Visual Odometry for Robust Rover Navigation by Binocular Stereo

ALDO CUMANI and ANTONIO GUIDUCCI
Istituto Nazionale di Ricerca Metrologica
Str. delle Cacce, 91 - I-10135 Torino
ITALY

Abstract: This work deals with the problem of estimating the trajectory of an autonomous rover by passive stereo vision only (visual odometry). The proposed method relies on the tracking of pointwise image features and on the estimation of the robot motion by robust bundle adjustment of the stereo matched features, before and after the motion. Preliminary results from the application of the algorithm both to simulated data and to actual image sequences acquired by a mobile platform are presented and discussed.

Key-Words: Robot localisation, Egomotion, Stereo vision, Bundle adjustment, SLAM

1 Introduction

It is well known that estimating the actual trajectory of a mobile robotic platform by pure dead reckoning, i.e. by integration of wheel motions over time, is highly prone to error accumulation. Wheel odometry alone is not even sufficient in principle when navigating on a non-planar surface, and must be complemented by other inputs (e.g. inclinometers).

On the other hand, there are several applications where teleoperation of the robot is difficult, if at all possible. Such applications require truly autonomous long range navigation in a partially or totally unknown environment. This is the case for a planetary rover, which typically can communicate with the Earth station once per day, in order to transmit acquired data and to receive commands, e.g. to reach some far target on the planet surface. In such a scenario, it is clear that high navigation errors may mean loss of entire days, while keeping such errors to a minimum can boost the scientific throughput of the rover mission.

The use of vision techniques for improving the rover's trajectory estimate has been studied rather extensively. There have been many proposals for using ego-motion estimation from either monocular or stereo image sequences [1, 2, 3, 4], possibly integrated with dead reckoning as well as with other kinds of sensorial inputs. An affine field of research is that of *Simultaneous Localisation And Mapping* (SLAM) techniques [5, 6, 7, 8, 9]. Most of the published SLAM algorithms, however, are designed for indoor environments, and make use of environment representations (maps) which are essentially 2D. Their extension to navigation on a generic rough terrain is therefore not straightforward.

In this paper we present some results on a trajectory estimation method which relies on visual odometry from stereo. From pairs of images taken by a stereo head fixed on the rover, a dense depth map is estimated by standard correlation techniques. Point features are then extracted in the left image and matched against the right image, using the depth map as a disparity cue. Image features are then tracked along the sequence grabbed during the rover motion, and after a suitable number of frames the relative motion is estimated by bundle adjustment over the two (left and right) initial images and the two final ones. The latter step consists in estimating both the 3D positions of all observed features, and the relative rototranslation of the stereo head between the two rover positions, by minimizing a robust cost function of the reprojection error of feature points in all four images. The computed rototranslation constitutes the visual odometry estimate which is then summed over to determine the rover trajectory.

This approach has some resemblance both to a SLAM method previously proposed by the authors [10] (though in that case only the angular part of visual odometry was used), and to the one in [4, 11]. In both the cited references, however, visual odometry is obtained by first making two 3D reconstructions from stereo, one before and one after the motion, and then registering the two point clouds by minimizing a fitting criterion defined in 3D space. This has the distinct drawback of requiring an accurate modeling of the distribution of the result of stereo reconstruction, not to mention the problem of outliers (i.e. mismatched features). On the contrary, a bundle adjustment approach, like the one described here, takes advantage of a more sound definition of fitting error (i.e. on the image plane), and can easily accommodate the use of a robust cost function to deal with outliers.

It should be remarked that this kind of approach is

purely *passive*, relying only on available images acquired during rover motion, so it should not interfere with other visual tasks as e.g. obstacle avoidance.

In the following, the method is explained in some detail. Results from its application to image sequences, both simulated and actually acquired by a mobile robotic platform, are presented and discussed.

