Edge Landmarks in Monocular SLAM

Ethan Eade and Tom Drummond
Cambridge University

{ee231, twd20}@cam.ac.uk

Abstract

While many visual simultaneous localisation and mapping (SLAM) systems use point features as landmarks, few take advantage of the edge information in images. Those SLAM systems that do observe edge features do not consider edges with all degrees of freedom. Edges are difficult to use in vision SLAM because of selection, observation, initialisation and data association challenges. However, a map that includes edge features contains higher-order geometric information useful both during and after SLAM. We define a well-localised edge landmark and present an efficient algorithm for selecting such landmarks. Further, we describe how to initialise new landmarks, observe mapped landmarks in subsequent images, and deal with the data association challenges of edges. Initial operation of these methods in a particle-filter based SLAM system is presented.

1 Introduction

Much work in visual SLAM systems focuses on mapping point-based landmarks. Point landmarks have desirable properties in the context of visual SLAM: Point feature selection and description is well studied, the resulting feature descriptors are well-localisable in images, and they are highly distinctive, easing the task of data association. However, many environments have abundant edges and edge-like features. By tracking edges in the world, a SLAM system can build richer maps containing higher-level geometric information, and need not rely on an abundance of good point features. In contrast to point features, edges are well-localisable in only one image dimension, and often have non-local extent in the other image dimension. Though highly invariant to lighting, edges are also difficult to distinguish from each other locally. Such characteristics make the incorporation of edge landmarks into a visual SLAM system challenging.

This paper focuses on SLAM with edges. We encourage the reader to examine [2, 13, 9, 7] for detailed discussions of the general operation of visual SLAM systems. Our work is implemented within the system described in [4].

Edges have been recognised as critical features in image processing since the beginning of computer vision. While edge detection methods abound, Canny's algorithm[1] for choosing edgels in an image has emerged as the standard technique, and consistently ranks well in comparisons[6, 12]. We use it as a starting point for our edge feature selection algorithm.

The invariance of edges to lighting, orientation and scale makes them good candidates for tracking. Model-based trackers such as [3] and [11] use edge models to permit highly

BMVC 2006 doi:10.5244/C.20.2

efficient tracking of moving objects. Model-based tracking with edges, where structure is known but camera or object position is unknown, can be considered a subset of our SLAM problem.

The structure-from-motion algorithm described in [14] operates solely on edges detected in a video sequence. A global cost function is optimised to yield camera trajectory and line parameters. Results of this work are also shown in [12]. There are two important differences between this and the problem of SLAM with edges: SLAM must maintain estimates of camera motion and world structure online, not in a global optimisation, and a SLAM system must maintain uncertainty information regarding its estimates of motion and structure.

The vision SLAM system of [5] is designed to support heterogeneous landmark types in a common framework. While lines are used as features, they are assumed to be confined to planes of known orientation. Lines are also tracked by the SLAM system of [8], but only vertical lines are considered, as the camera is known to move in a plane. Our work is concerned with estimating edges of arbitrary location and orientation. To our knowledge, no such visual SLAM system exists.

In this paper, we show how to efficiently select, observe, and estimate local edge features in a real-time monocular SLAM system. In Sec. 2, we describe the basic operation of our SLAM implementation. In Sec. 3 we define the edge features estimated as landmarks, and describe their representation in the world and the image. In Sec. 4 we present a simple, effective, and efficient algorithm for selecting new edge features. Sec. 5 discusses the problem of partial initialisation and describes our approach. Sec. 6 addresses the problem of data association with edges and explains our straightforward scheme for robust association. In Sec. 7, we present qualitative performance results of the system, draw conclusions and discuss future work.

2 SLAM Model

Here we give an overview of the SLAM system to which we add edge landmarks. For a detailed description and evaluation, see [4].

The system is based on a FastSLAM-type Rao-Blackwellised particle filter [10], which exploits probabilistic independence properties of the SLAM problem: Given a set of exactly determined camera poses, landmark estimates are probabilistically independent of each other. Thus, the uncertainty of the system can be maintained as a set of hypotheses, each one containing a camera trajectory and a set of (independent) estimates of landmarks given the trajectory. Landmark estimates are represented analytically as gaussian distributions within each hypothesis, while the current camera pose uncertainty is spread over the set of hypotheses. Each hypothesis is a particle in the filter, representing a full structure estimate and associated camera trajectory.

