

Document downloaded from the institutional repository of the University of Alcalá: <http://ebuah.uah.es/dspace/>

This is a preprint version of the following published document:

Alcantarilla, P.F., Bergasa, L.M. & Davison, A.J. 2013, "Gauge-SURF descriptors", *Image and Vision Computing*, vol. 31, no. 1, pp. 103-116

Available at <http://dx.doi.org/10.1016/j.imavis.2012.11.001>

© 2013 Elsevier

(Article begins on next page)



This work is licensed under a

Creative Commons Attribution-NonCommercial-NoDerivatives
4.0 International License.

Gauge-SURF Descriptors

Pablo F. Alcantarilla, Luis M. Bergasa, Andrew Davison

{pablo.alcantarilla,bergasa}@depeca.uah.es

{ajd}@doc.ic.ac.uk

Abstract

1 In this paper, we present a novel family of multiscale local feature descrip-
2 tors, a theoretically and intuitively well justified variant of SURF which is
3 straightforward to implement but which nevertheless is capable of demon-
4 strably better performance with comparable computational cost. Our family
5 of descriptors, called Gauge-SURF (G-SURF), are based on second-order
6 multiscale gauge derivatives. While the standard derivatives used to build a
7 SURF descriptor are all relative to a single chosen orientation, gauge deriva-
8 tives are evaluated relative to the gradient direction at every pixel. Like
9 standard SURF descriptors, G-SURF descriptors are fast to compute due to
10 the use of integral images, but have extra matching robustness due to the
11 extra invariance offered by gauge derivatives. We present extensive experi-
12 mental image matching results on the Mikolajczyk and Schmid dataset which
13 show the clear advantages of our family of descriptors against first-order lo-
14 cal derivatives based descriptors such as: SURF, Modified-SURF (M-SURF)
15 and SIFT, in both standard and upright forms. In addition, we also show ex-
16 perimental results on large-scale 3D Structure from Motion (SfM) and visual
17 categorization applications.

Keywords: Gauge coordinates, scale space, feature descriptors, integral

18 **1. Introduction**

19 Given two images of the same scene, image matching is the problem of
20 establishing correspondence and is a core component of all sorts of computer
21 vision systems, particularly in classic problems such as Structure from Mo-
22 tion (SfM) [1], visual categorization [2] or object recognition [3]. There has
23 been a wealth of work in particular on matching image keypoints, and the
24 key advances have been in multiscale feature detectors and invariant descrip-
25 tors which permit robust matching even under significant changes in viewing
26 conditions.

27 We have studied the use of gauge coordinates [4] for image matching and
28 SfM applications and incorporated them into a Speeded-Up Robust Features
29 (SURF) [5] descriptor framework to produce a family of descriptors of dif-
30 ferent dimensions which we call Gauge-SURF (G-SURF) descriptors. With
31 gauge coordinates, every pixel in the image is described in such a way that
32 if we have the same 2D local structure, the description of the structure is
33 always the same, even if the image is rotated. This is possible since multi-
34 scale gauge derivatives are rotation and translation invariant. In addition,
35 gauge derivatives play a key-role in the formulation of non-linear diffusion
36 processes, as will be explained in Section 3.1. By using gauge derivatives,
37 we can make blurring locally adaptive to the image itself, without affecting
38 image details.

39 The G-SURF descriptors are very related to non-linear diffusion [6, 7]
40 processes in image processing and computer vision. In the typical Gaussian

41 scale-space [8] framework, details are blurred during evolution (i.e. the con-
42 volution of the original image with Gaussian kernels of increasing standard
43 deviation). The advantage of blurring is the removal of noise, but relevant
44 image structures like edges are blurred and drift away from their original lo-
45 cations during evolution. In general, a good solution should be to make the
46 blurring locally adaptive to the image yielding the blurring of noise, while
47 retaining details or edges. Instead of local first-order spatial derivatives, G-
48 SURF descriptors measure per pixel information about image blurring and
49 edge or detail enhancing, resulting in a more discriminative descriptors.

50 We have obtained notable results in an extensive image matching evalua-
51 tion using the standard evaluation framework of Mikolajczyk and Schmid [9].
52 In addition, we have tested our family of descriptors in large-scale 3D SfM
53 datasets [10] and visual categorization experiments [2] with satisfactory re-
54 sults. Our results show that G-SURF descriptors outperform or approximate
55 state of the art methods in accuracy while exhibiting low computational de-
56 mands making it suitable for real-time applications.

57 We are interested in robust multiscale feature descriptors, to reliably
58 match two images in real-time for visual odometry [11] and large-scale 3D
59 SfM [10] applications. Image matching here, is in fact a difficult task to solve
60 due to the large motion between frames and the high variability of camera
61 movements. For this purpose, we need descriptors that are fast to compute
62 and at the same time exhibit high performance.

63 In addition, we have created an open-source library called *OpenGSURF*
64 that contains all the family of G-SURF descriptors and we plan to make it
65 publicly available. This family of descriptors comprises several descriptors

66 of different dimensions based on second-order multiscale gauge derivatives.
67 Depending on the application some descriptors may be preferred instead of
68 others. For example, for real-time applications a low-dimensional descriptor
69 should be preferred instead of a high-dimensional one, whereas for image-
70 matching applications considering severe image transformations one can ex-
71 pect a higher recall by using high-dimensional descriptors. To the best of our
72 knowledge, this is the first open source library that allows the user to choose
73 between different dimensional descriptors. Current open source descriptors
74 libraries [12, 13] just have implementations for the standard SURF and Scale
75 Invariant Feature Transform (SIFT) [14] descriptors’ default dimensions (64
76 and 128 respectively). This can be a limitation and a computational bot-
77 tleneck for some real-time applications that do not necessarily need those
78 default descriptor dimensions.

79 The rest of the paper is organized as follows: Related work is described in
80 Section 2. Gauge coordinates are introduced in Section 3 and the importance
81 of gauge derivatives in non-linear diffusion schemes is reviewed in Section 3.1.
82 Then we briefly discuss SURF based descriptors in Section 4. The overall
83 framework of our family of descriptors is explained in Section 5. Finally, we
84 show extensive experimental results in image matching, large-scale 3D SfM
85 and visual categorization applications in Section 6.

86 2. Related Work

87 The highly influential SIFT [14] features have been widely used in applica-
88 tions from mobile robotics to object recognition, but are relatively expensive
89 to compute and are not suitable for some applications with real-time de-

90 mands. Inspired by SIFT, Bay et al. [5] proposed SURF features, which
91 define both a detector and a descriptor. SURF features exhibit better re-
92 sults than previous schemes with respect to repeatability, distinctiveness and
93 robustness, but at the same time can be computed much faster thanks to the
94 use of integral images [15]. Recently, Agrawal et al. [16] proposed some mod-
95 ifications of SURF in both the detection and description steps. They intro-
96 duced Center Surround Extremas (CenSurE) features and showed that they
97 outperform previous detectors and have better computational characteristics
98 for real-time applications. Their variant of the SURF descriptor, Modified-
99 SURF (M-SURF), efficiently handles the descriptor boundaries problem and
100 uses a more intelligent two-stage Gaussian weighting scheme in contrast to
101 the original implementation which uses a single Gaussian weighting step.

102 All the mentioned approaches rely on the use of the Gaussian scale-
103 space [8] framework to extract features at different scales. An original image
104 is blurred by convolution with Gaussian kernels of successively large standard
105 deviation to identify features at increasingly large scales. The main drawback
106 of the Gaussian kernel and its set of partial derivatives is that both interest-
107 ing details and noise are blurred away to the same degree. It seems to be
108 more appropriate in feature description to make blurring locally adaptive to
109 the image data so that noise will be blurred, while at the same time details
110 or edges will remain unaffected. In this way, we can increase distinctiveness
111 when describing an image region at different scale levels. In spirit, non-linear
112 diffusion shares some similarities with the *geometric blur* proposed by Berg
113 and Malik [17], in where the the amount of Gaussian blurring is proportional
114 to the distance from the point of interest.

115 From their definition, gauge derivatives are local invariants. Matching by
116 local invariants has previously been studied in the literature. In [18], Schmid
117 and Mohr used the family of local invariants known as *local jet* [19] for image
118 matching applications. Their descriptor vector contained 8 invariants up to
119 third order for every point of interest in the image. This work represented a
120 step forward over previous invariant recognition schemes [20]. In [9], Mikola-
121 jczyk and Schmid compared the performance of the *local jet* (with invariants
122 up to third order) against other descriptors such as steerable filters [21], im-
123 age moments [22] or SIFT. In their experiments the local jet exhibits poor
124 performance compared to SIFT. We hypothesize that this poor performance
125 is due to the fixed settings used in the experiments, such as a fixed image
126 patch size and a fixed Gaussian derivative scale. In addition, invariants of
127 high order are more sensitive to geometric and photometric distortions than
128 first-order methods. In [23], the local jet was again used for matching ap-
129 plications, and they showed that even a descriptor vector of dimension 6
130 can outperform SIFT for small perspective changes. By a suitable scaling
131 and normalization, the authors obtained invariance to spatial zooming and
132 intensity scaling. Although these results were encouraging, a more detailed
133 comparison with other descriptors would have been desirable. However, this
134 work motivated us to incorporate gauge invariants into the SURF descriptor
135 framework.

136 Brown et al. [10], proposed a framework for learning discriminative local
137 dense image descriptors from training data. The training data was obtained
138 from large-scale real 3D SfM scenarios, and accurate ground truth corre-
139 spondences were generated by means of multi-view stereo matching tech-

140 niques [24, 25] that allow to obtain very accurate correspondences between
 141 3D points. They describe a set of building blocks for building discrimina-
 142 tive local descriptors that can be combined together and jointly optimized
 143 to minimize the error of a nearest-neighbor classifier. In this paper, we use
 144 the evaluation framework of Brown et al. to evaluate the performance of
 145 multiscale gauge derivatives under real large-scale 3D SfM scenarios.

