In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010

Estimating Camera Pose from a Single Urban Ground-View Omnidirectional Image and a 2D Building Outline Map

Tat-Jen Cham astjcham@ntu.edu.sg
Arridhana Ciptadi carridhana@ntu.edu.sg
Wei-Chian Tan wctan@ntu.edu.sg
Minh-Tri Pham[1] t.pham@surrey.ac.uk
Liang-Tien Chia asltchia@ntu.edu.sg

School of Computer Engineering, Nanyang Technological University; Singapore
[1]Surrey Space Centre, University of Surrey; Guildford, England

Abstract

A framework is presented for estimating the pose of a camera based on images extracted from a single omnidirectional image of an urban scene, given a 2D map with building outlines with no 3D geometric information nor appearance data. The framework attempts to identify vertical corner edges of buildings in the query image, which we term VCLH, as well as the neighboring plane normals, through vanishing point analysis. A bottom-up process further groups VCLH into elemental planes and subsequently into 3D structural fragments modulo a similarity transformation. A geometric hashing lookup allows us to rapidly establish multiple candidate correspondences between the structural fragments and the 2D map building contours. A voting-based camera pose estimation method is then employed to recover the correspondences admitting a camera pose solution with high consensus. In a dataset that is even challenging for humans, the system returned a top-30 ranking for correct matches out of 3600 camera pose hypotheses (0.83% selectivity) for 50.9% of queries.

1. Introduction

When tourists face problems navigating around a new city, it is usually not because they are unable to recognize famous landmarks or unique buildings, but rather they are in an area where buildings appear bland and similar to each other. Concurrently, most city maps today show only building outlines in plan view, and do not contain 3D geometry nor appearance information except for unique landmarks.

The problem posed is: *determine the camera pose based on a single omnidirectional image and given apriori a 2D building outline map of the locale, without appearance data.* See figure 1. We consider a map of 111 buildings with nondescript shape and appearance.

Figure 1. Estimating camera location (blue circle) and pose from four images extracted from a **single** omnidirectional image, given **only** a 2D plan-view map of 111 building outlines. Here we represent an omnidirectional image as a collection of planar perspective images with a common optical center. Google StreetView images are used in the experiments.

While GPS is popularly used for location-awareness, it is notoriously unreliable in high-rise urban areas where line-of-sight to GPS satellites are disrupted. The military will also be interested in vision-based location technologies as GPS can be spoofed or jammed in hostile territory.

1.1. Related Work

In structure-from-motion and wide-baseline stereo problems where finding correspondence is challenging, the matching of geometric features such as lines [11, 10], planes [13] and rectangles [7] have proven useful. Appearance information can be used to help discriminate line matching [1], but can only be applied to matching two or more real images. Alternatively, generative models may also be exploited [12].

In problems that involve searching through larger databases containing ground view appearance of buildings, matching approaches based on filter measures [15], SIFT

features [16] and facade line structures [8] have been explored.

A number of papers have addressed the problem of matching ground view images to aerial images [4], but these assume that 3D models in the aerial image are available, and focuses on specific buildings rather than a broad search across the entire aerial image. Doing this returns to a problem of wide-baseline stereo matching. Tracking using line correspondences between ground view video and an aerial image was carried out in [5].

Partially related are the works that attempt 3D reconstruction from a single image [2, 9], although these do not involve a search problem.

To the best of our knowledge, this is the first paper to address localization based on a single ground view omnidirectional image and a 2D plan-view building outline map with a large number of buildings (111). No appearance dataset is used, nor are any apriori 3D models of the buildings.

2. Framework Overview

Given an omnidirectional image, the goal is to estimate the extrinsic camera pose (up to an ambiguity in elevation from the ground plane), with the aid of a 2D plan view map of building outlines. The overall approach of our system involves identifying vertical corner edges of buildings in the image together with the plane normals of the neighboring walls, and subsequently grouping these into elemental planes and 3D structural fragments. These are matched to the 2D map building contours, and finally high consensus camera poses are obtained. A collection of planar perspective images sharing the same optical center is used to represent the omnidirectional image, where the principal points are known. See figure 2.

In this paper, we adopt two key assumptions: (1) the scene is quasi-Manhattan world, with a difference being that horizontal plane normals from different vertical planes need not be orthogonal; and (2) buildings are reasonably modeled as 2D extrusions of a ground-plane cross-section[1].

3. Line Descriptors in a 2 1/2-D Sketch

Under the above assumptions, each line in the world is either the result of (1) linear albedo changes across a single 3D plane, or (2) the intersection of two 3D planes. While such a line can be expressed solely as a line segment in 3D space, we can further imbue the line with additional geometric descriptors of one or two planes that are associated with the line as described.

