Urban Position Estimation from One Dimensional Visual Cues

Derek Johns and Gregory Dudek
Center for Intelligent Machines, McGill University
Montreal, Quebec H3A 2A7, CA
{djohns,dudek}@cim.mcgill.ca

Abstract

We consider the problem of vision-based position estimation in urban environments. In particular, we are interested in position estimation from visual cues, but using only limited computational resources. Our particular solution to this problem is based on representing the variability of the "horizon" of the cityscape when seen from within the city; that is, the outlines of the rooftops of adjacent buildings. By encoding the image using only such a one-dimensional contour, we obtain an image encoding that is exceedingly compact. This, in turn, allows us to both efficiently transmit this representation to a remote "recognition engine" as well as allowing for an efficient storage and matching process. We outline our approach and representation, and provide experimental data supporting its feasibility.
Keywords: Vision-based localization

1 Introduction

In this paper we consider how to estimate the position of an observer in an urban environment. We are particularly interested a solution based on computer vision since this type of sensing has become commonplace and seems to provide sufficient information to solve the problem. We are further interested in position estimation mechanisms that can be employed in a highly efficient platform-independent manner, such as either on a cellular telephone or on a small autonomous vehicle. The particular problem of mobile telephone localization motivates our work, but we are also interested in related applications. To be feasible, such a solution needs to be highly efficient in terms of both computational cost as well as communications bandwidth.

To help motivate our approach, we note that two alternative methods for outdoor position estimation are the use of radio signals from the global positioning system (GPS satellites) or via triangulation from cellular telephone towers. Each of these approaches has severe limitations in terms of both accuracy and practicality. GPS signals are susceptible to various forms of interference and can be quite unreliable in urban settings. In addition, even though many mobile telephones are becoming GPS enabled, the signals themselves are not regularly available for consumer applications. Cell tower triangulation is similarly problematic with respect to accuracy as well as the need for access to geometric data that is often difficult to access. Finally, in situations where either of these radio-frequency technologies are appropriate, supplementing them with a complementary information source can increase robustness and accuracy.

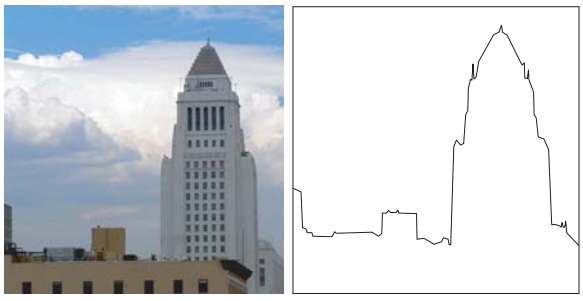

Figure 1: A typical cityscape image on the top, and its corresponding skyline contour.

1.1 Approach

Position estimation is sometimes subdivided into two sub-problems, local pose refinement (given a good initial guess) and global localization. The former problem is often solved using methods such as Kalman filtering. The latter problem refers to selecting a probable position estimate from a large range of candidate locations that may be widely separated in space; it is this global problem we are interested in. Note that this reduces position estimation to a specialized scene recognition problem. In particu-

lar, in urban environments position estimation can be achieved by determining which building(s) we are standing in front of. Thus, our challenge is to recognize the specific building, or set of buildings, located in front of the observer. For this paper, we assume that we have a cooperative observer who will acquire an approximately fronto-parallel image of a set of nearby buildings, and that the sky above these buildings will be, by and large, visible. While several authors have considered the problem of directly recognizing the facade of a building (see below), our approach is to recognize merely the horizon line separating the top of the building from the sky (see Figure 1). This minimalist approach can suffer from ambiguity since the horizon line above a building contains less information than the facade, and hence may be less likely to be unique, but it has the compensatory advantages of being extremely efficient as well as being highly robust to lighting variations. In fact, even the ambiguity problem (the lack of uniqueness) is not as bad as it might appear at first glance since we are able to exploit the positional constraints offered by an *ensemble* of buildings at any one location.