146 3. Gauge Coordinates and Multiscale Gauge Derivatives

147 Gauge coordinates are a very useful tool in computer vision and image
 148 processing. Using gauge coordinates, every pixel in the image is described
 149 in such a way that if we have the same 2D local structure, the description
 150 of the structure is always the same, even if the image is rotated. This is
 151 possible since every pixel in the image is fixed separately in its own local
 152 coordinate frame defined by the local structure itself and consisting of the
 153 gradient vector \vec{w} and its perpendicular direction \vec{v} :

$$\begin{aligned}\vec{w} &= \left(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y} \right) = \frac{1}{\sqrt{L_x^2 + L_y^2}} \cdot (L_x, L_y) \\ \vec{v} &= \left(\frac{\partial L}{\partial y}, -\frac{\partial L}{\partial x} \right) = \frac{1}{\sqrt{L_x^2 + L_y^2}} \cdot (L_y, -L_x)\end{aligned}\tag{1}$$

154 In Equation 1, L denotes the convolution of the image I with a 2D Gaussian
 155 kernel $g(x, y, \sigma)$, where σ is the kernel’s standard deviation or scale parame-
 156 ter:

$$L(x, y, \sigma) = I(x, y) * g(x, y, \sigma)\tag{2}$$

157 Derivatives can be taken up to any order and at multiple scales for detecting
 158 features of different sizes. Raw image derivatives can only be computed in
 159 terms of the Cartesian coordinate frame x and y , so in order to obtain gauge

160 derivatives we need to use directional derivatives with respect to a fixed
 161 gradient direction (L_x, L_y) . The \vec{v} direction is tangent to the isophotes or
 162 lines of constant intensity, whereas \vec{w} points in the direction of the gradient,
 163 thus $L_v = 0$ and $L_w = \sqrt{L_x^2 + L_y^2}$. If we take derivatives with respect to
 164 first-order gauge coordinates, since these are fixed to the object, irrespective
 165 of rotation or translation, we obtain the following interesting results:

- 166 1. Every derivative expressed in gauge coordinates is an orthogonal in-
 167 variant. The first-order derivative $\frac{\partial L}{\partial \vec{w}}$ is the derivative in the gradient
 168 direction, and in fact the gradient is an invariant itself.
- 169 2. Since $\frac{\partial L}{\partial \vec{v}} = 0$, this implies that there is no change in the luminance if
 170 we move tangentially to the constant intensity lines.

171 By using gauge coordinates, we can obtain a set of invariant derivatives
 172 up to any order and scale that can be used efficiently for image description
 173 and matching. Of special interest, are the second-order gauge derivatives
 174 L_{ww} and L_{vv} :

$$L_{ww} = \frac{L_x^2 L_{xx} + 2 \cdot L_x L_{xy} L_y + L_y^2 L_{yy}}{L_x^2 + L_y^2} \quad (3)$$

$$L_{vv} = \frac{L_y^2 L_{xx} - 2 \cdot L_x L_{xy} L_y + L_x^2 L_{yy}}{L_x^2 + L_y^2} \quad (4)$$

176 These two gauge derivatives can be obtained as the product of gradients
 177 in \vec{w} and \vec{v} directions and the 2×2 second-order derivatives or Hessian matrix.

$$L_{ww} = \frac{1}{L_x^2 + L_y^2} \begin{pmatrix} L_x & L_y \end{pmatrix} \begin{pmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{pmatrix} \begin{pmatrix} L_x \\ L_y \end{pmatrix} \quad (5)$$

$$L_{vv} = \frac{1}{L_x^2 + L_y^2} \begin{pmatrix} L_y & -L_x \end{pmatrix} \begin{pmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{pmatrix} \begin{pmatrix} L_y \\ -L_x \end{pmatrix} \quad (6)$$

179 L_{vv} is often used as a ridge detector. Ridges are elongated regions of
 180 approximately constant width and intensity, and at these points the curvature
 181 of the isophotes is high. L_{ww} gives information about gradient changes in
 182 the gradient direction.

183 Figure 1(a) illustrates first-order gauge coordinates. Unit vector \vec{v} is
 184 always tangential to lines of constant image intensity (isophotes), while unit
 185 vector \vec{w} is perpendicular and points in the gradient direction. Figure 1(b)
 186 depicts an example of the resulting second-order gauge derivative L_{ww} on one
 of the images from the Mikolajczyk and Schmid’s standard dataset [9].

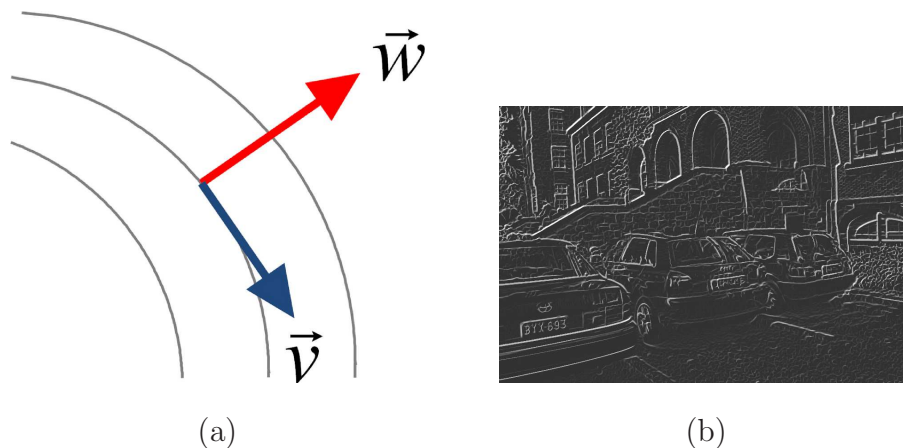


Figure 1: (a) Local first-order gauge coordinates. (b) Resulting gauge derivative L_{ww} applied on the first image of the Leuven dataset, at a fixed scale $\sigma = 2$ pixels.

187

188 According to [26], where Schmid and Mohr explicitly describe the set of
 189 second-order invariants used in the local jet, we can find two main differences
 190 between the second-order gauge derivatives L_{ww} , L_{vv} and the local jet. The
 191 first difference is that by definition gauge derivatives are normalized with
 192 respect to the modulus of the gradient at each pixel. However, this normal-
 193 ization can be also included in the local jet formulation as shown in [23]. The

194 second difference, and the most important one, is that the invariant L_{vv} is not
195 included in the set of second-order derivatives of the local jet. The invariant
196 L_{vv} plays a fundamental role in non-linear diffusion processes [7, 27]. Typi-
197 cally, Equation 4 is used to evolve the image in a way that locally adapts the
198 amount of blurring to differential invariant structure in the image in order
199 to perform edge-preserving smoothing [4].

200 *3.1. Importance of Gauge Derivatives in Non-Linear Diffusion Schemes*

201 In this section we aim to throw some more light on our decision to use
202 gauge derivatives in a feature descriptor by briefly reviewing non-linear image
203 diffusion, and highlighting the important role of gauge derivatives in these
204 schemes. Koendenrik [28] and Lindeberg [8] showed that the Gaussian kernel
205 and its set of partial derivatives provide the unique set of operators for the
206 construction of linear scale-space under certain conditions. Some examples
207 of algorithms that rely on the Gaussian scale-space framework are SIFT [14]
208 and SURF [5] invariant features.

209 However, to repeat, details are blurred in Gaussian scale-space during
210 evolution. The advantage of blurring is the removal of noise, but relevant
211 image structures like edges are blurred and drift away from their original
212 locations during evolution. In general, a good solution should be to make
213 the blurring locally adaptive to the image yielding the blurring of noise, while
214 retaining details or edges.

215 In the early nineties, several Partial Differential Equations (PDEs) were
216 proposed for dealing with the mentioned Gaussian scale-space problem. Some
217 famous examples are the Perona-Malik equation [6] and the Mean Curvature
218 Motion (MCM) [7]. Note that in general, non-linear diffusion approaches

219 perform better than linear diffusion schemes [4, 29]. Recently, Kuijper showed
220 in [29] that the evolution of an image can be expressed as a linear combination
221 of the two different second-order gauge derivatives L_{ww} and L_{vv} . According
222 to this, we can conclude that non-linear approaches steer between blurring
223 L_{ww} and edge regularising L_{vv} . Some examples of practical applications of
224 L_{ww} flow are image impaiting [30]. For L_{vv} flow an example is the cited
225 MCM [7].

226 Based on this, we can think about a local invariant descriptor that takes
227 into account the information encoded in the two gauge derivatives L_{vv} and
228 L_{ww} while the image evolves according to a scale σ . Notice that in our family
229 of descriptors we just replace the first-order local derivatives L_x and L_y with
230 the gauge derivatives L_{vv} and L_{ww} and do not perform any image evolution
231 through a non-linear scale space. That is, our descriptors will measure in-
232 formation about blurring (L_{ww}) and edge enhancing (L_{vv}) for different scale
233 levels.

234 Another difference between first-order local derivatives and gauge ones
235 is that gauge derivatives are intrisically weighted with the strength of the
236 gradient L_w . That is, the weighting is intrinsically related to the image
237 structure itself, and no artificial weighting such as Gaussian weighting is
238 needed. This is an important advantage over other descriptors, such as for
239 example SURF, where different Gaussian weighting schemes [16] have been
240 proposed to improve the performance of the original descriptor.

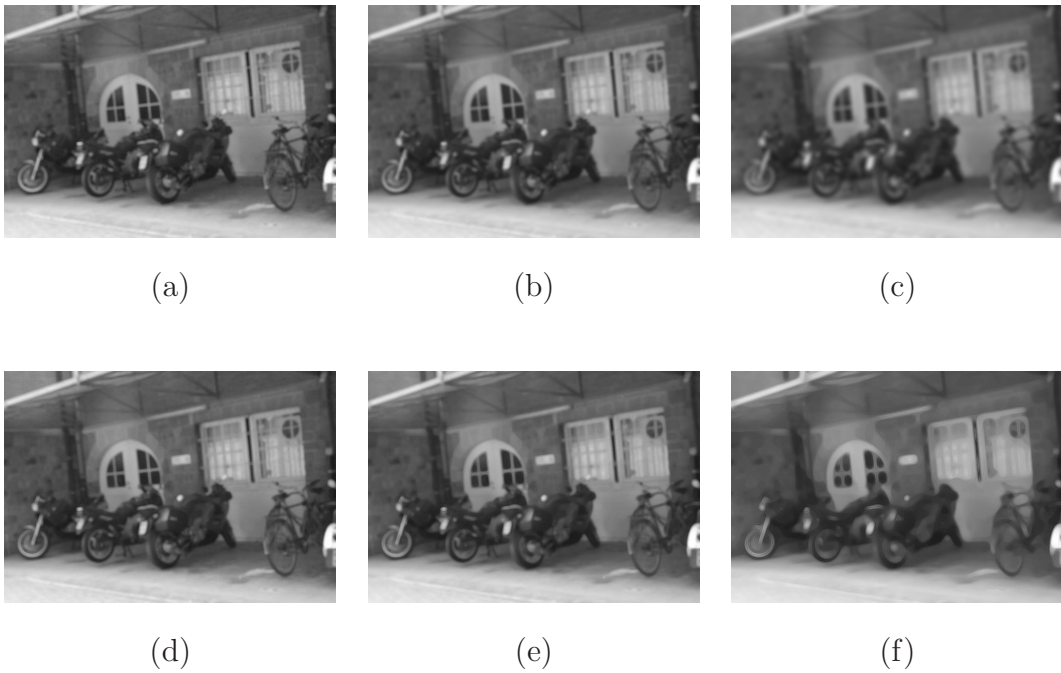


Figure 2: Gaussian scale-space versus Non-Linear diffusion schemes. The first row depicts the evolution of the sixth image from the Mikolajczyk and Schmid's Bikes dataset considering a Gaussian scale space of increasing σ in pixels. (a) $\sigma = 2$ (b) $\sigma = 4$ (c) $\sigma = 8$. The second row depicts the evolution of the same reference image but considering the MCM non-linear diffusion flow. (d) $\sigma = 2$ (e) $\sigma = 4$ (f) $\sigma = 8$. Notice how with non-linear diffusion schemes, details are enhanced and noise is removed, whereas for the Gaussian scale-space, details and noise are blurred in the same degree.