With images captured from a single optical center, the 3D positions of image lines generally cannot be recovered. Nevertheless, through vanishing point analysis, particularly in a quasi Manhattan world context, it is possible to calibrate cameras to recover 3D directions of the image lines.

Marr's theory of visual processing includes an intermediate representation known as the $2\frac{1}{2}$-D sketch [6], in which local surface orientations have been estimated. In a similar spirit, an analogous version of the $2\frac{1}{2}$-D sketch is used in our framework, in which image lines have intermediate-level features comprising the 3D direction of the lines, as well as neighboring plane normals. Formally, we define:

Definition 1. ***Augmented Image Line***. *An augmented image line $\mathcal{L}$ is expressed as a tuple:*

$$\mathcal{L} = (\mathbf{x}_a, \mathbf{x}_b, \mathbf{u}, \mathbf{n}_l, \mathbf{n}_r) \tag{1}$$

where $\mathbf{x}_a$ and $\mathbf{x}_b$ are two end-points of the line on the virtual image plane in 3D camera coordinates, $\mathbf{u}$ is the 3D direction vector of the line, while $\mathbf{n}_l$ and $\mathbf{n}_r$ are the 3D normals of the neighboring planes on the left and right sides of the image line respectively. If the plane normals are unknown, then they are set as null vectors. Coplanarity of $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{u}$ is required, i.e. $(\mathbf{x}_a \times \mathbf{x}_b) \cdot \mathbf{u} = 0$.

As is the case with the traditional notion of the $2\frac{1}{2}$-D sketch, no estimation is made on 3D depth of the scene; positional data is limited to the image plane, while 3D information is limited only to directions and plane normals.

Prior to a formal definition, a few other basic notations and definitions are in order:

- *Vanishing points (vp).* The vertical normal to the ground plane is a 3D vector $\mathbf{v}_v$, and horizontal normals corresponding to horizontal vp's are $\mathbf{v}_i, i$=1, .., H, where H is the number of horizontal vp's.

- *Line segment sets*. Suppose different sets of edgel-based 2D line segments have been recovered in the image, where extrapolations of the line segments within each set intersect at a common vp. The set of line segments associated with $\mathbf{v}_v$ is defined as $\mathcal{S}_v = \{(\mathbf{x}_{ak}, \mathbf{x}_{bk}) : (\mathbf{x}_{ak} \times \mathbf{x}_{bk}) \cdot \mathbf{v}_v = 0, k = 1, .., K\}$ where $(\mathbf{x}_{ak}, \mathbf{x}_{bk})$ are the end points of a line segment in 3D camera coordinates. We similarly define other sets corresponding to horizontal vp's as $\mathcal{S}_1, .., \mathcal{S}_H$.

- *Sets of line segment end points*. We also define sets $\mathcal{X}_i$ to contain all end point vectors of line segments in sets $\mathcal{S}_i$, for $i = 1, .., H$ respectively.

- *Intersection points*. We define $\mathcal{I}_{ij}$ as the set of all intersection points from extensions of any pair of line segments from two different $\mathcal{S}_i$ and $\mathcal{S}_j$, *i.e.* $\mathcal{I}_{ij} = \{\mathbf{z}_{ij} : (\mathbf{x}_{ai} \times \mathbf{x}_{bi}) \cdot \mathbf{z}_{ij} = 0, (\mathbf{x}_{aj} \times \mathbf{x}_{bj}) \cdot \mathbf{z}_{ij} = 0, \forall_{(\mathbf{x}_{ai}, \mathbf{x}_{bi}) \in \mathcal{S}_i, (\mathbf{x}_{aj}, \mathbf{x}_{bj}) \in \mathcal{S}_j, i \neq j}\}$.

Next we introduce an intermediate level feature called a Vertical Corner Line Hypothesis (VCLH):

[1] Google Earth also appears to simulate some minor 3D buildings in various cities by extruding 2D polygonal outlines to building heights.

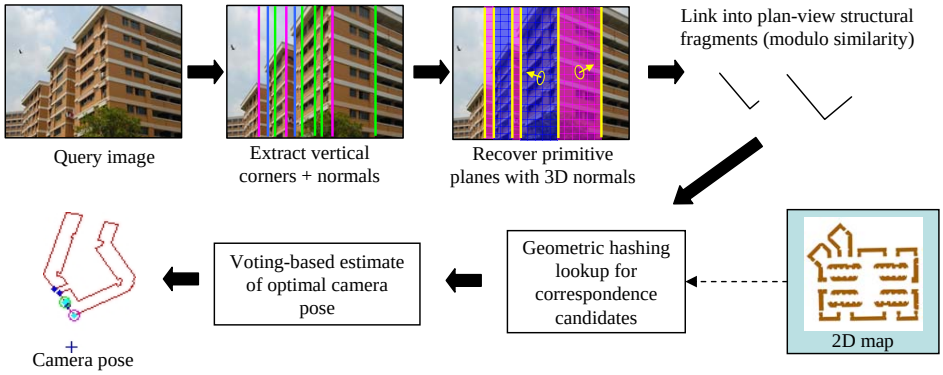


Figure 2. Illustrative block diagram of system. In practice, four query images extracted from a single omnidirectional image is used.