2 Visual odometry

As said above, the algorithm relies on estimating relative motion from tracked features in the images acquired while the robot is moving. In principle, such estimate is not computed for every image, but only at *key frames*, whose spacing is a compromise between larger intervals, which are desirable both for numerical accuracy and for economy of computations, and the need for a sufficiently large number of features, which naturally tend to be lost either because going out of view and because of tracking errors. The algorithm can be summarized in the following steps:

Depth map computation. A dense depth map is estimated at each key frame for easing stereo matching of features.

Feature extraction. Point features are extracted at each key frame and left-right matched.

Feature tracking. Detected features are tracked in the left image up to the next key frame, where stereo matching of tracked features is performed again.

Motion estimation. The relative rototranslation of the robot between consecutive key frames is estimated by bundle adjustment of the tracked and matched feature points.

The actual implementation of the algorithm must take into account some features that make it just a bit more complicated than the above sketch. In particular, tracking failure cannot be detected until it occurs, which forces to store the previous stereo pair, so as to be able to restart from there. Also the bundle adjustment step may fail, either for numerical errors or because of an excessive number of outliers (see Sec. 2.4), forcing to declare a key frame at the previous frame; this means that bundle adjustment must be performed at every frame, although this is not a big problem, as the bundle adjustment step is usually rather fast.

The following paragraphs describe the above steps in some detail.

2.1 Depth map

Image pairs from the stereo head are warped to remove both lens distortion and camera misalignment, so that the output of this process is a stereo pair with horizontal epipolar lines, as would be obtained from an ideal binocular head with parallel optic axes. The images are then passed through a Laplacian of Gaussian (LOG) filter. We use LOG filtered images instead of the original intensity images to reduce the effect of exposure differences.

A dense disparity map is then computed by correlation matching, using the well known minimum Sum-of-Absolute-Differences (SAD) method [12]. We have also tested more refined algorithms like the one in [13]; however, the depth map in our case is only used as a cue for predicting stereo disparity of point features, and for this purpose the simple and fast SAD method is adequate. For the same reasons, the algorithm may be run on subsampled images, which both reduces the computational load and helps the correlation matching algorithm by smoothing out high frequencies.

2.2 Features

The current implementation uses Shi-Tomasi features [14], i.e. small textured image patches, whose centers yield pointwise measurements, similarly to SIFT features used by other researchers [9]. A significant advantage of Shi-Tomasi features is that their definition implicitly provides an efficient frame-to-frame tracking algorithm, provided that the image-plane displacement be small, i.e. well within the size of the patch. Using the same algorithm for stereo matching, where the displacement (disparity) may be rather large, especially for near objects, is still feasible if some coarse initial estimate of disparity is available. In our implementation, such estimate is provided by the dense disparity map computed as said above.

Matched point pairs are then backprojected to a 3D point estimate, in the camera reference. This 3D estimate is used both as a starting point for the bundle adjustment step, and for guiding the tracking of features in subsequent frames.

2.3 Tracking

The features detected in the left image at a key frame are tracked along the sequence up to the next key frame. As said in Sec. 2.2, the frame-to-frame tracking algorithm for point features expects limited feature displacements between subsequent frames. This is seldom the case, especially when the robot is rotating, so for reliable tracking we need some prediction of frame-to-frame feature displacements. We currently use two methods to this end.

Wheel odometry. As long as the robot motion is sufficiently smooth (and planar), differential wheel odometry already yields an estimate of robot motion between consecutive images. Combining the latter with the estimated 3D feature positions from stereo

allows to predict feature point displacements. This is the normal mode of operation.

Optic flow. When wheel odometry fails, as indicated by loss of many (sometimes all) feature points, the predicted displacements are estimated by computing a dense optic flow from suitably subsampled versions of the two consecutive left images. This has the advantage of not depending upon 3D point estimates, but is computationally expensive and therefore avoided as long as possible.

