Local-to-Global Point Cloud Registration using a Dictionary of Viewpoint Descriptors

David Avidar, David Malah, and Meir Barzohar
Department of Electrical Engineering, Technion
Haifa 32000, Israel
davidar@campus.technion.ac.il, malah@ee.technion.ac.il, barzoharmeir@gmail.com

Abstract

Local-to global point cloud registration is a challenging task due to the substantial differences between these two types of data, and the different techniques used to acquire them. Global clouds cover large-scale environments and are usually acquired aerially, e.g.*, 3D modeling of a city using Airborne Laser Scanning (ALS). In contrast, local clouds are often acquired from ground level and at a much smaller range, for example, using Terrestrial Laser Scanning (TLS). The differences are often manifested in point density distribution, occlusions nature, and measurement noise. As a result of these differences, existing point cloud registration approaches, such as keypoint-based registration, tend to fail. We improve upon a different approach, recently proposed, based on converting the global cloud into a viewpoint-based cloud dictionary. We propose a local-to-global registration method where we replace the dictionary clouds with viewpoint descriptors, consisting of panoramic range-images. We then use an efficient dictionary search in the Discrete Fourier Transform (DFT) domain, using phase correlation, to rapidly find plausible transformations from the local to the global reference frame. We demonstrate our method's significant advantages over the previous cloud dictionary approach, in terms of computational efficiency and memory requirements. In addition, We show its superior registration performance in comparison to a state-of-the-art, keypoint-based method (FPFH). For the evaluation, we use a challenging dataset of TLS local clouds and an ALS large-scale global cloud, in an urban environment.*

1. Introduction

Local-to-global 3D point cloud registration involves finding the rigid transformation between the reference frame of a local point cloud and that of a large-scale, global point cloud. A global point cloud is a representation of a large-scale scene (e.g., neighborhood, town, or city), acquired from multiple viewpoints, and united into a comprehensive 3D model. On the other hand, a local point cloud is a representation of a smaller environment, contained within the large-scale scene, and is acquired from a single viewpoint, or possibly a small set of nearby viewpoints.

The distinction between these two types of point cloud data is a result of employing different acquisition techniques. For example, typical means of 3D data acquisition of outdoor large-scale environments are Airborne Laser Scanning (ALS), also known as airborne LiDAR (Light Detection and Ranging), or photogrammetry based on aerial imagery. Examples of outdoor local cloud acquisition techniques are Terrestrial Laser Scanning (TLS) and stereo reconstruction. Due to the differences between the local and global clouds acquisition methods, the two types of point cloud data tend to have significantly different properties. For example, airborne and terrestrial scans suffer from inherently different kinds of occlusion, due to the difference in typical scanning angles. In addition, while airborne scans typically better capture horizontal surfaces (such as rooftops) over vertical ones, the opposite is true for terrestrial scans. This often leads to substantially different point density distribution between global and local clouds. See examples of an ALS global cloud and a TLS local cloud in Fig. 1 and Fig. 2, respectively.

Because of these local-vs-global dissimilarities, standard point cloud registration methods, such as keypoint-based registration, tend to fail or require impractical computational resources. Keypoint-based methods rely on detecting locally unique points, in both clouds, and characterizing them, using local 3D descriptors. Then, initial registration is carried out based on finding correspondences between descriptors, often followed by an iterative registration refinement step. However, due to the differences between local and global clouds, the tasks of finding repeatable keypoints in both clouds, and establishing correct correspondences between their descriptors, become much more challenging.

In this work, we propose an approach to solving the local-to-global point cloud registration problem, which

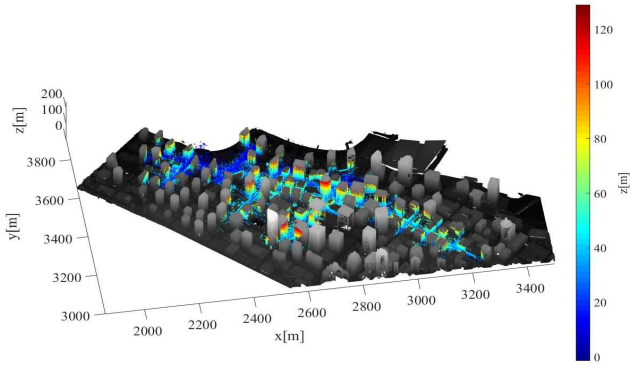

Figure 1. Local-to-global registration result, obtained using the proposed method, between 108 TLS clouds (shown in color), and the ALS cloud (shown in grayscale). Mean localization error is $0.43m$ ($STD = 0.27m$), and the mean RRE is 0.76^o ($STD = 0.37^o$).