Definition 2. ***Vertical Corner Line Hypothesis (VCLH).*** *A Vertical Corner Line Hypothesis is an augmented image line that represents a hypothesis for a vertical corner edge of a building. Formally, it is an augmented line* $\mathcal{L}_q = (\mathbf{x}_{qa}, \mathbf{x}_{qb}, \mathbf{u}_q, \mathbf{n}_{ql}, \mathbf{n}_{qr})$ *which intersects the vertical vp,* i.e. $(\mathbf{x}_{qa} \times \mathbf{x}_{qb}) \cdot \mathbf{v}_v = 0$. *This admits a hypothesis for the 3D direction of the line* $\mathbf{u}_q$ *to be perpendicular to the ground plane,* i.e. $\mathbf{u}_q \parallel \mathbf{v}_v$. *Additionally, any one (or more) of the following conditions have to be satisfied:*

1. *Collinearity with one or more long vertically-aligned image line segments,* i.e. *there exists* $(\mathbf{x}_a, \mathbf{x}_b) \in \mathcal{S}_v$ *such that* $\|\mathbf{x}_a - \mathbf{x}_b\| \geq T$ *for some* T *and* $(\mathbf{x}_{qa} \times \mathbf{x}_{qb}) \times (\mathbf{x}_a \times \mathbf{x}_b) = \mathbf{0}$.

2. *Collinearity with end points of some line segments sharing a common horizontal vp,* i.e. *there exists* $\mathcal{Y} \subseteq \mathcal{X}_i$ *for some* i *where* $|\mathcal{Y}| \geq 2$ *and* $\forall_{1 \leq j \leq |\mathcal{Y}|, \mathbf{y}_j \in \mathcal{Y}}, (\mathbf{x}_{qa} \times \mathbf{x}_{qb}) \cdot \mathbf{y}_j = 0$.

3. *Collinearity to some intersections of extrapolated horizontal line segments belonging to 2 different vp's,* i.e. *there exists* $\mathcal{W} \subseteq \mathcal{I}_{ij}$ *for some* i *and* j *such that* $|\mathcal{W}| \geq 2$ *and* $\forall_{1 \leq j \leq |\mathcal{W}|, \mathbf{w}_j \in \mathcal{W}}, (\mathbf{x}_{qa} \times \mathbf{x}_{qb}) \cdot \mathbf{w}_j = 0$.

Condition 1 does not postulate any neighboring plane normals. Condition 2 postulates either neighboring plane normals $\mathbf{n}_{ql}$ *or* $\mathbf{n}_{qr}$ *to be* $\mathbf{v}_i \times \mathbf{v}_v$. *Condition 3 postulates* $\mathbf{n}_{ql}$ *and* $\mathbf{n}_{qr}$ *to be* $\mathbf{v}_i \times \mathbf{v}_v$ *and* $\mathbf{v}_j \times \mathbf{v}_v$, *or vice versa.*

We are particularly interested in vertical corner lines of buildings, because in the ideal case they represent a one-to-one mapping to 2D corner points on a map. Although horizontal lines have been considered in [4, 8, 5, 7], such lines have a many-to-one mapping to a 2D map. Grouping them would entail some form of facade segmentation which is quite challenging in our dataset due to incomplete views and occlusions (*e.g.* cars, trees), and is therefore beyond the scope of this paper.

Figure 3. VCLH conditions. Left: condition 1, middle: condition 2, right: condition 3. See definition 2.

3.1. Extraction of Vertical Corner Line Hypotheses

The following stages are carried out to extract VCLH's.

1. **Edge detection and local linearity filtering**. Edge detection is carried out using a sub-pixel accurate Canny implementation. The detected edgels are then filtered for local linearity, which is excellent for removing non-building edges such as trees.

2. **EM-based estimation of vanishing points**. This procedure is similar to the method in [10], where edgels are soft-labeled to vp's, while vp positions and focal length of the camera are estimated in an alternating Expectation-Maximization procedure. The difference is that our calibration model enforces only orthogonality of vertical and horizontal vp vectors, but not between horizontal vp's.

