Selecting Feature Detectors for Accurate Visual Odometry

ALDO CUMANI and ANTONIO GUIDUCCI
Istituto Nazionale di Ricerca Metrologica
Str. delle Cacce, 91, I-10135 Torino
ITALY
a.cumani@inrim.it, a.guiducci@inrim.it

Abstract: - This work analyzes the performances of different feature detectors/descriptors in the context of incremental path estimation from passive stereo vision (Visual Odometry). Several state-of-the-art approaches have been tested, including a fast Hessian-based feature detector/descriptor developed at INRIM. Tests on both synthetic image sequences and real data show that in this particular application our approach yields results of accuracy comparable to the others, while being substantially faster and much more reliable.

Key–Words: - Robot localisation, Stereo vision, Visual odometry, Feature descriptors, Feature detectors

1 Introduction

Research in autonomous sensor-based navigation has received considerable attention in recent years, particularly in connection with planetary exploration tasks [1]. A mobile platform (rover) on, say, Mars must be capable of truly autonomous navigation in a mostly unknown environment, as its continuous teleoperation from an Earth station is clearly out of question. This requires that the rover be able to build, based on sensor measurements, a metric and topological model of its environment, while simultaneously estimating its position relative to the environment itself. This is the task of Simultaneous Localisation and Mapping (SLAM) studies [2]. SLAM algorithms have been proposed, using various kinds of sensors; among these the vision sensor, being the one that provides the largest amount of information, has been extensively studied both in single-camera [3, 4, 5] and in multi-camera setups [6, 7, 8].

Some navigation tasks, however, may require accurate localisation, i.e. accurate estimation of the rover path. Indeed, pure dead reckoning (wheel odometry) usually yields quite poor estimates (due e.g. to wheel slippage); also, wheel odometry alone can at most yield a 2D path estimate, so it is not even sufficient in principle when navigating on a non-planar surface, and must be complemented by other independent inputs (e.g. an absolute orientation sensor as in [1]).

On the other hand, vision-based path estimation (*Visual Odometry*) is able to yield accurate results while being intrinsically 3D. In this regard, it is important to note the following points:

- Any path estimate from onboard visual measurements is necessarily *incremental*, i.e. resulting from the sum of smaller motion estimates. It is therefore of utmost importance that each individual step be as accurate as possible, to reduce error accumulation.
- Monocular vision is subject to scale uncertainty, so, in absence of landmarks of known size (as is the case in planetary exploration), stereo or other multi-camera setups are needed.
- As error accumulation in the long run is unavoidable, it is important that the rover be able to recognise places where it has been before, and to use such information to correct its pose estimate.

In this context, our group at INRIM has developed a visual odometry algorithm [9, 10, 11, 12] which relies on the tracking of pointwise image features extracted from the images acquired by an onboard binocular stereo head. At intervals along the rover trajectory, its motion is estimated by robust bundle adjustment of the tracked features in the four images (two before and two after the motion). Several kinds of point features have been tested to this end, and a new Fast-Hessian based feature detector/descriptor, similar to SURF [13] has been developed.

A *déjà vu* mechanism for exploiting cyclic paths has also been devised, by periodically storing observed features and pose estimates, and comparing currently observed features to stored ones when near a saved position.

This work focusses on the problem of choosing the best feature detector/descriptor method, presenting the results of several tests both on simulated and real image data. Sec. 2 summarizes the algorithm and the main sources of inaccuracy; Sec. 3 focusses on

feature detectors and descriptors; finally, Sec. 4 reports and discusses the performance of the algorithm on both simulated and real image sequences.

2 Visual odometry

Our algorithm relies on accumulating relative motions, estimated from corresponding features in the images acquired while the rover is moving. Such estimates are computed at *key frames*, whose spacing is a compromise between larger intervals, desirable both for numerical accuracy and for reducing computations, and the need for a sufficiently large number of features, which naturally tend to be lost because going out of view. The algorithm can be so summarized:

Feature extraction and tracking. Point features are extracted at each key frame and left-right matched. Matched features are then tracked in subsequent frames.

Motion estimation. The relative motion of the rover between the current frame and the last key frame is estimated by robust bundle adjustment of matched feature points.

Déjà vu correction. Features and pose estimates are periodically saved. When the rover believes to be near a saved position, observed features are compared to the stored ones, and a pose correction is possibly computed by bundle adjustment.

The motion estimation and the *déjà vu* mechanism are described in more detail elsewhere [14]. To summarize, the relative roto-translation $(\mathbf{r}, \mathbf{t})$ of the rover between two keyframes is found by minimizing a suitable function of the image-plane backprojection errors of the observed points. For robustness, a Lorentzian cost function $f(e^2) = \log(1 + e^2/\sigma^2)$ is used, with σ chosen as a function of the expected image-plane error. The residual errors are then compared against a threshold θ (proportional to the Lorentz σ), and those exceeding that threshold are marked as outliers. Bundle adjustment is then run again on the inliers only, using the standard sum-of-squares error function.

As concerns keyframe determination, in our approach we have adopted the following strategy:

- start with a predefined maximum inter-frame step, say 20 frames;
- if the number of tracked points is enough, go on; else halve the step and retry.

This strategy can be used in a realtime application, provided that a sufficiently large memory buffer be available to store the incoming max number of images between two keyframes.

The above algorithm is subject to the accumulation of the errors made in each individual step. Such errors essentially come from three distinct sources:

Matching errors: feature points in distinct images can be wrongly matched. As long as there are enough matched points, this problem can be alleviated by using a *robust* estimation method, able to detect such wrong matches (outliers), as explained above. It is anyway desirable to start with as many good matches as possible, and this is influenced by the choice of feature descriptors.