2.4 Bundle adjustment

At every key frame, say $k+1$, except the first, we have a set of N_k features $\mathcal{F}_i$, left-right matched and tracked from the previous key frame k to the next one $k+1$ (for the sake of simplicity, and without loss of generality, from now on we shall assume $k=1$ and drop the k index). Let $\mathbf{X}_i = [x_i, y_i, z_i, t_i]^\top$ be the unknown 3D projective coordinates of feature $\mathcal{F}_i$ in the reference of the left camera at frame 1. Note that the scale ambiguity on $\mathbf{X}_i$ can be fixed by taking $z_i = 1$ since all the points must be visible in the first left image, hence they have strictly positive depth z. Let $\mathbf{u}_{iq} = [u_{iq}, v_{iq}]^\top$ be the 2D image coordinates of the feature point in image $q \in \{1L, 1R, 2L, 2R\}$, and $\mathbf{x}_{iq} = [su_{iq}, sv_{iq}, s]^\top$ the same in projective representation. Then

$$\begin{array}{rcl} \mathbf{x}_{i,1L} & = & \mathsf{P}_L \mathbf{X}_i \\ \mathbf{x}_{i,1R} & = & \mathsf{P}_R \mathsf{M}_S \mathbf{X}_i \\ \mathbf{x}_{i,2L} & = & \mathsf{P}_L \mathsf{M}_{12} \mathbf{X}_i \\ \mathbf{x}_{i,2R} & = & \mathsf{P}_R \mathsf{M}_S \mathsf{M}_{12} \mathbf{X}_i \end{array} \quad (1)$$

where P_L and P_R are the 3×4 intrinsic camera matrices, and $\mathsf{M}_S, \mathsf{M}_{12}$ are 4×4 Euclidean transformation matrices representing the stereo (left-to-right) mapping and the motion (frame 1 to frame 2) mapping, respectively, and are of the form

$$\mathsf{M} = \begin{bmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} e^{[\mathbf{r}]_\times} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (2)$$

where $\mathbf{r}$ is the vector representation of rotation R, i.e. $\mathbf{r}/\|\mathbf{r}\|$ = rotation axis, $\|\mathbf{r}\|$ = rotation angle, and $[\mathbf{r}]_\times$ is the antisymmetric matrix representation of vector cross product by $\mathbf{r}$.

Note that in the above both P_L, P_R and the stereo transform M_S are known from calibration data; in fact, due to image warping $\mathsf{R}_S \equiv \mathsf{I}_{3\times3}$ and $\mathbf{t}_S \equiv [b, 0, 0]^\top$ with b the stereo baseline length. We can also take $\mathsf{P}_{L,R} \equiv [\mathsf{I}_{3\times3}, \mathbf{0}_3]$ in Eq. (1) if the pixel-based feature coordinates are preliminarly transformed to *calibrated* coordinates using the actual intrinsic camera matrices. Now, for each pair of consecutive key frames we have measurements $\mathbf{u}^*_{iq}$ of the quantities $\mathbf{u}_{iq}$ corresponding to Eq. (1), so we can define an error measure

$$\begin{array}{rcl} J(\mathbf{p}) & = & \sum_i \sum_q f(\|\mathbf{e}_{iq}\|^2) \\ & = & \sum_i \sum_q f(\|\mathbf{u}_{iq} - \mathbf{u}^*_{iq}\|^2) \end{array} \quad (3)$$

parametrized as a function of the $6 + 3N$ unknowns

$$\mathbf{p} = [\mathbf{r}_{12}, \mathbf{t}_{12}, x_1, y_1, t_1, ...x_N, y_N, t_N]^\top \quad (4)$$

with $\mathbf{r}_{12}, \mathbf{t}_{12}$ the generators of M_{12}. The visual odometry estimate is then given by the $\mathbf{r}_{12}, \mathbf{t}_{12}$ entries of the $\mathbf{p}$ value that minimizes J.