241 4. SURF Based Descriptors

242 Agrawal et al. proposed in [16] the Modified Upright-SURF descriptor
243 (MU-SURF) which is a variant of the original U-SURF descriptor. MU-
244 SURF handles descriptor boundary effects and uses a more robust and in-
245 telligent two-stage Gaussian weighting scheme. For a detected feature at
246 scale s , Haar wavelet responses L_x and L_y of size $2s$ are computed over a
247 $24s \times 24s$ region. This region is divided into $9s \times 9s$ subregions with an
248 overlap of $2s$. The Haar wavelet responses in each subregion are weighted
249 with a Gaussian ($\sigma_1 = 2.5s$) centered on the subregion center and summed
250 into a descriptor vector $d_v = (\sum L_x, \sum L_y, \sum |L_x|, \sum |L_y|)$. Then, each sub-
251 region vector is weighted using a Gaussian ($\sigma_2 = 1.5s$) defined over a mask of
252 4×4 and centered on the interest keypoint. Finally, the descriptor vector of
253 length 64 is normalized into a unit vector to achieve invariance to contrast.
254 Figure 3(a) depicts the involved regions and subregions in the MU-SURF
255 descriptor building process.

256 The main difference between the MU-SURF and U-SURF descriptor is
257 that the size of the region is reduced to $20s \times 20s$ divided into $5s \times 5s$ sub-
258 regions without any overlap between subregions. In addition, Haar wavelet
259 responses in each subregion are weighted by a Gaussian ($\sigma = 3.3s$) centered
260 at the interest keypoint. This is a very small standard deviation considering
261 that the square grid size is $20s \times 20s$. Figure 3(b) depicts a normalized 2D
262 Gaussian kernel considering a standard deviation $\sigma = 3.3$. Notice how this
263 weighting scheme smoothes completely the contribution of points far from
264 the point of interest. Therefore, only points within a distance of ± 5 pixels
265 have a significant influence in the whole descriptor.

266 The upright version of SURF-based descriptors (U-SURF) is faster to
 267 compute and usually exhibits higher performance (compared to their corre-
 268 sponding rotation invariant version, SURF) in applications where invariance
 269 to rotation is not necessary. Some examples of these applications are 3D
 270 reconstruction [5] or face recognition [31]. Although the MU-SURF descrip-
 271 tor is not invariant to rotation, it can be easily adapted for this purpose by
 272 interpolating Haar wavelet responses according to a dominant orientation in
 273 the same way as is done in the original SURF descriptor.

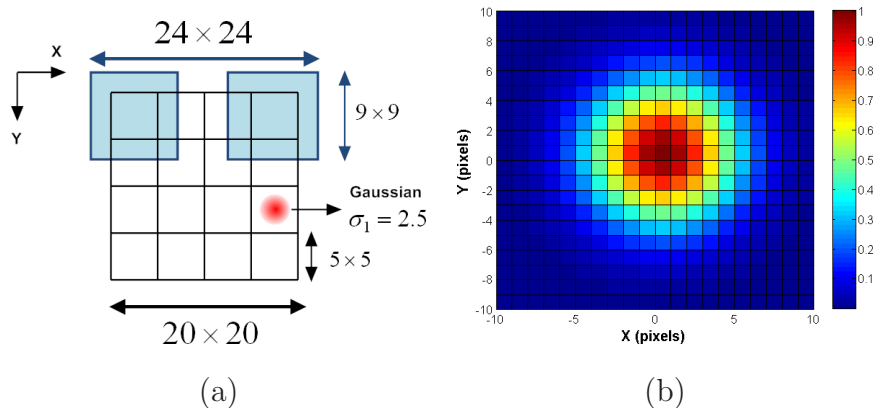


Figure 3: (a) MU-SURF descriptor building process. All sizes are relative to the scale of the feature. (b) The single Gaussian weighting scheme proposed in the original SURF descriptor. Normalized 2D gaussian kernel values considering a Gaussian kernel of standard deviation $\sigma = 3.3$ centered at the interest keypoint. Best viewed in color.

274 5. Gauge-SURF Descriptors

275 Our family of G-SURF descriptors is based on the original SURF descrip-
 276 tor. However, instead of using the local first-order derivatives L_x and L_y , we
 277 replace these two derivatives by the second-order gauge derivatives L_{ww} and

278 L_{vv} . For computing multiscale gauge derivatives, we always need to compute
 279 the derivatives first in the Cartesian coordinate frame (x, y) , and then fix the
 280 gradient direction (L_x, L_y) for every pixel. After these computations, we can
 281 obtain invariant gauge derivatives up to any order and scale with respect to
 282 the new gauge coordinate frame (\vec{w}, \vec{v}) .

283 From the definition of gauge coordinates in Equation 1, it can be observed
 284 that these coordinates are not defined at pixel locations where $\sqrt{L_x^2 + L_y^2} = 0$,
 285 i.e. at saddle points and extrema of the image. In practice this is not a
 286 problem as ter Haar Romeny states in [4], since we have a small number
 287 of such points, and according to Morse theory [32] we can get rid of such
 288 singularities by infinitesimally small local changes in the intensity landscape.
 289 What we do in practice is to not sum the contributions of these points into
 290 the final descriptor vector.

291 Now, we will describe the building process of a GU-SURF descriptor of
 292 dimension 64. For a detected feature at scale s , we compute first and second-
 293 order Haar wavelet responses $L_x, L_y, L_{xx}, L_{xy}, L_{yy}$ over a $20s \times 20s$ region.
 294 We call L_x the Haar wavelet response in the horizontal direction and L_y the
 295 response in the vertical direction. The descriptor window is divided into 4×4
 296 regular subregions without any overlap. Within each of these subregions
 297 Haar wavelets of size $2s$ are computed for 25 regularly distributed sample
 298 points. Once we have fixed the gauge coordinate frame for each of the pixels,
 299 we compute the gauge invariants $|L_{ww}|$ and $|L_{vv}|$. Each subregion yields a
 300 four-dimensional descriptor vector $d_v = (\sum L_{ww}, \sum L_{vv}, \sum |L_{ww}|, \sum |L_{vv}|)$.
 301 Finally, the total length of the unitary descriptor vector is 64.

302 Figure 4 depicts an example of the GU-SURF descriptor building process.

303 For simplicity reasons, we only show one gauge coordinate frame for each of
 304 the 4×4 subregions. Note that if we want to compute a descriptor which is
 305 invariant to rotation, we do not need to interpolate the value of the invariants
 306 L_{ww} and L_{vv} according to a dominant orientation as in SURF or M-SURF.
 307 Due to the rotation invariance of gauge derivatives, we only have to rotate
 the square grid.

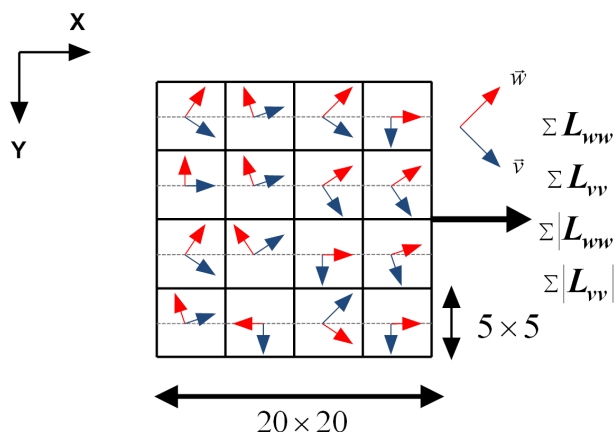


Figure 4: GU-SURF descriptor building process. Note that for the rotationally-invariant version of the descriptor we just have to rotate the square grid.

308

309 In the same way as proposed in SURF, we use box-filters to approximate
 310 first and second-order Gaussian derivatives. These box-filters are constructed
 311 through the use of integral images [15], which allows the approximation of
 312 Gaussian derivatives with low computational demands.

313 In Section 5.1, we describe the rest of descriptors of the G-SURF family
 314 included in the *OpenGSURF* library and the notation of the descriptors we
 315 will use throughout the rest of the paper.

316 *5.1. Descriptors Notation*

317 Similar to [5], we can modify the number of divisions of the square grid
318 and the size of each subregion in Figure 4 to obtain descriptors of different
319 dimensions. The descriptor size has a major impact on the matching speed
320 and recall rates. We also tested the extended version of the descriptors [5].
321 Due to space limitations, we will not evaluate this version of the descriptors
322 in this paper. However, this option is included in the OpenGSURF library.
323 As shown in [5], the overall effect of the extended descriptor is minimal.

324 Now, we will describe the notation for the set of descriptors we use
325 throughout the rest of the paper, with the number of dimensions of the
326 descriptors in parenthesis. For the SURF-based descriptors the default di-
327 mension is 64, whereas for SIFT the default dimension is 128.

- 328 • **SURF (64)**: Original SURF implementation as described in [33] that
329 uses a single Gaussian weighting scheme of a standard deviation $\sigma =$
330 $3.3s$ centered at the interest keypoint and a square grid of $20s \times 20s$.
- 331 • **M-SURF (64)**: Modified-SURF descriptor as described in [16]. This
332 descriptor uses a square grid of $24s \times 24s$ considering an overlap of Haar
333 wavelets responses and two Gaussian weighting steps.
- 334 • **G-SURF (64)**: Gauge-SURF descriptor, that uses second-order mul-
335 tiscala gauge derivatives and a square grid of $20s \times 20s$ without any
336 additional Gaussian weighting step.
- 337 • **MG-SURF (64)**: Modified Gauge-SURF descriptor, that uses the
338 same scheme as the M-SURF but replacing first-order local derivatives
339 (L_x, L_y) for second-order gauge ones (L_{ww}, L_{vv}) .

340 • **NG-SURF (64)**: No Gaussian Weighting-SURF descriptor. This de-
341 scriptor is exactly the same as the original SURF descriptor, with the
342 difference that no Gaussian weighting step is applied. In this way, we
343 can perform a fair comparison between gauge derivatives and first-order
344 local derivatives based descriptors without any additional weighting
345 scheme.

346 • **SIFT (128)**: The SIFT descriptor as described in [14]. This descriptor
347 has a dimension of 128.

