Geo-Localization of Street Views with Aerial Image Databases *

Mayank Bansal Harpreet S. Sawhney Hui Cheng Kostas Daniilidis
{mayank.bansal, harpreet.sawhney, hui.cheng}@sri.com kostas@cis.upenn.edu
Vision Technologies Center
SRI International Sarnoff
Princeton, NJ 08540
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

We study the feasibility of solving the challenging problem of geo-localizing *ground level* images in urban areas with respect to a database of images captured from the air such as *satellite* and *oblique aerial images*. We observe that comprehensive aerial image databases are widely available while complete coverage of urban areas from the ground is at best spotty. As a result, localization of ground level imagery with respect to aerial collections is a technically important and practically significant problem. We exploit two key insights: (1) satellite image to oblique aerial image correspondences are used to extract building facades, and (2) building facades are matched between oblique aerial and ground images for geo-localization. Key contributions include: (1) A novel method for extracting building facades using building outlines; (2) Correspondence of building facades between oblique aerial and ground images without direct matching; and (3) Position and orientation estimation of ground images. We show results of ground image localization in a dense urban area.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*; I.4.9 [**Image Processing and Computer Vision**]: Applications; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision, signal processing*

General Terms

Algorithms, Theory

1. INTRODUCTION

Given a ground level street view (SV) image in an urban area, we want to determine the geo-location of the camera in the absence of any metadata (GPS or camera parameters). We explore a novel approach: use commonly available satellite and oblique aerial image databases (e.g. Microsoft Bing [bing.com/maps], Google Maps [maps.google.com]) to match the SV image and in turn geo-locate the SV image. This problem is technically challenging and has not been addressed before, and practically satellite and aerial databases provide comprehensive coverage of large areas whereas SV databases are spotty at best.

Given an area-of-interest (AOI) such as a city, we create a feature database using both satellite (SAT) and oblique bird's eye-view (BEV) imagery covering the area. SAT provides orthographic top-down views of the scene while BEV provides oblique viewpoints. By combining the two, building outlines as well as building facades are extracted. SAT images provide the outlines while one or more of the BEV images are used to locate the corresponding buildings and extract facades. Appearance matching between images with widely different viewpoints (e.g. SV and SAT) is a challenging problem. Prior work using SIFT, MSER etc. has a low success rate, and is not robust to partial occlusions and appearance differences. Also matchable features must be discriminative enough for retrieval from a large database of indexable features along with the geo-coding (latitude-longitude) information in the BEV imagery. We approach the matching problem from a joint statistical and appearance viewpoint. We compute features that capture the statistical self-similarity (or dis-similarity) of local patches on a building facade with respect to other patches on the facade. Since these features essentially capture the local appearance statistics, they are robust to viewpoint and global appearance changes and can be computed in a similar manner for the SV image as well and then robustly matched with the features stored in the database.

Related Work: In [16] and [12], a street view image is matched to geo-tagged street view images. Chung et al. [2] extract MSER regions that are clustered to build adjacency matrices for matching with spectral graph approach. In [1], omnidirectional views are matched to building outline maps by detecting the tallest vertical corners of the buildings and matching them through 2D to 1D projection. Coorg & Teller [3] used vertical facades in combination with GPS and inertial sensors, to determine building orientations. Kosecka & Zhang [7] use facades to compute accurate camera orientation. The facades themselves are detected either by reasoning about their vertical projections [8] or by directly detecting lattice patterns after searching for repetitive features or image structure [10, 4, 6, 13, 9] followed by geometric or appearance-based validation. Schindler et al. [14] automatically geo-localize street images by matching repeated patterns on building facades. Recently, Park et al.[11] have focused on recovering the camera direction of a geo-tagged street image using either street views or satellite imagery. However, they rely on segmenting ground plane in the satellite imagery.

*Area Chair: Gang Hua

This material is based upon the work supported by the US Army RDECOM Acq Ctr under Contract No. 08-C-0117.

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

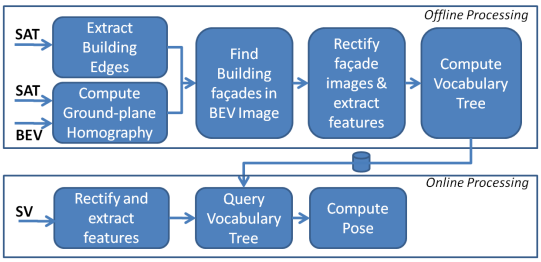

Figure 1: Overall system block diagram.