In the above, choosing $f(\cdot) =$ identity corresponds to standard least squares, which would be optimal for independent Gaussian distributed image errors. For reasons of robustness against outliers (mismatched points), it is however preferable to use a robust cost function such as the Lorentzian cost:

$$f(e^2) = \log(1 + e^2/\sigma^2) \quad (5)$$

with σ chosen as a function of the expected image-plane error.

Since the first derivatives of J with respect to $\mathbf{p}$ can be computed analytically, the minimization of J can be achieved by a gradient-based method such as Levenberg-Marquardt. Moreover, for given i the corresponding terms in Eq. (3) only depend upon one of the triples (x_i, y_i, t_i), which allows to use a sparse algorithm like the one described in [15, A4.3].

For better accuracy, the residuals $\mathbf{e}_{iq}$ are compared against a threshold (proportional to the Lorentz σ), and those exceeding that threshold are marked as outliers. The bundle adjustment is then run again on the inliers only.

3 Results

The proposed method has been implemented in C and MATLAB, and tested on a number of simulated and actual image sequences.

3.1 Simulation

In order to validate the algorithm, it has been run on a sequence of synthetic images of a Mars-like environment as shown in Fig. 1. The images were synthesized using the free raytracer POV-Ray [16]. Lens distortion and acquisition noise were not included in the simulation, but camera calibration was performed, using Bouguet's method [17] on five synthesized images of a checkerboard.

The simulated robot was sent on a closed trajectory with some sharp turns, as shown in Fig. 2, at a speed of 16 mm/frame (similar to that of the actual robot),

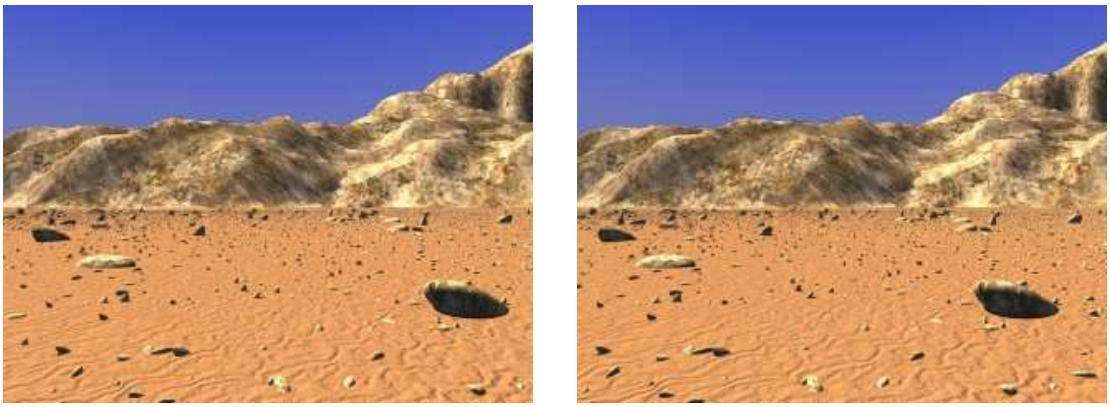

Figure 1: Simulated environment.

for a total of 1150 frames. Fig. 2 shows both the odometric trajectory (which in this case is exact!) and the estimates provided by the algorithm. The final pose was off by about 0.16° in angle and 0.014 m in position over a trajectory length of about 18.5 m (i.e. a position error less than 0.1% of travelled length).

This small but not negligible error is mostly due to the finite spatial sampling of the images, and to calibration errors (again due, in this case, to finite spatial resolution).