Localization errors: even neglecting mismatches, feature points must be *repeatable*, i.e. they must not disappear in different views, and *robust* against viewpoint changes, i.e. the image plane locations of matched features must be the projections of the same 3D point. This problem can be tackled by a careful choice of feature detector.

Insufficient features: detected features must both be numerous and rich in 3D structure, i.e. the corresponding 3D points must neither lie on some singular configuration, nor be too spatially concentrated. The distribution of visual features in the scene is obviously not under control, but again the choice of feature detector may heavily affect the result.

It must also be noted that each step estimates a full *pose* change of the robot (3D rotation $\mathbf{r}$ and 3D translation $\mathbf{t}$). The effects of errors in the rotation and translation parts are not the same, however, on the subsequent estimated path. While an error in $\mathbf{t}$ at a particular step just induces an equivalent position error on the portion of path afterwards, an error in $\mathbf{r}$ induces a position error which grows linearly with the distance from that point. This means that rotation error is generally more important than translation error.

It is clear from the above discussion that a key issue for getting reliable and accurate results is the way image features are detected and represented. This is the subject of the next Section.

3 Feature detectors and descriptors

In general, (point) feature extraction consists of two steps:

Detection, aiming at the localisation of visually salient points in the image, and at the determination of the apparent size (scale) of the visual feature.

Description, aiming at representing in a compact form the image behaviour in the vicinity of the detected point. This representation typically consists of a fixed number N of values, that can be interpreted as points in a Euclidean space E^N. This way, the similarity of features from different images can be

easily computed from Euclidean distance of their representations.

Good references on state-of-the-art detectors are in [15, 16]. Usually, detectors act by searching the image for the extrema of some local function of the smoothed luminance, such as the Harris operator or the Hessian (though other criteria may be used, e.g. edge line intersections). The size (scale) of the feature is then found at the scale-space extrema of some other operator (e.g. the Laplacian), possibly followed by an iterative refinement aiming at finding the affine shape of the feature, as in [17].

There is as well a vast literature on the subject of feature descriptors; again, a good reference is [18].

Following the terminology of [16] and [18], we have tested the following detectors and descriptors (the italicized acronyms are for subsequent reference):

Detectors:

- Harris-Laplace detector (*harlap*)
- Hessian-Laplace detector (*heslap*)
- Harris-Affine detector (*haraff*)
- Hessian-Affine detector (*hesaff*)
- Harris-Hessian-Laplace detector (*harhes*)
- Edge-Laplace detector (*sedgelap*)

Descriptors:

- Freeman's steerable filters (*jla*)
- Lowe's Scale Invariant Feature Transform (*sift*)
- Gradient Location-Orientation Histogram (extended SIFT) (*gloh*)
- Van Gool's moment invariants (*mom*)
- Spin image (*spin*)
- cross-correlation of image patches (*cc*)

In addition to the above, we have as well tested **SURF** features [13] and our homebrew **CVL** features [19]. Speeded-Up Robust Features use the Hessian for detecting interest points both in image space and in scale space. A rotation-invariant (but not affine-invariant) descriptor of size 64 or 128 is then computed by wavelet analysis of the luminance in an image patch, around the detected point. There is also a non-rotationally-invariant version (U-SURF) which has the advantage of faster computation.

Our own feature detector/descriptor [19] (in the following referred to as CVL) is a simplified form of SURF: detection is accomplished, as in the latter, by searching for image (x, y) and scale (σ) extrema of the Hessian. A descriptor of size 64 is then computed by mapping a square image area of size 10σ around (x, y) to size 8×8, then normalizing these pixel values to zero-mean and unit variance. This descriptor is somehow similar to the cross-correlation (*cc*) one mentioned above; it is neither rotation- nor affine-invariant, but is quite fast to compute (four times faster than SURF) and has proven quite good for the visual odometry application.

Figure 1: The ActivMedia P2AT rover with stereo head and acquisition PC.

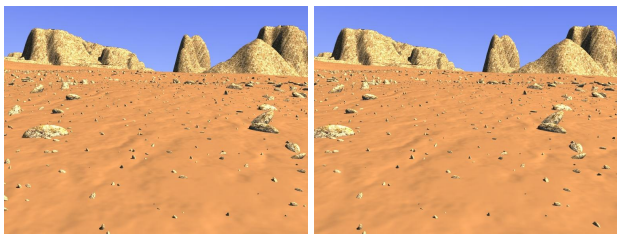

Figure 2: Sample stereo pair of the simulated environment.

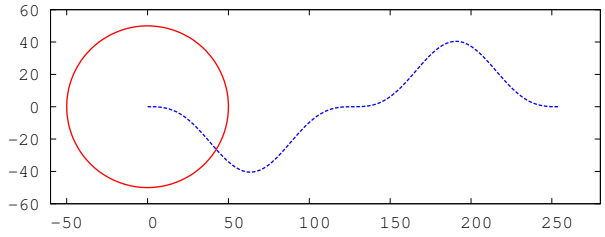

Figure 3: Simulated paths (coordinates in m).

4 Simulation

The visual odometry algorithm has been implemented as an application running in real time (when using our CVL features), on a Linux laptop onboard a small commercial rover (ActivMedia Pioneer 2AT, Fig. 1).