2. APPROACH

Fig. 1 depicts the flow of key processing modules in our approach. First, we align the ground plane between SAT and BEV imagery. Thus, BEV images can be rectified with respect to the ground plane with canonical axes (N-S, E-W) aligned. Then, we match building outlines extracted from SAT imagery with the corresponding outlines in the rectified BEV images. Subsequently, we use the identified building outlines to find the roofs of buildings thus identifying the facades. This allows extraction of orthorectified building facades from the BEV. Features that capture facade self-similarity are then extracted and stored in an adaptive vocabulary tree that hashes these features on the basis of mutual similarity. At run-time, for a query SV image, we first extract an orthorectified dominant facade region and then compute self-similarity features which are queried against the database to obtain a matching facade and its geo-location. Finally, point correspondences between the SV image and the BEV image for the identified location and direction are used to compute the 6 DOF pose of the SV camera using standard techniques.

2.1 Alignment

Input Imagery: We download both the overhead SAT and the oblique BEV imagery using Microsoft's Bing Web service. For computational management, each of these image types are downloaded as fixed size tiles, typically 100 meter square. Additionally, for both image types, we use the Web service to establish a map between pixel locations and their lat-long coordinates. For experiments in this work, we downloaded imagery for a 1 Km $\times$ 1 Km region in Ottawa, Canada. A few samples are shown in Fig. 2.

Imagery Alignment: Given the set of SAT and BEV image tiles and the mapping of their pixel coordinates to lat-long coordinates, we can warp the BEV images to the SAT coordinate system. To compute the warping transformation, we approximate it as a projective transformation between pixels in SAT and BEV – thus approximating the Earth's surface within each tile as a flat plane. Using the computed transformations, we warp each of the images to the SAT image coordinate system. As a result, the ground plane gets aligned well in all the images as shown in Fig. 3. To aid further processing, we also compute the dominant city block direction in the SAT imagery and rotate this image before warping the other images to its coordinate system. This renders most of the buildings parallel to the scan-lines in the image – a feature which will be exploited in further processing.

2.2 Facade Extraction

After initial imagery rectification, we extract regions from the BEV imagery corresponding to building facades. To ensure least distortion, we concentrate only on the facade planes which face the heading direction of the particular BEV image. Since the SAT imagery was previously rotated to align the city blocks with the image scan-lines, we can now restrict our attention to facade planes whose 2D projections are horizontal in the SAT images.

Figure 2: Sample SAT tiles (left) and BEV imagery (right) from Ottawa, Canada.

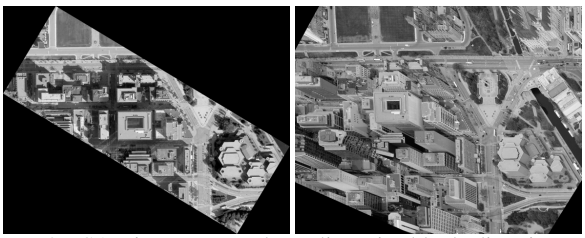

Figure 3: SAT image rotated to align city-block direction with the x-axis (left) and the corresponding BEV image automatically aligned to the SAT image w.r.t the ground-plane using the geo-coordinate information (right).

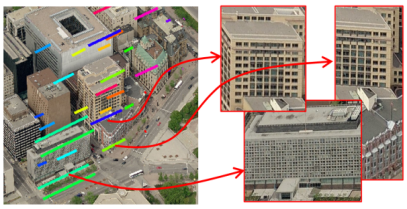

Figure 4: Example of the facade extraction process. a) detected building tops and bottoms, and b) extracted facade tiles.

Vertical Vanishing Point Estimation: In the ground-aligned BEV imagery, lines along the vertical (gravity) vanishing direction can be seen to be convergent. Before extracting affine corrected facades, we first rectify the BEV imagery so that these lines are rendered parallel. We detect canny edges in the BEV image and then group these edges into line segments. Lines along horizontal and vertical directions correspond to city block axes and can hence be rejected. From the remaining line segments, a RANSAC-based process then determines the inlier set of lines that intersect at the required vanishing point.

Image Rectification: Given the computed vanishing point, we now rectify the BEV image by mapping this vanishing point to a point at infinity (in particular to $v_x = [1, 0, 0]^t$), thus making the building edges parallel. This rectifying transformation is a projective warp which is computed by a method similar to the epipolar rectification method described in [5]. Due to the choice of v_x, the building facade edges in the rectified BEV become parallel to the image scan-lines.

