EVALUATION OF FEATURE-BASED METHODS FOR AUTOMATED NETWORK ORIENTATION

F. I. Apollonio [1], A. Ballabeni [1], M. Gaiani [1], F. Remondino [2]

[1] Dept. of Architecture – University of Bologna, Italy - (fabrizio.apollonio, andrea.ballabeni, marco.gaiani)@unibo.it

[2] 3D Optical Metrology (3DOM) unit, Bruno Kessler Foundation (FBK), Trento, Italy
remondino@fbk.eu, http://3dom.fbk.eu

Commission V, WG 1

KEY WORDS: Orientation, Automation, Feature extraction, Calibration

ABSTRACT

Every day new tools and algorithms for automated image processing and 3D reconstruction purposes become available, giving the possibility to process large networks of unoriented and markerless images, delivering sparse 3D point clouds at reasonable processing time. In this paper we evaluate some feature-based methods used to automatically extract the tie points necessary for calibration and orientation procedures, in order to better understand their performances for 3D reconstruction purposes. The performed tests - based on the analysis of the SIFT algorithm and its most used variants - processed some datasets and analysed various interesting parameters and outcomes (e.g. number of oriented cameras, average rays per 3D points, average intersection angles per 3D points, theoretical precision of the computed 3D object coordinates, etc.).

1. INTRODUCTION

Automated image processing for 3D reconstruction purposes is flooding every day with new tools and algorithms. Camera calibration, image orientation and dense matching methods are more and more hidden behind one-click button software and so affordable to non-expert users. In particular, automated image orientation approaches, built on feature-based methods for tie point extraction and pushed by the great developments in the Computer Vision community, are nowadays able to process large networks of unoriented and markerless images delivering sparse 3D reconstruction at reasonable processing time (Snavely et al., 2008; Agarwal et al., 2009; Frahm et al., 2010; Wu, 2013). This has led to the well know *Structure from Motion* (SfM) concept (firstly introduced by Ullman, 1979), i.e. the automated and simultaneous determination of camera parameters together with scene's geometry. SfM has been adopted also in the photogrammetric community (Barazzetti et al., 2010; Del Pizzo & Troisi, 2011; Deseilligny & Clery, 2011; Roncella et al, 2011) although camera calibration and image orientation are normally kept separate unless the image network is acceptable for self-calibration (Barazzetti et al., 2011). In all automated approaches, image correspondences are normally extracted using feature-based methods and then the unknown camera parameters and 3D object coordinates are determined using a bundle adjustment method. Commercial and open-source solution exist with performances sometimes unclear and often low reliability and repeatability (Remondino et al., 2012). Moreover a deep and metric evaluation of the different (hidden) steps is still missing.

In this work we evaluate some feature-based methods used to automatically extract the tie points necessary for calibration / orientation procedures. An automated calibration / orientation procedure is normally based on the following steps: feature detection, feature description, detector comparison, outlier removal, tie point transfer throughout the images, bundle adjustment and determination of unknown parameters. The detection and description steps are salient stages for the performances of an automated procedure. Recent investigations and comparisons of detectors and descriptors were presented in (Burghouts & Geusebroek, 2009; Juan & Gwon, 2009; Aanæs et al., 2012; Heinly et al., 2012; Oyallon & Rabin, 2013; Wu et al., 2013) mainly on indoor datasets, planar surfaces, low-resolution images and without geometric analyses with respect to 3D object coordinates. Therefore an in-depth analysis and comparisons in terms of photogrammetric parameters is needed.

In this contribution the Scale-invariant Feature Transform (SIFT) algorithm (Lowe, 2004) and its most interested variants are considered, paying great attention to the description phase of each method. The considered feature-based methods are (Section 2): SIFT (in the VLFeat implementation), SIFT-GPU, ASIFT, ColSIFT, DAISY, LDAHash, SGLOH and SURF. The employed feature-based methods are proper implementations adapted from the open-source domain. For the presented evaluation, different image networks are used (Section 3 and Fig. 1). Evaluation results and critical comments are reported.

2. FEATURE-BASED METHODS: DETECTORS & DESCRIPTORS

Feature identification and matching is at the base of many automated photogrammetric and computer vision problems and applications like 3D reconstruction, dense point cloud generation, object recognition or tracking, etc.

A feature detector (or extractor) is an algorithm that takes an image as input and delivers a set of local features (or regions) while a descriptor computes on each extracted region a specific representation of the extraction. Good image features should be independent from any geometric transformation applied to the image, they should be robust to illumination changes and they should have a low feature dimension in order to perform a quick matching.

Interest point or corner detectors (MORAVEC: Moravec, 1979; FOERSTNER: Foerstner & Guelch, 1987; HARRIS: Harris & Stephens, 1988; SUSAN: Smith & Brady, 1997; AGAST: Mair et al., 2010; FAST: Rosten et al., 2010; etc.) are very common but not really suitable for fully automated procedures and wide baselines matching as they lack of repeatability and orientation information (Schmid et al., 2000; Rodehorst & Koschan, 2006). Therefore scale- and affine-invariant region (or blob) detectors were developed (Fraundorfer & Bischof, 2004), contrary to points and corners which are less distinctive under different

viewpoints (Lindeberg, 1998; Kadir et al., 2004; Mikolajczyk & Schmid, 2004; Tuytelaars & Van Gool, 2004; Klein & Murray, 2008). The detection is usually performed with a Difference of Gaussian (DoG) or Histograms of Gradients (HoG) or Hessian methods.

Once points and regions (invariant to a class of transformations) are detected, (invariant) descriptors are computed to characterize the feature. The descriptors are a variable number of elements (from 64 to 512) computed with histogram of gradient location and orientation (Lowe, 2004), moment invariant (Van Gool et al., 1996), linear filters (Schaffalitzky & Zissermann, 2002), PCA (Mikolajczyk, K. & Schmid, C., 2005), intensity comparison and binary encoding (Calonder et al., 2010; Leutenegger et al., 2011), etc. Descriptors have proved to successfully allow (or simplify) complex operations like wide baseline matching, robot localization, object recognition, etc.