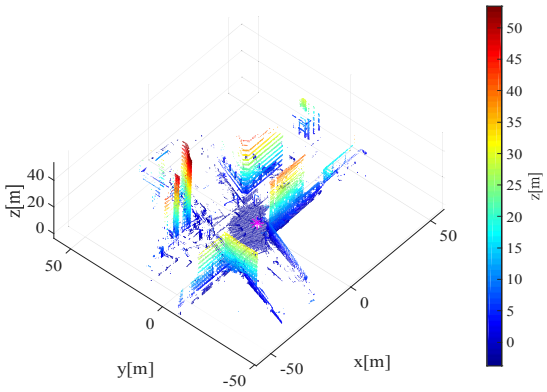

Figure 2. Local cloud example, acquired by a terrestrial LiDAR scanner. The scanner location is marked with a magenta asterisk.

builds upon the method proposed in [1]. As in [1], our approach involves creating a viewpoint dictionary over the global cloud, and solving the local-to-global registration problem through dictionary search. However, while in [1] the dictionary consists of multiple dictionary clouds per viewpoint, our dictionary is comprised of a single panoramic range-image per viewpoint. We show that panoramic range-images can be used to capture discriminative geometric information of the global cloud, with respect to the dictionary viewpoints, such that they enable efficient dictionary search. We show that our method greatly reduces computational complexity and memory requirements in comparison to [1], without loss in registration accuracy.

Our main contributions in this paper are:

1. We introduce the use of viewpoint descriptors within the viewpoint-dictionary based registration framework, proposed in [1]. We demonstrate that replacing the dictionary clouds, used in [1], with panoramic range-images, used as viewpoint descriptors, leads to considerable reduction in memory requirements and computational complexity, without loss in registration accuracy.

2. We propose the use of phase-correlation-based image registration [5, 10], for panoramic range-image matching, and to enable efficient dictionary search and rapid local-to-global initial (coarse) registration, with or without prior knowledge such as GPS data).

2. Related work

2.1. Keypoint-based point cloud registration

One of the most commonly used point cloud registration approaches is keypoint-based registration. Generally, keypoint-based registration methods include the following main steps: keypoint detection, keypoint descriptors computation, establishing keypoint correspondences based on descriptor matching, coarse registration, often based on a variation of RANSAC, and registration refinement, usually with a variation of ICP. Several comparative works have been published in the last few years regarding keypoint detection [14, 21], keypoint descriptors [7], and ICP [16]. The Fast Point Feature Histogram (FPFH) [17], a commonly used keypoint descriptor, has been shown in [7] to be memory efficient, reasonably descriptive and computationally efficient. In [17], multi-scale FPFH descriptors are also used for persistent keypoint detection, by selecting points whose FPFH descriptors are consistently unique across different scales. In the same work, it has also been shown that FPFH descriptors may be used for registration of large-scale TLS scans. We demonstrate that using an FPFH-based registration process, as described in [17], is less reliable for a large-scale local-to-global registration scenario such as TLS-to-ALS registration.

2.2. Line/plane based point cloud registration

A common strategy for large-scale local-to-global point cloud registration is matching sets of linear or planar features. In [8], line-features are detected in both TLS and ALS point clouds, and are used for registration in separate rotation and translation steps. However, it was found in [8] that this resulted in unstable registration results, because line features do not capture the sense of outside vs. inside in a scene (sometimes leading to erroneously placing the TLS sensor inside a building rather than outside). In our viewpoint-grid-based method, this problem is avoided by creating dictionary viewpoints only outside (as described in 4.1). A plane-feature based registration method between

Mobile Laser scanning (MLS) clouds and ALS clouds, was proposed in [20]. Although the method achieves low localization errors, it heavily relies on odometry and GPS for initial registration. Plane-based point cloud registration was also explored by Pathak et al., in [15]. However, their method tackles registration of *sequences* of local clouds, and not local-to-global registration.

2.3. Cloud dictionary based registration

Since our work is based on the principle of using a viewpoint dictionary, we briefly review here the point cloud registration approach proposed in [1], which is based on converting the global cloud into a dictionary of viewpoint-based smaller clouds. Then, local-to-global point cloud registration is carried out via a constrained dictionary search.

