Surface-Based Structure-from-Motion Using Feature Groupings

Phil McLauchlan, Xinquan Shen, Anastasios Manessis, Phil Palmer and Adrian Hilton
School of Electronic Engineering,
Information Technology & Mathematics
University of Surrey
Guildford GU2 5XH, UK

ABSTRACT

In this paper we describe a complete system from feature extraction to reconstruction of 3D models of indoor environments. The system uses a novel matching algorithm which matches groupings of features associated with boundaries of objects in the scene. We also present an extension of our structure-from-motion algorithm to incorporate surface constraints. We describe how planar surfaces can be incorporated into the model update procedure, and are hypothesised from the matched groupings of features between image frames. We present reconstructions of environments taken by an autonomous robot to demonstrate the improvement that can be achieved by this approach.

Keywords: Environment modelling, structure from motion, feature matching, mobile robotics

1. INTRODUCTION

In this paper we consider the problem of reconstructing 3D indoor environments, in near real time, using an autonomous mobile robot. The reconstruction is based upon a structure-from-motion algorithm using data from feature extraction software. Unlike other reconstruction methods, an intrinsic part of our problem is coping with inaccuracy in the features and their matching. To build in robustness to the reconstruction process, we use groupings of features which are related to boundaries of objects in the scene. The use of more complex features aids the matching reliability, and can provide important geometric information for structure-from-motion.

We advocate an integrated approach to feature matching and surface recovery. Automatic feature matching and grouping provides hypotheses for features that lie on the same surface. This aids the matching process and provides high-level information for structure recovery which is then passed to the variable state dimension filter (VSDF) [13, 12]. We shall demonstrate that the introduction of constraints between features which lie on a common surface reduces the error in reconstruction of individual features, and considerably richens the scene representation.

Surface-based structure-from-motion is not a new concept. Some work has used the planarity constraint to aid feature matching, surface recovery and calibration [20, 23]. The "plane-plus-parallax" concept [17] and the model-based stereo algorithm of [5] are based on the idea of a reference planar surface. The approaches closest to the current work are those of Bondyfalat & Bougnoux [3], who attempt to build and compute a minimal representation of a scene given multiple constraints, and Szeliski & Torr [21], who were the first to consider the introduction of plane constraints directly in structure-from-motion, although in a non-optimal statistical framework.

In this paper we generalise surface-based structure-from-motion to enable constraints associated with features on multiple surfaces to be integrated together in the recovery of scene structure, and extend the VSDF algorithm to use the "recursive partitioning" sparse matrix inversion algorithm [19], whose computational complexity is linear in the number of scene features and surfaces, and thus is applicable for long image sequences. This contrasts with the cubic (at least) complexity of approaches which ignore the sparse partitioned nature of the problem such as [3].

The state-of-the-art in feature-based structure-from-motion is based on tracking corner and/or line features over two or more images, and reconstructing each feature independently in 3D under the assumption that the camera is moving in a stationary scene [7]. Geometric constraints are encoded using the fundamental matrix and tri-focal tensor [11, 8]. Methods for incorporating surface constraints in stereo vision [9] do not easily generalise to image sequences of complex scenes.

To exploit the geometric approach in complex environments, we present a method of matching groupings of features associated with lines and corners representing the outlines of 3D objects. We match the whole grouping between successive frames and then exploit the relationship between these features to generate planar hypotheses. The matching of feature groupings provides a robust dataset for the reconstruction and closely inter-relates the feature matching with 3D reconstruction.

Although surface fitting could be performed as a post-process this approach fails to exploit the context of matched features which can resolve otherwise ambiguous interpretation, and does not allow us to reduce the degrees-of-freedom in both matching and structure estimation.

We believe this symbiotic approach is necessary if structure-from-motion is to develop into a practical tool for recovering 3D models of the real world. We demonstrate this belief by evaluating the recovered structure of unknown environments explored by an autonomous robot.

2. SURFACE AND THE VSDF

To handle the various parameter vectors and matrices, we employ the variable state dimension filter (VSDF), which has been developed [13, 12] for a variety of camera calibration and scene reconstruction applications. It is designed as a statistical tool for efficiently combining data from multiple images, or other sensors. We shall briefly summarise the VSDF approach to the SFM problem, before describing how we have extended it to build in surface information.

The VSDF is a batch/recursive estimation algorithm, based on the correspondences of features across multiple images. Given a reasonable initial starting point for all the unknown parameters (obtained in our case from a combination of robot odometry and triangulation), the images are combined using a standard batch Levenberg-Marquardt algorithm to estimate the 3D structure and motion.

In our previous work we noted the sparse structure of the Jacobian product matrix that results when Gauss-Newton methods (such as Levenberg-Marquardt) are applied to the structure-from-motion problem, and applied a simple partitioning scheme to achieve the requisite performance gain. This approach works well for the existing point/line feature models typically used in structure-from-motion, but we show below that when we consider surface constraints, the sparseness takes on a more complex shape, and to maintain efficiency we have now implemented a block-matrix version of the "recursive partitioning" algorithm matrix [19], which takes full advantage of the modified sparse matrix structure.