In the detection phase, in order to produce translation and scale invariant descriptors, structures must be unambiguously located, both in scale and position. This excludes image edges and corners since they are translation-, view- and scale-variant features. Therefore image blobs located on flat areas are the most suitable structures although not so precisely located as interest points and corners (Remondino, 2006).

Nowadays the most popular and used operator is the SIFT method. SIFT has good stability and invariance and it detects local keypoints with a large amount of information using the DoG method. As reported in literature (Remondino et al., 2012; Zhao et al., 2012; Apollonio et al. 2013; Morel & Yu, 2009), the typical failure cases of the SIFT algorithm are changes in the illumination conditions, reflecting surfaces (e.g. cars or windows), object / scene with strong 3D aspect, highly repeated structures in the scene and very different viewing angle between the images. In order to overcome these failures but also to quickly derive compact descriptor representations, many variants and alternatives of the SIFT algorithms were developed in the last years (Ke & Sukthankar, 2004; Brown et al., 2005; Bay et al., 2008; Morel & Yu, 2009; Bellavia et al., 2010; Tola et al., 2010; Vedaldi & Fulkerson, 2010; Rublee et al., 2011; Yeo et al., 2011; Strecha et al., 2012; Wu, 2014) and nowadays used in many open-source and commercial solutions which offer automated calibration / orientation procedures (VisualSFM, Apero, Eos Photomodeler, Microsoft Photosynth, Agisoft Photoscan, Photometrix Iwitness, 3DF Zephyr, etc.).

Between the available feature-based detector and descriptor algorithms, the evaluated methods are afterwards reported.

2.1 Scale Invariant Feature Transform (SIFT)

SIFT (Lowe, 2004) derives a large set of compact descriptors starting from a multi-scale representation of the image (i.e. a stack of images with increasing blur simulating the family of all possible zooms). In this multi-scale framework, the Gaussian kernel acts as an approximation of the optical blur introduced by a camera. The detection and location of keypoints is done by extracting the 3D extrema with a DoG operator. SIFT detects a series of keypoints mostly in the form of small patch structures, locating their centre (x,y) and characteristic scale (σ) and then it computes the dominant orientation (θ) from the gradient orientation over a region surrounding each patch. Given 8 bins for quantizing the gradient directions, the dominant orientation (responsible for the rotation invariance of the keypoint) is given by the bin with the maximum value.

The knowledge of (x, y, σ, θ) allows to compute a local descriptor of each keypoint's neighbourhood that encodes the spatial gradient distribution by a 128-dimensional vector. This compact feature vector is used to match the keypoints extracted from different images.

Since there are many phenomena that can lead to the detection of unstable keypoints, SIFT incorporates a cascade of tests to discard the less reliable points. Only those that are precisely located and sufficiently contrasted are retained. Main parameters that control the detection of points are:

- local extrema threshold (*contrast threshold*): points with a lower local extrema value are rejected but, since this threshold is closely related to the level of noise in the input image, no universal value can be set. Additionally, the image contrast of the input image plays the inverse role of the noise level therefore the contrast threshold should be set depending on the signal to noise ratio of the input image.
- local extrema localization threshold (*edge threshold*): it is used to discard unstable points, i.e. if the local extremum is on a valley. Extrema are associated with a score proportional to their sharpness and rejected if the score is below this threshold. The number of remaining features increases as the parameter is increased. The original value in Lowe (2004) is 10.

Calibration of these parameters is fundamental for the efficiency of the detection mechanism. Following the literature (May et al., 2010) and our practical experiences with different dataset (from 1x1x1 to 10x20x50 meters), a value of 6 for the *contrast threshold* and of 10 for the *edge threshold* appear to be very suitable choices.

In the presented tests, the VLFeat implementation (http://www.vlfeat.org) was used (Vedaldi & Fulkerson, 2010) in two versions: VLFeat3.4 (*contrast threshold* value set as in Lowe, 2004) and VLFeat6.0 (*contrast threshold* value adjusted from various experiences).

2.2 SIFT-GPU

SIFT-GPU (Sinha et al, 2006) is a SIFT implementation on the GPU based on the following steps:

1. convert colour to intensity and up-sample or down-sample the input image;
2. build Gaussian image pyramids (Intensity, Gradient, DoG);
3. detect keypoint with sub-pixel and sub-scale localization;
4. generate a compact list of features with GPU histogram reduction;
5. compute feature orientations and descriptors.

SIFT descriptors cannot be efficiently and completely computed on the GPU as histogram bins must be blended to remove quantization noise. Hence this step is normally partitioned between the CPU and the GPU. SIFT-GPU uses a GPU/CPU mixed method to build compact keypoint lists and to process keypoints getting their orientations and descriptors. SIFT-GPU, particularly on large size images, may get slightly different results on different GPUs due to the different floating point precision.

In the presented tests, the SIFT-GPU implementation available at http://cs.unc.edu/~ccwu/siftgpu was used. To speed-up the computation, it presents some changes in the parameter values compared with the original implementation:

- in the orientation computation, a factor σ=2.0 for the sample window size is used (typical value is σ=3.0) to increase the speed of 40%;
- the keypoint's location is refined only once and without adjusting it with respect to the Gaussian pyramids;
- the image up-sampling is not performed;
- the number of detected features (max 8000) and the image size (max 3200 pixel) are limited;
- the local extrema threshold (*contrast threshold*) is set to 5.16 instead 3.4.

In the presented tests, an optimized implementation of SIFT-GPU is also experimented with these specifications:
- the orientation computation uses σ=3.0 as in the original paper (Lowe, 2004);
- image up-sampling is performed as in Lowe (2004);
- the number of detected features and the image size are not limited;
- the local extrema threshold is set to 6;
- the detection is performed using GLSL, with an adaptation of the DoG threshold to detect more features in dark regions;
- matching is done using CPU and not limiting the number of matches.