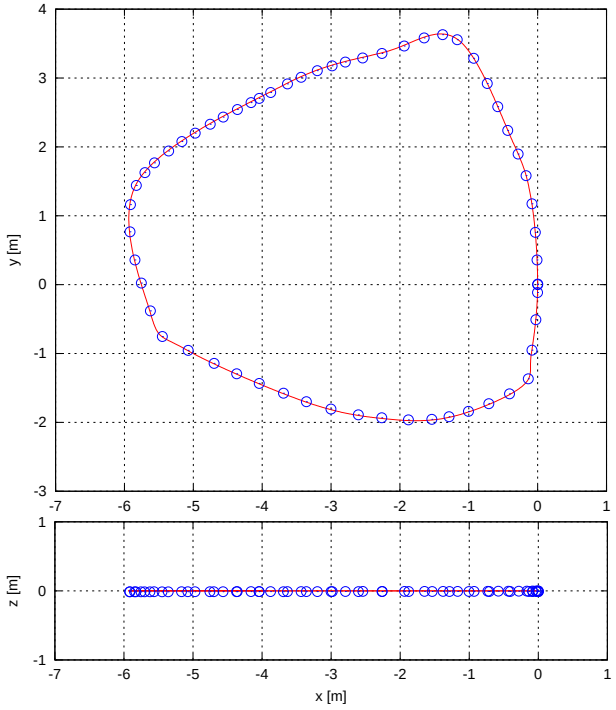

Figure 2: Trajectory for simulated data, top and side view. Red line: odometry, blue dots: estimates at keyframes. The initial position is at (0,0).

3.2 Actual rover tests

This section presents some results obtained by processing sequences of images acquired with our ActivMedia Pioneer 3-DX robot, equipped with a Videre Design STH-MDCS stereo head (Fig. 3). The latter is a low-cost commercial product nominally capable of yielding pairs of 1280×960 colour images at 7.5 fps, or lower resolution images (640×480 and 320×240) at up to 30 fps. A serious limitation of this device is its small stereo baseline (88 mm, non-adjustable). The head mounted a pair of consumer-grade 5mm lenses with a field of view of about 63°, and was calibrated before and after each run.

Figure 3: The mobile robot with stereo head.

In our experiments, the robot wandered on an asphalted surface as shown in Fig. 4. Since we currently have no means to assess the robot position with a reasonable accuracy, the initial position was marked, and the rover was then manually driven to run along an approximately closed path, so as to be able to check

Figure 4: Real environment. The two bricks mark the starting position.

the estimate at least on every return of the rover to its starting position. In the experiment presented here, the robot described a sort of double figure of eight, with four returns to the starting point, as shown in Fig. 5. The processed sequence counted a total of 6258 stereo frames acquired at a mean spacing of about 11 mm/frame. As can be seen from the images, the surface on which the robot travelled was quite rough, causing a rather irregular motion. It is worth noting that the odometric estimate of the trajectory, as shown in Fig. 5 is not only all but closed, but the final heading is off by more than 100°.

In order to have a rough estimate of the algorithm accuracy, the four frames of the sequence nearest to the four returns of the rover to the starting point were selected, and the robot pose relative to the initial one was estimated independently. Table I compares the latter to the actual algorithm output at the same frames. Note that the column labeled 'length' reports the path length estimated by the algorithm. The position error is always well below 1% of the path length, with an attitude error less than 6° over almost 70m of travel.

4 Conclusion

This work has presented a method for estimating the egomotion of a rover vehicle purely from passive stereo, without any other sensor input. In spite of the use of low-cost vision equipment, the preliminary results from field tests compare favorably to those repoted e.g. in [11].

References:

[1] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains.," *Int. Journ. Robotic Research*, vol. 21, no. 10-11, pp. 917–942, 2002.

[2] Z. Zhang and O. D. Faugeras, "Estimation of displacements from two 3-d frames obtained from stereo.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 12, pp. 1141–1156, 1992.

[3] A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision.," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 3519–3524, 2000.

[4] C. F. Olson, L. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion.," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, 2003.

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

[6] M. C. Martin and H. P. Moravec, "Robot evidence grids," Tech. Rep. CMU-RI-TR-96-06, Carnegie Mellon University, 1996.

[7] P. Jensfelt, H. Christensen, and G. Zunino, "Integrated systems for mapping and localization," in *ICRA-02 SLAM Workshop* (J. Leonard and H. Durrant-Whyte, eds.), IEEE, May 2002.