3. **Recovery of vp-labeled linear segments**. After step 2, collinear neighboring edgels sharing the same vp label are linked to form strong edge segments. This leads to the creation of the sets $\mathcal{S}_v$ and various $\mathcal{S}_i$.

4. **Image rectification**. The step warps the linear segments such that the vertical ground plane direction is aligned with the negative image y-axis, using the standard rectification homography.

5. **VCLH search**. An automated search is carried out by looking separately for the special conditions 1, 2 and 3 in definition 2. This process can be implemented by: (1) recovering the end points and intersection points, *i.e*. various sets $\mathcal{X}_i$ and $\mathcal{I}_{ij}$, (2) considering a vertical line (post-rectification) to each of these points, and (3) computing consensus from other points.

Figure 4 shows intermediate results.

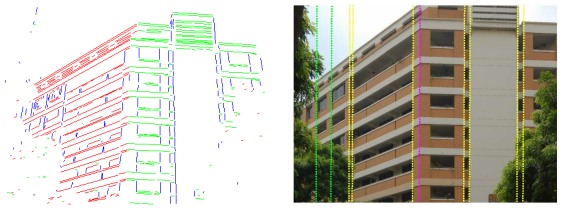


Figure 4. Left: color-coded line segments based on vp label. Right: sample VCLH result.

3.2. 2D-to-1D Perspective Projection

Our framework assumes that a 2D map consisting of various building outlines in plan view is available. Given the VCLH's extracted as described in the previous section, each VCLH is therefore a hypothesis for correspondence to a 2D corner of a building outline. In the rectified reference frame in which all VCLH (and therefore also the ground plane normal) are parallel to the image y-axis, the problem reduces to a *1D perspective projection*.

By projecting all entities to the ground plane, the rectified image plane is reduced to a single *image line*. All VCLH's extracted are simply points on this 1D image line. The main challenge here is finding correspondences between the VCLH points on the image line to the correct 2D corners. The 1D camera has only 2 intrinsic and 3 extrinsic parameters, with the intrinsic parameters being the focal length and principal abscissa which have been determined. The extrinsic parameters to be estimated are:

$$\mathcal{C}_{\text{ext}} = (x_c, y_c, \alpha_c) \tag{2}$$

where (x_c, y_c) is the camera position on the ground plane, and α_c is the camera orientation angle.

The reduction to a 2D-to-1D perspective projection also has implications for the relation to full extrinsic camera calibration. Estimating $\mathcal{C}_{\text{ext}}$ fixes 3 parameters, while rectification fixes another 2 parameters. The only inestimable parameter is elevation from ground plane. This is expected based on the information available.

Under the reduction to 1D projection, each neighboring plane normal of a VCLH can be simplified to a direction angle. Hence the reduced VCLH can now be compactly expressed in terms of 3 angles:

$$\mathcal{L}'_q = (\phi_q, \theta_{ql}, \theta_{qr}), \quad \text{where } \phi_q = \tan^{-1}\frac{-x_q}{f} \tag{3}$$

in which x_q is the abscissa on the image line, while θ_{ql} and θ_{qr} are the left and right neighboring plane angles (if available). All the angles are measured w.r.t. the optical axis. This representation is focal-length normalized and can be used to compare 1D scenes with different focal lengths.

4. Bottom-Up Reasoning into Higher-Order Entities

Although the neighboring planar angles of the VCLH's add significantly more discriminative power compared to using 1D positional information only, it remains insufficient for identifying the correct match exactly, especially as there is a significant number of false or missing VCLH's. Instead, there is a further need to develop higher-order features from lower level cues, partially analogous to the organizational process of moving from a $2\frac{1}{2}$-D sketch to a full 3D model representation by including non-local cues, as proposed by Marr. We propose the following 2-step mechanism:

1. form **elemental planes** from VCLH pairs;

2. group such plane hypotheses into **structural fragments** by exploiting in-plane invariant depth-ratios.

4.1. Forming Elemental Planes

Here we seek to recover pairs of VCLH's that: (i) share a common neighboring plane normal, and also (ii) lie in a same plane perpendicular to the common plane normal.

VCLH's supported by condition 2 (end points) or condition 3 (intersection points) of definition 2 have their horizontal supporting line segments further investigated. By considering pairs of such VCLH's, if a significant subset of horizontal line segments supporting one VCLH is also colinear with horizontal line segments supporting the second VCLH, the two VCLH's are postulated to share the same plane. We call these the *elemental planes*. Elemental planes in the reduced 2D-to-1D projection framework can be considered as line fragments between two VCLH points.

4.2. Invariant In-Plane Depth Ratios

A property that can be used to further reason about elemental planes is the invariant in-plane depth ratio. This ratio relates 2 VCLH's that are part of an elemental plane. Ratios of distance measurements in the image plane have been analyzed [2] for 2D images, but we provide an analysis for depth ratios from 1D image plane measurements.