2.3 Affine-SIFT (ASIFT)

ASIFT (Morel & Yu, 2009) aims to corrects the SIFT problem in case of very different viewing angles, i.e. it aims to be more affine invariant than SIFT by simulating the rotation of camera axes. ASIFT first adds rotation transformation to an image. Then, it further obtains a series of affine images by a tilt transformation operation $u(x, y) \rightarrow u(tx, y)$ on the image in x direction. From a technical point of view, unlike SIFT which normalize all six affine parameters, ASIFT simulates three parameters (the scale and the two rotations along the camera vertical and horizontal axes) and normalizes the other parameters (rotation along the axis orthogonal to the image plane and the two horizontal and vertical translations). ASIFT detects many feature points (as the detection is repeated several times), but the detection time rises significantly and matching time rises even more (Mikolajczyk et al., 2010). Comparing many pairs of putative homologous points, ASIFT can accumulate many wrong matches. Furthermore it shows many wrong points when used on repeated patterns.
In the presented test, the ASIFT implementation available at https://github.com/Itseez/opencv/blob/master/samples/python2/asift.py was used. As suggested in Morel & Yu (2009) the number of tilts was set to 7.

2.4 Colour SIFT

Colour SIFT expresses different ways of extending the SIFT descriptor from grey-level to colour images using colour moments and moment invariants (Mindru et al., 2004). The main goal is to (i) obtain invariance from colour description instead of grey-values description in particular for photometric events (such as shadow and highlights) and (ii) exploit the colour information to solve possible problems arising from the colour to grey conversion.
Bosch et al. (2006) compute SIFT descriptors over all three channels in the HSV colour space, resulting in a 3×128-dimensional HSV-SIFT image descriptor. Van de Weijer and Schmid (2006) concatenate the SIFT descriptor with a weighted Hue histogram. But this revealed some instabilities of the hue around the grey axis and that the hue histogram component of the descriptor is not invariant to illumination color changes or shifts.
Burghouts and Geusebroek (2009) defined a set of descriptors with 3 vectors of 128 values (following the opponent model of Eward Hering theory): the first vector is exactly the original intensity-based SIFT descriptor (representing the intensity, shadow and shading information), whereas the second and third vectors contain pure chromatic information as opponent colour channels (yellow–blue and red–green).
Other approaches are presented in Geusebroek et al. (2001) and Van de Sande et al. (2010).
In the presented tests, the implementation available at http://staff.science.uva.nl/~mark/downloads.html was used.

2.5 Shifting Gradient Location an Orientation Histogram (SGLOH)

SGLOH (Bellavia et al., 2010) is a modification of the GLOH descriptor (Mikolajczyk and Schmid, 2005) based on *n* circular grids centered on the feature point. SGLOH checks the similarity between two features not only in the gradient dominant orientation but also according to a set of discrete rotations. This is achieved by shifting the descriptor vector and by using an improved feature distance. This improves the descriptor stability to rotation for a reasonable computational cost. SGLOH descriptor is normally couple with the HarrisZ detector (Bellavia et al., 2008) for the extraction of the keypoints.
In the presented tests, the SGLOH implementation of Bellavia et al. (2010) is used with 3 circular rings centred on the feature point and 8 radial sectors per ring. Images needed to be down-sampled to 800x600 pixels.

2.6 DAISY

DAISY (Tola et al., 2010) is a local descriptor inspired by SIFT and GLOH but faster and more robust. In SIFT each bin contains a weighted sum of the norms of the image gradients around its centre, where the weights roughly depend on the distance to the bin centre. In DAISY these descriptors are reformulated so that they can be efficiently computed at every pixel location. This means that the histograms are computed only once per region and reused for all neighbouring pixels. To this end, the weighted sum of the norms is replaced with convolutions of the gradients in specific directions (normally 8) with several Gaussian filters. DAISY provides for a 264 dimensional vector and this formulation gives the descriptor the appearance of a flower, hence its name. DAISY gives the same kind of invariance as the SIFT and GLOH but is much faster for dense-matching purposes and allows the computation of the descriptors in all directions with little overhead (Winder et al., 2009).
In the presented tests, the DAISY implementation available at http://cvlab.epfl.ch/software/daisy was used.

2.7 Linear Discriminant Analysis (LDAHash)

LDAHash (Strecha et al. 2012) is a SIFT-like local binary feature descriptor that maps the descriptor vectors into the Hamming space, where the Hamming metric used to compare the resulting representations. LDAHash introduces a global optimization scheme to better take advantage of training data composed of interest point descriptors corresponding to multiple 3D points seen under different views. LDAHash performs a Linear Discriminant Analysis (LDA) on the descriptors before the binarization. Binarization techniques take advantage of training data to learn short binary codes whose distances are small for positive training pairs and large for others. This is useful to reduce the descriptor size and increase the performances of the descriptor. However LDAHash uses an exhaustive linear search to find the matching points, which reduces significantly its efficiency. Moreover LDAHash is a supervised and data-dependent approach that needs additional human labelling in the needed training stage. The approach is then fast and usable only when similar training data are available. In the presented tests the implementation available at http://cvlab.epfl.ch/research/detect/ldahash was used.

2.8 Speeded Up Robust Features (SURF)

The SURF descriptor (Bay et al. 2008) implements a similar algorithm to SIFT but reduces the processing time by simplifying and approximating the steps. All layers of the pyramid are generated from the original image by up-scaling the

filter size rather than taking the output from a previous filtered layer. The final descriptor vector has 64 dimensions. SURF can be computed efficiently at every pixel, but it introduces artefacts that can degrade the matching performance when used densely.

In the presented tests, the implementation available at http://www.mathworks.com/matlabcentral/fileexchange/13006-fast-corner-detector was used. We coupled SURF with the FAST detector (Rosten et al., 2010).

Dataset	*# img*	*Dimensions (WxHxD)[m]*	*Camera model*	*Sensor size [mm]*	*Image resol. [pixel]*	*Pixel size [mm]*	*Nominal focal length [mm]*
A	8	4 x 2 x 0.5	Kodak DSC Pro	36 x 24	4500x3000	0.008	35
B	21	1 x 1 x 0.2	Nikon D3X	35,9 x 24	6048x4032	0,006	50
C	39	54 x 19 x 1	Nikon D3100	23,1 x 15,4	4608x3072	0,005	18
D	48	19 x 11 x 5	Nikon D3100	23,1 x 15,4	4608x3072	0,005	18

Table 1: Main characteristics of the employed datasets for the evaluation of feature-based methods for tie point extraction.