Assessing the accuracy of the path estimate, however, would require good measurements of the rover position (and possibly attitude), at least at every keyframe. While we have done some tests on rover paths in a limited area (size $\sim$ 10m), using optical

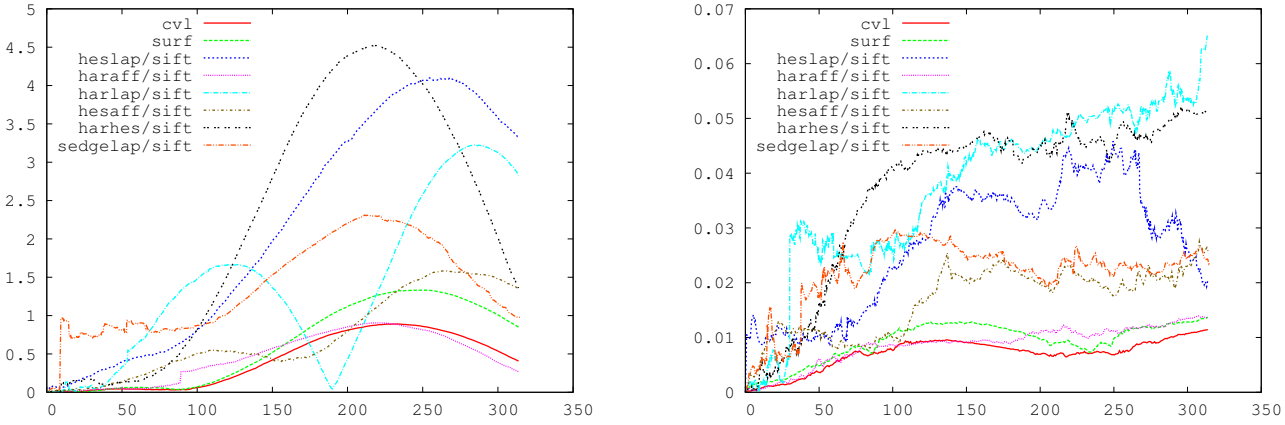

Figure 4: Simulated circular path. Left: position error (m) vs. path length (m), right: rotation error (rad) vs. path length (m), for various detectors.

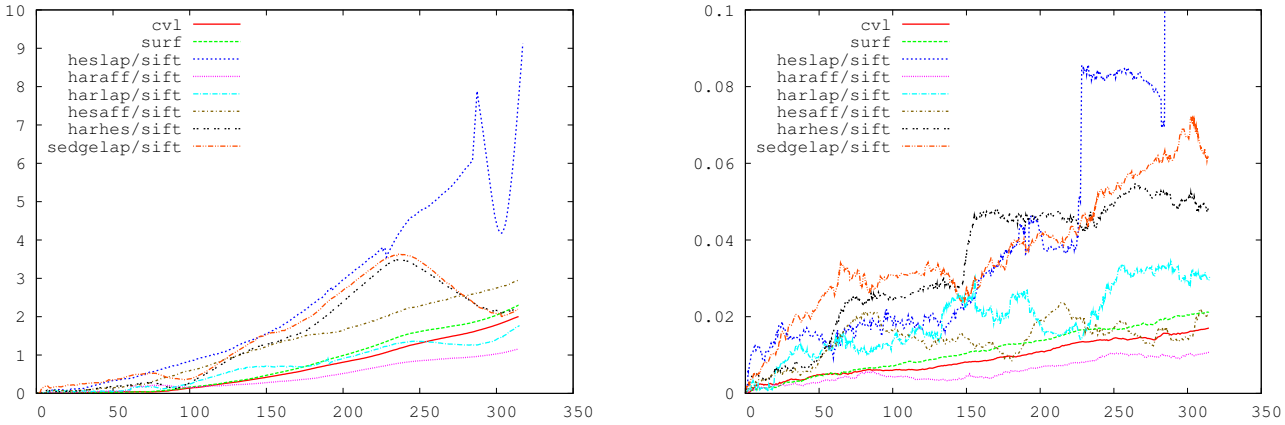

Figure 5: Simulated waving path. Left: position error (m) vs. path length (m), right: rotation error (rad) vs. path length (m), for various detectors.

targets for independent measurements of the rover position [19], this is impractical for paths of any useful size.

We have therefore resorted to simulation. Image sequences of a synthetic Mars-like landscape (see Fig. 2) have been generated using the free raytracer POV-Ray [20]. Lens distortion and acquisition noise were not included in the simulation. The stereo base (0.24m), head inclination (about 13°) and imaging parameters (image size, focal length and rate) were chosen to mimick the actual behaviour of our rover.

Two different trajectories were simulated: a circular one, with radius 50m, and a waving one of the same total length (see Fig. 3). Each sequence consisted of 7392 stereo pairs of size 768×576, imaged with an equivalent focal length of 766 pels (horizontal FOV $\sim 53°$).

In the following we compare the results of the visual odometry algorithm using the different feature detectors/descriptors discussed above. For SURFs we have used the author's implementation available at [13], and for all other feature detectors/descriptors mentioned in Sec. 3, the `extract_features.ln` package available at [21].

In our implementation, feature matching is based on the nearest-neighbor-ratio strategy, i.e. two features a and b in different images are matched if b is the nearest neighbor of a in feature parameter space, *and* their distance d_{ab} is less then r times that of the second nearest neighbor. A lower ratio r discriminates better among similar features, but obviously yields a lower number of matches. In the tests here reported, we have set $r = 0.9$; although our CVL features gave significantly better results at $r = 0.7$, with the latter value most of the other features did not yield a sufficient number of 4-matches, i.e. features matched over all the four images of a keyframe pair, over the whole trajectory.