348 For all the descriptors mentioned above, we denote the *upright* version
349 of the descriptors (not invariant to rotation) by adding the prefix U to the
350 name of the descriptor. For example, GU-SURF is the upright version of the
351 G-SURF descriptor. By modifying the number of divisions of the square grid
352 and the size of each of the subregions, we can obtain descriptors of different
353 dimensions. Now, we will describe the number of divisions of the square grid
354 and the size of each subregion for each of the descriptor sizes we evaluate in
355 this paper. The first number in parenthesis indicates the dimension of the
356 descriptor with the new square grid and subregion size.

357 • **(36)**: Square grid of size $18s \times 18s$ yielding 3×3 subregions each of
358 size $6s \times 6s$.

359 • **(144)**: Square grid of size $24s \times 24s$ yielding 6×6 subregions each of
360 size $4s \times 4s$.

361 6. Results and Discussion

362 In this section, we present extensive experimental image matching results
363 obtained on the standard evaluation set of Mikolajczyk and Schmid [9], large-
364 scale 3D SfM applications [10] and visual categorization experiments [2]. In
365 addition, we introduce a new dataset named *Iguazu* that consist of a series
366 of six images with the addition of increasing random Gaussian noise levels
367 with respect to the first image of the dataset. In some research areas such
368 medical imaging, RADAR or astronomy, images are usually corrupted by
369 different types of random noise. Therefore, we think that the evaluation of
370 local descriptors in these kind of datasets is of interest.

371 Our family of G-SURF descriptors implementation is based on the Open-
372 SURF library¹. The source code of our library is attached as supplementary
373 paper material. OpenSURF is an open source C++ based library with de-
374 tailed documentation and a reference paper [12]. To our knowledge, this
375 library is widely used in the computer vision and robotics community and
376 exhibits good performance, while having speed similar to the original SURF
377 library which is only available as a binary. Currently, OpenSURF uses by
378 default the M-SURF descriptor, since performance is much higher than when
379 using the single weighting Gaussian scheme. We think that OpenSURF is a
380 good open source library for performing an evaluation and comparison of a
381 set of descriptors that are all based on the same source code framework.

382 We also show comparison results with respect to SIFT descriptor, using
383 Vedaldi's implementation [13]. In all SIFT experiments we used the default

¹Available from <http://code.google.com/p/opensurf1/>

384 magnification factor $m = 3.0$, i.e. each spatial bin of the histogram has
385 support of size $m \cdot \sigma$ where σ is the scale of the point of interest. This
386 parameter has an important effect in descriptor performance. See [34] for
387 more details.

388 We have compared G-SURF descriptors to SURF, M-SURF, NG-SURF
389 (all based on OpenSURF implementation) and SIFT (based on Vedaldi's
390 implementation), in both standard and upright forms. Agrawal et al. [16]
391 claim that M-SURF's performance is similar to the original SURF library,
392 although their implementation is much faster than the original one. Like
393 Agrawal et al., we also noticed that the standard single Gaussian weighting
394 scheme as proposed in the original SURF algorithm [5] gives poor results.
395 However, we also include in our comparison the standard SURF method
396 based on the OpenSURF implementations, since this single Gaussian scheme
397 is still used in practically all of the open source libraries that include the
398 SURF algorithm, such as OpenCV or dlib C++². In addition, in Section 6.2
399 we also show some comparison results with respect to the OpenCV SURF
400 implementation, since this library has become a de facto standard for fast-
401 to-compute descriptors.

402 The rest of the experimental results and discussion section is organized as
403 follows: In Section 6.1 we show extensive image matching experiments based
404 on the standard evaluation framework of Mikolajczyk and Schmid [9], with
405 the addition of a new dataset for evaluating descriptor performance under
406 different image noise settings. Then, in Section 6.3 we evaluate the perfor-

²Available from <http://dlib.sourceforge.net/>

407 mance of G-SURF descriptors in large-scale 3D SfM scenarios. In Section 6.4
408 we show some results on visual categorization applications, and finally in
409 Section 6.5 we describe some implementation details and timing evaluation
410 results.

411 6.1. Image Matching Experiments

412 We tested our descriptors using the image sequences and testing software
413 provided by Mikolajczyk ³. We used OpenSURF’s Fast Hessian to extract
414 the keypoints in every image and then compute the descriptors, setting the
415 number of octaves and number of intervals to 4 and 2 respectively.

416 The standard dataset includes several image sets (each sequence generally
417 contains 6 images) with different geometric and photometric transformations
418 such as image blur, lighting, viewpoint, scale changes, zoom, rotation and
419 JPEG compression. In addition, the ground truth homographies are also
420 available for every image transformation with respect to the first image of
421 every sequence. We show results on eight sequences of the dataset. Table 1
422 gives information about the datasets and the image pairs we evaluated for
423 each of the selected sequences. We also provide the number of keypoints de-
424 tected for each image and the Hessian threshold value to permit reproduction
425 of our results.

426 Descriptors are evaluated by means of *recall versus 1 - precision* graphs
427 as proposed in [9]. This criterion is based on the number of correct matches

³Available from <http://www.robots.ox.ac.uk/~vgg/research/affine/>

428 and the number of false matches obtained for an image pair:

$$recall = \frac{\#correct\ matches}{\#correspondences} \quad (7)$$

$$1 - precision = \frac{\#false\ matches}{\#all\ matches}$$

429 The number of correct matches and correspondences is determined by the
 430 overlap error. Two regions (A, B) are deemed to correspond if the overlap
 431 error ϵ_0 , defined as the error in the image area covered by the regions, is
 432 sufficiently small, as shown in Equation 8:

$$\epsilon_0 < 1 - \frac{A \cap H^T \cdot B \cdot H}{A \cup H^T \cdot B \cdot H} \quad (8)$$

433 In [9] there were shown some examples of the error in relative point location
 434 and recall considering different overlap errors. They found that for overlap
 435 errors smaller than 20% one can obtain the maximum number of correct
 436 matches. In addition, they showed that recall decreases with increasing over-
 437 lap errors. Larger overlap errors result in a large number of correspondences
 438 and general low recall. Based on this, we decided to use an overlap error
 439 threshold of $\epsilon_0 < 20\%$, since we think this overlap error is reasonable for SfM
 440 applications, where you are only interested in very accurate matches. Fur-
 441 thermore, as in [35] we also impose that the error in relative point location for
 442 two corresponding regions has to be less than 2.5 pixels: $\|x_a - H \cdot x_b\| < 2.5$,
 443 where H is the homography between the images. Due to space limitations,
 444 we only show results on similarity threshold based matching, since this tech-
 445 nique is better suited for representing the distribution of the descriptor in its
 446 feature space [9].

447 Figure 5 depicts *recall versus 1-precision* graphs for the selected pairs of
 448 images. This figure suggests the following conclusions:

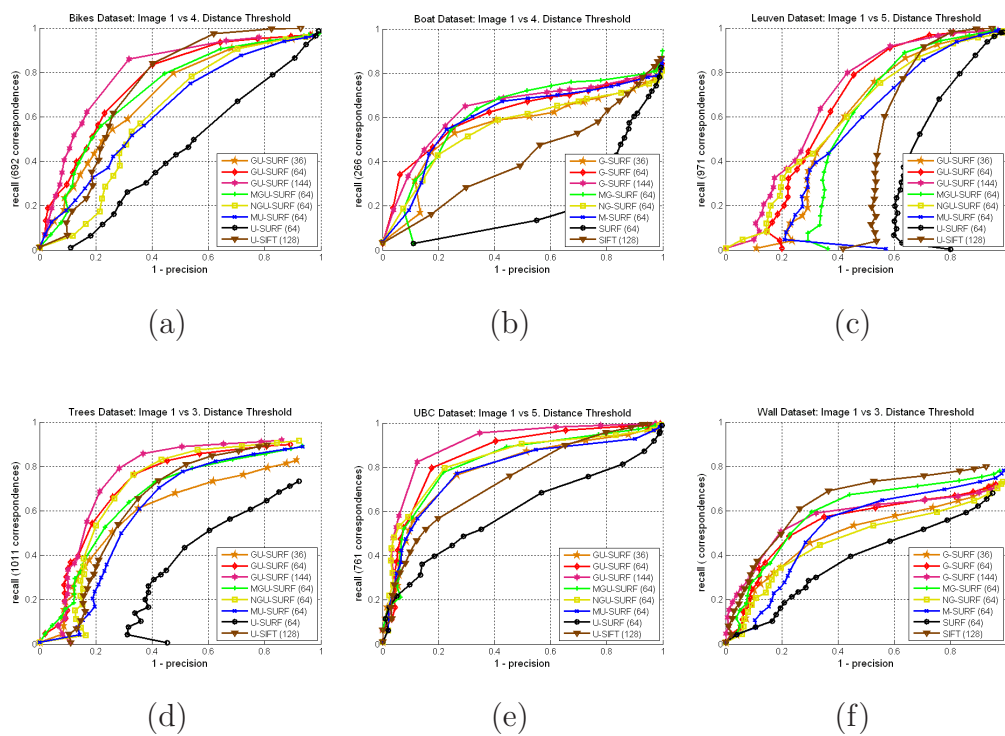


Figure 5: Image matching experiments: Recall versus 1-precision graphs, Similarity threshold based matching. (a) Bikes 1 vs 4. (b) Boat 1 vs 4. (c) Leuven 1 vs 5. (d) Trees 1 vs 3. (e) UBC 1 vs 5. (f) Wall 1 vs 3. Best viewed in color.

Dataset	Image Change	Image N	# Keypoints Image 1	# Keypoints Image N	Hessian Threshold
Bikes	Blur	4	2275	1538	0.0001
Bikes	Blur	5	2275	1210	0.0001
Boat	Zoom+Rotation	4	2676	1659	0.0001
Graffiti	Viewpoint	2	1229	1349	0.001
Leuven	Illumination	5	2705	2009	0.00001
Trees	Blur	3	3975	4072	0.0001
UBC	JPEG Compression	5	2106	2171	0.0001
Van Gogh	Rotation	10	864	782	0.00005
Van Gogh	Rotation	18	864	855	0.00005
Wall	Viewpoint	3	3974	3344	0.0001
Iguazu	Gaussian Noise	3	1603	2820	0.0001
Iguazu	Gaussian Noise	4	1603	3281	0.0001
Iguazu	Gaussian Noise	5	1603	3581	0.0001

Table 1: Sequences and image pairs used for image matching experiments: Image change, image number, keypoints number and Hessian threshold value.