The only sensor in the system is a single calibrated camera, delivering 30 frames per second. Processing occurs in stages: first, the current camera pose and uncertainty for each hypothesis is estimated from a dynamic model. Then observations are taken from the latest video image, and the observations are used to optimise the gaussian pose estimates. Next, samples are drawn from the distribution of poses, based on the likelihood of observations, and lastly, the same set of observations is used to update landmark estimates within each sampled particle.

Prediction: When a new frame is retrieved from the camera, the current distribution of particles is determined according to a constant-velocity dynamic model. Before the prediction step, each particle represents an exact pose with an associated structure estimate, which is a gaussian over landmarks with block-diagonal covariance. After the prediction, the particle set represents a gaussian mixture model over poses and maps: Each particle's pose is moved according to its velocity and the elapsed time, and the result is taken as the mean of a gaussian with covariance given by the process noise, **Q**. The landmark estimates in each particle remain unchanged.

Observation: A subset of landmarks to observe in the current frame is chosen based on expected visibility. For each landmark to be observed, the expected image location and appearance is calculated from the filter's estimates of landmark and camera states. A search in the image determines the content of the observation. In the simple case of point features, the search yields a location in the image (with associated measurement noise) where the landmark's descriptor is localised. A top-down active-search observation framework limits the search region based on the uncertainty in the current estimate of the landmark. The observation stage yields a list of landmark identifiers and associated observations.

Particle Update and Resampling: At the point that observations are made from the latest image, the particle distribution actually represents a gaussian mixture over poses and landmarks. The observations are used within each particle (or component of the mixture) to update the pose estimate, using a standard EKF update. Additionally, a weight is assigned to each component according to the likelihood of the set of observations under the *unoptimised* components. Then particles are drawn from the posterior mixture according to the component weights, with poses drawn from the component gaussians.

Landmark Update: Finally, the same set of observations is used to update landmark estimates within each particle. After resampling, there is no uncertainty associated with each particle's pose hypothesis, so the conditional independence of landmark estimates within each particle still holds. In a given particle, the gaussian estimate of each observed landmark is updated according to a standard extended Kalman filter. Once the landmark estimates have been updated, the system is ready to process the next frame.

Point Landmarks: Point landmarks are considered three-dimensional points in the world with a locally planar structure, represented by an image patch. In the filter, estimates of landmarks are stored as three-dimensional gaussians. To localise a point landmark in an image, its three-sigma uncertainty ellipse, including uncertainty due to camera pose, is projected into a two-dimensional ellipse in the image. The descriptor patch is warped according to the expected camera pose, and the patch is localised inside the ellipse using normalised cross correlation.

New point landmarks are chosen using a feature selection algorithm, such as [11]. Initially, the system has no information about the landmarks depth in the world, necessitating a partial initialisation scheme, of which several exist. Davison's system uses a separate particle filter to estimate the depth of each new landmark[2]. Our system maintains an estimate of the landmark's inverse depth in the initial frame[4]. When the estimate is

well approximated by a gaussian, it is converted to the world frame and considered fully initialised.

3 Edgelet Landmarks

Point landmarks fit well into a SLAM system because they have a well-defined representation, both in image space and world space. In the image, a point landmark is represented as a locally planar patch with a distinct, but view-dependent, appearance. In the world, it is estimated as a three-dimensional point with gaussian uncertainty. In order to use edge features, we must also define their image and world representations.

Definition: We define our edge features, which we call *edgelets*, with an analogous property in mind. An edgelet is a local portion of an edge, with an edge being a strong, one-dimensional intensity change. Thus, given an edge, which may have significant extent in a given image, we can take any small segment on the edge as an edgelet observation. Furthermore, the edge need only be locally straight: a slow curve has many locally linear pieces, all of which can be considered edgelets. Tracking only local edge segments avoids several problems of trying to estimate full edges in the world. Full edges, because they are not local quantities in an image, may be partially occluded, or broken into pieces in the image. They might never be wholly visible, so determining their full extent and actual endpoints may be impossible. The locality of edgelets means that assumptions made about an edgelet as a unit (for instance, that it is straight) is much more likely to be satisfied than the same assumption made about a long edge.

