A Vision-based 2D-3D Registration System
Quan Wang, Suya You
CGIT/IMSC USC
Los Angeles, CA 90089
quanwang@usc.edu, suyay@graphics.usc.edu

Abstract

In this paper, we propose an automatic system for robust alignment of 2D optical images with 3D LiDAR (Light Detection and Ranging) data. Focusing on applications such as data fusion and rapid updating for GIS (geographic information systems) from diverse sources when accurate georeference is not available, our goal is a vision-based approach to recover the 2D to 3D transformation including orientation, scale and location.

Two major challenges of 2D-3D registration systems are different sensors problem and large 3D viewpoint changes. Due to the challenging nature of the problem, we achieve the registration goal through a two-stage solution. The first stage of the proposed system uses a robust region matching method to handle different sensor problems registering 3D data onto 2D images with similar viewing directions. Robustness to 3D viewpoint changes is achieved by the second stage where 2D views with sensor variations compensated by the first stage are trained. Other views are rapidly matched to them and therefore indirectly registered with 3D LiDAR data.

Experimental results using four cities’ datasets have illustrated the potential of the proposed system.

1. Introduction and related works

The traditional image matching problem is to find matching primitives between digital images from optical sensors. Evaluation criteria include invariance to lighting, orientation, scale and location changes as well as other geometric distortions. Many well-known existing methods are based on local texture analysis around interest points.

Recently, there has been a growing need to register 3D range sensing data with digital photography in nadir, oblique or even ground views. For example, the photorealistic modeling and real-time rendering of urban scenes requires the efficient registration of 3D range data or derived models onto aerial or ground 2D images. In the medical image processing domain, there is a long standing concern about how to automatically align CT (Computed Tomography) or MR (Magnetic Resonance) images with optical camera images. Frequently, those data are captured not only by different sensors, but also at significant time differences and possible large 3D view point changes.

Figure 1: Overview of the proposed 2D-3D registration system

This paper proposes a robust and automatic 2D-3D registration system (overview in figure 1), which efficiently registers 3D LiDAR data with 2D images of diverse viewing directions and from various sources. Figure 2 shows typically inputs and outputs of our system. More application results are in the supplemental materials.

The first stage focuses on different sensors problem and assume similar viewing directions (nadir view) of the inputs. Based on our basic assumption: the dominant contours of many interest regions (ROI), which typically are well-separated regions of individual buildings, are repeatable under both optical and range sensors, we extract, describe and match ROI from optical images and projected depth image of LiDAR data. This paper provides details of our LiDAR segmentation method (section 2.1) and key steps of the region matching component (2.2). We also introduce several techniques that significantly improve the whole stage’s efficiency (2.3).

The inputs to the second stage (section 3) are all from optical sensors but could have large viewpoint changes. In order to match other images with those already registered nadir images, we first obtain distinctive and invariance features of the nadir images through a multiview-based training process. Next, other images with different viewing
directions are rapidly matched with those nadir images and indirectly registered with 3D range data. Excluding the training process, which happens offline and only once for each location, our system is able to process 10 frames per second at this particular stage, which makes it ideal for efficient data fusion and urban scene rendering. Furthermore, the proposed LiDAR segmentation method is an important component for a variety of tasks such as recognition, understanding and modeling of urban scenes.

Traditional texture-based image matching approaches such as [1], [2] and [3] can not be directly adapted to this newly emerged registration problem, basically because range sensors capture no texture information. To tackle the problem, many recently developed methods first reconstruct dense or sparse 3D point clouds from 2D images then use 3D features to establish correspondences with initial alignment provided by positioning hardware. Zhao, et al. [4] use motion stereo to recover dense 3D point clouds from video clips. ICP (Iterative Closest Point) algorithm is used to register those reconstructed 3D points to LiDAR data with initial alignment provided by positioning hardware such as GPS (Global Positioning System) and IMU (Inertial Measurement Unit). Later work of Ding, Lyngbaek and Zakhor [5] detects 2D orthogonal corners (2DOC) as primitives to match single oblique image onto LiDAR data using similar positioning hardware. It achieves overall 61% accuracy during a test of 358 images and the processing time of each image is only several minutes in contrast to 20 hours in [6]. We agree that accurate and uniform georeference can be assumed to associate with particular inputs for certain stage of the registration to significantly simplify the problem, so that there only exist small offset errors to correct. However it is not reasonable to assume such assistant information always available for general inputs from diverse sources, especially for data fusion and GIS updating using historic data and oblique or ground photos from common users.