- 449
- 450
- 451
- 452
- 453
- 454
- 455
- 456
- 457
- 458
- 459
- 460
- 461
- 462
- In general, among the upright evaluation of the descriptors, GU-SURF descriptors perform much better than their competitors, especially for high precision values, with sometimes more than 20% improvement in recall for the same level of precision with respect to MU-SURF (64) and U-SIFT (128) (e.g. Leuven, Bikes and Trees datasets), and even much more improvement with respect to U-SURF (64). GU-SURF (144) was the descriptor that normally achieved the highest recall for all the experiments, followed close by GU-SURF (64). GU-SURF (36) also exhibits good performance, on occasions even better than higher dimensional descriptors such as U-SIFT (128) or MU-SURF (64).
 - In the upright evaluation of the descriptors, one can obtain higher recall rates by means of descriptors that do not have any kind of Gaussian weighting or subregions overlap. For example, we can observe this effect between NGU-SURF (64) and U-SURF (64), where the only difference

463 between both descriptors is the Gaussian weighting step. Furthermore,
464 we can see that between GU-SURF (64) and MGU-SURF (64), GU-
465 SURF (64) obtained higher recall values than when using the modified
466 version of the descriptors.

467 • With respect to the rotation invariant version of the descriptors, in
468 these cases the modified descriptor version plays a more important role.
469 The use of two Gaussian weighting steps and subregions overlap yields
470 a more robust descriptor with respect to large geometric deformations
471 and non-planar rotations. In addition, the Gaussian weighting helps in
472 reducing possible computation errors when interpolating Haar wavelets
473 responses according to a dominant orientation. This interpolation of
474 the responses is not necessary in the case of gauge derivatives, since
475 by definition they are rotation invariant. We can observe that MG-
476 SURF (64) obtained slightly better results compared to M-SURF (64)
477 and SIFT (128) for the Boat dataset (Zoom+Rotation). For the Wall
478 dataset (changes in viewpoint), SIFT (128) was the descriptor that
479 obtained the best results, and MG-SURF (64) obtained better results
480 compared to M-SURF (64), especially for high precision values.

481 • When comparing gauge-based descriptors and first-order local deriva-
482 tives descriptors, we can observe that gauge-based descriptors always
483 obtained higher recall values, both in the standard and upright form of
484 the descriptors. We can observe this behaviour between G-SURF (64)
485 versus NG-SURF (64), and MG-SURF (64) versus M-SURF (64) and
486 also depending on the upright version of the descriptors. One of the

487 reasons why gauge derivatives obtained better performance is because
488 they are intrinsically weighted by the strength of the gradient L_w per
489 pixel, and thus the resulting descriptor exhibits a higher discriminative
490 power.

- 491 • In all the sequences the worst results were obtained by OpenSURF's
492 SURF implementation, which uses the single Gaussian weighting scheme
493 that gives poor results.

494 6.1.1. Evaluation under image noise transformations

495 In this section, we evaluate the performance of the descriptors under im-
496 age noise transformations. For this purpose, we created a new dataset named
497 *Iguazu*. This dataset consists of 6 images, and the image transformation in
498 this case is the progressive addition of random Gaussian noise. For each pixel
499 of the transformed images, we add random Gaussian noise with increasing
500 variance considering grey scale value images. The noise variances for each
501 of the images are the following: Image 2 ± 2.55 , Image 3 ± 12.75 , Image 4
502 ± 15.00 , Image 5 ± 51.0 and Image 6 ± 102.00 , considering that the grey value
503 of each pixel in the image ranges from 0 to 255. This new dataset is available
504 as supplementary paper material. Noisy images are very common in fields
505 such as biomedical imaging [4] and other research areas such as Synthetic
506 Aperture RADAR imaging (SAR) [36]. We think that for these applications,
507 a descriptor which is robust to different noise settings is very desirable. Fig-
508 ure 6 depicts three images of the Iguazu dataset for image random noise
509 transformations, and the *recall versus 1-precision* for three image pairs of
510 the sequence.

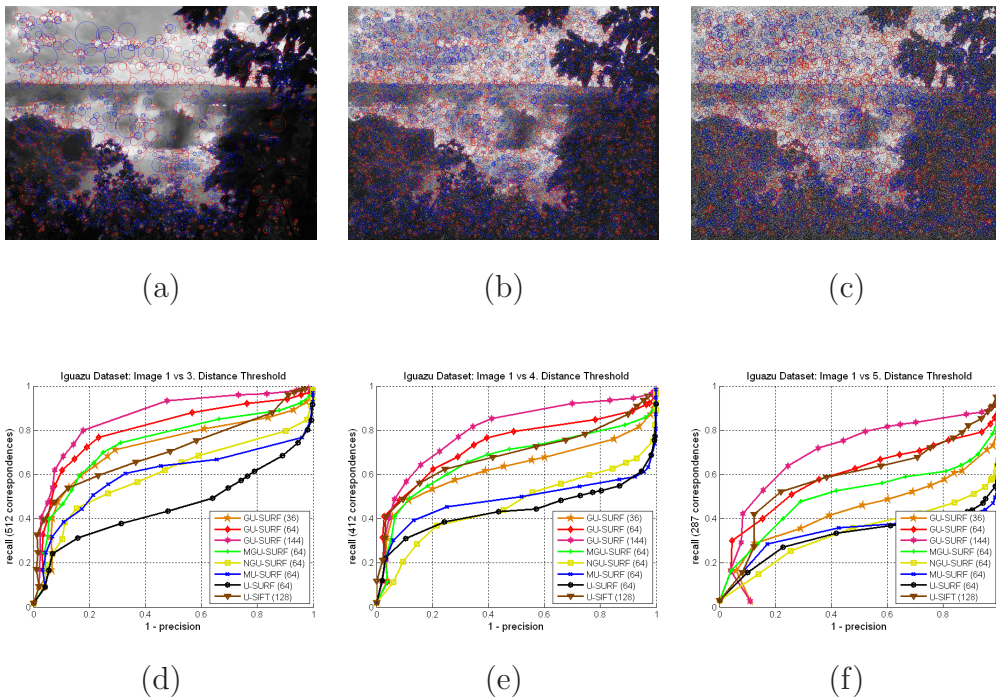


Figure 6: In the first row (a,b,c), we show some images from the Iguazu dataset, with incrementally increasing random Gaussian noise values per image. Notice that when severe random noise is added to the image, the number of detected blobs increases, mainly at small scales. The detected keypoints are shown in red or blue depending on the sign of the Laplacian. (a) Iguazu 1 (b) Iguazu 3 (c) Iguazu 5. In the second row (d,e,f), Image matching experiments: Recall versus 1-precision graphs, Similarity threshold based matching. (d) Iguazu 1 vs 3 (e) Iguazu 1 vs 4 (f) Iguazu 1 vs 5. Best viewed in color.

511 According to the graphs, we can observe that for this dataset, the dif-
512 ference between gauge-derivatives and first-order local derivatives based de-
513 scriptors is much more significant than in the previous image transformations
514 evaluation. The best results were obtained again with the GU-SURF (144)
515 descriptor. In this experiment, U-SIFT (128) obtained also good results,
516 with higher recall values than MU-SURF (64), U-SURF (64) and NGU-
517 SURF (64). Notice that in these experiments, GU-SURF (36) obtained bet-
518 ter results for the three image pairs than MU-SURF (64), U-SURF (64) and
519 NGU-SURF (64). This is remarkable, due to the low dimension of the de-
520 scriptor, and this clearly demonstrates the discriminative properties of gauge
521 derivatives against first-order ones. The main reason why G-SURF descrip-
522 tors exhibit good performance against image noise settings and higher recall
523 rates compared to first-order local derivatives methods is because G-SURF
524 descriptors measure information about the amount of blurring (L_{ww}) and
525 details or edge enhancing (L_{vv}) in the image at different scale levels.

526 6.1.2. Evaluation under pure rotation sequences

527 One of the nicest properties of gauge derivatives is their invariance against
528 rotation. In this section, we compare G-SURF descriptors against first-order
529 local derivatives descriptors, to highlight the rotation invariance properties
530 of gauge derivatives. For this purpose, we decided to use the Van Gogh
531 sequence that consists of pure rotation image transformations. This sequence
532 and the ground truth homographies relating the images can be downloaded
533 from Mykolajczyk's older webpage⁴. In order to show the performance of

⁴<http://lear.inrialpes.fr/people/mikolajczyk/Database/rotation.html>

534 G-SURF descriptor under pure rotation transformation, we evaluated two
 535 image pairs from the Van Gogh sequence. Figure 7 depicts the reference
 536 image and the other two images that are related by a pure rotation of 90°
 and 180° with respect to the reference image.

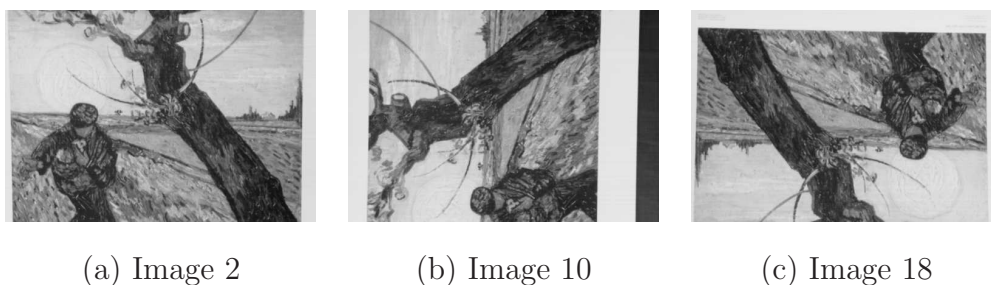


Figure 7: Van Gogh rotation dataset. Images 2 and 10 are related by a pure rotation of 90° , whereas Images 2 and 18 are related by a pure rotation of 180° .

537

538 Figure 8 depicts the *recall versus 1-precision* for the selected image pairs
 539 from the Van Gogh dataset. In this experiment, we compared only G-SURF
 540 (64) versus NG-SURF (64) and SURF (64). According to the results, we can
 541 observe that for some points in the graphs, by using G-SURF (64) there is an
 542 improvement in recall of about the 20% with respect to NG-SURF (64) and
 543 approximately double 40%, with respect to SURF (64) for the same preci-
 544 sion values. These results highlight the effect of the nice rotation invariance
 545 property of gauge-derivatives in the matching capabilities of the descriptors.

546 6.2. Comparison to OpenCV

547 In this section, we also compare our G-SURF descriptors with the lat-
 548 est OpenCV⁵ implementation of the SURF descriptor. According to [37],

⁵Available from <http://sourceforge.net/projects/opencvlibrary/>

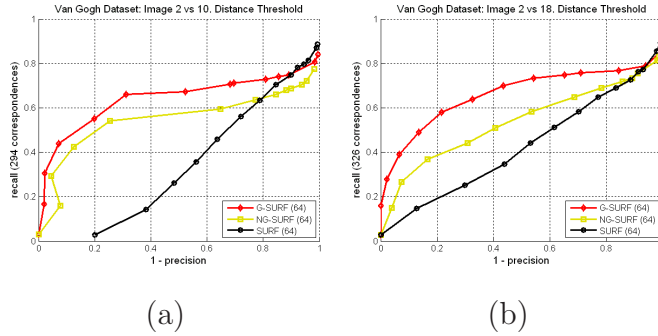


Figure 8: Image matching experiments: Recall versus 1-precision graphs, Similarity threshold based matching. (a) Van Gogh 2 vs 10 (b) Van Gogh 2 vs 18. Best viewed in color.