Representation: As a world representation of an edgelet, we use a three-dimensional point $\mathbf{x}$ corresponding with the center of the edgelet, and a three-dimensional unit vector $\hat{\mathbf{d}}$ describing the direction of the edgelet in the world. Note that this representation is not minimal: $\hat{\mathbf{d}}$ has only two degrees of freedom. However, we find the cartesian representation more convenient in calculations. The uncertainty in these six parameters is represented as a gaussian with covariance $\mathbf{P}$. Given a camera pose $C = (\mathbf{R}, \mathbf{T}) \in SE(3)$, the observation function $\mathbf{h}_1$ sending $\mathbf{x}$ to a point in the image plane (the plane $z = 1$ in the camera frame) is identical to the observation function for points in general:

$$\mathbf{h}_1(\mathbf{x}) = project(\mathbf{R}\mathbf{x} + \mathbf{T}) \tag{1}$$

$$project\left(\begin{pmatrix} x & y & z \end{pmatrix}^T\right) = \begin{pmatrix} x/z & y/z \end{pmatrix}^T \tag{2}$$

For the direction, $\hat{\mathbf{d}}$, we have a unit vector in the image plane:

$$\mathbf{h}_2(\hat{\mathbf{d}}) = \begin{pmatrix} \mathbf{X}_3\mathbf{D}_1 - \mathbf{X}_1\mathbf{D}_3 \\ \mathbf{X}_3\mathbf{D}_2 - \mathbf{X}_2\mathbf{D}_3 \end{pmatrix} \Big/ \left| \begin{pmatrix} \mathbf{X}_3\mathbf{D}_1 - \mathbf{X}_1\mathbf{D}_3 \\ \mathbf{X}_3\mathbf{D}_2 - \mathbf{X}_2\mathbf{D}_3 \end{pmatrix} \right| \tag{3}$$

$$\mathbf{X} = \mathbf{R}\mathbf{x} + \mathbf{T} \tag{4}$$

$$\mathbf{D} = \mathbf{R}\hat{\mathbf{d}} \tag{5}$$

Because an edgelet is a local portion of a potentially longer edge, observations can not decrease the uncertainty of $\mathbf{x}$ along the direction $\hat{\mathbf{d}}$, because of the aperture problem. However, the location of the edgelet along the edge to pixel accuracy is determined during initialisation (Sec. 5), which is sufficiently accurate to allow subsequent observation.

Observation: Given a gaussian estimate of an edgelet $(\mathbf{x}, \hat{\mathbf{d}})$, we observe the landmark by predicting its location in an image, searching for the edgelet, and then feeding the observation to the filter. The edgelet's parameters project into the image plane according to 1 and 3, and its covariance projects through a linearisation of the observation functions $\mathbf{h}_1$ and $\mathbf{h}_2$. Then the image plane quantities project into the image (pixel space) according to the calibrated camera model. The result is a prediction of the edgelet in the image: an image location $\mathbf{x}_p$ and an image direction $\mathbf{d}_p$ with associated covariances.

The prediction implies that we expect to find a short edge segment centered at $\mathbf{x}_p$ with normal $\mathbf{n}_p$ perpendicular to $\mathbf{d}_p$. To locate the edgelet in the image, we consider the image region given by a three-standard-deviation variation of $\mathbf{x}_p$ in the direction of $\mathbf{n}_p$, with a predetermined local width (e.g. 15 pixels). We consider the set of edgels in this region with gradient direction similar to $\mathbf{n}_p$, and take straight segments within this local set of edgels as possible observations of the edgelet.

We use an approximation of the Hough transform to find such segments: First, we bin the edgels according to the gradient angle θ. Sliding a window over the bins, we consider all peaks in total edgel count. For each peak group of edgels, we compute a histogram of edgel location component in the direction of the bins' average gradient angle. The two histograms correspond to the two dimensions of the angle-radius representation of the Hough line tranform. Thresholding the peaks in the radius histograms yields sets of edgels that form straight segments in the region.