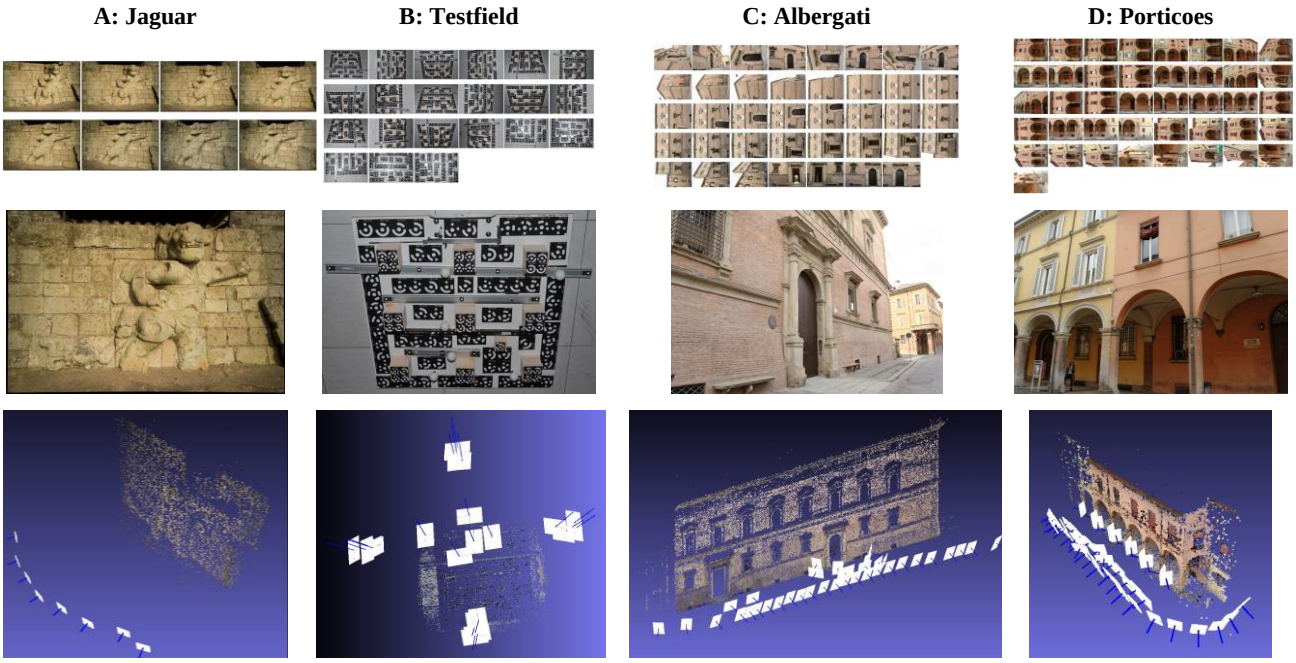


Figure 1: The employed datasets with their images and different camera networks: Jaguar (A), Testfield (B), Albergati (C), Porticoes (D). They feature high-resolution images, convergent acquisitions, variable image overlap, camera rolls, flat and textureless surfaces as well as repeated patterns and illumination changes.

3. DATASETS DESCRIPTION

The evaluation of the feature-based methods performances and potentialities is performed with four datasets (Fig. 1):
A) *Jaguar bass-relief*, a heritage monument located in Copan (Honduras) with uniform texture and highly overlapping convergent images. Ground truth measurements are available.
B) *Calibration testfield*, with coded targets and scale bars, imaged by highly overlapped and convergent images. Ground truth measurements are available for this datasets.
C) *Albergati building*, a three floors historical palace (54 x 19 m) characterized by repeated brick walls, stone cornices and a flat facade. The camera was moving along the façade of the building, with some closer shots of the entrances.
D) *Building with porticoes*, a three floors historical building (19 x 10 m) characterized by arches, pillars/columns, cross vault and plastered wall. The camera was moving along the porticoes, with some closer shots of the columns.
Other characteristics of the employed photogrammetric datasets are summarized in Tab.1. The datasets are characterized by different image scales (ranging from 1/800 for the Albergati case to 1/30 for the calibration dataset), image resolution, number of images, camera network, object texture and size. The employed datasets try to verify the efficiency of different techniques in different situations (scale variation, camera rotation, affine transformations, etc.). In particular datasets C and D represent urban test frameworks summarizing scenarios typical of historical urban environments. The datasets contain, besides convergent imaging configurations and some orthogonal camera rolls, a variety of situations typical of failure cases, i.e. 3D scenes (non-coplanar) with homogeneous regions, distinctive edge boundaries (e.g. buildings, windows/doors, cornices, arcades), repeated patterns (recurrent architectural elements), textureless surfaces and illumination changes. With respect to other evaluations where synthetic datasets, indoor scenarios, low resolution images, flat objects or simple 2-view matching procedures are used and tested, our datasets are more varied and our aim is the final scene's 3D reconstruction. The datasets are available to the scientific community for research purposes.

4. EXPERIMENTAL SETUP AND EVALUATION RESULTS

In order to have a common evaluation procedure, once the feature points are extracted and described with the aforementioned algorithm implementations, the descriptors matching procedure, the outlier detection phase and the final bundle adjustment are run inside the same software environment. Particularly, the generation of the correct image

correspondences is performed following (Agarwal et al., 2009; Frahm et al., 2010) and then RANSAC to eliminate possible mismatches - or one of its variants (Chum et al., 2004; Chum et al., 2005; Chum & Matas, 2005, 2008). Other similar approaches are presented in (Nister & Stewenius, 2006; Farenzena et al., 2009).

The performed tests compare the results achieved at the end of the descriptor matching phase and after the bundle solution to better understand the performances of the feature-based methods for 3D reconstruction purposes.