An automatic system for texture mapping 2D images onto ground-scanned 3D range data is proposed by Liu, et al. High level features such as rectangles are extracted and matched for 2D-3D registration [7]. Ground-based manual range scan is able to provide rich 3D details about a particular building’s structure. However, it is not feasible to obtain such data for large urban scene efficiently. Moreover, because both range and optical data are ground-captured focusing on one building for each test, the range of possible initial misalignments is restricted. Their experiments report that only a portion of 2D images can be independently registered with 3D range data and the system will fail in parts of the scene without sufficient features for matching. The working range can be expanded [8] by applying structure from motion technique to a sequence of images to produce sparse point clouds and aligning dense and sparse point clouds.

Both [4] and [8] try to use multiview geometry to recover 3D point clouds from image sequences. The availability of appropriate multiple 2D images associated with 3D range data, the well-known challenge of inferring 3D from 2D, and the difficulty of finding correspondences among 3D primitives without good initial pose estimations all raise practical restrictions to such approaches.

Figure 2: (a) inputs; (b) registration results illustrating our two-stage system; (c) applications to urban modeling and rendering;

Other related works include: BAE-systems [10] uses high-resolution stereo images to reconstruct 3D. Specially designed hardware is needed for inputs, the problem of fusing other views still remains, and unlike LiDAR, such approaches typically have difficult handling vegetation area. Viola and Wells [11] use mutual information to align un-textured models to a scene. The first problem is that a clean model could be difficult to obtain from noisy, low resolution 3D range data. Moreover, using mutual information as similarity measurement lacks spatial information since it processes each pixel independently, consequently tends to produce unstable results. Recent work of Yang et al. [21] uses 2D-3D matching to refine model to image mapping estimates. In [22], 3D point clouds are aligned to nadir images by matching 3D points onto 2D edges under camera constraints.

Our registration system has been tested using datasets containing nearly 1,000 images. Experimental results in terms of robustness to different sensors, viewpoint changes, large geometric distortions and missing of partial data demonstrate the potential of our system.
2. Registration handling different sensors

This section presents the first stage of our two-stage registration system, handling different sensors problems. Inputs are nadir view photography of urban scenes without georeference and depth images projected from LiDAR data. Their relative scale, rotation and translation are unknown. Outputs are initial point-to-point registration and the recovered transformation.

2.1. LiDAR segmentation

The first component of this stage is ROI extraction from 3D range data. Similar topic has been intensively studied ever since the range sensing technology emerges. Early work of Vestri and Devernay [12] extracts planar surface patches from Digital Elevation Model, which is obtained from laser altimetry. Tall trees which produce a region of high range responses similar to buildings have always been a major problem for LiDAR data segmentation. Kraus and Pfeifer [13] propose an iterative approach, in which 3D measurements with residuals to current surface larger than a threshold are removed and the surface is re-computed using the remaining data. Recent work of Verma et al. [14] proposes a region-growing based approach using Total Least Squares to extract local planar patches and merge consistent patches into larger regions. The largest region is simply labeled as ground while others are building, which is not true for heavily urbanized areas. Later work by Matei et al. [15] overcomes this limitation by applying ground classification prior to building segmentation with the help of minimum ground filtering and surfels grouping.

Different from previous approaches which use LiDAR segmentations directly for modeling and desire as many details as possible to produce accurate models, our goal is to extract the most external boundaries of interest regions, which will be used for region matching later in the proposed registration system. Based on this goal, we propose our own method based on local consistent check (LCC) and region growing with watershed.