For each resulting edgel set, we first unproject edgel locations into the image plane, and compute the offset in the direction of $\mathbf{n}_p$ of each edgel from the line given by $\mathbf{x}_p$ and $\mathbf{n}_p$. We fit a least-squares slope-intercept line to the resulting values. The fitted two-parameter lines, all relative to $\mathbf{x}_p$ and $\mathbf{n}_p$, are the two-dimensional observation hypotheses used to update the pose and landmark estimates of the filter. The observation noise is calculated by mapping the edgel location uncertainty through the line fit algorithm to yield uncertainty in the slope-intercept fit. In the simplest case, the observation hypothesis closest to the predicted edgelet location can be taken as the single hypothesis. However, when there is clutter around the edgelet in the image, this can lead to incorrect data association. Instead, we use all of the hypotheses, deciding maximum likelihood data association as described in Sec. 6.

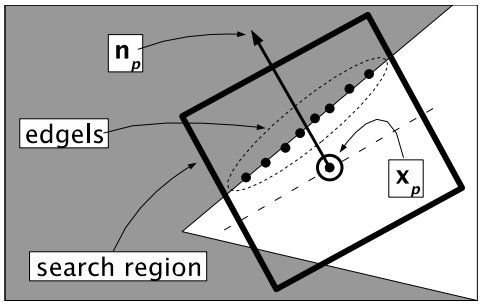


Figure 1: Edgelet observation

4 Finding new Edgelets

Our map of the environment is initially empty, except for fiducial landmarks used to bootstrap the system. We must acquire new landmarks to populate the map as we go. Point features are acquired using feature selection algorithms, and edgelets must be chosen with an analogous method. The edgelet selection algorithm should select edgelets that can be easily localised in subsequent frames, and it must be efficient, so as not to be a burden on the real-time operation of the system. We describe a simple, effective, and efficient method for choosing edgelets to track. Our method yields the locations of short, straight edge segments that are well-separated from nearby edges of similar direction.

Given an intensity image, we first identify all the edgels in the image with a minimum gradient magnitude that is maximal along the direction of the gradient. The output of the first steps of the Canny algorithm[1] is sufficient for this stage. The edgels are considered in subsets determined by placing a grid of a fixed size over the image. We use a grid with boxes of size 16×16 pixels. All subsequent processing happens within each grid subset.

For a subset of edgels $\{\mathbf{e}_i\}$, we compute the average second moment, $\mathbf{M}$ of the gradients $\mathbf{g}_i$ at the edgels:

$$\mathbf{M} \quad = \quad \left\langle \mathbf{g}_i \mathbf{g}_i^T \right\rangle \tag{6}$$

The eigenvectors of $\mathbf{M}$ describe the dominant directions of intensity change in the image patch. For a patch containing a single edgelet, the eigenvector corresponding to the larger eigenvalue should be normal to the edgelet. Let this dominant eigenvector be $\hat{\mathbf{n}}$. For each edgel, the angle θ between $\hat{\mathbf{n}}$ and the edgel's gradient satisfies

$$\cos\theta \quad = \quad \mathbf{g}_i^T \hat{\mathbf{n}} / |\mathbf{g}_i| \tag{7}$$

To select those edgels with gradients in agreement with $\hat{\mathbf{n}}$, we choose a minimal $\cos\theta$ and threshold according to

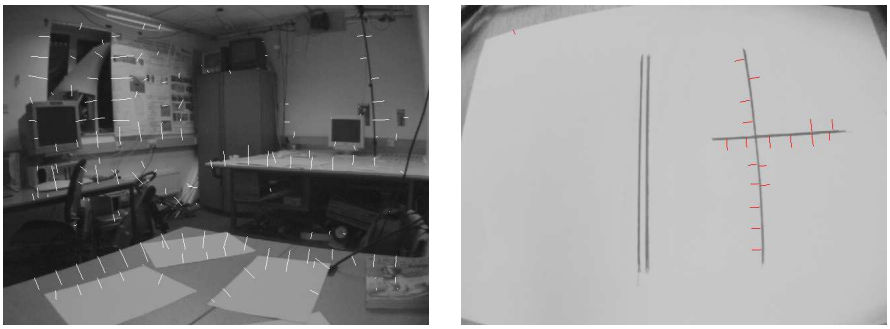
$$\left(\mathbf{g}_i^T \mathbf{g}_i\right) \cos^2\theta \quad > \quad \left(\mathbf{g}_i^T \mathbf{n}\right)^2 \tag{8}$$