As concerns motion estimation, the Lorentzian scale σ and outlier detection threshold θ (Sec. 2) were

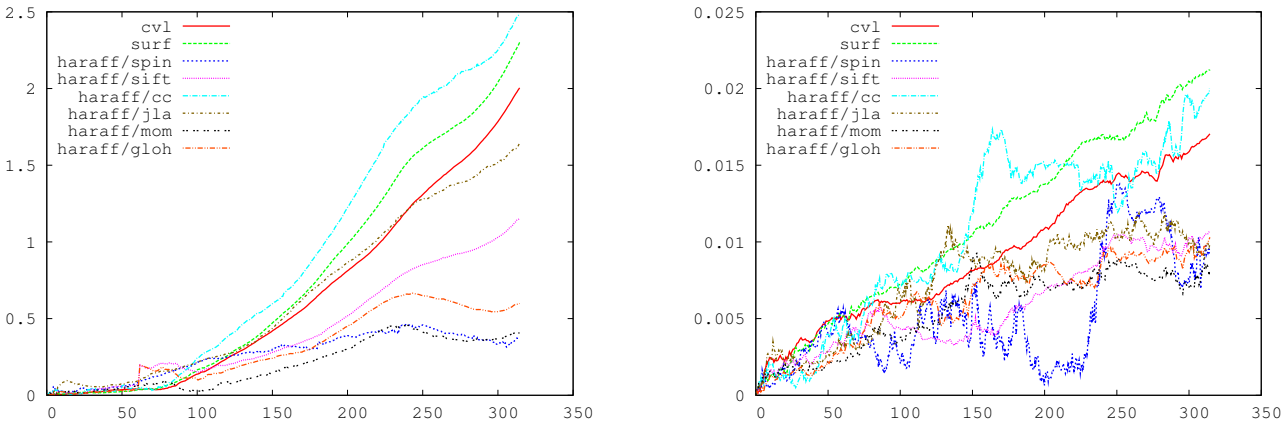

Figure 6: Simulated waving path. Left: position error (m) vs. path length (m), right: rotation error (rad) vs. path length (m), for various descriptors.

set at $\sigma = 1$ pel and $\theta = 1$ pel. The adaptive keyframe approach was used, with maximum keyframe step 20.

Fig. 4, referring to the circular path, reports (left) the position error (distance between the estimated head position and its true value) versus the estimated path length, using the detectors of Sec. 3 in combination with the SIFT descriptor, plus SURF and CVL. For the same tests, Fig. 4 (right) plots the rotation error, defined as the absolute value of the angle defined by the rotation matrix $R_E^{-1} R_T$, with R_T the true rotation of the stereo head with respect to the initial pose, and R_E the same as estimated by the visual odometry algorithm.

Similarly, Fig. 5 shows the position and rotation errors for the waving path, again using the detectors of Sec. 3 in combination with the SIFT descriptor, plus SURF and CVL.

As these graphs suggest a better accuracy of the Harris-Affine detector, more tests have been done using that detector in combination with several of the descriptors mentioned in Sec. 3. The results are shown in Fig. 6, with SURF and CVL added for comparison.

Table 1 summarises some relevant statistics, i.e. the average inter-keyframe step, and average numbers of features, namely: total detected features in each image, 4-matched features and inliers over each keyframe pair.

These graphs, particularly those of the circular path, confirm what said in Sec. 2 about the impact of rotation errors; indeed, the sinusoidal behaviour of the position error in Fig. 4 looks like a long-term effect of an early accumulated angular error.

These results show that the CVL detector/descriptor accuracy is comparable to that of the Harris-Affine detector, although the latter, with some descriptors, seems to perform better. However, CVL

features	*step 0..20*	*total 0..5000*	*4-match 0..500*	*inliers 0..250*
circular path				
haraff/cc				
haraff/gloh				
haraff/jla				
haraff/mom				
haraff/sift				
haraff/spin				
harhes/sift				
harlap/sift				
hesaff/sift				
heslap/sift				
sedgelap/sift				
surf				
cvl				
waving path				
haraff/cc				
haraff/gloh				
haraff/jla				
haraff/mom				
haraff/sift				
haraff/spin				
harhes/sift				
harlap/sift				
hesaff/sift				
heslap/sift				
sedgelap/sift				
surf				
cvl				

Table 1: Average keyframe step, total features, 4-matched features and inliers for all the simulation tests.

yields many more good matches (inliers) than *haraff* (Table 1), a big advantage in case of large image changes, and with some tuning (lower r) CVL performs significantly better than shown in Fig. 6. This is confirmed by the tests on real data reported in the next Section.

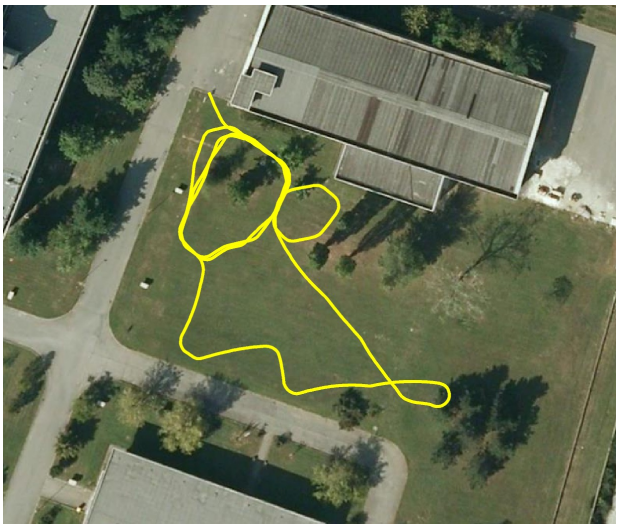

Figure 7: INRIM data I: top view of location.