The VSDF applies a state vector approach analogous to the Kalman filter (not directly applicable for both theoretical and efficiency reasons), and separates out components of the structure (*local* states $\boldsymbol{x}_l$); plane parameters (*constraint* states $\boldsymbol{x}_c$); motion parameters (*dynamic* states $\boldsymbol{x}_d$) and calibration parameters (*fixed* parameters $\boldsymbol{x}_f$). The parameters are automatically ordered in a beneficial way for solution of updates by recursive partitioning. This allows virtually any reconstruction problem to be solved in a general-purpose and efficient statistical framework, and allows controlled studies to be made of the trade-off between batch and recursive algorithms.

Suppose we have then n structure vectors $\boldsymbol{x}_{l1}, \ldots, \boldsymbol{x}_{ln}$ and motion parameter vectors $\boldsymbol{x}_{d(1)}, \ldots, \boldsymbol{x}_{d(k)}$ for k images, the image feature observation $\boldsymbol{z}_{i(j)}$ for feature i in image j takes the general form

$$\boldsymbol{z}_{i(j)} = \boldsymbol{h}_i(j; \boldsymbol{x}_{li}, \boldsymbol{x}_{d(j)}) + \boldsymbol{w}_{i(j)}$$

for some measurement (projection) function $\boldsymbol{h}_{i(j)}$ and zero-mean Gaussian random noise vector $\boldsymbol{w}_{i(j)}$, with covariance matrix $R_{i(j)}$. The maximum likelihood solution for $\boldsymbol{x}_l$ and $\boldsymbol{x}_d$ minimises the log-likelihood

$$J(\boldsymbol{x}_l, \boldsymbol{x}_d) = \sum_{i=1}^{n} \sum_{j=1}^{k} {\boldsymbol{\nu}_{i(j)}}^\top {R_{i(j)}}^{-1} \boldsymbol{\nu}_{i(j)} \qquad (1)$$

where $\boldsymbol{\nu}_{i(j)}$ is the innovation vector

$$\boldsymbol{\nu}_{i(j)} = \boldsymbol{z}_{i(j)} - \boldsymbol{h}_i(j; \boldsymbol{x}_{li}, \boldsymbol{x}_{d(j)})$$

that measure the image-plane error between the actual feature position $\boldsymbol{z}_{i(j)}$ and the projected 3D feature position $\boldsymbol{h}(\boldsymbol{x}_{li}^-, \boldsymbol{x}_{d\,(j)}^-)$.

It is then straightforward to show that batch Levenberg-Marquadt updates from $\boldsymbol{x}_l^-/\boldsymbol{x}_d^-$ to $\boldsymbol{x}_l^+/\boldsymbol{x}_d^+$ take the form

$$\begin{pmatrix} \boldsymbol{x}_l^+ \\ \boldsymbol{x}_d^+ \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_l^- \\ \boldsymbol{x}_d^- \end{pmatrix} + \begin{pmatrix} A_{ll} & A_{ld} \\ A_{ld}^\top & A_{dd} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{a}_l \\ \boldsymbol{a}_d \end{pmatrix} \qquad (2)$$

where

$$\boldsymbol{x}_l^{+/-} = \begin{pmatrix} \boldsymbol{x}_{l1}^{+/-} \\ \vdots \\ \boldsymbol{x}_{ln}^{+/-} \end{pmatrix}, \quad \boldsymbol{x}_d^{+/-} = \begin{pmatrix} \boldsymbol{x}_d^{+/-}{}_{(j)} \\ \vdots \\ \boldsymbol{x}_d^{+/-}{}_{(k)} \end{pmatrix},$$

$$A_{ll} = \begin{pmatrix} A_{ll1} & 0 & \ldots & 0 \\ 0 & A_{ll2} & . & \vdots \\ \vdots & . & . & \vdots \\ 0 & \ldots & 0 & A_{lln} \end{pmatrix}.$$

The blocks of the $n \times n$ block diagonal matrix A_{ll} are

$$A_{lli} = \sum_{j=1}^{k} {H_{li(j)}}^\top {R_{i(j)}}^{-1} H_{li(j)},$$

where $H_{li(j)} = \partial \boldsymbol{h} / \partial \boldsymbol{x}_l$. A_{ld} and A_{dd} are $n \times k$ and $k \times k$ block matrices formed from the blocks

$$\begin{aligned} A_{ldi(j)} &= {H_{li(j)}}^\top {R_{i(j)}}^{-1} H_{di(j)} \\ A_{dd(j)} &= \sum_{i=1}^{n} {H_{di(j)}}^\top {R_{i(j)}}^{-1} H_{di(j)} \end{aligned}$$

where $H_{di(j)} = \partial \boldsymbol{h} / \partial \boldsymbol{x}_d$. Both $H_{li(j)}$ and $H_{di(j)}$ are evaluated at $(\boldsymbol{x}_{li}^-, \boldsymbol{x}_{d\,(j)}^-)$. Finally $\boldsymbol{a}_l$ and $\boldsymbol{a}_d$ are $n \times 1$ and $k \times 1$ block vectors formed from blocks

$$\begin{aligned} \boldsymbol{a}_{li} &= \sum_{j=1}^{k} {H_{li(j)}}^\top {R_{i(j)}}^{-1} \boldsymbol{\nu}_{i(j)} \\ \boldsymbol{a}_{d(j)} &= \sum_{i=1}^{n} {H_{di(j)}}^\top {R_{i(j)}}^{-1} \boldsymbol{\nu}_{i(j)} \end{aligned}$$