Pragmatically then, our goal is to match a query view of the nearby skyline to a database of similar images taken from previously known positions. Each of these images, including the query view, will be represented by a simple one dimensional horizon line extracted from the image. This is accomplished by a combination of adaptive thresholding, relaxation labeling, and segmentation. Our approach assumes that the skyline contour is a relatively easy feature to extract, as compared to other image features such as windows or doorways. A compact representation of the skyline contour is achieved by storing a list of linked edges that approximates the signal. Typically the list contains some 40 line segments for a given image. This edge list is then matched to a library of known buildings (and locations). The best matching building provides a position estimate.

1.2 Paper Outline

This following sections outline the framework for our pose estimation system, as well as the various issues encountered at each stage. We begin with a discussion of related work in Section 1.3. Section 2 discusses the extraction of a skyline contour from an input image. The goal of this stage is to be able to accurately find the skyline under different lighting and weather conditions, and to generate a compact representation of the contour to be used in the latter stages of the system.

Section 3 outlines the contour matching process. A discussion of the information contained in skyline contours is provided, leading to an outline of the algorithm that exploits this information to provide for accurate matching across views. An efficient dynamic programming approach is used to perform this match. Section 4 provides a discussion for how a contour is selected as a"match" from our database. Section 5 provides preliminary experimental results. Finally, concluding remarks and a discussion of future work follows in Section 6.

1.3 Background and Related Work

Pose estimation in computer vision has a long history. Classic approaches to pose estimation dealt with surface reconstruction and thus, implicitly with the recovery of observer pose. Such work goes back to early methods from shape from motion [11], and shape from stereo [8]. Several systems also considered the use of stereo data in robot navigation [5]. Such early work, including developments on vision for mobile robotics, did not address what is commonly referred to as the localization problem, whereby the current location of an observer must be recognized as well as numerically optimized with respect to a stored map.

Some of the earliest work on robot localization was developed by Cox [1] and by Leonard [9]. Position estimation using recognition-like methods applied to range data was considered by Crowley and Pourraz who used a global method based on a principal components representation [12]. Nayar also considered pose estimation using PCA over images, but the technique was only well-suited to scenes with essentially no ambiguous locations and controlled lighting (for example a robot arm moving over a small workspace). Jugessur and Dudek considered a voting scheme for robot position estimation from images based on machine learning and substantial computational expense. Subsequent work on voting-based scene recognition and localization included that of Sim and Dudek [15], Se *et al* [14] and others. In general, these methods are based on image filtering and have multiple drawbacks such as sensitivity to weather, visibility (e.g haze), illumination variations, viewpoint, or are computationally costly.

Recent research in vision-based position estimation in urban environments focuses on the recognition and matching of building facades. In work by Robertson and Cipolla [13], viewpoint invariant matching of buildings is explored using a wide-baseline matching technique. After a candidate building has been selected from a database, the relative pose of the camera is determined by computing the transformation required to match a rectified

view of the building facade. The approach taken by Zhang and Košeckà [17] is similar but uses powerful supplementary information provided by local color histograms to identify candidate building matches.

Horizon-based localization methods have mainly been studied for use away from urban centers. In [16] localization is performed using a topological map of the surrounding terrain. Possible horizon contours at different locations are precomputed off-line using the topological description of the surrounding landscape. The system performs localization by matching an input panoramic horizon contour to these precomputed signals.

Our approach focuses on compactness and efficiency by avoiding computationally expensive image processing methods and using compact, one-dimensional skyline contours in our database.

2 Segmentation

We begin by describing our skyline extraction algorithm. This stage of the process takes as input a digital image of an urban scene. The goal is to extract the one-dimensional contour from the image that is the boundary between the sky and ground objects.

Ideally, the target contour would be a piecewise linear function with a start and end position corresponding to where the skyline enters and exits the camera's field of view. There are several situations where the true signal might not be piecewise linear, such as buildings with curved structure or if trees or mountains were present in the image. However, these exceptions occur infrequently and can be adequately approximated.