Figure 8: INRIM data I: sample stereo pair of the environment.

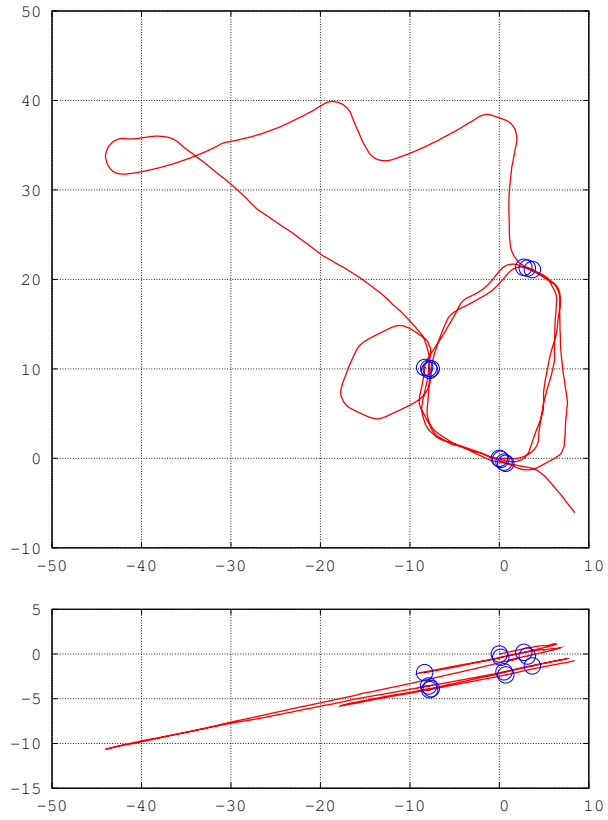

Figure 9: INRIM data I: CVL-estimated path, top and side view (coordinates in m). Circles indicate stop positions.

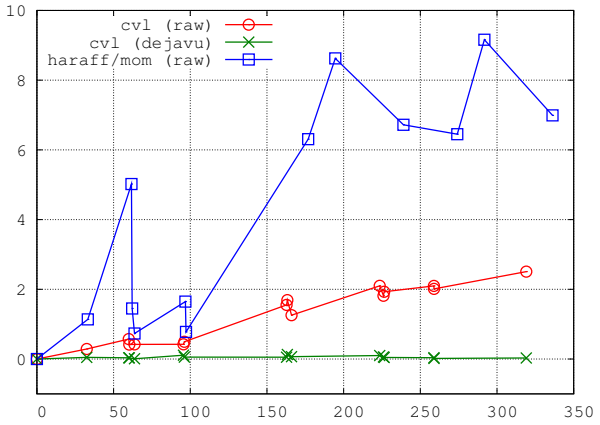

Figure 10: INRIM data I: return position error (m) vs. path length (m).

5 Results on real data

5.1 INRIM data I

A stereo sequence was acquired by our Pioneer 2AT rover (Fig. 1) on a grassy plain within the INRIM campus (Fig. 7 and Fig. 8). The stereo head consists in a pair of Basler A312f digital cameras, equipped with 6mm lenses. The cameras are mounted on an aluminum slab, as shown in Fig. 1. Each camera provides a stream of CCIR-size (720×576) 8-bit graylevel images on a IEEE1394 bus. The two cameras can be accurately synchronized by an external trigger signal, provided in our case by the controlling PC through the parallel port. The cameras are mounted at a nominal center-to-center distance (stereo baseline) of 236 mm, and the head was calibrated using Bouguet's method [22], with a 0.6×0.6 m checkerboard pattern. Note that the two cameras are mounted with parallel axes, but the head is slightly tilted (about 13°) towards the ground. In fact, a good estimate of robot motion requires both distant features, providing reliable hints about direction, as well as near ones, which yield more reliable hints about the distance traveled between keyframes. The arrangement used allows to increment the number of usable near features, while cutting off most of the sky area, which does not provide much useful information.

The rover was sent along a winding path of about 320m total length (Fig. 9), framing a total of 9200 stereo pairs. Since in this case we had no independent measure of the rover pose, the rover was driven to pass multiple times through some marked spots (cir-

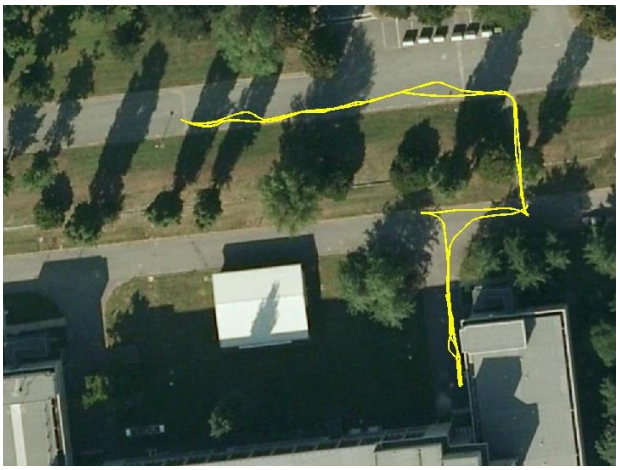

Figure 11: INRIM data II: top view of location.