The full system matrix in (2), can be identified as the inverse covariance matrix of the state parameters. Now because A_{ll} is block diagonal (due simply to the fact that each image feature observation involves a single image and a single scene feature), we can invert A_{ll}, and hence solve this set of equations, in time $\mathcal{O}(n)$, instead of $\mathcal{O}(n^3)$ as would be the case for a general system. This is achieved by solving the partitioned system first for the update of $\boldsymbol{x}_d$, then for $\boldsymbol{x}_l$, as follows:

$$\begin{aligned} \boldsymbol{x}_d^+ &= \boldsymbol{x}_d^- + (A_{dd} - A_{ld}^\top A_{ll}^{-1} A_{ld})^{-1} (\boldsymbol{a}_d - A_{ld}^\top A_{ll}^{-1} \boldsymbol{a}_l) \\ \boldsymbol{x}_l^+ &= \boldsymbol{x}_l^- + A_{ll}^{-1} (\boldsymbol{a}_l - A_{ld} (\boldsymbol{x}_d^+ - \boldsymbol{x}_d^-)) \end{aligned}$$

Note that in a naive implementation (2) is singular, because of the arbitrary choice of the world coordinate frame used to represent the $\boldsymbol{x}_l$ and $\boldsymbol{x}_d$, and special constraining techniques are required to remove the "unwanted" degrees of freedom; see [10, 22].

The VSDF iterates the above update to convergence. So long as the strict block-diagonal shape of A_{ll} is maintained, and no extra parameters are introduced, the simple partitioning scheme described above can yield an efficient and accurate solution.

Incorporating surface constraints

In order to incorporate surfaces into structure-from-motion,

we need to specify how the surfaces relate to simpler features such as points and lines. We shall treat here the simplest case of one or more planar surfaces in the scene, but it should be clear that other constraints may be implemented in a similar manner.

We represent a plane by its normal vector $\boldsymbol{n}$ and its distance d from the origin, so that 3D points $\boldsymbol{X}$ lying on the plane obey the equation $\boldsymbol{n}.\boldsymbol{X} = d$. Together $\boldsymbol{n}$ and d constitute a constraint state vector $\boldsymbol{x}_c$, to be estimated along with the structure and motion. To impose the plane constraints we apply the technique of Lagrange multipliers. Considering a single plane $\boldsymbol{x}_c$ constraining r scene points $i_1, i_2, \ldots, i_r$, for the sake of simplicity, we identify $\boldsymbol{x}_{li} = \boldsymbol{X}_i$, and incorporate an extra term into the functional J in (1):

$$J'(\boldsymbol{x}_l, \boldsymbol{x}_d) = \sum_{i,j} \boldsymbol{\nu}_{i(j)}{}^\top R_{i(j)}{}^{-1} \boldsymbol{\nu}_{i(j)} + \sum_{p=1}^{r} \lambda_p (\boldsymbol{n}.\boldsymbol{X}_{i_p} - d),$$

where λ_p are the Lagrange multipliers. Then minimising J' over $\boldsymbol{x}_l$, $\boldsymbol{x}_c$, $\boldsymbol{x}_d$ and the λ_p's simultaneously will achieve our aim of minimising the original J in (1) subject to all the points $i_1, i_2, \ldots, i_r$ lying on the same plane. This is because differentiation w.r.t. each λ_p reduces to the plane constraint equation

$$\boldsymbol{n}.\boldsymbol{X}_{i_p} = d,$$

and so the constraints are satisfied at the solution. We "attach" the Lagrange coefficient λ_p to each structure vector $\boldsymbol{x}_{li_p}$, generating augmented local states

$$\boldsymbol{y}_{li_p}^\top = (\boldsymbol{x}_{li_p}^\top \ \lambda_p),$$

and insert them and the plane parameters $\boldsymbol{x}_c$ into the state vector. We also write the general scene constraint for a feature i with constraint $\boldsymbol{x}_c$ as $\boldsymbol{f}_i(\boldsymbol{x}_{li}, \boldsymbol{x}_c) = \boldsymbol{k}_i$ (in this case $\boldsymbol{f}_i(.) = \boldsymbol{n}.\boldsymbol{X}_i - d$ and $\boldsymbol{k}_i = (0)$). We will need the Jacobians $F_{li} = \partial \boldsymbol{f}_i / \partial \boldsymbol{x}_{li}$ and $F_{ci} = \partial \boldsymbol{f}_i / \partial \boldsymbol{x}_c$. Then the stationary point of J', computed to first order, yields the modified update rule

$$\begin{pmatrix} \boldsymbol{y}_l^+ \\ \boldsymbol{x}_c^+ \\ \boldsymbol{x}_d^+ \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_l^- \\ \boldsymbol{x}_c^- \\ \boldsymbol{x}_d^- \end{pmatrix} + \begin{pmatrix} B_{ll} & B_{lc} & B_{ld} \\ B_{lc}^\top & 0 & 0 \\ B_{ld}^\top & 0 & A_{dd} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{b}_l \\ \boldsymbol{0} \\ \boldsymbol{a}_d \end{pmatrix}$$