A viewpoint grid is created over the global cloud P, in a way that aims to capture possible viewpoints of a sensor, carried by a pedestrian, or mounted on a vehicle moving through the scene. Each viewpoint $\mathbf{v} \in \Re^{3\times 1}$ is placed at a certain height above ground in the direction of the ground's normal vector. For each grid viewpoint $\mathbf{v}$, a part of the global cloud, which includes the points whose distance from $\mathbf{v}$ is smaller than a certain radius r_{max}, is cropped. Let us denote the cropped part of the global cloud as $P_{\mathbf{v}}$. Next, $P_{\mathbf{v}}$ is used to create a set of possibly overlapping dictionary clouds. This is done by defining a set of N_{dir} reference frames $\{O_{\mathbf{v},i}\}_{i=1,...,N_{dir}}$, whose origin is $\mathbf{v}$. The z-axes of the reference frames are all aligned with the local ground's normal vector. The x-axes rotation angles around the common z-axis are evenly divided between $-\pi$ and π. The y-axes are defined using cross-products to define right-hand reference frames. A dictionary cloud is created for each reference frame $O_{\mathbf{v},i}$ by treating the x-axis as a "viewing direction" and by cropping $P_{\mathbf{v}}$ according to a desired horizontal Field-of-View (FoV) angle. Then, the cropped part of $P_{\mathbf{v}}$ is transformed from the global reference frame to $O_{\mathbf{v},i}$. The FoV angle and radius r_{max} of each dictionary cloud are set such that they resemble those of the local clouds. Since each dictionary cloud is stored in its own reference frame, it is possible to compare a local cloud to each of the dictionary clouds without transforming it to numerous poses. It only needs to be transformed once, such that the local cloud viewpoint is translated to the origin, the z-axis is aligned with the local cloud ground normal vector, and the x-axis points in the direction of the middle of the local FoV.

The comparison is done using a Root Mean Square Error (RMSE) criterion over the nearest-neighbor distances from each point in the local cloud to a dictionary cloud. The best matching dictionary clouds (e.g., a small percentage or fixed number of clouds with the lowest RMSE values) are selected as candidate matches to the local cloud. Based on each of these candidates, a coarse registration is computed between the local and global clouds. Then, each coarse registration is refined using the iterative closest point (ICP) algorithm [3] between the local cloud and the corresponding candidate dictionary cloud. Finally, the best registration is selected based on the lowest RMSE score after using ICP.

In this work, we demonstrate that replacing the dictionary clouds of [1] with viewpoint descriptors, *i.e.*, panoramic range-images, greatly reduces the memory requirements of storing the dictionary and also accelerates dictionary search.

3. Viewpoint descriptor

In this section, we introduce several concepts that will be used in section 4, where we describe our proposed local-to-global registration pipeline. These concepts serve the purpose of improving the registration method in [1], by replacing dictionary clouds with viewpoint descriptors. Instead of creating a set of dictionary clouds for each viewpoint, as in [1], we compute a single descriptor per viewpoint. In using such descriptors, our aim is to capture discriminative geometric information of the 3D environment *with respect to a specific viewpoint*. We require the descriptor to allow efficient viewpoint dictionary search, while maintaining low memory usage. Additional desirable properties are robustness to noise, occlusions, and clutter. In sub-sections 3.1 and 3.2 we describe a viewpoint descriptor which fulfills these requirements, and propose an efficient method for matching pairs of descriptors.

As in [1], we also use a grid of viewpoints, spread over the global cloud P, given in a reference frame O, in a way that aims to capture possible viewpoints of the sensor used to acquire the local clouds. Then, For each grid viewpoint $\mathbf{v}$, we define an individual reference frame $O_{\mathbf{v}}$. The origin of $O_{\mathbf{v}}$ is located at the viewpoint $\mathbf{v}$, its z-axis is aligned with the normal vector to the ground in the vicinity of $\mathbf{v}$, and the x-axis direction is chosen arbitrarily. Before computing a viewpoint descriptor for $\mathbf{v}$, a part of the global cloud is cropped to obtain $P_{\mathbf{v}} = \{\mathbf{p}_{\mathbf{v},i} \in \Re^{3\times 1}\}_{i=1,...,N_{pts}}$, as described in section 2.3, where N_{pts} is the number of points in $P_{\mathbf{v}}$. $P_{\mathbf{v}}$ is then transformed to the new reference frame to obtain $\tilde{P}_{\mathbf{v}}$:

$$\tilde{\mathbf{p}}_{\mathbf{v},i} = {R_{\mathbf{v}}}^T(\mathbf{p}_{\mathbf{v},i} - \mathbf{v}), \quad i = 1, ..., N_{pts}, \tag{1}$$

where the columns of $R_{\mathbf{v}}$ are unit vectors corresponding to the axes of $O_{\mathbf{v}}$. Before computing a viewpoint descriptor for a local cloud, similar steps are carried out.