2.1.1 Normalization

Because intensities of input depth images tend to be clustered, to enlarge the effective range and facilitate later processing, we normalize the depth images. Two different normalization methods are compared. The first one utilizes a lookup table mechanism based on integral of histograms. The other method simply translates and rescales all the intensities to the same range. Experiments show that the first method will reveal a lot of details inside building or ground regions, ideal for direct modeling but will cause confusion to our ROI extraction. So we apply the second method to normalize input depth images.

2.1.2 Region growing with local consistent check

Next, initial segmentations are generated using region growing with watershed and LCC to remove outliers like trees. The original watershed algorithm [16] treats the strength of detected edges as elevation information, which is used to hold the flooding water within separated "lakes". Concerning depth images, the intensity value of each pixel is already the hardware obtained elevation. So we simply negative all the intensity values in depth images and then fill in water from marker locations (section 2.1.3) in a natural bottom-up way. This corresponds to a recursive region-showing process starting from marker locations as seed points and depth information as boundary constraint.

Each current segmenting region is recursively expanded until one of three boundary conditions are met:

First, the current pixel's depth value in positive depth images is above a soft threshold \(WL_{\text{dyn}}\), which is dynamically fluctuated around the global water level \(W(L)\). A region with smaller current area \((\text{area}_{\text{cur}})\) will have a larger range so that the expansion won't be stuck in isolated small peak regions such as a high clock tower on top of a building. When the current area is relative large \((\text{area}_{\text{cur}} > \text{area}_{\text{MIN}})\), we have: \(WL_{\text{dyn}} = W\). Otherwise, \(WL_{\text{dyn}} = W - \left(-^{(\text{area}_{\text{cur}}-\text{area}_{\text{MIN}})} + 1\right) \cdot \text{range}_{\text{MAX}}\) where \(\text{area}_{\text{MIN}}\) is the minimum acceptable area and \(\text{range}_{\text{MAX}}\) is the maximum adjustable range.

Second, if neighboring pixels along both positive and negative directions of x/y axis fail the 1st condition, we say the central pixel is fully isolated. If only one direction is isolated, we call it partial isolation. A partial isolation on either x or y axis indicates a possible edge point while a partial isolation both x and y axes indicates a possible corner point. However, a fully isolation on either x or y axis likely indicates a noise pixel outside the ideal boundary and further expansion will be stopped.

Last, the current pixel \((I_i)\) should pass the LCC. Here a two dimensional Gaussian is centered at the current pixel and the neighboring pixels with approximately identical distances from \(I_i\) form neighboring circles (suppose \(I_i\) is the j-th pixel on the i-th circle away from \(I_i\)). We apply a stricter consistency requirement for closer pixels. The current pixel will pass the LCC only if the number of consistent neighboring pixels is above certain percentage \((P)\) of the total number of scanned pixels \((\text{Num})\).

Let \(T_i\) represents the consistency threshold for the i-th circle with radius \(r_i\), we have: \(T_i = (1 - \exp(-\frac{r_i^2}{2\sigma^2})) I_i\). Now define a binary function:

\[
\delta_{i,j} = \begin{cases} 
1 & |I_{i,j} - I_i| < T_i \\
0 & \text{otherwise}
\end{cases}
\]

So the last boundary condition can be expressed as:

\[
\int \int (\delta_{i,j})dijdj < P \cdot \text{Num}
\]

Trees, especially those high and close to buildings, have always been a challenge problem in LiDAR segmentation.
Many recent approaches (e.g. [17], [18]) compute expensive 3D normal vectors for each pixel to remove trees. Experimental results showed that over 95% of the trees can be removed by our method at a much lower computation cost. Figure 3 shows a challenging example where tall trees partially overlapped with the building.

Figure 3: from left to right, the original depth image, the segmentation with and without LCC.

2.1.3 Marker locations
To determine marker locations for watershed, a uniform grid is placed onto the depth image. The three marker conditions are: (1) the pixel should pass LCC. This will eliminate many tree markers because of the noise-like and inconsistent nature of tree's depth values. (2) The intensity of this pixel, which corresponds to the relative depth of the location, is above the global water level (WL). (3) In case when multiple pixels inside the same cell satisfying condition (1) and (2), the one with highest intensity will be chosen. It is possible that all pixels fail certain condition and the corresponding cell has no marker at all (e.g. when the cell is placed on the ground or trees).