where

- B_{ll} is a block-diagonal matrix formed from the blocks $B_{lli} = \begin{pmatrix} A_{lli} & F_{li}^\top \\ F_{li} & 0 \end{pmatrix}$,
- B_{lc} is a column of blocks, each block being $B_{lci} = \begin{pmatrix} 0 \\ F_{ci} \end{pmatrix}$,
- B_{ld} is formed as a grid of blocks $B_{ldi(j)} = \begin{pmatrix} A_{ldi(j)} \\ \boldsymbol{0}^\top \end{pmatrix}$,
- $\boldsymbol{b}_{li} = \begin{pmatrix} \boldsymbol{a}_{li} \\ \boldsymbol{k}_i - \boldsymbol{f}_i(\boldsymbol{x}_{li}^-, \boldsymbol{x}_c^-) \end{pmatrix}$

Thus the modified problem with a single plane and Lagrange multipliers has a very similar structure to the basic reconstruction problem, with a block-diagonal matrix B_{ll} allowing updates in time $\mathcal{O}(n)$. This scheme clearly generalises easily to different structure vector types and constraints. For instance, perpendicularity constraints between pairs of planes can easily be incorporated by attaching Lagrange multipliers to the appropriate plane parameter vectors.

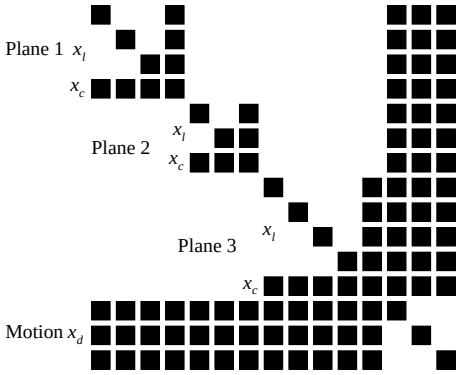


Figure 1. The shape of the inverse covariance matrix when surface constraints are included. Three surfaces are illustrated, each with their own constraint parameters $\boldsymbol{x}_c$ and structure parameters $\boldsymbol{x}_l$. The motion parameters $\boldsymbol{x}_d$ take the bottom rows of blocks.

Recursive partitioning

The simple partitioning scheme used above, whereby the state vector is split into the local states $\boldsymbol{x}_l$ and the rest, works fine so long as the number of local states is much greater than the number of constraint vectors $\boldsymbol{x}_{cl}$ or images $\boldsymbol{x}_{d(j)}$. With multiple planes it is not difficult to see that the shape of the system matrix takes the form shown in figure 1. Each surface (three are shown here) creates its own partitionable system, and where surfaces are linked by common features, extra blocks in the matrix are filled in. This more general form is much more difficult to handle efficiently, but the photogrammetric technique of *recursive partitioning* [19] turns out to be ideal in solving such matrix inverse problems, so long as the blocks are ordered in the right way. Recursive partitioning performs the partitioning of the system matrix one row & column of blocks at a time, starting from the top-left, in a manner analogous to block-based Gaussian elimination and back-substitution. It incorporates the simple partitioning scheme above as a special case, and in our view should become the standard tool for sparse symmetric bundle adjustment problems in computer vision. For the simple "block-diagonal with spikes" matrix form illustrated in figure 1, recursive partitioning maintains computational complexity proportional to n and also proportional to the number of spikes (surfaces). We have also developed a novel approach to computing the covariance matrix without inducing undesirable fill-in of the sparse structure.

More details of the VSDF and the implementation of recursive partitioning may be found in the "Horatio" software library documentation [14].

3. FEATURE MATCHING AND GROUPING

The determination of the structure in the scene is very dependent upon consistent tracking of features through the

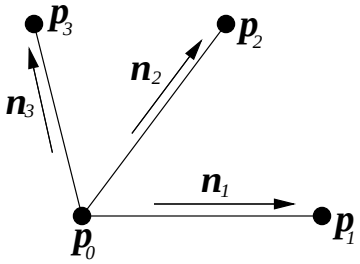


Figure 2. Representation of junctions

sequence. For the efficiency and robustness, we use feature groupings as matching primitives instead of using lines and/or corners as matching primitives [18, 1, 4]. A group of line segments intersected at a corner point is detected as a junction. Feature matching is then determined by matching junctions. Epipolar constraint directly determined from the odometry of the robot is used to derive matching criterion. The tolerance of the matching algorithm to the inaccuracy of the odometry is achieved by exploiting the complexity of the junction structure and investigating the similarity in junction topologies. The operation of junction matching produces a set of matched line segments and points from which line segments are further grouped in terms of their 3D co-planarity.