Lemma 1. ***Invariant Depth Ratio***. *Suppose two VCLH's are denoted by* $\mathcal{L}'_a = (\phi_a, \theta_{al}, \theta_{ar})$ *and* $\mathcal{L}'_b = (\phi_b, \theta_{bl}, \theta_{br})$, *and they share a common* $\bar{\theta} = \theta_{al} = \theta_{bl}$. *If they are also known to be lying in the same 2D line with normal angle* $\bar{\theta}$, *then the depths of the VCLH's, given by* z_a *and* z_b, *have*

a ratio invariant to the actual placement of the line. This ratio is given by

$$\frac{z_a}{z_b} = \frac{\tan(\frac{\pi}{2} - \bar{\theta} + \phi_a)\cos\bar{\theta} + sin\bar{\theta}}{\tan(\frac{\pi}{2} - \bar{\theta} + \phi_b)\cos\bar{\theta} + sin\bar{\theta}} \quad (4)$$

Proof. See figure 5 for illustration and notation.

$$z' = L\cos\bar{\theta}, \quad z'' = R\sin\bar{\theta}$$
$$L = R\tan(\tfrac{\pi}{2} - \bar{\theta} + \phi)$$

Then

$$\begin{aligned} z &= z' + z'' \\ &= R\left(\tan(\frac{\pi}{2} - \bar{\theta} + \phi_a)\cos\bar{\theta} + sin\bar{\theta}\right) \end{aligned} \quad (5)$$

Since R is unknown but constant for points on the same line, taking ratios of (5) for different points leads to (4); thus lemma 1 holds. □

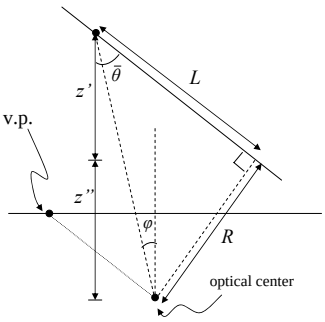


Figure 5. Diagram for depth ratio derivation

4.3. Linking into Structural Fragments

Depth ratios are not immediately useful when considering two or more separate elemental planes, as these hypotheses can have different relative-depth ambiguities independent of each other.

Instead, we further group elemental planes into *structural fragments* by linking each adjacent pair of elemental planes that share a common VCLH, thus forming a chain. Note that such adjacent pairs of elemental planes do not share the same plane; rather it is intuitive to think of a 3D structural fragment as a vertical facade of linked planar faces in 3D space, and in the reduced 2D-to-1D projection framework as a piecewise linear contour.

While depth ratios are only invariant for VCLH's on a plane with a known normal, it is straightforward to stack ratios in a structural fragment since the topological connectivity ensures equality of depths at VCLH "joints" in the chain. Suppose in the reduced 2D-to-1D projection framework we denote the depth of the first VCLH in a fragment as z_0, and subsequent depths as $z_k, k = i, \ldots, N$, for a structural fragment consisting of N linked elemental planes. Then

$$z_k = z_0 \prod_{i=1}^{k} \frac{\tan(\frac{\pi}{2} - \bar{\theta}_i + \phi_i)\cos\bar{\theta}_i + \sin\bar{\theta}_i}{\tan(\frac{\pi}{2} - \bar{\theta}_i + \phi_{i\text{-}1})\cos\bar{\theta}_i + \sin\bar{\theta}_i} \quad (6)$$

where $\bar{\theta}_i$ are the common plane directions along each link of the fragment, while ϕ_i are the projected angular displacement of the VCLH's in the image as illustrated in figure 5.

The scale-depth ambiguity for all VCLH's in a structural fragment is reduced to a common z_0 scaling factor. Combined with the known planar angles, a structural fragment obtained from an image is directly related to a building portion of the 2D map in plan view via a similarity transform (see figure 6). Note that scale-depth ambiguities remain independent across different structural fragments.

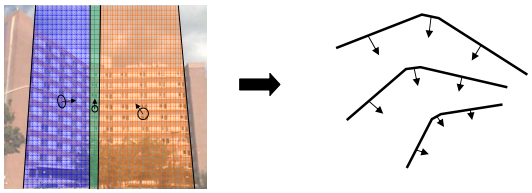


Figure 6. Forming structural fragments. In the left image, 3 elemental planes are recovered and grouped into a 3-link structural fragment. The boundaries of the planes are not assumed to be known except at the VCLH joints. This allows the plan-view structure of the fragment to be estimated modulo a similarity transformation (as illustrated by the 3 contours on the right).