For all edgel locations $\mathbf{e}_i$ with gradient in agreement with $\hat{\mathbf{n}}$, we consider the distribution of location in the direction of $\hat{\mathbf{n}}$, given by

$$b_i = \mathbf{e}_i^T \hat{\mathbf{n}} \tag{9}$$

The mean and variance of $\{b_i\}$ describe the location and agreement of edgels along the dominant direction. For a grid element with one clear, single straight edge, the variance will be on the order of a single pixel. We threshold on this variance to identify grid elements containing edgelets. Note that edgels with gradient directions not similar to the dominant gradient direction do not affect the edge-normal variance, as they are culled from the calculations early. Thus, a grid patch can contain two orthogonal segments and the stronger one will be chosen as an edgelet. Each grid element contributes either one or zero edgelets, with associated location, direction, and strength.

On a Pentium IV 2.8 GHz workstation, using grid elements of size 16×16, the entire algorithm, including non-max-suppressed edgel detection, processes a typical 320×240 grayscale image in 2-3 ms, yielding up to 300 edgelets. We choose new edgelets by taking edgelets sufficiently distant in the image from all recently observed landmarks. We further



(a) (b)

Figure 2: (a) Detected edgelets. The white line segments are normal to each detected edgelet in the direction dark-to-light, and their length is proportional to the strength of the edgelet. (b) The double lines on the left are rejected by the edgelet selector because they are too close, and might be confused in a search.

guide edgelet selection by choosing edgelets with direction more orthogonal to the image motion at the center of the edgelet given by the current average pose velocity. When the image motion is orthogonal to the edge direction, the landmark's depth can be recovered more rapidly.

5 Initialising Edgelets

A new edgelet cannot be added to the map as a fully-initialised landmark described in Sec. 3 until enough is known about its location and direction to make its estimate gaussian. While its location and direction in the image plane is well-determined from one observation, the components along the viewing ray are unknown. Thus, the landmark must remain partially initialised until all of its dimensions are well-represented by a gaussian. We represent a partially initialised edgelet in its initial observation frame with inverse depth. That is, instead of world coordinates $\mathbf{x} = (x, y, z)$, the edgelet position is given as (u, v, q), where (u, v) is the position of the edgelet in the camera plane and $q = z^{-1}$, all with respect to the first camera pose from which the landmark was observed. As shown in [4], the observation function $\mathbf{h}_1$ of a point expressed in inverse depth is nearly linear in the coordinates (u, v, q). Thus linear techniques such as the Kalman filter can be used to estimate $\mathbf{x}_q = (u, v, q)$. Just as $\hat{\mathbf{d}}$ is the unit differential of $\mathbf{x}$ along the edge, we estimate the unit differential $\hat{\mathbf{d}}_q$ of $\mathbf{x}_q$ along the edge for partially initialised edgelets.

As the landmark is repeatedly observed in subsequent images, the estimates of $\mathbf{x}_q$ and $\hat{\mathbf{d}}_q$ are updated using the Kalman filter framework. When the estimate is nearly gaussian in world coordinates, a change-of-variables is performed using the unscented transform, and the landmark's mean and covariance is thereafter expressed in world coordinates $\mathbf{x}$, $\hat{\mathbf{d}}$, and $\mathbf{P}$. Typically, new edgelets are fully initialised in fewer than ten frames, given non-degenerate camera motion. Note that each particle maintains a separate estimate of each partially initialised landmark (and each fully initialised landmark), so the change-of-variables occurs independently, sometimes at different times and with different results for different particles.

6 Robust Data Association

Edges are characterised only by the direction of their intensity change from low to high. If an edgelet's prediction uncertainty is large, there may be several possible edges in the image search region. We wish to choose the set of observation associations that has the maximum likelihood given our given our current estimates of poses and landmarks.

However, for m observations, there are 2^m subsets to consider. This number grows when multiple hypotheses exist for some observations. We use RANSAC to sample from the subsets. Taking observations three at a time, we consider the posterior estimate of a single particle's pose given those observations. The combined likelihood of the whole set of observations under that particle is computed. For each observation, we consider the most likely hypothesis, if there is more than one. If that likelihood is below a common threshold, the observation is considered an outlier, and the threshold likelihood is used in place of the observation's likelihood.