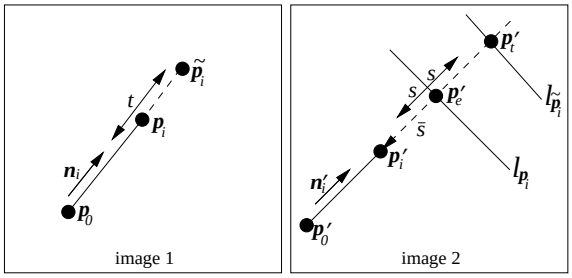
Junction detection and matching

Junctions are detected in three stages. We have implemented the optimised edge detector described by [16], which then provides data to a robust Hough transform line detector [15]. The line segments produced are then used in a junction finder which locates normal V type junctions, triple junctions and occluded T junctions between the sets of Hough lines. Each junction is described by a junction location and a set of line segments (called junction segments) associated with the junction location. Figure 2 shows a junction with junction location $\boldsymbol{p}_0$ and three associated line segments with end-points $\boldsymbol{p}_i$ and orientations $\boldsymbol{n}_i$, $i = 1, 2, 3$. The propagation of uncertainty in each stage of junction detection is investigated. Since line features are robust, we assume that line segments are detected with only uncertainty at the location of their end-points. For each junction, the detected end-point $\boldsymbol{p}_i$ is then randomly distributed about its true position $\tilde{\boldsymbol{p}}_i$ along the direction $\boldsymbol{n}_i$, and the distribution of the true end-point $\tilde{\boldsymbol{p}}_i$ is characterised by the following probability density function:

$$f(t) = \begin{cases} e^{-\alpha t}, & t \geq 0 \\ e^{\beta t}, & t \leq 0 \end{cases} \tag{3}$$

where α and β are parameters determining the strength of the uncertainty in extending and shortening the end-point along the line segment by a distance t.

Matching between two junctions is established by determining a correspondence between both junction locations and junction segments. For each junction in image 1, its matching candidates in image 2 are first determined by checking the epipolar constraint on junction locations. For each potential match, we then investigate the correspondence between the junction segments by determining a probability that two junction segments describe the same 3D structure.



(a) (b)

Figure 3. Epipolar constraint on junction end-points

Let $J = \{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_K\}$ be a junction detected in image 1, where, $\boldsymbol{p}_0$ is the junction location, and $\boldsymbol{p}_i$ is the end-point of the ith junction segment with orientation $\boldsymbol{n}_i$, $1 \leq i \leq K$. According to the previous discussion on the uncertainty of junctions, $\tilde{\boldsymbol{p}}_i$, the true position of the detected end-point $\boldsymbol{p}_i$ can expressed as follows:

$$\tilde{\boldsymbol{p}}_i = \boldsymbol{p}_i + t\boldsymbol{n}_i \tag{4}$$

where, t is a random variable describing a signed distance from $\boldsymbol{p}_i$ to $\tilde{\boldsymbol{p}}_i$ along the direction parallel to $\boldsymbol{n}_i$ (see Fig. 3(a)). According to the epipolar geometry [6], given a point $\boldsymbol{p}_i = [x_i\, y_i]^\top$ in image 1, its corresponding point in image 2 is on a line (called the epipolar line associated with the point $\boldsymbol{p}_i$) determined by $F[x_i\, y_i\, 1]^\top$, where F is the fundamental matrix which, in this work, is directly derived from the odometry of the robot to achieve an efficient matching operation. Let $J' = \{\boldsymbol{p}'_0, \boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_K\}$ be a junction detected in image 2 which we hypothesise should match junction J in image 1, and $l_{\boldsymbol{p}_i}$ and $l_{\tilde{\boldsymbol{p}}_i}$ denote epipolar lines in image 2 associated with the detected end-point $\boldsymbol{p}_i$ and true (unknown) end-point $\tilde{\boldsymbol{p}}_i$ in image 1 respectively. Assume that $\boldsymbol{p}'_t$ is the intersection point between the line $\overline{\boldsymbol{p}'_0\boldsymbol{p}'_i}$ and the epipolar line $l_{\tilde{\boldsymbol{p}}_i}$. Then:

$$\boldsymbol{p}'_t = \boldsymbol{p}'_e + s\boldsymbol{n}'_i \tag{5}$$

where, $\boldsymbol{p}'_e$ is the intersection point between the line $\overline{\boldsymbol{p}'_0\boldsymbol{p}'_i}$ and the epipolar line $l_{\boldsymbol{p}_i}$, s is the signed distance from $\boldsymbol{p}'_e$ to $\boldsymbol{p}'_t$ (see Fig. 3(b)). Clearly, s is a function of the random variable t. The distribution of $\boldsymbol{p}'_t$ is then characterised by the probability density function of the random variable s, $g(s)$ which is determined as follows [2]:

$$g(s) = f(t)\left|\frac{dt}{ds}\right| \tag{6}$$

The relation between random variables s and t can be determined by the epipolar constraint on two images.

To quantify how likely two junction segments $\overline{\boldsymbol{p}_0\boldsymbol{p}_i}$ and $\overline{\boldsymbol{p}'_0\boldsymbol{p}'_i}$ actually describe the same 3D structure, we determine a probability $\mathcal{H}(\boldsymbol{p}_i, \boldsymbol{p}'_i)$ that the epipolar line $l_{\tilde{\boldsymbol{p}}_i}$ passes through the true end-point $\tilde{\boldsymbol{p}}'_i$ given the detected end-points $\boldsymbol{p}_i$ and $\boldsymbol{p}'_i$. Let r be a random variable describing a signed distance from $\boldsymbol{p}'_i$ in image 2 to the corresponding true end-point $\tilde{\boldsymbol{p}}'_i$, i.e.