3.1. Panoramic range-images

A possible way to capture discriminative geometric information of a 3D environment, with respect to a certain viewpoint, is creating a panoramic range-image. We describe the creation of a panoramic range-image of a point cloud $\tilde{P}_{\mathbf{v}}$, given in a viewpoint reference frame $O_{\mathbf{v}}$.

The following process is the same for either a part of the global cloud or for a local cloud. The points of $\tilde{P}_{\mathbf{v}}$, $\tilde{\mathbf{p}}_{\mathbf{v},i} = [x_i\ y_i\ z_i]^T$, are converted from Cartesian to spherical coordinates:

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \tag{2a}$$

$$\theta_i = \arctan(z_i / \sqrt{x_i^2 + y_i^2}), \quad \theta_i \in [-\pi/2, \pi/2] \tag{2b}$$

$$\phi_i = \arctan(y_i / x_i), \quad \phi_i \in [-\pi, \pi] \tag{2c}$$

where r_i, θ_i, and ϕ_i respectively represent range, elevation angle, and azimuth of the point $\tilde{\mathbf{p}}_{\mathbf{v},i}$. Note that in our selected spherical coordinates convention, θ is measured from the $z = 0$ plane. A rectangular grid, with angular resolution α, is defined over the (θ, ϕ) space such that the domain $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\pi, \pi]$ is divided into M by N rectangular bins. In each bin, or range-image pixel, we assign the *minimal* range value of all points that fall within. We have found that using the minimum range, which is simple to compute, is suitable for capturing enough geometric information for range-image matching (section 3.2), and dictionary search (section 4). Pixels not containing any points are assigned zero range values. The time complexity of creating a range-image in this way is $O(N_{pts})$. We note that a more accurate method of creating a range-image may be to use ray intersections with a reconstructed mesh. However, doing so based on [13], has not led to a significant improvement in registration results due to the robustness of the proposed phase-correlation-based range-image matching.

We note that instead of using 2D range-images, we have also used skyline-like 1D descriptors with some success. Such 1D descriptors consist of the height values corresponding only to the topmost non-zero pixels in each column of the range-image. However, for increased robustness and descriptive power we opted for the 2D range-images.

Another viewpoint descriptor option is an Extended Gaussian Image (EGI) [9]. EGIs can be used for point cloud alignment via correlation in the Fourier domain, as proposed in [12]. However, computing an EGI requires normal estimation for *all* cloud points, which may be computationally prohibitive for large-scale clouds. In contrast, the proposed method requires normal estimation only for a small set of viewpoints, as described in 4.2.

Examples of panoramic range-images created from a local cloud (terrestrial LiDAR scan) and a corresponding part of the global cloud (created from airborne LiDAR scans), are shown in Fig. 3. It can be seen that the range-image created from part of global cloud contains artifacts ("holes" in building walls), which occur due to the low point density on vertical surfaces in airborne LiDAR scans. As we describe in section 3.2, when we are given a local range-image, the phase-correlation-based method that we use for range-image matching, not only allows us to detect relevant dictionary viewpoints despite these artifacts, but also

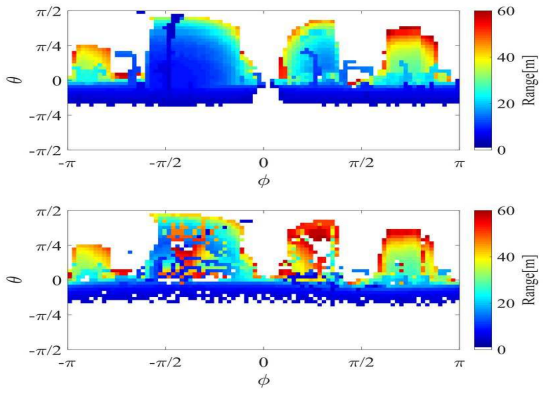

Figure 3. Panoramic range-images created from a local cloud (top) and a corresponding part of the global cloud (bottom). The distance between the two different viewpoints is $1.64m$. Both range-image sizes are 45×90. While the local range-image is relatively smooth, the dictionary range-image contains artifacts such as "holes" in building walls. This is a result of the low point density on vertical surfaces, in ALS scans.

to gain information regarding the necessary alignment from the local reference frame to the global one.