We repeat this process with random subsets of three observations. After a fixed number of tries, we take the maximum-likelihood set of inliers as our observations, and use that set to update all pose and landmark estimates. The number of random subsets tried is limited by computation time; in our system, we find that 30 tries gives good results. Even when only one observation hypothesis is found in the image for each observation, there are in fact two possibilities for each observation: inlier or outlier. This data association framework greatly improves the reliability of our system when viewing cluttered scenes or when partial initialisation gives spurious estimates.

7 Results and Conclusions

Our implementation of SLAM with edgelets runs at frame rate on a Pentium IV 2.8 GHz workstation, while measuring more than 20 landmarks each frame, and considering 30 data association RANSAC hypotheses. Figure 2 shows the output of the edgelet detection algorithm in an indoor scene.

The average search time for locating mapped edgelets in video frames in 0.25 ms per edgelet. This cost of finding the edge is dominated by the much greater cost of incorporating the landmark into the filter estimates, which currently requires 0.9 ms per landmark. The tracking is noticeably more reliable when using the RANSAC-based maximum-likelihood data association described in Sec. 6.

The initial performance results of our system using edgelets show that edges can be successfully tracked and mapped in an active-search monocular SLAM setting. Furthermore, the use of only local portions of possibly extended edges yields a framework flexible enough to map curved intensity changes. The main failure mode of the system is partial initialisation; eliminating the occasional spuriously initialised landmark and associated bad initial parameter estimates will significantly improve the system's robustness.

References

[1] J. F. Canny. A computational approach to edge detection. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 8, pages 679–698, 1986.

[2] Andrew Davison. Real time simultaneous localisation and mapping with a single camera. In *ICCV*, Nice, France, July 2003.

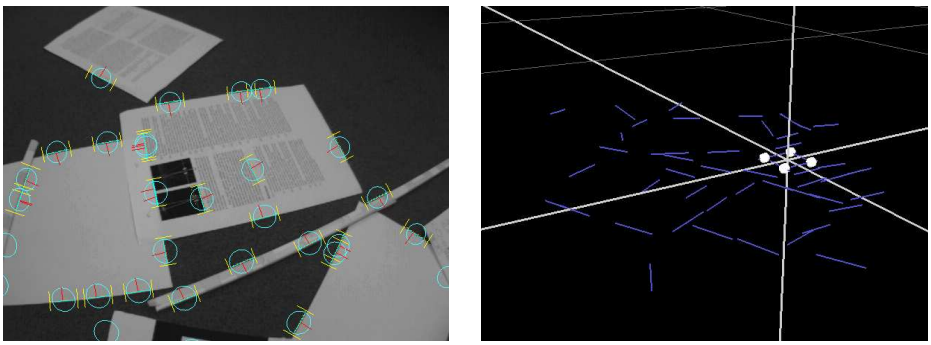


Figure 3: Mapping a planar scene with 51 edgelets: The mean displacement of the edgelet centers from the ground plane is $8.58 \cdot 10^{-5}$m. The standard deviation is 2.5mm. The standard deviation of edge angles out of the plane is 0.0331rad, or 1.9 degrees.

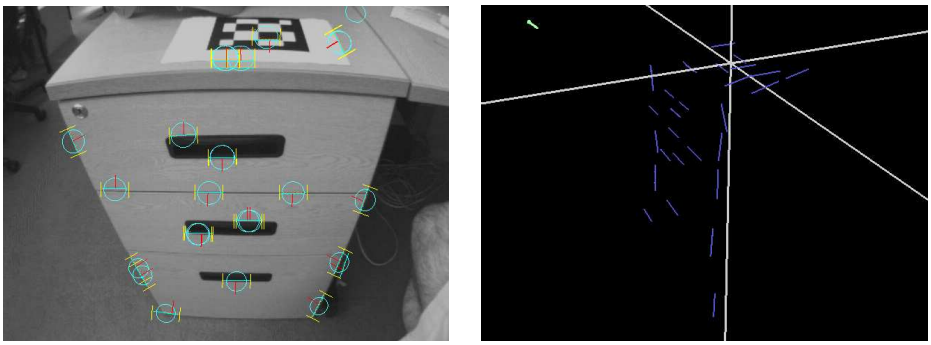


Figure 4: Edgelets of varying orientations: The system correctly captures the structure of the cabinet top and face.