$$\tilde{\boldsymbol{p}}'_i = \boldsymbol{p}'_i + r\boldsymbol{n}'_i \tag{7}$$

The probability density function of r is then given by $f(r)$ defined in (3). Since r and s are independent, the probability

that the epipolar line $l_{\tilde{\boldsymbol{p}}_i}$ passes through $\tilde{\boldsymbol{p}}'_i$ which is located at a singed distance s away from the point $\boldsymbol{p}'_e$ can be taken as[1]:

$$f(r)g(s)\Delta r\Delta s\,\big|_{r=s-\bar{s}} = f(s-\bar{s})g(s)(\Delta s)^2 \qquad (8)$$

where, Δr and Δs are differential elements of variables r and s respectively, and $\bar{s}$ is the signed distance from the point $\boldsymbol{p}'_e$ to the point $\boldsymbol{p}'_i$ along the direction $\boldsymbol{n}'_i$. The maximum value of (8) is then defined as the probability $\mathcal{H}(\boldsymbol{p}_i, \boldsymbol{p}'_i)$, i.e.

$$\mathcal{H}(\boldsymbol{p}_i, \boldsymbol{p}'_i) = \max_s \left\{ f(s-\bar{s})g(s)(\Delta s)^2 \right\} \qquad (9)$$

Let $\mathcal{H}(\boldsymbol{p}'_i, \boldsymbol{p}_i)$ be the probability that the epipolar line $l_{\tilde{\boldsymbol{p}}'_i}$ passes through the true end-point $\tilde{\boldsymbol{p}}_i$ given the detected end-points $\boldsymbol{p}_i$ and $\boldsymbol{p}'_i$. The value $\mathcal{H}(\boldsymbol{p}_i, \boldsymbol{p}'_i)\mathcal{H}(\boldsymbol{p}'_i, \boldsymbol{p}_i)$ then characterises the likelihood that the junction segment $\overline{\boldsymbol{p}_0\boldsymbol{p}_i}$ matches with junction segment $\overline{\boldsymbol{p}'_0\boldsymbol{p}'_i}$. A matching strength is then derived by combining this likelihood associated with each pair of corresponding line segments, and the proximity of the junction point to the epipolar line associated with the matched junction point. The derived matching strength associated with two junctions is further combined with the matching on junction topologies to achieve a robust matching. Junctions in the image are topologically described by a graph in which each vertex represents a junction and each edge linking two vertices corresponds to the line segment connecting the junctions. A connected sub-graph of this graph then describes the topology of a junction network in which each junction shares a line segment with another junctions. A maximal structural matching between a junction network associated with a junction in one image and a junction network associated with a junction in another image is then determined by simultaneously traversing two graphs to find a matched sub-graph. Matching strength for each pair of corresponding junctions within the matched sub-graph is then accumulated to produce a similarity value for two junctions. The matching is then determined by finding a candidate matching with a maximal similarity value.

Feature grouping

The aim of feature grouping is to establish features which correspond to the same scene plane in order to provide constraints to VSDF. The line segments in the junction groupings are themselves grouped in terms of 3D co-planarity. This is achieved, prior to structure from motion, by using odometry information only.

Let junction segments $\overline{\boldsymbol{p}_0\boldsymbol{p}_1}$ and $\overline{\boldsymbol{p}'_0\boldsymbol{p}'_1}$ be the projection of a 3D line L onto images 1 and 2 respectively. Assume that R_i and $\boldsymbol{T}_i$, $i = 1, 2$ are the rotation and translation associated with the two images, and are known from odometry. By using homogeneous coordinates to represent entities in image plane, the orientation $\boldsymbol{N}$ of the line L can be determined from the following equation:

$$\xi_1\boldsymbol{p}_1 - \xi_0\boldsymbol{p}_0 = R_1\boldsymbol{N} \qquad (10)$$

[1] This is a probability element [2] corresponding to the probability that the values of random variables r and s are located within small intervals $[r-\Delta r/2, r+\Delta r/2]$ and $[s-\Delta s/2, s+\Delta s/2]$ respectively. In numerical implementation, we can compute this probability by giving a suitable value to differential elements Δr and Δs, e.g. $\Delta r = \Delta s = 1$.

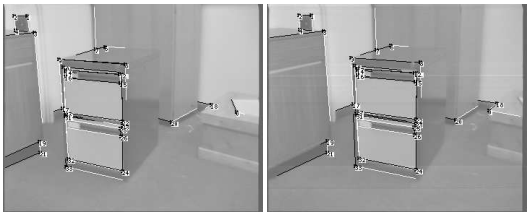


Figure 4. Matched junctions associated with the image pair $(6, 7)$ in the first sequence.

Features	Performance		
	total	errors	% errors
Junctions	209	29	13.88
Line segments	267	24	8.99
Junction + line segments	476	53	11.13

Table 1. Performance of the junction matching on the first sequence.

where, $\xi_i, i = 0, 1$ are determined from the correspondence of two line segments:

$$\begin{aligned} \xi_i\boldsymbol{p}_i &= R_1\boldsymbol{X} + \boldsymbol{T}_1 \\ \xi'_i\boldsymbol{p}'_0 + \delta'_i\boldsymbol{n}' &= R_2\boldsymbol{X} + \boldsymbol{T}_2 \end{aligned} \qquad (11)$$

where, ξ'_i and δ'_i are constants, $\boldsymbol{n}'$ is the line orientation, and $\boldsymbol{X}$ is a point on L.