2.1.4 Refinement
The initial segmentations, which generally have low false positive rate, may have jaggy edges along boundaries that are smooth in reality and small holes inside the region not from physical depth discontinuity but from remote sensing noises. To reduce false negatives, the refinement process scans background pixels within each segmented region's bounding box. If the majority neighbors of a background pixel are foreground pixels, then it will be re-labeled as foreground. Theoretically, the refinement process should run iteratively until the region's area become constant. In practice, we observe that segmented regions become stable after only two iterations.

Segmented regions with an extremely small size or area are either noise regions or not reliable for future registration, and therefore discarded. The rest are still highly redundant, because multiple markers placed on one building generate multiple regions. Since our algorithm can obtain consistent results for the same region from different markers, we merge regions with corner distance less than 3 pixels and area difference within 5% to remove the redundancy.

In terms of parameters sensitivity, most parameters can be fixed to some reasonable values. We find the only data-dependent parameter is the global water level (WL). Although ideally its value should be the shortest building's relative height, which depends on the details of re-sampling and normalizing the raw LiDAR, experiments indicate a large range of WL is acceptable by our system. Additionally, because we have single uniform source of LiDAR data for each city, the WL value only needs to be set at most once for each of the four cities' datasets.

Finally, all the ROI\textsubscript{range} extracted from 3D range data and their most-external contours (connected close contour guaranteed by the intrinsic properties of watershed method) represented by contour point lists are saved. We also develop an algorithm (not covered in this paper) to extract ROI\textsubscript{optical} from optical images. Figure 4 shows one result for each of our four testing cities.

Figure 4: color-coded LiDAR ROI extraction results

2.2 Region matching
We describe each ROI using 2D histograms of the relative distances and angles of its sampled contour points, similar to the shape context [19]. They are computed in relative frame for orientation invariance, either constructed by computing tangent vectors for each pixel [20] or by using principal directions when they can be reliably obtained (section 2.3). Other adaptations we introduced to the shape context include: additional partial descriptors are formed for ROI, using continuous subset of the original contour points containing a larger number of corner points. Scale invariance is enhanced by distance normalization using ROI bounding boxes' sizes, which we believe is more stable under imperfect segmentations.

To establish ROI correspondences, we measure the similarity of two ROI as the minimum average histogram distance (matching cost) of their corresponding sampled contour points. Because in our case, those points represent the most-external contours of ROI in relative frame, instead of solving the expensive general bipartite matching problem, searching complexity becomes a low constant by
first ordering the contour points in counter-clock manner and only searching close locations for correspondences. Once one point's matching is determined, the rest points are automatically corresponded. Additionally, we introduce "cost ratio" as the ratio of best matching's cost over the second best matching's. A lower cost ratio indicates a ROI correspondence with higher confidence. Rectangle ROI are generally ambiguous and produce higher cost ratio because many ROI in urban scenes have similar shapes, while ROI of special shapes will produce lower cost ratio and higher matching confidence.

![extracted ROI contours](Image 133x477 to 247x585)

Figure 5: ROI descriptors. The first column is one ROI extracted from LiDAR data. The second and third columns are the corresponding ROI extracted from an optical image. Notice the similarity between 1st and 2nd columns' descriptors.

With the help of cost ratio, we apply an RANSAC-like method to remove outliers among the initial ROI correspondences. The consistent set is used to compute a global transformation $(T_1)$ and with it the matchings are propagated across the entire scene. It is during this process, global context is implicitly taken into consideration. The first stage will claim no correspondence could be established without proper transformation found.