5. Matching Structural Fragments to 2D Map

Once the structural fragments are extracted from an input image, the challenge is to find correspondences between these fragments and building contour segments in the 2D plan view map. This is done in the reduced 2D-to-1D projection framework.

While the structural fragments are related to their corresponding map segments by a 4-dof similarity transform, note that the camera pose in this framework only has 3 dof's. This is because of the additional scale-depth ambiguity when constructing structural fragments from image features. Once correspondence between these fragments and the map contour segments, the scale as well as the camera pose parameters are established. Consequently, in a scenario where there are multiple structural fragments in a single view, the further challenge is to compute an optimal 3-dof camera pose that simultaneously fits all these structural fragments to plan view contour segments, while allowing for an independent scale parameter for each fragment.

The matching process is divided into two stages. The first stage obtains correspondence candidates for each structural element through a rapid geometric hashing lookup,

while the second stage simultaneously determines the best combination of correspondence candidates for all extracted structural fragments as well as an optimal camera pose.

5.1. Geometric Hashing

We employ geometric hashing [14] in the classical manner to rapidly test all possible correspondences between a structural fragment and different candidate contour segments from the 2D plan-view map.

In the offline preprocessing phase, the contour of each building is processed. Similarity bases are established from all adjacent pairs of building corners, while remaining building corners are used as support in the canonical reference frame. In the online lookup phase, a pair of adjacent VCLH's from each structural fragment is used to set the canonical reference frame and support computed from the remaining VCLH's and the geometric hashed database.

Correspondence candidates, comprising 2D building contour segments with corner points that correspond to each VCLH in a structural fragment, are subsequently passed to the following camera pose estimation stage.

5.2. Voting-based Camera Pose Estimation

After the previous geometric hashing lookup stage, correspondence candidates have been established independently for each separate structural fragment in a single view. However in order to compute the optimal camera pose, the combination of correspondence candidates with the best consensus across the different structural fragments must be determined. A voting-based camera pose estimation method is employed as described in algorithm 1.

Algorithm 1 Voting-based Camera Pose Estimation.

Initialize a 3D accumulator array Acc in the camera pose space
Obtain list SF of structural fragments from the input image
for each structural fragment in SF **do**
  Obtain a list $Corr$ of correspondence candidate contour segments from geometric hashing lookup
  for each contour segment in $Corr$ **do**
    Compute the least squares similarity transform mapping the structural fragment to the contour segment with error ϵ^2.
    Let length of segment be l and compute score $s = l/(1+\epsilon^2)$.
    Extract the 3 dof camera parameters (x_c, y_c, α_c) and add s into the corresponding bin in Acc.
  end for
end for
Find largest scoring bin in Acc to get optimal $(\hat{x_c}, \hat{y_c}, \hat{\alpha_c})$.

A weighted linear least squares solution is used in the algorithm. There are frequent situations when an end VCLH is not a true building corner, but a termination due to occlusion by other buildings. Hence in the 1D projection framework, the end VCLH points are in most situations only desired to be collinear to the plane-line rather than be coincidental with the map segment end-points. This is easily established through weighted linear least squares.

6. Experiments

For the purpose of creating a widely accessible public dataset that can be verifiable, we used Google Maps [3] and Google Earth. We selected an area in New York City of approximately 440m×440m size in the vicinity of $40°48'50"$N, $73°54'7"$W, and annotated 2D outlines of 111 buildings, with a total of 885 corners. The outlines are created by tracing the ground cross-sections of the synthetic building models embedded in Google Maps which are by default shown in oblique projection (ground plane angles are preserved) when in "Map" view. This forms the 2D map dataset. Additionally, we collected images from Google Earth in StreetView, comprising uniformly sampled locations (on the roads) and 4 orientations per location. The orientations were selected to be tilted upward and $45°$ in yaw angle from the road. This is so as to capture sufficient building details, and also to minimize ground level occlusions such as vehicles and trees. In total, there are 53 unique locations used as ground truth with 212 images in total.

In all our experiments, we tested on the 53 unique locations and *combined results from four views per location each as a single test*. The images for each location are assumed to share a common optical center, and represents our abstraction of an omnidirectional image. The combined four views are roughly equivalent to a $360°$ field of view thereby enabling a significantly more unique solution with many more structural fragments than in one view.

This is an extremely challenging dataset as the buildings are nondescript with very similar appearances — see figure 8, the images are from non-overlapping viewpoints, but they all look alike — and the image quality is quite poor.

6.1. Uniqueness Analysis

To informally evaluate the effectiveness of solely exploiting geometric relationships for localization without resorting to appearance data, we can inspect solution uniqueness by visualizing scores in a 2D camera-position-only version of the accumulator array. An example result is shown in figure 7. Given the very few bins with high scores, it is evident that solutions obtained from these geometric features are highly unique, although estimating such features reliably is a different matter.