In terms of image processing problems the skyline contour in an image is a relatively easy feature to automatically recognize. In essence, the problem is to find the sky region in the input image. This region is generally much brighter than the objects on the ground (during the day), and contains low frequency changes in intensity relative to ground objects. Variations in illumination, an aspect of image processing that remains a common obstacle, is much less of an issue in this problem domain.

If assumptions are made about the orientation of the camera when the image was obtained, the problem further simplifies to growing and linking the region at the uppermost of the image. Furthermore the existence of sky in the input image could be guaranteed by the user, avoiding the sometimes difficult problem of testing for existence.

Taking advantage of these considerations a robust skyline extraction algorithm is achieved in two stages. The first stage begins by using edge-detection to identify candidate skyline pixels. A Canny [2] edge detector is used on the grayscale image to identify dominant line structure in the image. The output is then analyzed per column, selecting the uppermost pixel with the highest gradient. The output from this stage is a noisy representation of the skyline contour (see Figure 2). This approach, while simple, provides an accurate approximation to the target skyline contour in most cases. The process could be improved through use of relaxation-labeling, allowing pixels to reinforce their neighbors based on their signal strength and relative placement.

The second stage approximates this noisy signal and simplifies its representation into an ordered sequence of linked edges using a recursive geometric fit algorithm [6]. Since we assume the target contour will be piecewise linear and continuous this is a suitable representation. The fit algorithm begins with a single edge whose endpoints are determined by the pixels closest to the left and right borders of the image. This edge serves as the first approximation to the desired contour. For each candidate skyline pixel from stage one the error is calculated as the euclidian distance to this approximating edge. A new vertex is created at the pixel location with the greatest error, dividing the initial edge into two linked edges. The process then repeats using the new approximating contour. The algorithm stops when the error drops below a threshold (see Figure 2).

The resulting contour usually consists of approximately 10-50 edges depending on the complexity of the input skyline. As can readily be seen this is an extremely efficient and compact representation compared to the number of pixels in the input images.

3 Line Matching

This section deals with the problem of matching components of two different skyline contours. The goal of this stage is to identify how well a given contour matches another. The skyline contours are described by a continuous piece-wise linear function, or chain-code. We encode a line segment using its ordinal position along the contour, n_i, its length, and its angle to the image horizontal:

$$s_i = (n_i, l_i, \theta_i). \tag{1}$$

The matching process will consider segment similarity in the length and angle dimension of this encoding, and preserve the relative ordering of the line segments. A dynamic programming approach is well suited for this type of contour matching [4, 10], as it implicitly encodes the segment's ordinal position and