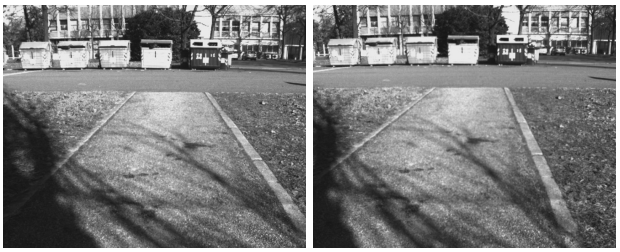

Figure 12: INRIM data II: sample stereo pair of the environment.

cles in Fig. 9), and the algorithm was forced to put a keyframe at that positions. This allowed to compute the accumulated position error at each return to the same spot. The latter is reported in Fig. 10 vs. the (estimated) length of path traversed before returning.

This plot compares the results obtained by our CVL features (with $r = 0.7$, $\sigma = 2$ pel and $\theta = 1.5$ pel) against the best of the other detectors/descriptors, i.e. Harris-Affine + Moments, with $r = 0.9$ and the same σ and θ. The reason for the different r is that CVL features yield slightly better results at $r = 0.7$, while Harris-Affine+Mom does not work at all with this value (too few good matches).

Note that the points in this graph have been joined by lines only for the sake of readability, nearby points on the plot are not necessarily near on the path, and the error between successive points is simply unknown..

Fig. 10 also reports the results of applying the mentioned déjà vu loop-closure algorithm (again with CVL features).

5.2 INRIM data II

This data set comprises 13347 stereo pairs, collected by our rover over a path of about 305 m, again within the INRIM area (Fig. 11). The main differences with

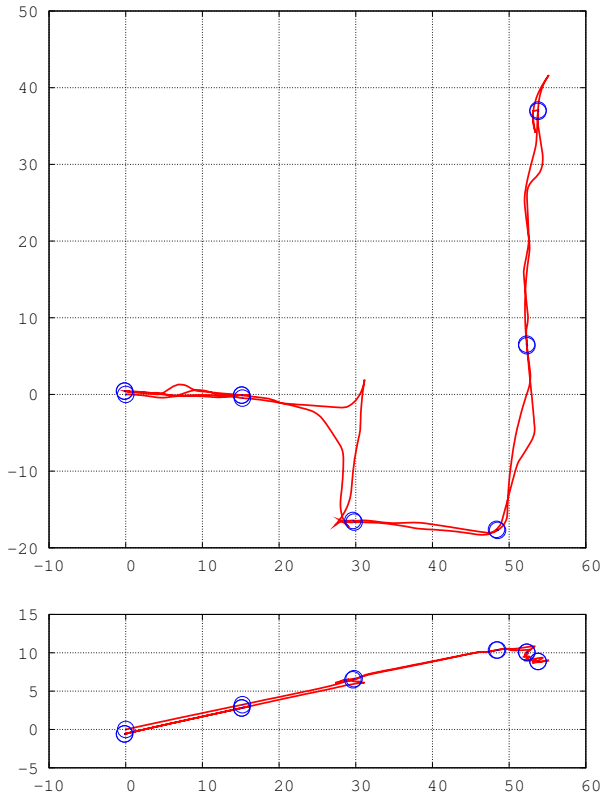

Figure 13: INRIM data II: CVL-estimated path, top and side view (coordinates in m). Circles indicate stop positions.

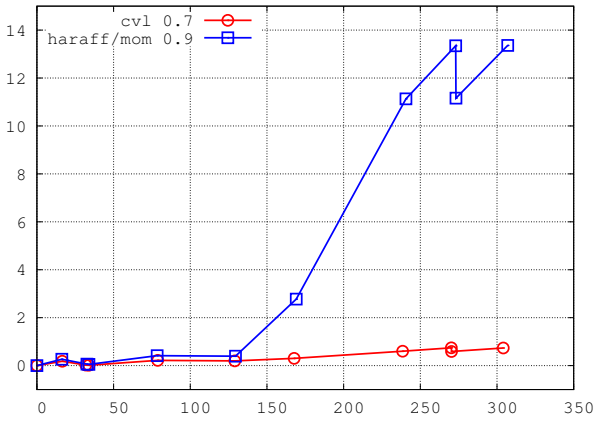

Figure 14: INRIM data II: return position error (m) vs. path length (m).

respect to the data presented above are the different season (longer shadows) and ground type (mainly paved road, see Fig. 12). An even more important feature, when evaluating the results, is that for most of its return travel the rover was driven to go approximately over the same path, running backwards. This means that the images acquired in the forward and

features	*step* 0..20	*total* 0..5000	*4-match* 0..500	*inliers* 0..250
INRIM path I				
cvl (0.7)				
cvl (0.9)				
haraff/mom				
INRIM path II				
cvl (0.7)				
haraff/mom				

Table 2: Average keyframe step, total features, 4-matched features and inliers for the real-world tests.

backward paths were much similar, so it may be expected that some systematic errors (due e.g. to calibration) could compensate. This is indeed confirmed by the results in Fig. 14, where the return position error with CVL features is about 1/3 of that on the data in Sec. 5.1 over a similar path length. However, the results with Harris-Affine + Moments are much worse, even worse than with the other data set.

Finally, Table 2 summarises the relevant statistics (step size and numbers of features) for both tests on INRIM data.

5.3 Oxford data

We have done some test on the "New College Dataset", kindly provided to the vision research community by the Oxford Mobile Robotics Group [23, 24]. The platform used to collect those data is a two-wheeled vehicle ("Lisa") built upon a Segway RMP200 base. A Point Grey Bumblebee camera, placed at about 1 m over ground and tilted about 13° towards ground, provides a 20 Hz stream of greylevel stereo pairs, with a resolution of 512×384 pixels. The platform is also equipped with other sensors, among which a GPS unit aimed at providing absolute positioning data at more or less regular intervals.