6.2. Extraction of VCLH

The results of vp estimation are generally accurate. Examples of the VCLHs obtained from the sample images are shown in figure 8. The results are color-coded with red indicating no associated normals (triggered by condition 1 of definition 2), green indicating associated vp is to the right, and blue indicating associated vp is to the left. Additionally,

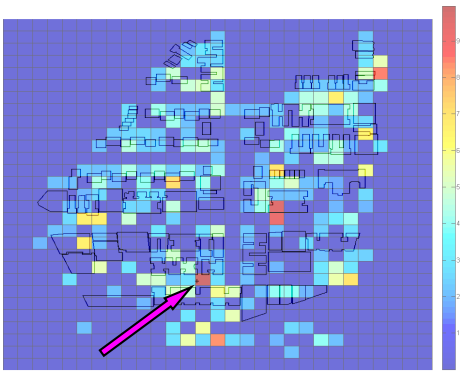


Figure 7. Example distribution of camera pose scores in a 2D position-only version of the accumulator array, with 2D map overlay. Red indicates high scores and blue low scores. Arrow shows ground truth pose (coincides with top score). Very few red spots show that use of structural fragments leads to unique solutions.

a small horizontal line next to the VCLH indicates which side of the line (and hence which plane) the normal is associated with. VCLHs with two normals are shown as neighboring VCLHs of blue and green codes.

Figure 8. Rectified images with overlay VCLHs.

The results indicate that almost all the significant building corners were correctly detected as a VCLH position. However, a number of correct VCLHs detected did not have any plane normals detected, or the normals were incorrectly computed due to the ambiguity between a concave corner and a partial occlusion of a wall by a different building corner. Conversely, there are many false positives for VCLH, *e.g*. VCLH that are associated with windows. This illustrates a highly asymmetric problem. One key observation that we made based on the experiments, is that the most reliable building corners are convex corners for which both neighboring planes were visible. This corresponds with VCLHs that are associated with two neighboring plane normals. Whenever such a VCLH is detected, it is most certainly a true positive detection.

6.3. Geometric Hashing for Matching Structural Fragments

The underlying correspondence problem here is matching the VCLH's in a structural fragment to any of the 885 building corners in the 2D map. We investigate two characteristics of this: (1) time savings from geometric hashing lookup compared to a brute force exhaustive search, and (2) the reduction in the number of correspondence candidates for processing in the subsequent pose estimation stage.

Please note that because structural fragments imposes a topological ordering for acceptable pointwise correspondences between VCLH and map building corners, the cost of brute force exhaustive search is only $O(n)$ where n is the number of building corners in the map – not particularly intensive. We exclude the number of structural fragment links for computational complexity because this is typically between 3-5 only. In practice, for our map of 885 building corners, running only stage 2 without geometric hashing lookup, *i.e*. a full exhaustive search, takes about 8s to compute for a single input image on Matlab. Nevertheless, geometric hashing lookup runs in time constant to the number of building corners, and takes less than 1s on Matlab.

The geometric hashing bins are set to be relatively large in order to minimize the number of missing correspondences (*i.e*. false negatives). Additionally since there is only one correct correspondence candidate, the shortlist of correspondence candidates is naturally dominated by false positives, but this is satisfactory as these will be culled in the pose estimation stage. The results of the number of correspondence candidates on average that are passed to the second stage is shown in table 1. It is clear that 4-linked and 5-linked fragments are more discriminative than 3-linked fragments. The results between 4-linked and 5-linked fragments are less easy to interpret, as we would expect 5-linked fragments to be more discriminative. It is likely that this is the result of significant noise levels in the extraction of VCLH's (and consequently structural fragments).

# of links per structural fragment	3	4	5
# of correspondence candidates	212.98	91.06	113.33
Selectivity (out of 885)	24.1%	10.3%	12.8%

Table 1. Number of correspondence candidates from geometric hashing lookup.

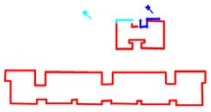
6.4. Voting-based Camera Pose Estimation

Finally, we tested our voting-based camera pose estimation method. For the 3D accumulator array for camera pose space, we used bins that are approximately 16m$\times$16m spatially, and 90° wide in rotation angle. This array covers the entire map area, resulting in a total of 3600 bins.

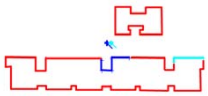
We show an illustrative result in figure 9 whereby the poses computed from two correspondence candidates are not in consensus and thereby rejected, while another two candidates have poses that are in consensus and accepted.