SAT Edge Extraction: To extract building facades from BEV, we start by detecting building contours in the overhead SAT imagery. The contours need to be detected as chains of line-segments, each segment corresponding to one face of a building. We developed an iterative algorithm to extract these line segments from an initial canny edge-detector processed SAT image. Briefly, the algorithm links edges into edge-chains based on proximity and then fits line segments to these edge-chains, splitting wherever the deviation of the edges from the fitted line segment becomes greater than a threshold. Consistent line segments are merged into longer line segments and the overall process is iterated a few times.

Facade ROI Search: From the line segments extracted in the SAT imagery, we keep only the segments along the dominant facade direction in the BEV. Using the ground plane homography

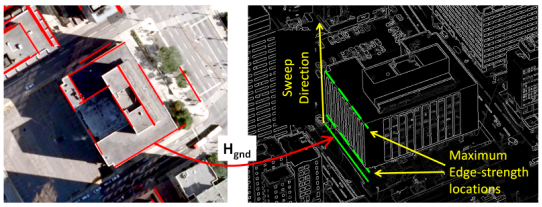

Figure 5: Building top search. Line segments extracted from the SAT imagery are projected to the canny edge map of BEV where a sweep along the gravity direction is expected to give a maximal point at the top edge of the building.

between SAT and BEV, we warp these segments into the rectified BEV image coordinate system. These segments then map to approximately the bottom of the buildings in the BEV image because the transformation corresponds to the ground plane. In the rectified BEV imagery, the gravity vanishing direction is aligned with the scan lines and therefore the tops of these buildings can be found by sliding the mapped line segments horizontally (Fig. 5). Our algorithm to determine the building tops is described below. Once the building tops are determined, we obtain the coordinates of the four corners of each facade which can then be mapped back to the original (unrectified) BEV imagery for high-resolution texture retrieval. For each facade, we crop the texture from the original BEV imagery and then warp it into a rectilinear coordinate system. Fig. 4 shows an example of this process where a few of the facades are extracted and rectified to their orthographic representations.

Computation of Building Tops using GC: Given the nature of the rectified BEV imagery, the top of each building can be determined as a translation $\delta(s)$ for each segment s projected to the building bottom. We formulate this problem as a Graph Cut (GC) optimization of an objective function that consists of the usual data and smoothness costs. The data cost for a line segment is strictly a function of the hypothesized translation and is computed by measuring the average edge strength in the rectified BEV image when the line segment is translated by this value. Thus, when the segment lands on the top of a building, we incur a lower cost due to the high edge strength. To ensure smoothness in the translation values for connected line segments, we add a smoothness cost that penalizes difference in translation values for line segments that are spatially close to each other at their endpoints. For the typical polygonal chains of line segments that we detect for each building, the smoothness cost enforces a strong constraint that the entire building top be at a single translation and avoids the problem of local optima occurring at the numerous edges in the middle of the building facade. Fig. 6 shows an example of how this optimization approach helps the building extraction process.

2.3 Feature Detection & Matching

We represent each rectified facade with appearance features that are extracted over a uniformly sampled grid on the facade. Features from all the facades populate a feature database. We extract a similar set of features for a SV query image. Best matches between the query features and the database features are used to index facades in the database. The location corresponding to the best matching facade are used used to estimate the location and orientation of the coordinates of the query image.

Features: Features like MSER and SIFT do not work well for matching between the BEV and SV imagery due to their large viewpoint and appearance differences. We observe that even under large appearance and viewpoint changes, the layout of local patches within each facade can be used to create a statistical description of the facade pattern. Such statistical descriptions do not

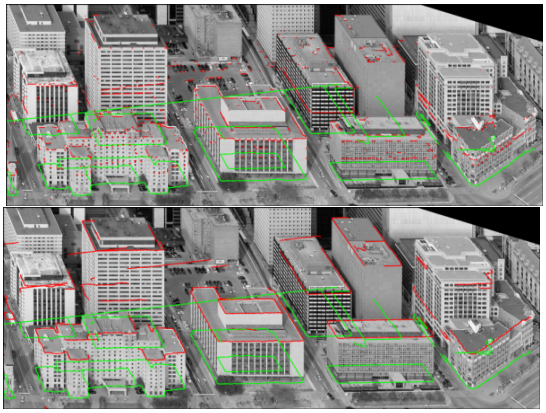

Figure 6: Effect of Graph-Cuts Optimization. The green edges are the SAT edges directly projected to this view and they lie in the ground plane. The red edges are the estimated building top edges. The top row shows the estimates obtained by picking the maximum score for each edge pixel independently; the bottom row shows these estimates refined by the GC optimization.