549 OpenCV’s SURF implementation has become a de facto standard for fast-
 550 to-compute descriptors. However as we will show in our results, the descrip-
 551 tor performance is poor and much lower compared to OpenSURF’s default
 552 M-SURF descriptor. This low performance is because the SURF implemen-
 553 tation in OpenCV uses also the single Gaussian weighting scheme as proposed
 554 in the original SURF paper [5].

555 Figure 9 depicts *recall versus 1-precision* graphs for two image pairs from
 556 the Bikes and Graffiti datasets. In this experiment, we compare G-SURF (64)
 557 with respect to M-SURF (64), SURF (64) and CV-SURF (64) both in the
 558 upright and standard forms of the descriptors. We denote by CV-SURF the
 559 OpenCV implementation of the SURF descriptor using the single weighting
 560 scheme as described in Section 4. According to the results, we can see that
 561 the OpenCV implementation gives poor results, comparable to SURF (64)
 562 in OpenSURF’s implementation, since both algorithms use the mentioned
 563 single Gaussian weighting scheme. We can appreciate a huge difference in
 564 recall with respect to G-SURF (64) and M-SURF (64).

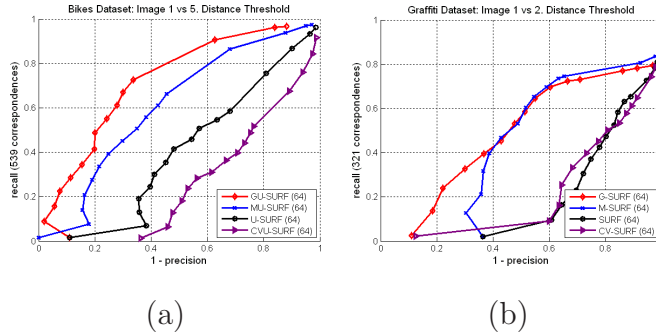


Figure 9: Image matching experiments: Recall versus 1-precision graphs, Similarity threshold based matching. (a) Bikes 1 vs 5 (b) Graffiti 1 vs 2. Best viewed in color.

565 6.3. Application to 3D Structure from Motion

566 In this section, we evaluate the performance of G-SURF based descriptors
 567 in large-scale 3D SfM applications. In particular, we use the learning local
 568 image descriptors dataset from [10]. In the mentioned work, Brown et al.
 569 proposed a framework for learning dense local image descriptors from training
 570 data using 3D correspondences from large-scale SfM datasets. For generating
 571 ground truth image correspondences between real interest points, the authors
 572 used multi-view stereo matching techniques [24, 25] that allow very accurate
 573 correspondences between 3D points to be obtained.

574 The available dataset consists of several scale and orientation normalized
 575 64×64 image patches centered around detected Harris corners or Difference
 576 of Gaussian (DoG) [14] features. Those patches were extracted from real 3D
 577 points of large-scale SfM scenarios. In our evaluation, we used 40,000 patch
 578 pairs centered on detected Harris corners from which 50% are match pairs
 579 and the other 50% are considered non-match pairs. We attach the set of
 580 matches/non-matches image patches used for the evaluation as supplement-
 581 ary material of the paper. In the evaluation framework of Brown et al.,

582 two patches are considered to be a match if the detected interest points are
 583 within 5 pixels in position, 0.25 octaves in scale and $\pi/8$ radians in angle.
 584 Figure 10 depicts some of the pre-defined match, non-match pairs from the
 585 Liberty dataset.



Figure 10: Some of the predefined match, non-match pairs from the Liberty dataset. Each row shows 3 pairs of image patches and the two image patches in each pair are shown in the same column. (a) Match pairs. (b) Non-match pairs.

586 We performed an evaluation of the upright version of the descriptors U-
 587 SURF (64), MU-SURF (64), GU-SURF (64), MGU-SURF (64), NGU-SURF
 588 (64) and U-SIFT (128) for both the Liberty and Notre Dame datasets. We
 589 chose a scale of 2.5 pixels to make sure that no Haar wavelet responses were
 590 computed outside the bounds of the image patch. For all the image pairs
 591 in the evaluation set, we computed the distance between descriptors and by
 592 means of sweeping a threshold on the descriptor distance we were able to
 593 generate ROC curves. Figure 11 depicts the ROC curves for the Liberty
 594 dataset, whereas Figure 12 depicts the ROC curves for the Notre Dame
 595 dataset.

596 In addition, in Table 2 we also show results in terms of the 95% error rate
 597 which is the percentage of incorrect matches obtained when the 95% of the

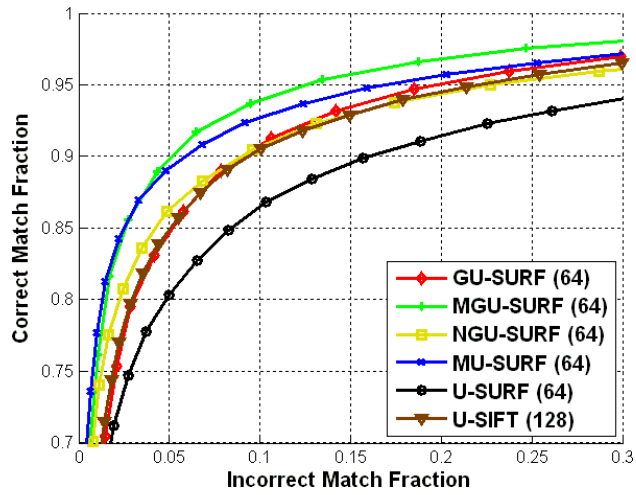


Figure 11: ROC curves for local image descriptors. Liberty dataset. Best viewed in color.

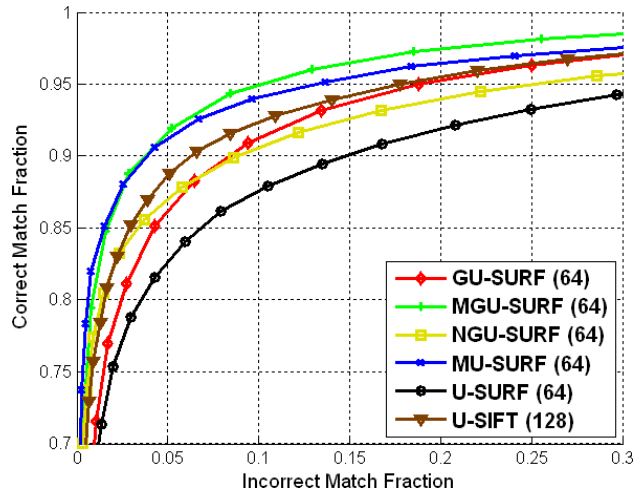


Figure 12: ROC curves for local image descriptors. Notre Dame dataset. Best viewed in color.

598 true matches are found.

Descriptor	Liberty	Notre Dame
GU-SURF (64)	19.78	18.95
MGU-SURF (64)	12.55	10.19
NGU-SURF (64)	22.95	25.22
MU-SURF (64)	16.88	13.17
U-SURF (64)	36.49	34.18
U-SIFT (128)	21.92	17.75

Table 2: Local image descriptors results. 95% error rates, with the number of descriptor dimension in parenthesis.

599 According to the results, we can observe that the lowest incorrect match
600 fraction rate for the 95% recognition rates was obtained by the MGU-SURF
601 (64) descriptor. This descriptor uses the same square grid configuration,
602 two Gaussian weighting steps and subregions overlap as proposed in [16] for
603 the MU-SURF descriptor. In typical large-scale 3D SfM scenarios, there
604 exist non-planar transformations and illumination changes resulting from
605 viewing a truly 3D scene [10]. In addition, second-order derivatives are more
606 sensitive to perspective or affine changes than first-order ones. Therefore,
607 on those scenarios where the affine changes or changes on perspective are
608 significant, the two-steps Gaussian weighting and subregions overlap seem to
609 have a good effect on the descriptor performance. This is the reason why
610 in this evaluation we obtained better results for MGU-SURF (64) and MU-
611 SURF (64) against GU-SURF (64) and NGU-SURF (64), which do not use
612 any kind of subregion overlap or Gaussian weighting steps. U-SIFT (128)

613 also obtained good results, always better than NGU-SURF (64) and very
614 similar results compared to GU-SURF (64), slightly better for the Notre
615 Dame dataset. U-SIFT (128) also uses biliner interpolation between the
616 bins of the descriptor histogram [14]. When comparing gauge-derivatives
617 based descriptors and first-order local derivatives ones, without any subregion
618 overlap nor any Gaussian weighting step, we can observe that GU-SURF (64)
619 obtained much better results than NGU-SURF (64). As expected, the worst
620 results were obtained for the U-SURF (64) descriptor, since in this descriptor
621 configuration the single Gaussian weighting step smoothes to a very high
622 degree the descriptor information, yielding lower recognition rates.

623 Besides, in the OpenGSURF library, the user can choose between the
624 SIFT-style clipping normalization or unit vector normalization of the descrip-
625 tor. This normalization can have a big impact on the matching performance
626 of the descriptors, as demonstrated in [38, 10], where one can obtain lower
627 error rates by using the SIFT-style clipping normalization. However, in order
628 to avoid the influence of this normalization style in our results, we just show
629 results using the standard unit vector normalization, except for the SIFT
630 descriptor, in which we use its default SIFT-style clipping normalization.

631 *6.4. Application to Visual Categorization Problems*

632 In this experiment, we show that G-SURF based descriptors can be used
633 efficiently in typical visual image categorization or object recognition prob-
634 lems. Bay et al. have shown in previous work [39, 33, 5] that SURF-based
635 descriptors can be used efficiently in this kind of applications. Nowadays,
636 SURF or SIFT invariant descriptors are of common use in typical visual cat-
637 egorization or object recognition schemes [2]. In a similar way to [40], we

638 performed our tests using the Caltech faces, airplanes and camels dataset ⁶.
639 Firstly, we resized all the images to a 640×480 resolution and selected 25% of
640 all the images (randomly distributed among the three categories) for training.
641 The rest of the images were used for test evaluation.

642 Even though this is a simple visual categorization problem, we want to
643 evaluate if G-SURF based descriptors can exhibit higher recognition rates
644 than traditional first-order spatial derivatives based approaches due to the
645 extra invariance offered by using gauge derivatives. Figure 13 depicts three
646 image pairs of the different categories that we used in our evaluation. In
647 particular, we can expect higher confusion between the faces and camels
648 categories. This is because in some images of the camels dataset we can
649 observe some human faces as shown for example in Figure 13(f), and also
650 that camel and human faces share some degree of similarity.