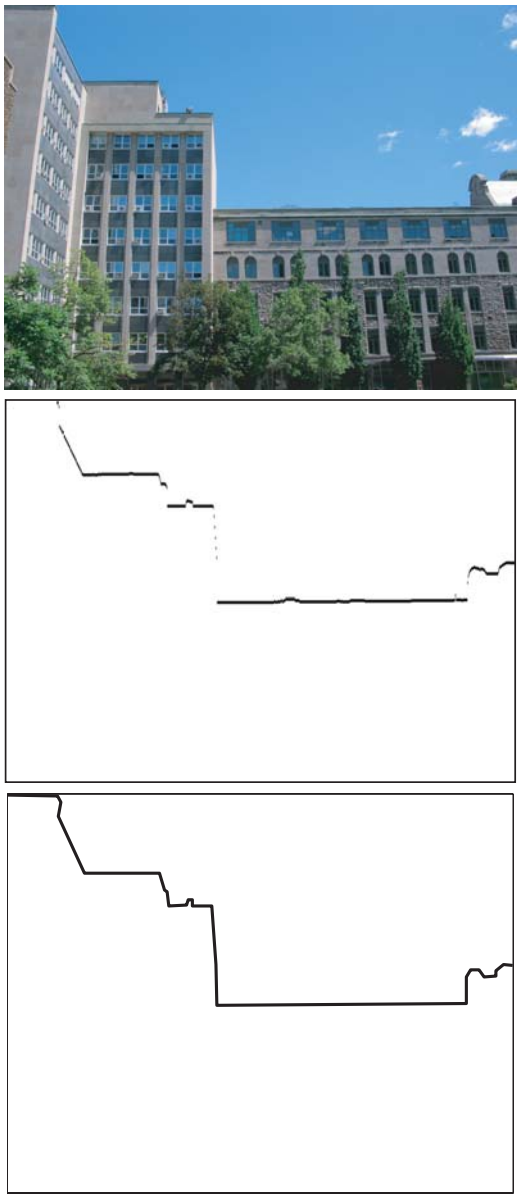

Figure 2: The segmentation process. At the top is the original image, the middle shows the noisy output from the edge detection and region-growing phase, and the bottom image displays the final approximating contour.

considers matches for segments based on a suitable metric for length and angle differences.

The similarity metrics for length and angle are described by Equations 2 and 3 respectively:

$$L(S_{a,i}, S_{b,j}) = \left[1 - \frac{||l_{a,i}| - |l_{b,j}||}{|l_{a,i}| + |l_{b,j}|}\right]^{-1} \quad (2)$$

$$A(S_{a,i}, S_{b,j}) = |cos^{\beta}(\Delta\theta_{a,b,i,j})|^{-1} \quad (3)$$

where $S_{m,k}$ refers to kth line segment from contour m, $\Delta\theta_{a,b,i,j} = \theta_{b,j} - \theta_{a,i}$ and the constant β tunes the penalty contribution of the angle difference. The range of functions L and A is therefore $[1, \infty)$, 1 being the result of an exact match. The final cost for matching these segment pairs, γ, is the product of the two metrics:

$$\gamma(S_{a,i}, S_{b,j}) = L(S_{a,i}, S_{b,j})A(S_{a,i}, S_{b,j}). \quad (4)$$

In determining the cost of omitting a segment in the matching process we can observe certain heuristics that apply to the skyline signals. A typical image can be seen in Figure 2. In this example, long segments in the skyline contour tend to represent dominant structure, such as large and spatially local buildings. Shorter segments tend to appear from noise and building detail. Longer segments are therefore a better characterization of a skyline as they embody more relevant information. By this reasoning omitting a long segment should incur a heavier cost penalty. We embody these principles in a linear function of length:

$$\gamma(S_a, nil) = 1 + \alpha l_a \quad (5)$$

where α is a constant.

Given two sequences of line segments, S_a and S_b of length n and m respectively, the algorithm proceeds by constructing a cost table of size n-by-m. A match is reflected by a path through this cost array. At entry $C(i, j)$ is the cost of matching the first i segments from contour S_a with the first j segments of contour S_b. Each entry in the cost array is generated using the following rule:

$$C(i,j) = min \left\{ \begin{array}{l} C(i-1, j-1) + \gamma(S_{a,i}, S_{b,j}) \\ C(i-1, j) + \gamma(S_{a,i}, nil) \\ C(i, j-1) + \gamma(nil, S_{b,j}) \end{array} \right\}. \quad (6)$$

The final cost for matching the two contours is stored at position $C(n, m)$ in the array. The minimum cost path through the table yields the optimal segment matches based on the current cost function. This algorithm has a complexity of $O(nm)$, making it efficient given that the input contours contain approximately 50 segments each.

There remains the problem of selecting constants β and α in Functions 3 and 5. Increasing β penalizes larger differences in angle between input line segments, and increasing α reduces the number of

skipped line segments during the matching process. Choosing values for these variables is difficult to do as they interact in a complicated manner, and in our experiments values that provided a balance between omitting and matching contour segments were chosen. Machine learning could be used to derive better values for these constants, and we are currently exploring the possibilities of this approach.