In particular, the following outcomes were analysed:

- *pairwise matching efficiency*: using a set of images (Fig. 2) featuring illumination differences, textureless surfaces, possible loss of information in the colour-to-grey conversion and elements with strong 3D features, we tested pairwise matching efficiency of the operators with respect to three camera movements: (i) parallel with limited baseline (00-01); (ii) rotation of 90° (00-03); (iii) tilt of more than 30° (01-02). The number of correct inlier matches (after the RANSAC phase) is then normalized with all putative correspondences:

$$efficiency = \frac{\#\ inliers}{\#\ putative\ corrispondances}$$

Figure 2: Images used to test pairwise matching efficiency.

The optimized SiftGPU obtained the higher number of correct inlier for each situation (Table 2). This is probably due to the variation of the DoG threshold in the dark areas, allowing a higher number of matching. Conversely, ASIFT seems the more efficient solution from the efficiency point of view (Table 3).

	PARALLEL 00 - 01	*ROTATE 90° 00 - 03*	*TILT 45° 01 - 02*
ASIFT	1354	1079	224
COLSIFT	590	263	52
DAISY	536	451	67
LDAHash	1482	1703	187
SGLOH	172	86	41
SIFT+GPU	964	1033	116
SIFTGPUoptim	**2222**	**2679**	**330**
FAST + SURF	80	36	13
VLFeat3.4	1022	1344	239
VLFeat 6	440	388	99

Table 2: Total number of correct inliers for each operator.

	PARALLEL 00 - 01	*ROTATE 90° 00 - 03*	*TILT 45° 01 - 02*
ASIFT	**0,880**	**0,769**	0,66
COLSIFT	0,693	0,553	0,65
DAISY	0,701	0,731	**0,663**
LDAHash	0,657	0,553	0,539
SGLOH	0,524	0,434	0,164
SIFT+GPU	0,593	0,532	0,370
SIFTGPUoptim	0,698	0,553	0,445
FAST + SURF	0,176	0,099	0,03
VLFeat3.4	0,652	0,556	0,571
VLFeat 6	0,652	0,617	0,526

Table 3: Efficiency of each operator.

- *number of oriented cameras*: Fig. 3 shows that ColSIFT and FAST+SURF achieve poor results in case of b/w objects (Testfield) and scenes with uniform colour (Portici). Best performances are obtained with ASIFT, LDAHash, SiftGPU and VLFeat.

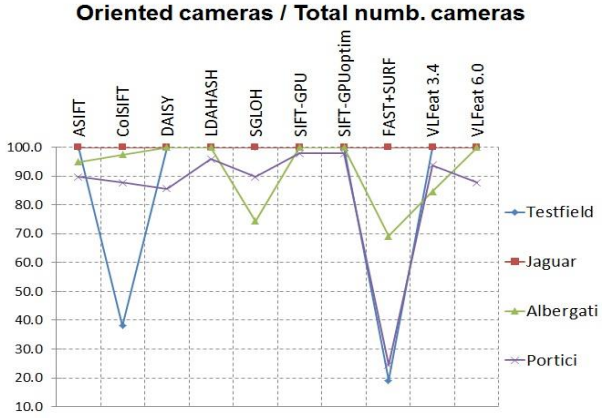


Figure 3: Percentage of oriented cameras for each dataset and feature-based method.

- *root mean square error of the bundle adjustment* (Fig. 4): it expresses the re-projection error of all computed 3D points. ASIFT and SIFT-GPU have limited results in all the datasets (last one probably due to changes in the parameter values compared with the original one). VLFeat shows rather good results as well as FAST+SURF although not able to orient all the images in the datasets (just 20% for the Testfield and Porticoes dataset).

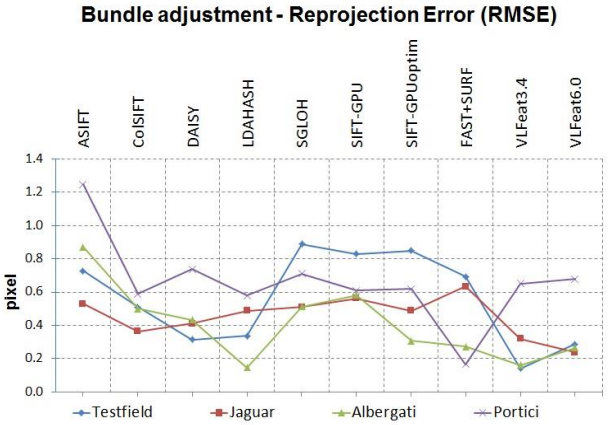


Figure 4: Results of the bundle adjustment for each dataset in terms of reprojection error.

- *visibility of 3D points in more than 3 images* (Fig. 5): the results show that more than 50% of the triangulated points are visible at least in 3 images, although for highly overlapped images (Testfield and Jaguar) this might not be so significant.

- *average rays per 3D points* (i.e. the redundancy of the computed 3D object coordinates): for the Jaguar and Testfield datasets the point multiplicity is shown in Fig. 6a and Fig. 7a. In comparison with the ground truth measurements and despite the high overlap of the images (almost 100% for the entire dataset), the feature-based methods show a low average multiplicity.

- *average intersection angles per 3D points*: as 3D from images is determined by the triangulation, a higher intersection angle of homologues rays provides for more accurate 3D information. The Testfield and Jaguar datasets consist of highly overlapping images (almost 100%) acquired with very convergent image,

therefore high intersection angles should be expected. On the other hand, the average angles (Fig. 6b and 7b) are always quite low, compared to the ground truth measurements.

- *points per intersection angle*: the analysis of the Jaguar dataset (Fig. 6c) shows that less than 10% of the correspondences provide for a 3D point under an angle larger than 60 degrees. Instead more than 30% of the 3D points are determined with an intersection angle smaller than 20 degrees (LDAHash up to 50%, VLFeat3.4 almost 40%). The results of the Testfield dataset are pretty similar and all the feature-based methods normally provide 3D points with a small intersection angle (Fig. 7c).

- *theoretical precision of the computed 3D object coordinates* (for the Testfield dataset): in comparison with the ground truth values (σx=0.01 mm, σy=0.009 mm, σz=0.017mm – Z is the depth axis), all the feature-based methods deliver much higher accuracy (Fig. 8a). SGLOH has so high values due that the used implementation process only low-resolution images. Observing Fig.6c and Fig. 7c - i.e. fact that a large number of 3D points are determined with an intersection angle smaller than 10 degrees – after removing all those points we obtained much better theoretical precisions of the object coordinates (Fig. 8b).