651 In order to perform an evaluation of the different local descriptors, we
652 used our own implementation of the visual bag of keypoints method de-
653 scribed in [2]. This implementation has been successfully tested before in an
654 occupant monitoring system based on visual categorization [41]. Basically,
655 we used the standard Fast-Hessian detector to detect features of interest at
656 different scale levels, and then we computed different local descriptors. In
657 this experiment, we only show a comparison between 64 dimensional descrip-
658 tors in their upright form (U-SURF, MU-SURF, GU-SURF, NGU-SURF).
659 Once the descriptors are extracted, the visual vocabulary is constructed by
660 means of the standard *k-means* clustering scheme [42]. This clustering al-

⁶<http://www.vision.caltech.edu/html-files/archive.html>



Figure 13: Three pairs of images from the Caltech dataset. (a,d) Faces. (b,e) Airplanes. (c,f) Camels. Notice the possible confusion between the faces and camels categories.

661 algorithm proceeds by iterated assignments of keypoints descriptors to their
 662 closest cluster centers and recomputation of the cluster centers. The selec-
 663 tion of the number of clusters and the initialization of the centers are of great
 664 importance in the performance of the algorithm. Finally, the visual catego-
 665 rization is done by using a simple N ave Bayes classifier [43]. In order to
 666 reduce the influence of the clustering method on the final results, we decided
 667 to use a small number of clusters $k = 20$ and performed a random initial-
 668 ization of the cluster centers. To avoid cluster initialization problems, the
 669 clusters were randomly initialized ten times in each of the experiments, re-
 670 porting categorization results just for the cluster initialization that obtained
 671 minimum compactness measure.

672 Tables 3, 4, 5, 6 show information about the performance of each of the
 673 different descriptors in the test evaluation. Similar to [2], we used three per-

674 performance measures to evaluate the performance in visual categorization: the
 675 confusion matrix, the overall error rate and the mean ranks. For more in-
 676 formation about the meaning of these performance measures, we recommend
 the reader to check the experiments section in [2].

True Classes	Faces	Airplanes	Camels
Faces	82.6531	0.8714	19.0000
Airplanes	1.3605	91.5033	12.0000
Camels	15.9864	7.6252	69.0000
Mean Ranks	1.1973	1.1154	1.3100
Overall Error Rate	0.1352		

Table 3: Confusion matrix, mean ranks and overall error rate for U-SURF (64).

True Classes	Faces	Airplanes	Camels
Faces	79.2517	0.3267	25.5000
Airplanes	0.6802	93.6819	7.0000
Camels	20.0680	5.9912	67.5000
Mean Ranks	1.2142	1.0824	1.3250
Overall Error Rate	0.1303		

Table 4: Confusion matrix, mean ranks and overall error rate for MU-SURF (64).

677

678 With respect to the confusion matrix, we can observe that GU-SURF
 679 (64) descriptor obtained higher recognition rates for the faces (85.3741%)
 680 and camels (72.0000%) categories. However, the MU-SURF (64) descriptor

True Classes	Faces	Airplanes	Camels
Faces	85.3741	0.2178	22.5000
Airplanes	0.3401	91.8301	5.5000
Camels	14.2857	7.9520	72.0000
Mean Ranks	1.1564	1.1132	1.2800
Overall Error Rate	0.1232		

Table 5: Confusion matrix, mean ranks and overall error rate for GU-SURF (64).

True Classes	Faces	Airplanes	Camels
Faces	80.6122	0.3267	20.0000
Airplanes	1.36054	93.3551	10.0000
Camels	18.0272	6.31808	70.0000
Mean Ranks	1.2074	1.0882	1.3
Overall Error Rate	0.1260		

Table 6: Confusion matrix, mean ranks and overall error rate for NGU-SURF (64).

681 obtained a higher recognition rate for the airplanes (93.68%) dataset. In
682 the same way, GU-SURF (64) obtained the lowest mean ranks for the faces
683 (1.1564) and camels (1.2800) datasets and MU-SURF (64) obtained the low-
684 est one for the airplanes dataset (1.0824). Regarding the overall error rate,
685 GU-SURF (64) was the descriptor that achieved the lowest error (0.1232).
686 There is a reduction in the overall error rate of 8.88% with respect to U-
687 SURF (64), 5.45% with respect to MU-SURF (64) and 2.22% with respect
688 to NGU-SURF (64). Even though the experimental evaluation was a simple
689 visual categorization problem, we can conclude that G-SURF based descrip-
690 tors can be used efficiently in these visual recognition schemes. In addition,
691 G-SURF descriptors can also obtain lower error rates and higher recognition
692 rates than traditional approaches that are based only on first-order local
693 derivatives.

694 *6.5. Implementation Details and Timing Evaluation*

695 In this section, we describe some implementation details of G-SURF de-
696 scriptors and perform a timing evaluation. One of the criticisms about using
697 second-order derivatives in the context of local descriptors, is the higher
698 computational cost that sometimes is not accompanied by a better perfor-
699 mance. In this section, we show that by means of using gauge derivatives
700 we can obtain much better performance than first-order based methods with
701 comparable computational cost. Table 7 shows timing results for descriptor
702 computation and also the number of the most important operations in the
703 process of building the upright SURF based descriptors. All timing results
704 were obtained on an Intel i7 2.8GHz computer.

705 In Table 7, the number of integral image areas means the number of

Case	U-SURF	MU-SURF	MGU-SURF	GU-SURF	GU-SURF	GU-SURF
Dimension	64	64	64	36	64	144
# First-Order Wavelets	800	2592	2592	648	800	1152
# Second-Order Wavelets	0	0	3888	972	1200	1728
# Gaussian Weights	800	2608	0	0	0	0
Square area	20×20	24×24	24×24	18×18	20×20	24×24
# Integral Image Areas	1600	5184	15552	3888	4800	6912
Time (ms)	0.03	0.16	0.30	0.06	0.07	0.10

Table 7: Descriptor Building Process: Number of operations, square area and average computation time per descriptor keypoint.

706 areas that we have to obtain in order to compute the descriptor. Based on
707 OpenSURF’s implementation details [12], one can estimate first-order Haar
708 wavelets L_x, L_y with just the difference of two areas of the integral image for
709 each of the first-order wavelets. For each of the second-order Haar wavelets
710 L_{xx}, L_{yy} it is necessary to compute two areas of the integral image and sum
711 these areas in a proper way. Finally, the most consuming Haar wavelet is
712 L_{xy} , since it requires the computation of 4 areas of the integral image. For
713 example, for the U-SURF (64) case, the total number of areas of the integral
714 image that we need to compute is: $(4 \times 4) \cdot (5 \times 5) \cdot (2 + 2) = 1600$. Due
715 to the extra-padding of $2s$, the MU-SURF (64) case yields: $(4 \times 4) \cdot (9 \times$
716 $9) \cdot (2 + 2) = 5184$. On the other hand, the GU-SURF (64) case yields:
717 $(4 \times 4) \cdot (5 \times 5) \cdot (2 + 2 + 2 + 2 + 4) = 4800$. However, the core observation
718 is that for the GU-SURF (64) descriptor one can obtain substantial speed-
719 up for those points in the rectangular grid where the gradient is equal to
720 zero. For those cases we do not need to compute the second-order wavelets,
721 since gauge coordinates are not defined for these points. This corresponds
722 to regions of the images of equal value, and therefore these regions are non-
723 Morse.

724 Using the same settings as described in Table 1, we can show the fraction
725 of non-Morse points among all the points where Haar wavelets were evalu-
726 ated. For example, for the following images the ratio is: Leuven Image 1
727 (17.96%), Bikes Image 1 (17.73%) and Iguazu Image 1 (32.43%). Another
728 computational advantage of the G-SURF descriptor is that it is not neces-
729 sary to interpolate the Haar wavelet responses with respect to a dominant
730 orientation, since gauge derivatives are rotation invariant.

731 As explained above, the number of operations for U-SURF (64) is the
732 smallest, yielding a small computation time per descriptor, but the perfor-
733 mance is the worst compared to the other SURF-based cases. NGU-SURF
734 (64) descriptor has similar computation times to the U-SURF descriptor,
735 with the advantage that no Gaussian weighting operations are necessary and
736 exhibiting much better performance. The modified version of the descrip-
737 tors introduces more computations in the descriptor building process, since
738 the square area is $24s \times 24s$. This yields higher computation times per de-
739 scriptor. In particular, for the MGU-SURF (64) descriptor, the number of
740 integral image areas is the highest (15552), and also the associated computa-
741 tion time per descriptor (0.30 ms). However, this descriptor only offers small
742 advantages in performance against GU-SURF (36), GU-SURF (64) and GU-
743 SURF (144) when we have sequences with strong changes in viewpoints and
744 non-planar rotations (e.g. Wall, Graffiti, Liberty and Notre Dame datasets).
745 In addition, GU-SURF (36), GU-SURF (64) and GU-SURF (144) are faster
746 to compute than MU-SURF (64) and also exhibit much better performance.
747 For the U-SIFT (128) descriptor, we obtained an average computation time
748 per keypoint of 0.42 ms. Besides, for any SIFT-based descriptor one needs

749 to compute the Gaussian scale space since the gradients are precomputed for
750 all levels of the pyramid [14]. Pre-computing the scale space is a highly con-
751 suming task in contrast to the fast integral image computation. We obtained
752 a computation time of 186 ms for the SIFT scale space generation, whereas
753 for the SURF integral image we obtained 2.62 ms. For the CVU-SURF case,
754 we obtained an average computation time per keypoint of 0.05 ms.

755 According to these results, it is clear that image matching using the G-
756 SURF descriptors can be accomplished in real-time, with high matching per-
757 formance. For example, we think that GU-SURF (36) and GU-SURF (64)
758 are of special interest to be used efficiently in real-time SfM and SLAM appli-
759 cations due to excellent matching performance and computational efficiency.

760 **7. Conclusions**

761 We have presented a new family of multiscale local descriptors, a novel
762 high performance SURF-inspired set of descriptors based on gauge coordi-
763 nates which are easy to implement but are theoretically and intuitively
764 highly appealing. Image matching quality is considerably improved rela-
765 tive to standard SURF and other state of the art techniques, especially for
766 those scenarios where the image transformation is small in terms of change in
767 viewpoint or the image transformation is related to blur, rotation, changes in
768 lighting, JPEG compression or random Gaussian noise. Our upright descrip-
769 tors GU-SURF (64) and GU-SURF (36) are highly suited to SfM and SLAM
770 applications due to excellent matching performance and computational ef-
771 ficiency. Furthermore, the rotation invariant form of the descriptors is not
772 necessary in applications where the camera only rotates around its vertical

773 axis, which is the typical case of visual odometry [11, 44] or SLAM [45] ap-
774 plications. We also showed successful results of our family of descriptors in
775 large-scale 3D SfM applications and visual categorization problems.