2.3. Principal directions

We also propose two techniques particularly designed for urban scenes to improve our system’s performance. In this stage of our system, when dealing with nadir or slightly oblique views, a reasonable assumption is many buildings can be aligned to two principal directions. To detect those principal directions, we first use Canny operator to produce edge responses on each pixel. Then probabilistic Hough transformation groups them into line segments. To handle irregular buildings and tolerate noises from edge detection and line grouping, directions of each line segment are placed into a histogram of 36 bins. The first two peaks of the histogram correspond to the two principal directions. For better accuracy, the 3 histogram bins closest to each peak will interpolate the peak position. Given the principal directions, two techniques can significantly reduce the computational cost of this stage.

First, the input depth and optical images will be rotated according to the positive and minimum of the two principal directions. When the angle difference between optical and depth images is large, a full search for all four possible directions (positive and negative for each principal direction) could be needed. Therefore, we can generate rotation invariant ROI descriptors by dealing with at most four possible directions without computing the expensive tangent vectors for each point.

Second, a learning-based method is used to prune those initial ROI too irregular to become building regions. To reduce the necessary searching space for later registration. Most lines belonging to regular building's contours should be compatible with the principal directions. For contour pixels of each ROI candidate, their x and y coordinates in the rotated relative frame are projected into two 10-bin histograms respectively. As a result of the region-growing process, each ROI candidate is a close external contour, so each histogram should have at least two distinct peaks.

The distinctiveness of highest peak is measured by peak neighborhood ratio ($PNR$), the average of the peak's two closest neighboring bin values and divided by the highest peak's bin value. The $PNR$ of the second peak is the ratio of its bin value to the highest peak's. Finally, the $PNR$ of each histogram is the average of two peaks' $PNR$.

Given two $PNR$ for each ROI, $PNR(H_i)$ and $PNR(H_j)$, our goal is to determine whether this region belongs to the building class ($c = 1$) or not ($c = 0$). We choose to use Linear Discriminant Analysis (LDA) to find a quadratic decision boundary for $((PNR(H_i), PNR(H_j))$, $c)$, because first, LDA gives better accuracy particularly when the amount of training data is small; second, LDA can be computed directly from training data in one pass.

For each city's depth images, we choose one with ROI candidates manually labeled as $c = 1$ or 0 to form the training set. The joint likelihood $P(PNR, c)$ is computed as:

$$P(c) = P(c) \cdot P(PNR | c) = \frac{1}{\sqrt{2\pi} \Sigma} \exp \left( -\frac{1}{2} (PNR - \mu_c)^{\top} \Sigma^{-1} (PNR - \mu_c) \right),$$

where $\mu_c$ is the mean vector for class $c$ and in our case $\Sigma$ is a $2 \times 2$ covariance matrix for linear model. Their estimated values are directly computed from training data:

A new test region $R_{test}$ will be classified into class 1 if:

$$\log \left( \frac{P(c = 1 | PNR_{test})}{P(c = 0 | PNR_{test})} \right) > 0.$$

Suppose $N$ is the total number of training regions among which $N_1$ has label 1 and $N_0$ label 0. Using the formula of conditional probability and the $P(PNR, c)$ equation above, the final condition for $R_{test}$ to be classified into class 1 is:

$$\log \left( \frac{P(c = 1 | PNR_{test})}{P(c = 0 | PNR_{test})} \right) = \log \left( \frac{N_1 \cdot \exp \left( -\frac{1}{2} (PNR_{test} - \mu_c)^{\top} \Sigma^{-1} (PNR_{test} - \mu_c) \right)}{N_0 \cdot \exp \left( -\frac{1}{2} (PNR_{test} - \mu_0)^{\top} \Sigma^{-1} (PNR_{test} - \mu_0) \right)} \right)$$

$$= \log \left( \frac{N_1}{N_0} \cdot \frac{\exp \left( -\frac{1}{2} (PNR_{test} - \mu_c)^{\top} \Sigma^{-1} (PNR_{test} - \mu_c) \right)}{\exp \left( -\frac{1}{2} (PNR_{test} - \mu_0)^{\top} \Sigma^{-1} (PNR_{test} - \mu_0) \right)} \right)$$