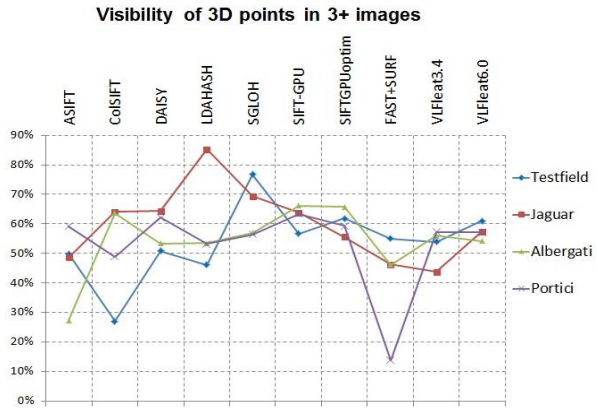
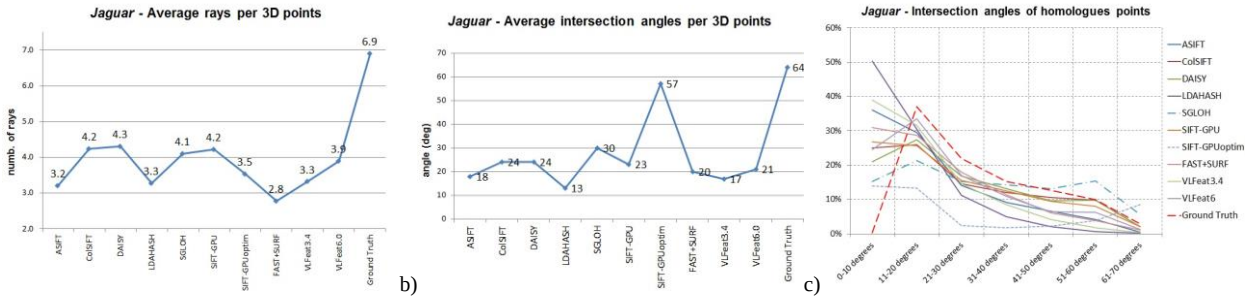
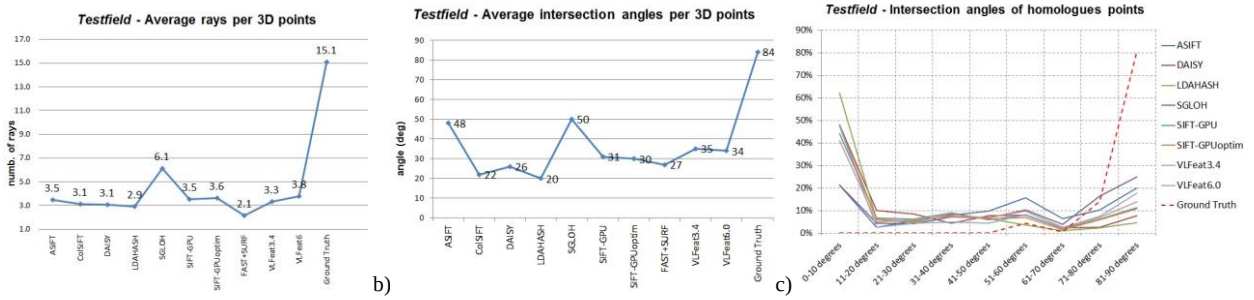


Figure 5: The visibility of the derived 3D points in more than 3 images (normalized with respect to the all extracted points).



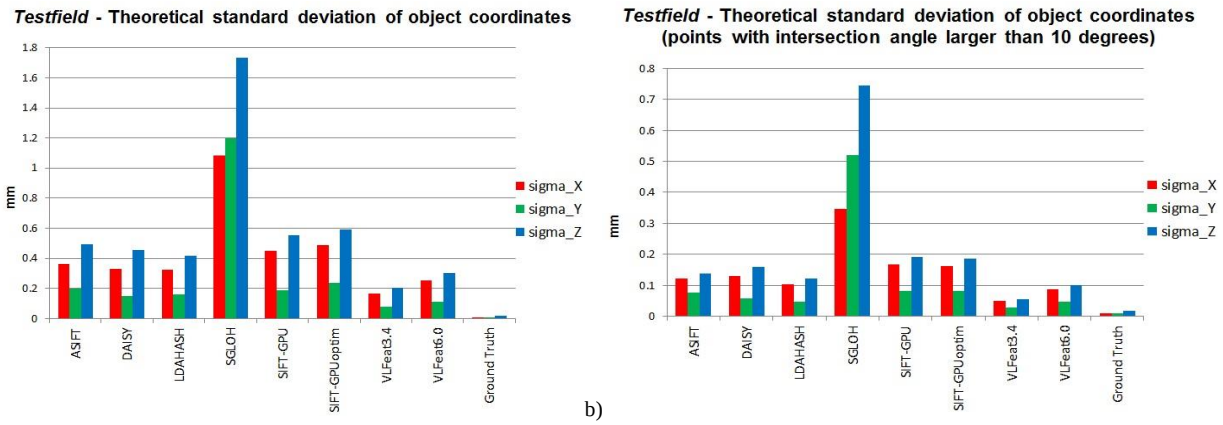
a) b) c)

Figure 6: The Jaguar dataset: average rays per computed 3D points (a), average intersection angles (b) and normalized number of points wrt the intersection angles (c).



a) b) c)

Figure 7: The Testfield dataset: average rays per computed 3D points (a), average intersection angles (b) and normalized number of points wrt the intersection angles (c).



a) b)

Figure 8: Derived theoretical precision of the computed object coordinates for the Testfield dataset: statistics for all 3D points (a) and for the 3D points with an intersection angle larger than 10 degrees (b). The SGLOH method has low accuracy results as the used implementation can process only low-resolution images (800x600 pixel). ColSIFT and SURF, as they oriented a very low number of cameras, were not included in this analysis.