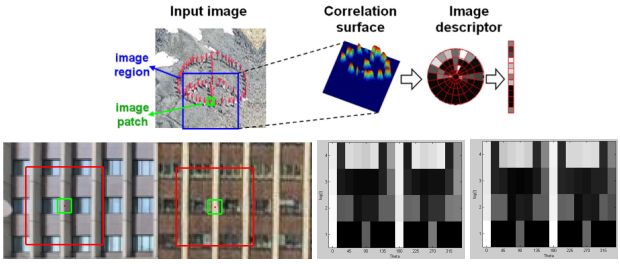

Figure 7: Self-similarity descriptor computation (top row), and example self-similarity descriptors for corresponding facades from the SV and BEV images respectively (bottom row).

get affected by the appearance and viewpoint changes as described by Shechtman et. al [15].

For a given pixel q, the local self-similarity descriptor d_q is computed by defining a patch centered at q and correlating it with a larger surrounding image region R_q to form a local 'correlation surface' (Fig. 7 top row) which is then transformed into a binned log-polar representation to account for local spatial affine deformations. For details, please refer to [15]. The bottom row of Fig. 7 shows rectified facades from corresponding SV and BEV images with the red ROI showing the region of support used for computing the self-similarity descriptor at the red pixel in the center. Also shown are the corresponding descriptors (computed at the center of the red ROIs, R_q) which are noticeably quite similar even with the large appearance difference between the images themselves.

Feature Detection: A single building facade may consist of multiple patterns at different scales. This variation is best captured by the self-similarity descriptor evaluated at the proper scale. We sample a uniform grid of points on the facade and extract the self-similarity descriptor at each of these points at a fixed set of scales. All the descriptors thus obtained are labeled with a unique number which we will refer to as the ID of the given facade. We repeat this process for each facade in our dataset and the descriptors are thus labeled with unique IDs equal in number to the number of facades.

Feature Matching: For scalable data retrieval using the self-similarity features, we use an Adaptive Vocabulary Tree (ADT). We push each of the feature vectors from each facade with the unique ID as the associated label into an ADT data structure. The ADT hashes the feature vectors according to the frequency of the IDs and the co-occurrence of the dimensions of the features resulting in a tree structure (Fig. 8). This tree structure is the database which we use at run-time for query search as described next.

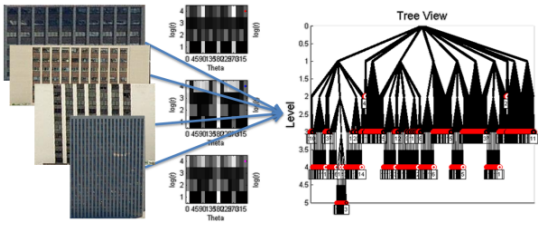

Figure 8: Self-similarity descriptor features extracted from various facades get pushed into an ADT data-structure.

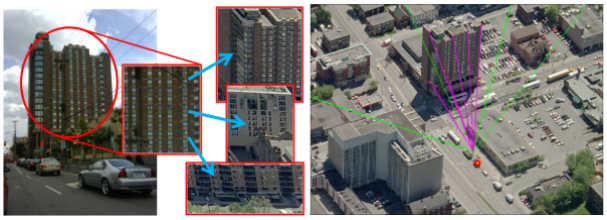

Figure 9: Online processing of a SV image from Panoramio. Extracted and rectified facade returns these top three matching facades from the database. Using the top match, we estimate the camera pose (the point with the emanating rays) and compare it with the ground-truth location (red circle).

2.4 Query Processing & Pose Estimation

At run-time, we first rectify the query SV image by computing the vanishing points in the horizontal and vertical directions, and then extract any facade occupying a substantial portion of the image. For extracting the facades, we exploit the dominance of horizontal and vertical facade patterns; we compute HoG descriptors at a uniform sampling of pixels in the image and then cluster the resulting descriptors using K-Means. Self-similarity features are then extracted on a uniform grid like for the BEV image. These features are queried against the ADT database which returns matching candidates as the top few best matches. The score differential between the top two hits gives us an estimate of the confidence in the retrieval. If the retrieval is confident, we proceed to the geometric localization step with the best match; otherwise, we process both matches for pose estimation and use geometric consistency to prune out the bad match.