The data set considered here was recorded in November, 2008, in New College, Oxford (see Figs. 15 and 16) and comprises a total of 52478 stereo pairs, collected in about 47 minutes over a total path length of about 2844 m (from dead reckoning data).

Actually, this data set has some undesirable features in view of applying our algorithm, namely:

- Camera calibration data are incomplete. The data set does provide software and data for rectifying the images, and also some intrinsic and extrinsic camera parameters, but curiously enough the focal length is given in mm, instead of pixels - an information quite useless for processing digitized images. We converted this datum to pixels using the nominal sensor pitch from the manufacturer data sheet, yet we believe that the converted value is not accurate (we had to slightly adjust that value in order to get reconstructed 3D angles right).

- The Bumblebee camera has a rather limited base (0.12 m) and relatively low resolution (512×384), which do not contribute to accuracy in 3D estimation.

- The platform runs at a relatively high speed. In spite of the 20Hz sampling rate, this may cause a severe loss of correspondences, especially in sharp turns. In the latter situation, moreover, the slow shutter speed of the camera yields strongly blurred images, which sometimes cause a complete loss of image correspondences.

- There is no reliable ground truth position data. The dataset does provide GPS data, but we were not able to find in the accompanying papers any assessment of the accuracy of GPS data, which can sometimes be grossly wrong, as seen in the plot of Fig. 17. Moreover, there is no way to link the GPS reference to the robot's one.

We have nevertheless tried to run our algorithm on the Oxford data. The results are shown in Fig. 17,

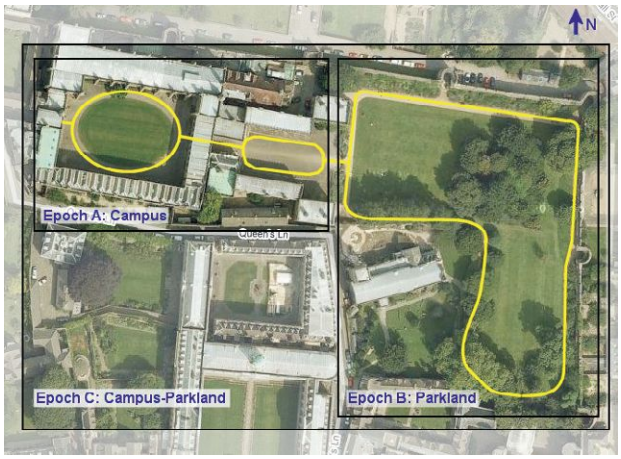

Figure 15: Oxford data: aerial view of location (from [23]).

Figure 16: Oxford data: sample stereo pair.

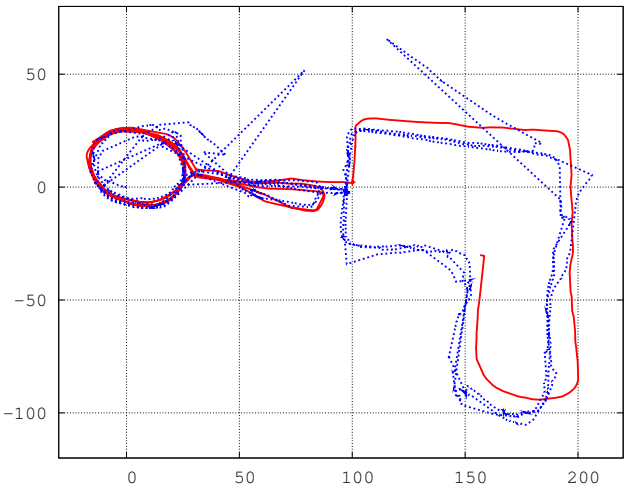

Figure 17: Oxford data: CVL-estimated path (solid red line) and GPS path (dotted blue line), top view. Coordinates in m.

where the visual odometry estimate, obtained using CVL features, has been superimposed to GPS data, after a rough manual alignment. Only CVL results are reported, because none of the other competitors were able to get outside Epoch A in Fig. 15; the best was again *haraff/mom*, reaching a path length of about 732 m, with an average keyframe step of 9, before algorithm failure due to lack of useful features.

Note that also the CVL visual odometry is interrupted, but after an estimated path length of about 1315m, and with an average keyframe step of 19. This interruption, due again to lack of good matches, happens in a sharp turn of the mobile platform. It should be noted that, in a real application, our visual odometry algorithm is not expected to be used alone, but supplemented with other modules able to catch and correct such situations using data from different sensors (at worst, wheel odometry).

From Fig. 18 the difference between the CVL estimate and GPS data for the first part (Epoch A in Fig. 15) appears, on the average, well under 5m, i.e. the order of magnitude of a commercial GPS device mounted on cars (the spikes are actually GPS errors, as can be seen from Fig. 17). The final rise of the curve may be partly imputed to the misalignment of visual and GPS reference frames.

6 Discussion and conclusion

This work certainly does not pretend to be exhaustive. The number of tested sequences is too small to allow a reliable statistical assessment of the behaviour of the different algorithms. Nevertheless, some conclusion can already be drawn from the results reported

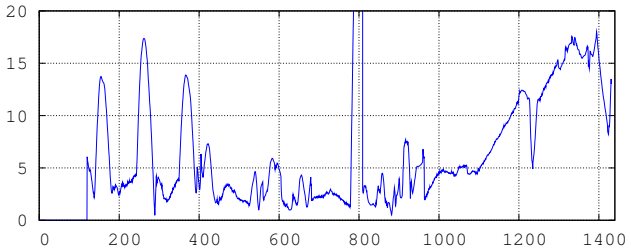

Figure 18: Oxford data: distance (m) between CVL-estimated position and GPS position, vs. time (s).

above, in particular from the statistics in Tables 1 and 2. In fact, while each error graph in Sec. 4 represents a global behaviour on a single sample of all possible trajectories in all possible environments, the statistics in the Tables are averages over many different keyframe pairs, with different imaged scenes.