5. CONCLUSIONS

The paper reports some experiments and evaluations carried out to test the performances and efficiency of common feature-based methods used for automated homologues point extraction. Different datasets were used featuring scale variations, camera rotations, illumination changes, affine transformations, variable image overlap and resolution, flat and textureless surfaces, repeated patterns. Image correspondences were extracted following the typical detection, description, matching and blunder detection phases. Then a bundle adjustment was used to derive the camera poses and sparse 3D reconstructions. The achieved results were compared and the following considerations can be summarized:

- real and complex scenarios show that the automated image orientation is still an open issue and that unsuccessful results can still be achieved;
- each method has a set of parameters which needs to be correctly set otherwise its performances can be very poor (no unique set of parameter is valid in all the situations);
- repeated patterns and 3D scenarios, very common in architectural scenes, show the necessity of more invariant descriptors vectors;
- all the methods cannot detect correspondences on longer track of images and deliver 3D points with small intersection angles;
- the small intersection angles affect negatively the quality of the 3D reconstruction but, given the large number of extracted correspondences, the low-angle intersections can be removed;
- the processing time can be considerably high in case of high-resolution images and descriptors involving multiple detection routines or involving multiple channels;
- comparing all the graphs, some operators have limited discrepancies, particularly SIFT-GPU and VLFeat which seem to be the more stable.

Beside these considerations, we cannot declare any winner. For sure fully automated feature-based methods combined with accurate and reliable results are still a hot research topic.

REFERENCES

Aanæs H., Lindbjerg Dahl A., Steenstrup Pedersen K., 2012: Interesting Interest Points - A comparative study of interest point performance on a unique data set. *Int. Journal of Computer Vision*, Vol. 97(1), pp. 18-35

Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R., 2009: Building Rome in a day. Proc. *ICCV*, Kyoto, Japan

Apollonio, F.I., Fallavollita, F., Gaiani, M., Sun, Z., 2013: A colour digital survey of arcades in Bologna. M. Rossi (ed.), *Colour and Colorimetry. Multidiscipliary contribution*, Vol. IXb, Rimini, pp. 58-68

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*, Vol. 110(3), pp. 346-359

Barazzetti, L., Scaioni, M., Remondino, F., 2010: Orientation and 3D modeling from markerless terrestrial images: combining accuracy with automation. *The Photogrammetric Record*, Vol. 25(132), pp. 356-381

Barazzetti, L., Mussio, L., Remondino, F., Scaioni, M., 2011: Targetless camera calibration. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 38(5/W16)

Bay H., Ess A., Tuytelaars T., Van Gool L., 2008: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, Vol. 110(3), pp. 346-359

Bellavia, F., Tegolo, D., Valenti, C., 2008: A non-parametric scale-based corner detector. Proc. ICPR, pp. 1-4

Bellavia, F., Tegolo, D., Trucco, E., 2010: Improving SIFT-based descriptors stability to rotations. Proc. *ICPR*

Bosch A., Zisserman A., Munoz X., 2006: Scene classification via pLSA. Proc *ECCV*, pp. 517-530

Brown, M., Winder, S., Szeliski, R., 2005: Multi-image matching using multi-scale oriented patches. Proc. *CVPR*, pp. 510-517

Burghouts, G. J., Geusebroek, J. M., 2009: Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, Vol. 113, pp. 48-62

Calonder, M., Lepetit, V., Fua, P., 2010: BRIEF: Binary Robust Independent Elementary Features. Proc. *ECCV*, pp. 778-792

Chum, O., Matas, J., Obdrzalek, S., 2004: Enhancing RANSAC by generalized model optimization. Proc. *ACCV*

Chum, O., Matas, J., 2005: Matching with PROSAC – progressive sampling consensus. Proc. *CVPR*

Chum, O., Werner, T., Matas, J., 2005: Two-view geometry estimation unaffected by a dominant plane. Proc. *CVPR*, pp. 772-779

Chum, O., Matas, J., 2008: Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,Vol.30(8), pp. 1472-1482

Del Pizzo, S., Troisi, S., 2011: Automatic orientation of image sequences in Cultural Heritage. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 38(5/W16)

Deseilligny, M. P., Clery, I., 2011. Apero, an open source bundle adjustment software for automatic calibration and orientation of set of images. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 38(5/W16)

Farenzena, A.M., Fusiello, A., Gherardi, R., 2009: Structure-and-Motion pipeline on a hierarchical cluster tree. Proc. of the IEEE *Int. Workshop on 3-D Digital Imaging and Modeling*

Foerstner, W., Guelch, E., 1987: A fast operator for detection and precise location of distinct points, corners and center of circular features. *ISPRS Conference on Fast Processing of Photogrammetric Data*, Interlaken, Switzerland, pp. 281-305

Frahm J.-M., Fite-Georgel P., Gallup D., Johnson T., Raguram R., Wu C., Jen Y.-H., Dunn E., Clipp B., Lazebnik S., Pollefeys M., 2010: Building Rome on a cloudless day. Proc. *ECCV*, pp. 368-381.

Fraundorfer, F., Bischof, H., 2004: Evaluation of local detectors on non-planar scenes. *Proc. Austrian Association for Pattern Recognition (AAPR)*, pp, 125-132.

Geusebroek, J. M., van den Boomgaard, R., Smeulders, A. W. M., Geerts, H., 2001: Color invariance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 23(12), pp.1338-1350

Harris, C., Stephens, M., 1988: A combined edge and corner detector. Proc. *Alvey Vision Conference*, pp. 147-151

Heinly J., Dunn E., Frahm J.-M., 2012: Comparative evaluation of binary features. Proc. 12th *ECCV*, pp. 759-773

Kadir, T., Zissermann, A., Brady, M., 2004: An affine invariant salient region detector. Proc. 8th *ECCV*

Ke, Y., Sukthankar, R., 2004: PCA-SIFT: A more distinctive representation for local image descriptors. Proc. *CVPR*, pp. 506-513

Klein, G., Murray, D., 2008: Improving the agility of keyframe-based SLAM. Proc. *ECCV*

Juan, L., Gwon, O., 2009: A comparison of SIFT, PCA-SIFT and SURF. *Int. Journal of Image Processing*, Vol. 3(4), pp. 143-152