Pose Estimation: Facade matching is in itself good enough to localize the SV image within a constrained visibility zone defined by the facade. However, for precise localization of the SV camera we compute the 6 DOF pose of the camera to establish the efficacy of our method. We manually identify 7 point correspondences between the SV and BEV image in the structure surrounding the matched facade. These correspondences are used to estimate the fundamental matrix F [5] between the SV and BEV images. The epipole of the BEV image, as computed from F, then corresponds to the SV camera location in the BEV coordinate system.

The SV camera location in the BEV image is mapped to absolute lat-long coordinates using the ground plane correspondence with the SAT imagery. Finally, the metric (cms/pixel) information in the SAT image is used to estimate the camera focal length which can be used in conjunction with any knowledge about the CCD array dimensions to establish the camera field-of-view as well.

3. EXPERIMENTAL RESULTS

We tested our system on a region around Ridieu St. in Ottawa, Canada with a sample shown in Fig. 2. We used BEV imagery from the west and south heading directions to capture facades as seen from these two directions. For the test imagery, we used imagery from Panoramio and screen-shots from Google Street-view both of which contain lat-long information. Fig. 9 shows an example processing flow for a query SV image from Panoramio. The scores for the top three BEV facade matches shown were 5632.0, 1452.0 and 290.0 respectively. The magnitude difference between the first and second score suggests a good degree of confidence in the match. Pose estimation using the matched facade then gives us the final camera location of the SV image which is also shown in the figure along with the ground-truth location depicted as a red circle with a yellow center. Fig. 10 shows results on four more examples, the right column queries being screen-shots from Google Street-view.

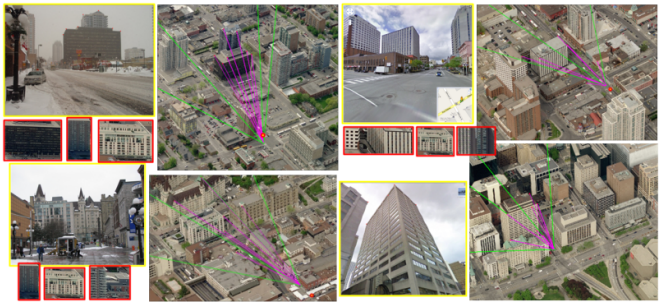

Figure 10: Localization Results. Yellow boxes: input SV images, Red boxes: top 3 facade matches, and estimated and ground-truth pose on the BEV for each input.

In summary, this work establishes the feasibility of matching highly disparate street view images to aerial image databases to precisely geo-localize SV images without the need for GPS or camera metadata. In future work we plan to establish the quantitative efficacy of our method through large city-scale experiments.

4. REFERENCES

[1] T. Cham, A. Ciptadi, W. Tan, M. Pham, and L. Chia. Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map. In *CVPR*, 2010.

[2] Y. Chung, T. Han, and Z. He. Building recognition using sketch based representations and spectral graph matching. In *ICCV*, 2010.

[3] S. Coorg and S. Teller. Extracting textured vertical facades from controlled close-range imagery. In *CVPR*, 1999.

[4] P. Doubek, J. Matas, M. Perdoch, and O. Chum. Image Matching and Retrieval by Repetitive Patterns. *ICPR*, 2010.

[5] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press New York, NY, USA, 2003.

[6] J. Hays, M. Leordeanu, A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. *ECCV*, 2006.

[7] J. Kosecka and W. Zhang. Video compass. In *ECCV*, 2002.

[8] J. Kosecka and W. Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *CVIU*, 2005.

[9] P. Mueller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. In *ACM SIGGRAPH*, 2007.

[10] M. Park, K. Brocklehurst, R. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *TPAMI*, 31(10):1804–1816, 2009.

[11] M. Park, J. Luo, R. Collins, and Y. Liu. Beyond GPS: determining the camera viewing direction of a geotagged image. In *ACM-MM*, 2010.

[12] D. Robertson and R. Cipolla. An Image-Based System for Urban Navigation. *BMVC*, pages 819–828, 2004.

[13] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. *Shape, Contour and Grouping in Computer Vision*, pages 81–81, 1999.

[14] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments. In *CVPR*, 2008.

[15] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[16] W. Zhang and J. Kosecka. Image Based Localization in Urban Environments. In *Proc. Int. Symp. on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2006.