This condition defines a linear decision boundary on the
selected feature point we extract 32x32 pixel image patches, points with high repeatability will be selected. Around each object location and group features accordingly. Feature different training views correspond to the same physical views. Since transformations used to generate training point detector will detect potential feature points for all using randomly generated transformations. An interest number of training views for each registered nadir image large number of independent frames per second, ideal for lighting and viewpoint changes, it can efficiently process a (MVKP) method [3], because besides robustness to choose to use the offline multiview training and online photography. Many widely-accepted methods exist. We is a typical wide baseline matching for conventional meaningful outputs. Unlike the first stage, our second stage we are not able find any existing method that can produce from different sensors with no initial alignment assumed, registered with 3D range data by the first stage. For inputs from optical sensors, one of them is nadir view already registered with 3D range data by the first stage. For inputs from different sensors with no initial alignment assumed, we are not able find any existing method that can produce meaningful outputs. Unlike the first stage, our second stage is a typical wide baseline matching for conventional photography. Many widely-accepted methods exist. We choose to use the offline multiview training and online query framework of Multiple View Kernel Projection (MVKP) method [3], because besides robustness to lighting and viewpoint changes, it can efficiently process a large number of independent frames per second, ideal for rapid data fusion and scene rendering applications.

![Figure 6: top-ranking features with (left) and without (right) the additional feature selection.](image)

During the offline training, MVKP method synthesizes a number of training views for each registered nadir image using randomly generated transformations. An interest point detector will detect potential feature points for all views. Since transformations used to generate training views are known, we are able to tell which feature points of different training views correspond to the same physical object location and group features accordingly. Feature points with high repeatability will be selected. Around each selected feature point we extract 32x32 pixel image patches, treated as a vector composed of 1024 pixel intensities. These vectors would be projected onto a much lower dimensional space using Walsh-Hadamard kernel projection and still preserve the distance relationship. Based on the characteristics of urban scenes, we also apply an additional feature selection method computing the variance inside the same view track and distinctiveness across different view tracks. Many feature points along the boundaries of buildings un-stable under 3D viewpoint changes are automatically discarded (figure 6).

The W.H. kernels are used because: (1) they are very fast to compute and apply because the coordinates of their basis vectors are either +1 or –1; (2) it has been shown in [3] when the kernels are ordered based on increasing frequency of sign changes, an accurate lower bound of the original feature vectors’ distance can be achieved using only a small number of kernels to greatly reduce the computational cost.

The output of the offline training stage is a feature database for nadir images, with feature descriptors labeled by the physical object locations they correspond to.

After images from other viewing directions enter the registration system and strong feature points are detected, normalized patches around those points produce feature vectors for those un-registered images using the same W.H. kernels. Initial matching are established by an efficient nearest neighbor search using two-layer structure and fast filtering with block distances. Finally, standard RANSAC is applied to detect the consistent matching.

If the second stage is not able to find a sufficient large consistent set after RANSAC, it will claim the current query image can not be matched to the registered nadir image and input the next query image. Otherwise, our system will compute a global transformation (T2) from the training image (nadir view) to the current query image using those consistent matching. Combined with the transformation T1 from the first stage, the query image is therefore indirectly registered with 3D range data.

4. Experimental results

The proposed whole system was intensively tested using LiDAR data of four cities: Atlanta, Baltimore, Denver and Los Angeles, mainly focusing on urban areas. We have the LiDAR datasets and many aerial images for those cities covering the downtown and surrounding rural areas. We have single source for each city’s LiDAR data. But there is distinctive difference between different cities. For example, range data for Los Angeles contains a large potion of ground and vegetation, and were captured years ago. The resolution is low with a lot of noises due to the technology restriction of early days and naturally, some data sections are out of date because old buildings are torn down and new buildings are erected over the years. Some data like Baltimore contain heavily urbanized areas and are very current with high resolution and low noise level.