[8] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Lausanne, Switzerland), pp. 226–231, October 2002.

[9] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proceedings of the IEEE*

frame		**c** [m]			$\|\Delta\mathbf{c}\|$ [m]	**r** [rad]			$\|\Delta\mathbf{r}\|$ [deg]	length [m]
1685	R	0.002	-0.003	0.000		0.0020	0.0137	0.0070		
	A	0.005	0.024	0.049	0.06	0.0011	0.0165	0.0078	0.2	18.6
2958	R	0.017	-0.019	0.003		-0.0015	0.0252	0.0037		
	A	0.078	-0.024	0.149	0.16	-0.0053	0.0735	-0.0014	2.8	32.6
4458	R	-0.029	0.007	0.005		0.0033	0.0243	0.0102		
	A	-0.054	-0.162	0.289	0.33	0.0115	0.1200	0.0121	5.5	48.9
6257	R	0.008	-0.058	0.001		0.0002	-0.0263	-0.0026		
	A	-0.012	-0.256	0.305	0.36	0.0166	0.0739	-0.0106	5.8	68.8

Table I: Comparison of algorithm estimates ('A' rows) with reference values ('R' rows) on return to the starting position. **c** is the rover position, and **r** its attitude, both relative to the rover's starting pose.

International Conference on Robotics and Automation, pp. 2051–2058, May 2001.

[10] 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.

[11] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the Mars exploration rovers," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, vol. 1, pp. 903–910, 2005.

[12] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[13] O. Faugeras, T. Vieville, E. Theron, J. Vuillemin, B. Hotz, Z. Zhang, L. Moll, P. Bertin, H. Mathieu, P. Fua, G. Berry, and C. Proy, "Real-time correlation-based stereo: algorithm, implementations and applications," Tech. Rep. RR-2013, INRIA, 1993.

[14] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2000.

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

[17] J. Y. Bouguet, "Complete camera calibration toolbox for MATLAB," tech. rep., `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`, 2004.

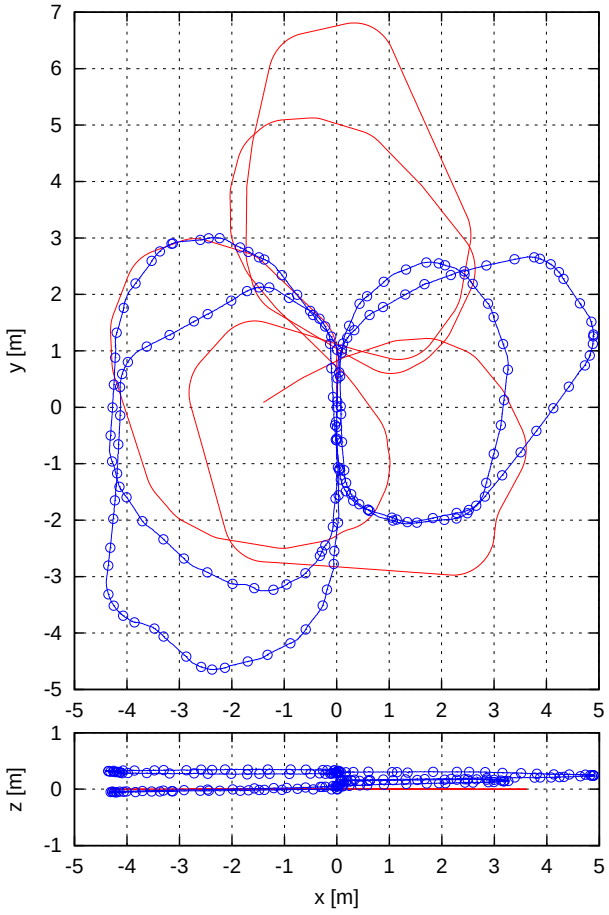

Figure 5: Trajectory for actual data, top and side view. Red line: odometry, blue line and dots: estimates. The starting point is at (0,0).