Leutenegger, S., Chli, M., Siegwart, R., 2011: BRISK: Binary Robust Invariant Scalable Keypoints. Proc. *ICCV*, pp. 2548-2555

Lindeberg, T., 1998: Feature detection with automatic scale selection. *Int. Journal of Computer Vision*, Vol. 30(2), pp. 79-116

Lowe, D., 2004: Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, Vol. 60(2), pp. 91-110

Mair, E., Hager, G., D., Burschka, D., Suppa, M., Hirzinger, G., 2010: Adaptive and generic corner detection based on the accelerated segment test. Proc. ECCV

May M., Turner M. J., Morris T., 2010: Scale Invariant Feature Transform: a graphical Parameter Analysis. Proc. *BMVC 2010 UK postgraduate workshop*

Mikolajczyk, K., Schmid, C., 2004: Scale and affine invariant Interest point detectors. *Int. Journal Computer Vision*, Vol. 60(1), pp. 63-86

Mikolajczyk, K., Schmid, C., 2005: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27(10)

Mikolajczyk, K, Kalal, Z, Matas, J., 2010: PN Learning: Bootstrapping binary classifiers by structural constraints. Proc. *CVPR*, pp. 49-56

Mindru, F., Tuytelaars, T., Van Gool, L., Moons, T., 2004: Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, Vol. 94(1-3), pp. 3-27

Morel J-M., Yu, G., 2009: ASIFT: a new framework for fully affine invariant comparison. *SIAM Journal on Imaging Sciences*, Vol. 2(2), pp. 438–469

Moravec, H.P., 1979: Visual mapping by a robot rover. Proc. *6th Int. Joint Conference on Artificial Intelligence*, pp. 598-600

Nister, D., 2003: Preemptive RANSAC for live structure and motion estimation. Proc. *ICCV*, pp. 199-206

Nister, D., Stewenius, H., 2006: Scalable recognition with a vocabulary tree. Proc. *CVPR*, Vol. 5

Oyallon, E., Rabin, J., 2013: An Analysis and implementation of the SURF method and its comparison to SIFT. *IPOL Journal - Image Processing On Line*, pre-print

Remondino, F., 2006: Detectors and descriptors for photogrammetric applications. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 36(3), pp. 49-54

Remondino, F., Del Pizzo, S., Kersten, T., Troisi, S., 2012: Low-cost and open-source solutions for automated image orientation – A critical overview. Proc. EuroMed 2012 Conference, *LNCS* 7616, pp. 40-54

Rodehorst, V., Koschan, A., 2006: Comparison and evaluation of feature point detectors. Proc. *5th International Symposium Turkish-German Joint Geodetic Days*

Roncella, R., Re, C., Forlani, G., 2011: Performance evaluation of a structure and motion strategy in architecture and Cultural Heritage. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 38(5/W16)

Rosten E., Porter, R., Drummond T., 2010: Faster and better: a machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 32, pp. 105-119

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011: ORB: an efficient alternative to SIFT or SURF. Proc. *ICCV*

Schaffalitzky, F. and Zisserman, A., 2002: Multi-view matching for unordered image sets. Proc. *ECCV*

Smith, S.M., Brady, J.M., 1997: SUSAN – a new approach to low level image processing. *Int. Journal Computer Vision*, Vol. 23(1), pp. 45-78

Schmid C., Mohr R., Bauckhage C., 2000. Evaluation of interest point detectors. *Int. Journal of Computer Vision*, Vol. 37(4), pp. 151-172

Snavely, N., Seitz, S.M., Szeliski, R., 2008: Modeling the world from internet photo collections. *Int. Journal of Computer Vision*, Vol. 80(2), pp. 189-210

Strecha, C., Bronstein, A., Bronstein, M., Fua P., 2012: LDAHash: Improved matching with smaller descriptors. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 34(1), pp. 66-78

Sinha, S.N., Frahm, J.M., Pollefeys, M., Genc, Y., 2006: GPU-based video feature tracking and matching. Proc. *Edge Computing Using New Commodity Architectures workshop*

Tola E., Lepetit V., Fua P., 2010: DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32(5)

Tuytelaars, T., Van Gool, L., 2004: Matching widely separated views based on affine invariant regions. *Int. Journal of Computer Vision*, Vol. 59(1), pp. 61-85

Ullman, S., 1979: The interpretation of Structure from Motion. Proc. *Royal Society London*, Vol. 203(1153), pp. 405-426

van de Sande, K. E. A., Gevers, T., Snoek C. G. M., 2010: Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32(9), pp. 1582-1596

van de Weijer, J., Schmid, C., 2006: Coloring local feature extraction. Proc. *ECCV*, pp. 334-348

Van Gool, L., Moons, T., Ungureanu, D., 1996: Affine / photometric invariants for planar intensity pattern. Proc. 4th *ECCV*, pp. 642-651

Vedaldi, A., Fulkerson, B., 2010: VLFeat - An open and portable library of computer vision algorithms. Proc.18th *ACM Intern. Conf. on Multimedia*

Yeo, C., Ahammad, P., Ramchandran, K., 2011: Coding of image feature descriptors for distributed rate-efficient visual correspondences. *Int. Journal of Computer Vision*, Vol. 94(3), pp. 267-281

Winder, S., Gang, H., Brown, M., 2009: Picking the best DAISY. Proc. *CVPR*, pp.178-185

Wu, J., Cui Z., Sheng V.S., Zhao P., Su D., Gong S., 2013. A comparative study of SIFT and its variants. *Measurement Science Review*, Vol. 13(3)

Wu, C., 2013: Towards linear-time incremental Structure from Motion. *Proc. 3D Vision*, pp. 127-134

Wu, C., 2014: SiftGPU: A GPU implementation of Scale Invariant Feature Transform (SIFT). http://cs.unc.edu/~ccwu/siftgpu/ (last access: 04 May 2014)

Zhao, X., Zhou, Z., Wu, W., 2012: Radiance-based color calibration for image-based modeling with multiple cameras. *Science China Information Sciences*, Vol. 55(7), pp.1509-1519.