The optical images we tested are even more diverse.
Many of them are from various sources (e.g. returned from online image search engine), captured in early years with low resolution. No georeference data can be tracked at all. When aligned with corresponding depth image, initial displacement of optical images is unknown. They may have any rotation (in-plane rotation for nadir images), even upside down. The scale errors range from 0.3 to 3. The location error could be comparable with the image size. The corresponding ROI might lie on the opposite corner of the image or the input optical image may not correspond to the depth image at all. Our testing set currently contains nearly 1,000 images, half of which are intentionally distorted with various geometric and viewpoint changes for performance evaluation.

4.1. LiDAR segmentation results

Our LiDAR segmentation algorithm has been tested on over 400 depth images projected from the four cities’ range dataset. Each contains around 30 buildings. Figure 4 shows one result for each of the four cites.

![Figure 7: segmentation comparison. (a) input depth images; (b) our extracted ROI; (c) graph-based segmentation.](image)

Figure 7 shows our segmentation results compared with Efficient Graph-based Segmentation [9]. Our extracted regions are more focused on interest buildings and can provide more accurate external contours. The computation time is roughly half of [9]. An average of more than 80% buildings can be correctly extracted by our algorithm. The number of false positives is generally around 10%. There could exist a few or several building in a scene wrongly labeled as background. Nonetheless, we believe it is neither reasonable today nor necessary to request perfect image segmentation. The important thing is how to make the best use of imperfect segmentation results. In our case, how to establish correct correspondences for parts of the scene and expand the partial results to the rest.

4.2. Whole system evaluation

First of all, the evaluation of a registration and matching system generally involves invariance to rotation, lighting and scale changes as well as other distortions. For the first stage, orientation invariance is achieved by generating ROI descriptors in relative frames. Scale invariance comes from distance normalization and by placing ROI of different scale into histograms with a fixed number of r bins. Distortion can be tolerated by histogram-based dense descriptors. Lighting issue doesn’t apply to this stage. For the second stage, invariance primarily comes from the multiview training while robustness to lighting comes from patch normalization and W.H. kernel projection (by discarding the first DC component). Both stages search local or semi-local matching primitives across the entire scene. Therefore, location invariance is naturally handled. Furthermore, both stages are robust again missing of partial data (e.g. due to occlusion or historic data). Experiments using nearly 1,000 images have demonstrated the invariance and robustness of our proposed system.

Second, concerning the registration success rate, currently the first stage achieves nearly 70% of success rate for our whole testing set of depth and nadir images. For example, for the city of Los Angeles, despite the low resolution and historic nature of the data, 230 out of the total 324 images are correctly registered. The registration failures are mainly due to insufficient matching primitives (ROI), either basically because the lack of such primitives in the scene, in which case even human found the registration difficult or impossible, or because such primitives can not be accurately acquired through segmentation techniques although it "seems" obvious to human observers. Still, to the best of our knowledge, there is no existing registration method that can achieve similar performance for the different sensors problem without support from positioning hardware. The success rate of the second stage largely depends on how wide the viewpoint changes from nadir view. Though wide baseline matching is still a challenging problem by itself, the second stage is typically able to correctly register two or three out of ten frames per second. Experimental results also show the success rate could be increase by up to 10% by using some other methods such as [1], [2], but the registration of each frame generally takes several seconds. Therefore, we decide MVKP is still best suitable for the second stage.

Finally, when it comes to the accuracy for those success registrations, the averaging pixel errors (APE) for the first stage is within 5 pixels even for propagated matchings (e.g. The APE for Baltimore dataset is 2.89). Those errors primarily comes from the difficulty of locating exact pixel locations inside high-level features due to shadows, the segmentation leaking and breaking, etc. However, the APE for the second stage could be as large of half of a building’s...
size. That is mainly because our current system simply fit a homography for \( T_2 \) using the consistent matchings instead of recovering the actual camera model. The current accuracy is sufficient for tasks like recognition. If needed by some applications, refinement process similar to [4] and [5] could be applied.

Figure 8: (a): initial correspondences of the first stage, which are used to compute \( T_1 \); (b) and (c): results after matching propagation, where \( T_1 \) and \( T_2 \) are applied respectively to the ROI contour points extracted by LiDAR segmentation. The results are visualized by the bounding boxes and centers of all interest regions’ point-to-point correspondences.

References