776 Another important conclusion that we showed in this paper, is that de-
777 scriptors based on gauge-derivatives can exhibit much higher performance
778 than first-order local derivatives based descriptors. This is possible, due
779 to the extra invariance offered by gauge-derivatives and also our G-SURF
780 descriptors have comparable computational cost with respect to other ap-
781 proaches.

782 As future work we are interested in testing the usefulness of G-SURF
783 descriptors for more challenging object recognition tasks (e.g. The PASCAL
784 Visual Object Classes Challenge). In addition, we also plan to incorporate
785 our descriptors into real-time SfM applications and evaluate them in loop
786 closure detection problems such as in [46]. Future work will aim at optimis-
787 ing the code for additional speed up and also we will exploit the use of gauge
788 coordinates in the detection of features in non-linear scale spaces. More-
789 over, we would like to introduce our gauge-based descriptors on a DAISY-
790 like framework [47] for performance evaluation on different computer vision
791 applications.

792 According to the obtained results and other successful approaches such
793 as *geometric blur*, we hope that in the near future we can break with the
794 standard scale-space paradigm in computer vision algorithms. In the stan-
795 dard scale-space paradigm the true location of a boundary at a coarse scale
796 is not directly available in the coarse scale image. The reason for this is
797 simply because Gaussian blurring does not respect the natural boundaries

798 of objects. We believe that introducing new invariant features that fully ex-
799 ploit non-linear diffusion scale spaces (both in detection and local description
800 of features) can represent step forward improvements on traditional image
801 matching and object recognition applications.

802 **References**

- 803 [1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, R. Szeliski, Building Rome
804 in a Day, in: Intl. Conf. on Computer Vision (ICCV), 2009.
- 805 [2] G. Csurka, C. Bray, C. Dance, L. Fan, Visual categorization with bags of
806 keypoints, in: In Workshop on Statistical Learning in Computer Vision,
807 ECCV, 2004, pp. 1–22.
- 808 [3] D. Lowe, Object recognition from local scale-invariant features, in: Intl.
809 Conf. on Computer Vision (ICCV), Corfu, Greece, 1999, pp. 1150–1157.
- 810 [4] B. M. ter Haar Romeny, Front-End Vision and Multi-Scale Image Anal-
811 ysis. Multi-Scale Computer Vision Theory and Applications, written in
812 Mathematica, Kluwer Academic Publishers, 2003.
- 813 [5] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, SURF: Speeded up robust
814 features, Computer Vision and Image Understanding 110 (3) (2008) 346–
815 359.
- 816 [6] P. Perona, J. Malik, Scale-space and edge detection using anisotropic
817 diffusion, IEEE Trans. Pattern Anal. Machine Intell. 12 (7) (1990) 1651–
818 1686.

- 819 [7] L. Álvarez, P. Lions, J. Morel, Image selective smoothing and edge de-
820 tection by nonlinear diffusion, *SIAM Journal on Numerical Analysis*
821 (SINUM) 29 (1992) 845–866.
- 822 [8] T. Lindeberg, Feature detection with automatic scale selection, *Intl. J.*
823 *of Computer Vision* 30 (2) (1998) 77–116.
- 824 [9] K. Mikolajczyk, C. Schmid, A performance evaluation of local descrip-
825 tors, *IEEE Trans. Pattern Anal. Machine Intell.* 27 (10) (2005) 1615–
826 1630.
- 827 [10] M. Brown, H. Gang, S. Winder, Discriminative learning of local image
828 descriptors, *IEEE Trans. Pattern Anal. Machine Intell.* 33 (1) (2011)
829 43–57.
- 830 [11] D. Nistér, O. Naroditsky, J. Bergen, Visual Odometry, in: *IEEE Conf.*
831 *on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- 832 [12] C. Evans, [Notes on the OpenSURF library](#), Tech. Rep. CSTR-09-001,
833 University of Bristol (January 2009).
834 URL <http://www.cs.bris.ac.uk/Publications/Papers/2000970.pdf>
- 835 [13] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of
836 computer vision algorithms, <http://www.vlfeat.org/> (2008).
- 837 [14] D. Lowe, Distinctive image features from scale-invariant keypoints, *Intl.*
838 *J. of Computer Vision* 60 (2) (2004) 91–110.
- 839 [15] P. Viola, M. J. Jones, Robust real-time face detection, *Intl. J. of Com-*
840 *puter Vision* 57 (2) (2004) 137–154.

- 841 [16] M. Agrawal, K. Konolige, M. R. Blas, CenSurE: Center Surround Ex-
842 tremas for realtime feature detection and matching, in: Eur. Conf. on
843 Computer Vision (ECCV), 2008.
- 844 [17] A. C. Berg, J. Malik, Geometric blur for template matching, in: IEEE
845 Conf. on Computer Vision and Pattern Recognition (CVPR), Hawaii,
846 USA, 2001, pp. 607–614.
- 847 [18] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval,
848 IEEE Trans. Pattern Anal. Machine Intell. 19 (5) (1997) 530–535.
- 849 [19] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, M. A.
850 Viergever, Cartesian differential invariants in scale-space, Journal of
851 Mathematical Imaging and Vision 3 (1993) 327–348.
- 852 [20] C. Rothwell, A. Zisserman, D. Forsyth, J. Mundy, Canonical frames for
853 planar object recognition, in: Eur. Conf. on Computer Vision (ECCV),
854 1992, pp. 757–772.
- 855 [21] W. Freeman, E. Adelson, The design and use of steerable filters, IEEE
856 Trans. Pattern Anal. Machine Intell. 13 (9) (1991) 891–906.
- 857 [22] L. V. Gool, T. Moons, D. Ungureanu, Affine/photometric invariants for
858 planar intensity patterns, in: Eur. Conf. on Computer Vision (ECCV),
859 1996, pp. 642–651.
- 860 [23] B. Platel, E. Balmachnova, L. Florack, B. ter Haar Romeny, Top-Points
861 as interest points for image matching, in: Eur. Conf. on Computer Vision
862 (ECCV), 2006.

- 863 [24] M. Goesele, B. Curless, S. Seitz, Multi-view stereo revisited, in: IEEE
864 Conf. on Computer Vision and Pattern Recognition (CVPR), New York,
865 USA, 2006, pp. 2402–2409.
- 866 [25] M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. Seitz, Multi-view stereo
867 for community photo collections, in: Intl. Conf. on Computer Vision
868 (ICCV), Rio de Janeiro, Brasil, 2007, pp. 14–20.
- 869 [26] C. Schmid, R. Mohr, Matching by local invariants, Tech. rep., INRIA
870 (Aug. 1995).
- 871 [27] L. Álvarez, F. Guichard, P. Lions, J. M. Morel, Axioms and fundamental
872 equations of image processing, Arch. for Rational Mechanics 123 (3)
873 (1993) 199–257.
- 874 [28] J. Koenderink, The structure of images, Biological Cybernetics 50 (1984)
875 363–370.
- 876 [29] A. Kuijper, Geometrical PDEs based on second-order derivatives of
877 gauge coordinates in image processing, Image and Vision Computing
878 27 (8) (2009) 1023–1034.
- 879 [30] V. Caselles, J.-M. Morel, C. Sbert, An axiomatic approach to image
880 interpolation, IEEE Trans. on Image Processing.
- 881 [31] P. Dreuw, P. Steingrube, H. Hanselmann, H. Ney, SURF-Face: Face
882 Recognition under Viewpoint Consistency Constraints, in: British Ma-
883 chine Vision Conf. (BMVC), 2009.

- 884 [32] J. Damon, Local Morse theory for solutions to the heat equation and
885 Gaussian blurring, *Journal of Differential Equations* 115 (2) (1995) 368–
886 401.
- 887 [33] H. Bay, T. Tuytelaars, L. V. Gool, SURF: Speeded up robust features,
888 in: *Eur. Conf. on Computer Vision (ECCV)*, 2006.
- 889 [34] A. Vedaldi, An open implementation of the SIFT detector and descrip-
890 tor, Tech. Rep. 070012, UCLA CSD (2007).
- 891 [35] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point
892 detectors, *Intl. J. of Computer Vision* 60 (2004) 63–86.
- 893 [36] R. Liu, Y. Wang, SAR image matching based on speeded up robust
894 feature, in: *WRI Global Congress on Intelligent Systems*, 2009.
- 895 [37] M. Calonder, V. Lepetit, P. Fua, BRIEF: Binary Robust Independent
896 Elementary Features, in: *Eur. Conf. on Computer Vision (ECCV)*, 2010.
- 897 [38] G. Hua, M. Brown, S. Winder, Discriminant embedding for local image
898 descriptors, in: *Intl. Conf. on Computer Vision (ICCV)*, Rio de Janeiro,
899 Brazil, 2007.
- 900 [39] H. Bay, B. Fasel, L. V. Gool, Interactive Museum Guide: Fast and
901 Robust Recognition of Museum Objects, in: *Proceedings of the first*
902 *international workshop on mobile vision*, 2006.
- 903 [40] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsu-
904 pervised scale-invariant learning, in: *IEEE Conf. on Computer Vision*
905 *and Pattern Recognition (CVPR)*, 2003, pp. 264–271.

- 906 [41] J. Yebes, P. Alcantarilla, L. Bergasa, Occupant monitoring system for
907 traffic control based on visual categorization, in: IEEE Intelligent Vehi-
908 cles Symposium (IV), 2011.
- 909 [42] C. M. Bishop, Pattern Recognition and Machine Learning, Springer,
910 2007.
- 911 [43] D. Lewis, Naive (bayes) at forty: The independence assumption in in-
912 formation retrieval, in: Eur. Conf. on Machine Learning (ECML), 1998,
913 pp. 4–15.
- 914 [44] M. Kaess, K. Ni, F. Dellaert, Flow separation for fast and robust stereo
915 odometry, in: IEEE Intl. Conf. on Robotics and Automation (ICRA),
916 Kobe, Japan, 2009.
- 917 [45] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, MonoSLAM: Real-
918 time single camera SLAM, IEEE Trans. Pattern Anal. Machine Intell.
919 29 (6).
- 920 [46] A. Angeli, D. Filliat, S. Doncieux, J. A. Meyer, Fast and Incremental
921 Method for Loop-Closure Detection using Bags of Visual Words, IEEE
922 Trans. Robotics 24 (2008) 1027–1037.
- 923 [47] E. Tola, V. Lepetit, P. Fua, DAISY: An efficient dense descriptor applied
924 to wide-baseline stereo, IEEE Trans. Pattern Anal. Machine Intell. 32 (5)
925 (2010) 815–830.