3.1 Discussion of Matching Process

There are a number of factors that make the matching problem relatively difficult. One issue is that there is an amount of ambiguity as to what the vertices of the contour represent, as they can result from several different physical causes. For example, consider the scenarios that could lead to a vertex in our skyline contour. It could be the result of an intersection of two different structures, a physical change on a single structure, or the occlusion of a distant building by one that is in the foreground. A vertex could also be the result of noise or due to small building details. These contributions are usually impossible to discriminate from the desired building structure.

The exact placement and orientation of the line segments can also change by large amounts if the camera is rotated or panned. Additionally, since these signals are acquired from a standard camera image there is the additional problem of perspective projection.

4 Global Localization

To achieve global localization, an input skyline contour needs to be matched to a skyline contour in a database. Using the matching process described above, it is likely that every candidate contour from the database contains a number of matching line segments. However many of these segment matches are false positives, which is to be expected given the simplicity of the input signals. In addition, the total number of line segments tends to increase the cost of a match regardless of how similar the signals are, due to the cost penalty incurred at each step through the cost table.

What we do expect is that the total cost of matching similar skylines will be less than those with large discrepancies. In addition, the number of matched segments relative to the total number of input segments should increase as the contours become more similar. The selection of a matching database contour is therefore based on:

$$Q(C, d) = \frac{C}{d} \tag{7}$$

where C is the final cost from Equation 6, and d is the number of matching line segments. The database contour with the lowest Q value is chosen as a match for the current query.

5 Evaluation

To evaluate our global localization system, the above framework was implemented in Matlab. A database of skyline contours was then generated using 30 images of McGill campus and downtown Montreal. The images were chosen to contain buildings of varying degrees of complexity, and at different distances from the camera.

Using the approach outlined above, we attempted to select the most similar skyline from our database for 17 query views. Each query took under a second on a 1.5 GHz desktop PC. Correct matches were determined through manual labeling of line segments. Out of the 17 query views, 15 were determined to be correct with 2 incorrect matches giving a success rate of 88%. Examples of query images and the match selected from the database can be seen in Figure 3.

Representative results of a query can be seen in Figure 4. Here the result of Equation 7 is plotted for each skyline contour in our database. The results are sorted to display the cost values in increasing order. The database contours marked with a circle specify which skylines are considered to match the current query, determined by inspection. It can be seen that the lowest two cost values correspond to matching skylines in our database, leading to a successful selection by our algorithm.

Based on the highly simplified representation of the scene, the matching process can be accomplished very rapidly and the data can efficiently be transmitted to a remote server for lookup, if desired. Given an actual horizon length of 40 segments, a compression factor of roughly 24390 (i.e. 0.0041 per cent) relative to the original image size is achieved, without bothering to resort to information theoretic encoding.

6 Conclusions and Future Work

In our work, it appears that qualitative localization in an urban environment can be accomplished using only skyline information. This has great advantages in terms of efficiency and robustness of the representation, but one might imagine that there may be difficulties in terms of accuracy and recall. While our present experimental data sets are insufficient to comment on these issues, it is clear that at some point (even if entire facades are used for recognition),

Figure 3: Examples of matches. The query images are displayed on the left, and the image selected from the database as a match is displayed to the right of each. The top two matches are correct, the bottom match is an example of where the algorithm fails.