A first conclusion is that, in an application like our visual odometry algorithm, a high degree of geometric invariance (rotation, affine etc.) of features is not necessarily a merit. Indeed, the point of view (PoV) changes very little among the four images of a keyframe pair, and scale invariance is enough to take into account the scale change of near features with the rover motion.

This is the reason why our CVL features (neither affine- nor even rotation-invariant) performs at a level of accuracy comparable to the best affine-invariant detectors, with the advantages of a much faster computation and greater reliability. The behaviour of CVL features is quite similar to that of SURF ones, but with a considerable speedup.

In fact, from Table 1 CVL and SURF appear to yield both a lower number of initial detected features (*total* column, around 1000 per image) and a quite higher number of useful ones (*inliers* column, around 200), i.e. of features matched over all the four images of a keyframe pair. A higher number of inliers contributes to accuracy and reliability of the estimate, while a lower number of detected features alleviates the computational load. Indeed, feature extraction and matching is the most resource-consuming step of the algorithm; extraction time grows linearly with the number of features, and matching time is quadratic.

All this is confirmed by the real-world tests. While the usual caveats still apply, in Fig. 10 and Fig. 14 *cvl* clearly outperforms *haraff/mom*. Table 2 indicates that the latter yields a lower number of usable features, and with a more frequent sampling (average step 6 against 14..16 for data set I, and 12 against 20 for data set II) - neither of these contribute to accuracy, and the lower step size greatly increases computations.

Affinely invariant features could be of some help

in the déjà vu loop-closure mechanism, where image matching has to be done from much more different points of view. However, our tests on other real image sequences, not reported here, indicate that the number of matches for quite different PoV's is usually not enough for a reliable pose estimation with the approach of Sec. 2. It could be enough, however, for place recognition and subsequent pose re-estimation with a different approach (e.g. map-matching).

Acknowledgment - This work has been partly supported by Regione Piemonte in the framework of the STEPS Project, led by Thales Alenia Space Italia.

References:

[1] C. F. Olson, L. Matthies, M. Schoppers, and M. W. Maimone, "Robust stereo ego-motion for long distance navigation." in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 2453–2458.

[2] S. Thrun, "Robotic mapping: A survey," Carnegie Mellon University, Tech. Rep. CMU-CS-02-111, 2002.

[3] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003.

[4] J. Neira, M. I. Ribeiro, and J. D. Tardos, "Mobile robot localization and map building using monocular vision," in *Proc. 5th Int. Symp. Intelligent Robotic Systems '97*, 1997, pp. 275–284.

[5] A. Cumani, S. Denasi, A. Guiducci, and G. Quaglia, "Integration of visual cues for mobile robot localization and map building," in *Measurement and Control in Robotics*, M. A. Armada, P. Gonzales de Santos, and S. Tachi, Eds. Instituto de Automatica Industrial, Madrid, 2003.

[6] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation"," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[7] H. Hirschmuller, "Real-time map building from a stereo camera under unconstrained 3d motion," De Montfort University, Leicester," Faculty Research Conference, 2003.

[8] A. Cumani, S. Denasi, A. Guiducci, and G. Quaglia, "Robot localisation and mapping with stereo vision," *WSEAS Trans. Circuits and Systems*, vol. 3, no. 10, pp. 2116–2121, 2004.

[9] A. Cumani and A. Guiducci, "Robot localisation error reduction by stereo vision," *WSEAS Trans. Circuits and Systems*, vol. 4, no. 10, pp. 1239–1245, 2005.

[10] ——, "Stereo-based visual odometry for robust rover navigation," *WSEAS Trans. Circuits and Systems*, vol. 5, no. 10, pp. 1556–1562, 2006.

[11] ——, "Visual odometry with SURFs and déjàvu correction," in *Proc. 8th Conf. Optical 3D Measurement Techniques*, 2007.

[12] ——, "Fast point features for accurate visual odometry," in *Proc. 12th WSEAS CSCC Multiconference*, 2008.

[13] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proceedings 9th European Conference on Computer Vision*, 2006, see also http://www.vision.ee.ethz.ch/~surf/.

[14] A. Cumani and A. Guiducci, "Robust and accurate visual odometry by stereo," in *Proceedings of the European Computing Conference*, N. Mastorakis, V. Mladenov, and V. T. Kontargyri, Eds. Springer, June 2009, vol. II, pp. 485–494.

[15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[16] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, 2008.

[17] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[18] ——, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[19] A. Cumani and A. Guiducci, "Fast stereo-based visual odometry for rover navigation," *WSEAS Trans. Circuits and Systems*, vol. 7, no. 7, pp. 648–657, 2008.

[20] Persistence of Vision Pty. Ltd. (`http://www.povray.org/`), 2004.

[21] http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html.

[22] J. Y. Bouguet, "Complete camera calibration toolbox for MATLAB," http://www.vision.caltech.edu/~bouguetj/calib_doc/index.html, Tech. Rep., 2004.

[23] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *International Journal for Robotics Research (IJRR)*, vol. 28, no. 5, pp. 595–599, May 2009.

[24] http://www.robots.ox.ac.uk/NewCollegeData/.