Two junction segments define a 3D plane, and their co-planarity with a third segment can be determined from the vector triple product of orientations of corresponding 3D lines and the normal distance of the 3rd line to the plane.

4. EXPERIMENTAL RESULTS

In this section we describe results obtained from two image sequences. The first shows the corner of the lab as the robot passes by, and the second sequence was taken as the robot moved along a corridor in which obstacles have been placed. Fig. 4 shows the matched junctions associated with the image pair $(6, 7)$ in the first sequence. The performance of the matching algorithm for the whole sequence is summarised in table 1. The reason that the number of mis-matched junctions is larger than the number of mis-matched line segments is because a false junction match may still match correct lines. Fig. 5 shows the experimental result on the image pair $(1, 2)$ in the corridor sequence. The performance for the whole sequence is summarised in the table 2.

Most of the false matchings are due to the inaccurate epipolar constraint derived from inaccurate odometry information. By removing outliers from the matching data, the structure from motion algorithm provides a better estimate of the camera motion. This can then be fed back to the matcher to improve performance and provide estimates for the structure.

In order to evaluate the effect on the reconstructed scene when plane constraints are enforced in the SFM, we have taken pairs of lines/planes from an image and found the angle between them in the 3D reconstruction. These have been chosen so that we know that the lines should be perpendicular or parallel, but not forced by the constraints themselves.

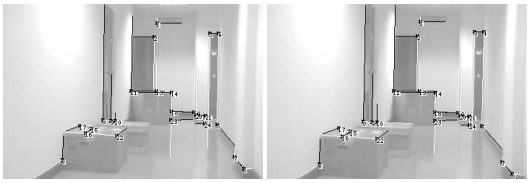


Figure 5. Matched junctions associated with the image pair $(1, 2)$ in the corridor sequence.

Features	Performance		
	total	errors	% errors
Junctions	419	50	11.93
Line segments	658	47	8.27
Junction + line segments	1077	97	9.01

Table 2. Performance of the junction matching on the corridor sequence.

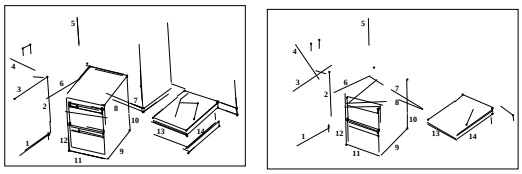
Fig. 6 shows reconstructions of the first sequence with surface constraint (Fig. 6a) and without surface constraint (Fig. 6b). 3D lines used for angle comparison are marked with numbers. Angles between line pairs are summarised in table 3. This table shows that a clear improvement in the angles has been achieved by enforcing surface constraints. The two outlier cases have both been resolved when constraints are used.

Fig. 7 shows two views from novel viewpoints of the 3D data obtained from the first image sequence. The texture maps are generated automatically from the plane data, utilising the convex hull of the features lying in the plane, and averaging the pixel values of all back-projected images in which the plane was visible.

Figure 8 shows two views of the 3D features and the texture-mapped planes obtained from the corridor image sequence. In this case, the relative lack of parallax in the sequence induced by the constant forward robot motion meant that the structure-from-motion could not reliably distinguish outliers in the data from the automatic feature matcher, so we used hand-matched feature data. In future we will use the on-board robot odometry to help constrain the camera motion, and so help with outlier detection and reconstruction. The well-known instability of SFM in the presence of weaker parallel signals is a major motivator for our approach of incorporating knowledge of self-motion into reconstruction algorithms, where possible. In both sequences, both line and point (junction) data are integrated with plane fitting in the VSDF bundle adjustment algorithm.

5. CONCLUSIONS

We have addressed the problem of using noisy data to reconstruct the 3D scene. We have extended the VSDF structure-from-motion algorithm to recover surfaces by exploiting the constraints available from 3D point and line scene features lying on the same planar surfaces. The features and surface parameters are computed optimally from image observations of the features. The scheme generalises naturally



(a) (b)

Figure 6. 3D lines used for the evaluation of structure from motion algorithm. (a) 3D lines reconstructed with the plane constraint; (b) 3D lines reconstructed without the plane constraint.

Pair no.	Angles with constraints	Angles without constraints
(1,2)	88.94	87.04
(2,3)	89.96	90.28
(3,4)	87.82	59.70
(5,6)	92.24	95.06
(5,7)	87.48	87.45
(7,8)	0.55	28.00
(9,10)	91.9	89.80
(11,12)	90.45	88.83
(13,14)	86.5	86.28

Table 3. Evaluation of SFM with and without plane constraint. The ground-truth angle between the lines in each pair is 90^o, except for pair (7.8) which are parallel lines in the world.

to other feature and surface types. We have combined the photogrammetric recursive partitioning algorithm for efficient inversion of sparse matrices, with a Lagrange multiplier technique to impose the constraints between surface features and the surface parameters themselves.

We have modelled the principal error sources from the feature extraction algorithms and used this to define a matching uncertainty to determine feature correspondences. The matching criterion is based upon the well known epipolar constraint, which only provides a necessary rather than sufficient condition for matching. We have extended the matching criterion to take account of local topology of junctions, providing a much greater robustness to the procedure.