We ran two experiments, one using correspondence candidates from the geometric hashing stage which is faster at

Two structural fragments recovered from an input image (in pink and cyan)



Correspondences with non-consensus on camera pose



Correspondences with consensus on camera pose

Figure 9. Pose space estimation from multiple structural fragment correspondence candidates. On the top-right, wrong correspondences leading to two disagreeing camera pose estimates (light blue and dark blue spots); bottom-right showing correct correspondences with consensus on camera pose estimates.

4s but less accurate, and the other without geometric hashing which runs in 8s. The quantitative results are shown in table 2. The obtained camera poses are ranked in terms of final consensus scores in the corresponding accumulator bins. We show the results where the correct solution is located in various rank categories. Without the use of geometric hashing (which is the better result, indicating that the speed of geometric hashing lookup comes with an accuracy penalty), we obtained 2 rank-1 correct poses from 53 queries, while half of correct poses are within rank-30.

Note that we are trying to determine the best bin from 3600 bins. This means that a correct solution in the top-30 represents a strong selectivity of 0.83%, while obtaining a top-ranked solution is a very high selectivity of 0.028%. While the selectivity of our method is already very strong, the main obstacle to better results is that the framework was simply unable to detect all the true vertical corner edges and/or neighboring plane normals of the buildings in the query images. The main cause is the poor, blurred quality of the images. Once these corner edges were missed, the extensive nature of the 2D map (111 buildings, 885 corners) and the similarity in building design meant that it was easy for an alternative incorrect match to be made. In any case, even humans will have a very hard time with this dataset.

Without geometric hashing:

Rank bands	Rank 1	Rank 2-5	Rank 6-10	Rank 11-15	Rank 16-20	Rank 21-25	Rank 26-30
# correct poses	2	9	3	4	4	2	3
Cumulative %	3.77%	20.75%	26.42%	33.96%	41.51%	45.28%	50.94%

With geometric hashing:

Rank bands	Rank 1	Rank 2-5	Rank 6-10	Rank 11-15	Rank 16-20	Rank 21-25	Rank 26-30
# correct poses	0	3	2	3	6	5	3
Cumulative %	0%	5.66%	9.43%	15.09%	26.42%	35.85%	41.51%

Table 2. Voting-based camera pose estimation. A correct solution in the top-30 represents a high selectivity of 0.83%, while a top-ranked solution is a very high selectivity of 0.028%. Half of correct solutions lie within rank-30 if geometric hashing is not used.

7. Conclusions and Future Work

We have clearly demonstrated a method to very substantially narrow down possible solutions for camera poses, given a single omnidirectional image and a large 2D building outline map with no appearance information. Higher-order image features, from vertical building corner hypotheses to elemental planes to structural fragments, were constructed in a bottom-up process. These features capture an increasing amount of partial 3D information, in a manner analogous to Marr's $2\frac{1}{2}$-D sketch.

There is substantial scope for improving the results. One possibility is to investigate a more robust form of plane analysis to reduce the ambiguity due to missing vertical hypotheses. Another possibility would be to consider how the skyline of a building may be identified to provide partial matches to building outlines on a map.

Acknowledgements

This research is supported by the Project Tacrea grant from the Defence Science & Technology Agency (DSTA), Singapore, and carried out at the Centre for Multimedia and Networking (CeMNet) in NTU. We would also like to thank contributions / advice from Rudianto Sugiyarto, Zahoor Zafrulla and Teck-Khim Ng.

References

[1] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *CVPR*, 2005.

[2] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 40(2):123–148, 2000.

[3] Google Maps. http://maps.google.com/.

[4] S. Lee, S. Jung, and R. Nevatia. Automatic pose estimation of complex 3d building models. In *WACV*, 2002.

[5] K. Leung, C. Clark, and J. Huisson. Localization in urban environments by matching ground level video images with an aerial image. In *ICRA*, 2008.

[6] D. Marr. *Vision*. Freeman, San Francisco, 1982.

[7] B. Micušík, H. Wildenauer, and J. Košecká. Detection and matching of rectilinear structures. In *CVPR*, 2008.

[8] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.

[9] A. Saxena, S. Chung, and A. Ng. 3-D depth reconstruction from a single still image. *IJCV*, 76(1):53–69, 2008.

[10] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. In *3DPVT*, 2006.

[11] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *IJCV*, 40(3):199–233, 2000.

[12] R. Sim and G. Dudek. Learning generative models of scene features. *IJCV*, 60(1):45–61, 2004.

[13] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *ECCV*, 2002.

[14] H. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE CSE*, 4(4):10–21, 1997.

[15] T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. In *CVPR*, 2004.

[16] W. Zhang and J. Košecká. Image based localization in urban environments. In *3DPVT*, 2006.