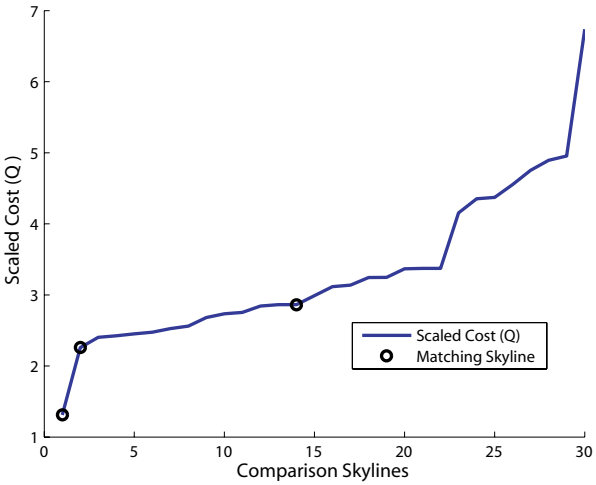

Figure 4: For a given query, the match cost divided by the number of matching line segments is plotted for each skyline contour in our database. The circles indicate database contours that match the query signal(chosen by inspection). The comparison yielding the minimum cost is chosen by the algorithm, in this case giving a correct match.

image-based positioning will have to deal explicitly with ambiguity. We envision two natural approaches to this problem, both based on constraint integration. A first task-specific solution would be to employ secondary sensors, such as cellular phone tower ID's, coarse GPS, or manual input to constrain the spatial region for the recognition problem. A second approach would be to allow a user to select from a series of alternative "guesses" regarding the building identity or position. Lastly, one could acquire a series of images by either looking in different directions, or while translating along a constrained trajectory and use a technique such as Markov localization [7] to probabilistically estimate the more likely location that could produce the ensemble of measurements.

We are currently conducting experimental trials to evaluate the scaling behavior and robustness of our approach.

References

[1] I. Cox, "Blanche - An experiment in guidance and navigation of an autonomous robot vehicle,", *IEEE Trans. on Robotics and Automation,* 7, pages 193-204, 1991.

[2] J. Canny, "A computational approach for edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence,*, 8(6), pages 679-698, 1986.

[3] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics,* Cambridge University Press, Cambridge, 2000.

[4] G. Dudek and J. K. Tsotsos, "Shape representation and recognition from curvature," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 35-41, June 1991.

[5] O. D. Faugeras, N. Ayache, and B. Faverjon, "Building visual maps by combining noisy stereo measturements," in *Proc. IEEE Conference on Robotics and Automation,* San Francisco, CA, 1986.

[6] J. Foley, A. V. Dam, S. Feiner, and J. Hughes. Computer Graphics. *Principles and Practice,* Addison Wesley, 2nd edition , 1990.

[7] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers, "Position estimation for mobile robots in dynamic environments, in *Proc. of the Fifteenth National Conference on Artifical Intelligence,* Madison, WI, 1998.

[8] B. K. P. Horn, "Shape from Shading: A Method for Obtaining the Shape of a Smoooth Opaque Object from One View," PhD Thesis, MIT, 1970.

[9] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. on Robotics and Automation,* 7(3), pages 376-382, 1991.

[10] E. Milios, "Recovering shape deformation by an extended circular image representation," *Proc. of 2nd ICCV,* pages 20-29, Dec. 1988.

[11] K. Prazdny, "Motion and Structure from Optical Flow," in *Proc. 6th International Conference on Artifical Intelligence,*, Tokyo, Japan, Aug. 1979.

[12] F. Pourraz and J. L. Crowley, "Continuity properties of the appearance manifold for mobile robot position estimation," *Proc. of the IEEE Conference on Pattern Recognition Workshop on Perception for Mobile Agents,* Ft. Collins, CO, June 1999.

[13] D. Robertson and R. Cipolla, "An Image-Based System for Urban Navigation", in *BMVC,*, 2004.

[14] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robitcs Research,* 21(8), pages 735-758, 2002.

[15] R. Sim and G. Dudek, "Learning environmental features for pose estimation," *Image and Vision Computing,* 19(11), pages 733-739, Elsevier Press, 2001.

[16] F. Stein and G. Medioni, "Map-Based Localization using the Panoramic Horizon", *Proc. of the IEEE Conference on Robotics and Automation,* pages 2631-2637, May 1992.

[17] Wei Zhang and J. Košeckà, "Localization Based on Building Recognition", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 21-30, June 2005.