[3] Tom Drummond and Roberto Cipolla. Application of lie algebras to visual servoing. *Int. J. Comput. Vision*, 37(1):21–41, 2000.

[4] Ethan Eade and Tom Drummond. Scalable monocular slam. To appear in CVPR'06, June 2006.

[5] John Folkesson, Patric Jensfelt, and Henrik Christensen. Vision slam in the measurement subspace. In *ICRA-05*, Barcelona, Spain, April 2005. IEEE.

[6] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. Comparison of edge detectors: A methodology and initial study. In *CVPR'96*, pages 143–148, 1996.

[7] Hailin Jin, Paolo Favaro, and Stefano Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6):377–394, Oct 2003.

[8] N Kwok and G Dissanayake. Bearing-only slam in indoor environments using a modified particle filter. In *ACRA'03*, 2003.

[9] Thomas Lemaire, Simon Lacroix, and Joan Sola. A practical 3d bearing-only slam algorithm. In *IROS 2005*, August 2005.

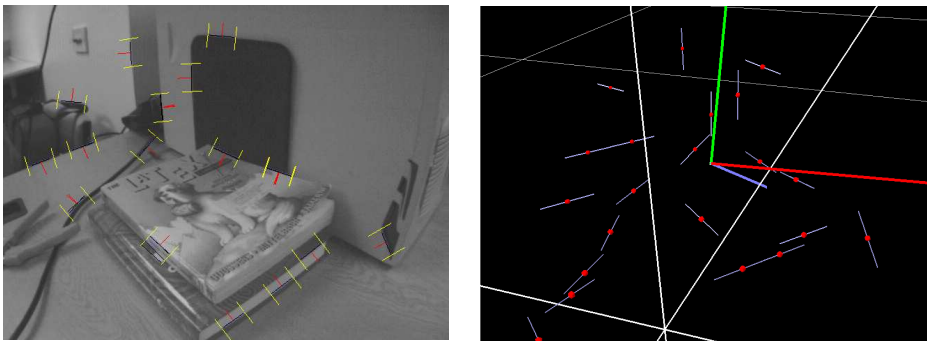


Figure 5: A scene with 3D structure, and the map, shown in clockwise order from side, overhead, and perspective views.

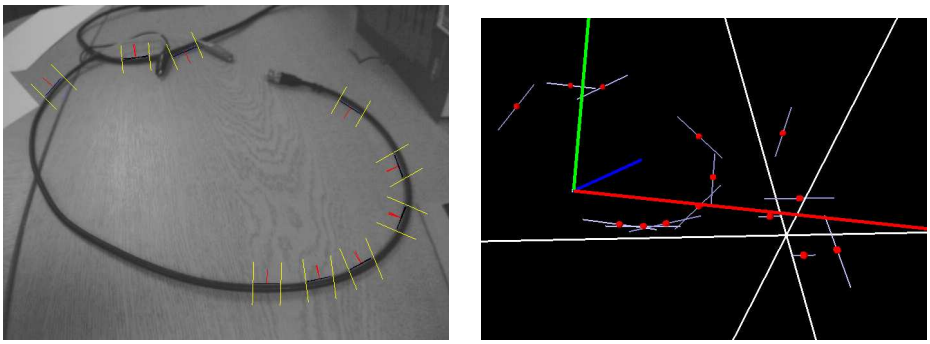


Figure 6: Edgelets on curved surfaces.

[10] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI*, pages 1151–1156, 2003.

[11] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *ICCV 2005*, volume 2, pages 1508–1515, October 2005.

[12] M. C. Shin, D. Goldgof, and K. W. Bowyer. An objective comparison methodology of edge detection algorithms using a structure from motion task. In *CVPR '98*, page 190, Washington, DC, USA, 1998. IEEE Computer Society.

[13] Robert Sim, Pantelis Elinas, Matt Griffin, and James J. Little. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI RUR*, Edinburgh, Scotland, 2005.

[14] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(11):1021–1032, 1995.