Comput. Sci.* **6314**, 778–792 (2010).
11. E. Rublee et al., "ORB: an efficient alternative to SIFT or SURF," in *Proc. of Int. Conference on Computer Vision (ICCV)*, pp. 2564–2571 (2011).
12. R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," *Lect. Not. Comput. Sci.* **801**, 151–158 (1994).
13. T. Mäenpää, "The local binary pattern approach to texture analysis - extensions and applications," PhD Thesis, Machine Vision and Media Processing Unit, Infotech Oulu, University of Oulu, Finland (2003).

14. S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2548–2555 (2011).
15. A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: fast retina keypoint," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 510–517 (2012).
16. R. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.* **24**, 118–121 (2007).
17. C. A. Luna et al., "Calibration of line-scan cameras," *IEEE Trans. Instrum. Meas.* **59**, 2185–2190 (2010).
18. M. Pietikäinen et al., *Computer Vision Using Local Binary Patterns*, Springer, London (2011).
19. R. W. Frischholz and K. P. Spinnler, "Class of algorithms for real-time subpixel registration," *Proc. SPIE* **1989**, 50–59 (1993).
20. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2005).
21. D. Scharstein et al., "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. of German Conf. on Pattern Recognition (DAGM)*, pp. 31–42, Springer (2014).
22. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D* **60**(1), 259–268 (1992).

Kristián Valentín received his PhD in computer science from Comenius University in Bratislava, Slovakia, in 2015. Since 2014, he has worked at AIT, Vienna, Austria, in the field of computational imaging and computer vision.

Reinhold Huber-Mörk received his PhD in computer science from the University of Salzburg, Austria, in 1999. Since then he has worked at the Aerosensing GmbH, Oberpfaffenhofen, Germany, in remote sensing image analysis, at the Advanced Computer Vision GmbH, Vienna, Austria, in computer vision, and in 2006 he joined the AIT, Vienna, Austria, where he is currently a senior scientist in the field of machine vision.

Svorad Štolc received his master's degree in computer science from Comenius University, Bratislava, in 2002, and his PhD in bionics and biomechanics from the Technical University of Košice and Slovak Academy of Sciences, Bratislava, in 2009. He is a researcher at the Digital Safety and Security Department of AIT GmbH, Vienna. His main research areas are image processing and computational imaging.