Acknowledgements

We acknowledge the award of EPSRC grant GR/L34099 which has funded the work described herein.

References

[1] P.A. Beardsley, P. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. In *Proc. 4th European Conf. on Computer Vision, Cambridge*, volume 2, pages 683–695. Springer-Verlag, 1996.

[2] Z. W. Birnbaum. *Introduction to Probability and Mathematical Statistics*. Harper & Brothers, 1962.

[3] D. Bondyfalat and S. Bougnoux. Imposing euclidean constraints during self-calibration processes. In *Proc.*

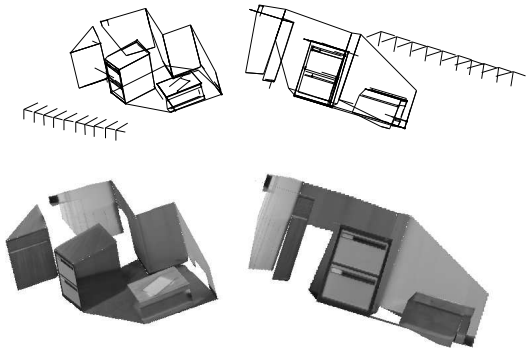


Figure 7. Two views of the 3D point and line structure and rendered planes obtained from the first image sequence. The reconstructed camera positions are also shown.

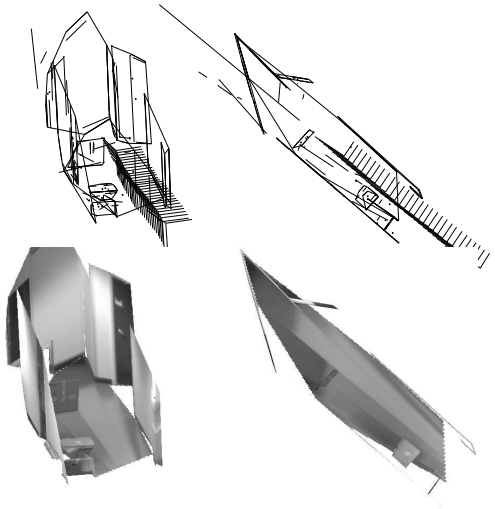


Figure 8. Two views of the 3D structure and planes obtained from the corridor image sequence

SMILE Workshop (associated with ECCV'98), pages 69–72, 1998.

[4] Y. Chang and J. K. Aggarwal. Line Correspondences from Cooperating Spatial and Temporal Grouping Processes for a Sequence of Images. *Computer Vision and Image Understanding*, 67(2):186–201, 1997.

[5] P. Debevec, C.J. Taylor, and J. Malik. Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIG-GRAPH Conf.*, 1996.

[6] Oliver Faugeras. *Three Dimensional Computer Vision*. The MIT Press, 1993.

[7] A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. 5th European Conf. on Computer Vision, Freiburg*, volume 1, pages 311–326. Springer-Verlag, June 1998.

[8] R.I. Hartley. A linear method for reconstruction from lines and points. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, 1995.

[9] W. Hoff and N. Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):121–136, February 1989.

[10] H. M. Karara. *Non-Topographic Photogrammetry*. Americal Society for Photogrammetry and Remote Sensing, 1989.

[11] Q.-T. Luong and O.D. Faugeras. A stability analysis of the fundamental matrix. In *Proc. 3rd European Conf. on Computer Vision, Stockholm*, pages 577–588, May 1994.

[12] P.F. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, pages 314–320, June 1995.

[13] P.F. McLauchlan and D.W. Murray. Active camera calibration for a Head-Eye platform using the variable State-Dimension filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):15–22, 1996.

[14] P.F. McLauchlan and A. Rahimi. Horatio: the efficient vision library; on-line documentation. http://www.ee.surrey.ac.uk/Personal/P.McLauchlan/, 1998.

[15] P. L. Palmer, J. Kittler, and M. Petrou. An optimising line finder using a hough transform algorithm. *CVGIP: Image Understanding*, 67(1):1–23, 1997.

[16] M. Petrou and J. Kittler. Optimal edge detectors for ramp edges. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:483–491, 1991.

[17] H.S. Sawhney. Simplifying motion and structure analysis using planar parallax. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[18] C. Schmid and A. Zisserman. Automatic Line Matching across Views. In *Computer Vision and Pattern Recognition*, pages 666–671, Puerto Rico, June 1997. IEEE Computer Society Press.

[19] C.C. Slama, C. Theurer, and S.W. Henriksen, editors. *Manual of Photogrammetry*. American Society of Photogrammetry, 1980.

[20] S. Sull and N. Ahuja. Integrated 3D analysis of flight image sequences. In *Proc. 3rd European Conf. on Computer Vision, Stockholm*, pages 211–216, May 1994.

[21] R. Szeliski and P. Torr. Geometrically constrained structure from motion: Points on planes. In *Proc. SMILE Workshop (associated with ECCV'98)*, pages 33–36, 1998.

[22] B. Triggs. Optimal estimation of matching constraints. In *Proc. SMILE Workshop (associated with ECCV'98)*, 1998.

[23] W. Triggs. Autocalibration from planar scenes. In *Proc. 5th European Conf. on Computer Vision, Freiburg*, volume 1, pages 89–105. Springer-Verlag, June 1998.