3.2. Phase correlation

As mentioned above, we propose to find matching grid viewpoints to a given local cloud by applying phase correlation between the local cloud range-image and each of the dictionary range-images, within a relevant search area. Image alignment methods based on phase correlation have been shown to have good accuracy and robustness to occlusion and to narrow-band noise [5, 10], especially between images whose registration can be solved by translation. Note that since we use phase correlation between *panoramic* range-images, translation in the image plane corresponds to changes in elevation angle and in azimuth.

We briefly describe the computation of phase correlation between two images, $g_a(m, n)$ and $g_b(m, n)$, where m and n are row and column indices. The 2D Discrete Fourier Transform is computed for each image:

$$G_s(u, v) = DFT\{g_s(m, n)\}, \quad s \in \{a, b\}, \tag{3}$$

followed by the computation of the normalized cross-power spectrum:

$$R(u, v) = G_a^* G_b / |G_a^* G_b| \ , \tag{4}$$

where G_a^* represents the complex conjugate of G_a. Note that the numerator in the fraction above is the DFT of a cyclic cross-correlation between g_a and g_b. The phase correlation is found by computing the inverse DFT of $R(u, v)$:

$$r(m, n) = IDFT\{R(u, v)\}. \tag{5}$$

In order to find the shift between the two images, we search for the peak value of $r(m, n)$:

$$(m^*, n^*) = argmax_{m,n}\{r(m, n)\}. \quad (6)$$

Using the DFT shift theorem, it can be shown that if g_b is a cyclic shift of g_a by integer values $(\Delta m, \Delta n)$, then their phase correlation is a Kronecker delta at $m^* = \Delta m, n^* = \Delta n$, which allows to accurately estimate the cyclic shift between the images. It is known that the time complexity of computing the phase correlation between two images of size M by N, when using radix-2 FFT, is $O(\hat{M}\hat{N}log_2\hat{M}\hat{N})$, where $\hat{M}$ and $\hat{N}$ are the closest powers of 2 that are larger than M and N respectively.

Let us assume the images g_a and g_b are panoramic range-images, created using the same viewpoint and the same reference frame, up to rotation around the z-axis. Then, a horizontal shift Δn between them, is equivalent to rotation between the corresponding reference frames by angle $\Delta\phi$:

$$\Delta\phi = 2\pi\Delta n/N. \quad (7)$$

However, when the viewpoints of each of the range-images are not exactly the same, or if the z-axes of the two reference frames that were used to create them are not exactly aligned, the rotation is only approximately correct.

Panoramic range-images are cyclic in the horizontal direction (azimuth), but not in the vertical direction (elevation angle). However, in typical urban scenes, several upper and lower rows of a range-image, which correspond to looking almost straight up or down, typically do not contain significant geometric information and so can be assigned with zeros. This is similar to multiplying the image with a window function in the vertical direction, which mitigates the effect of the image being non-cyclic in that direction.

Fig. 4 demonstrates the phase correlation between the two panoramic range-images from Fig. 3. It can be seen that despite the dissimilarities between the two images, caused by occlusions, viewpoint difference, and lower point density on vertical surfaces in the global cloud, the peak phase correlation, whose location indicates the shift between the two images, is easily discernible.

4. Proposed registration pipeline

In this section, we describe the overall local-to-global registration pipeline of our proposed method (see block diagram in Fig. 5). We distinguish between two types of algorithmic steps, carried out offline or online. The offline steps are done only once and involve downsampling and ground detection in the large-scale global cloud, followed by creation of the viewpoint dictionary. The online steps are done for each local cloud after it is acquired. These steps include denoising and downsampling, viewpoint descriptor creation, selection of candidate dictionary viewpoints (based on phase correlation results), coarse registration, RMSE verification (whose purpose is rejection of unlikely candidates), and finally, registration refinement. Next, we provide additional details on each of these steps.

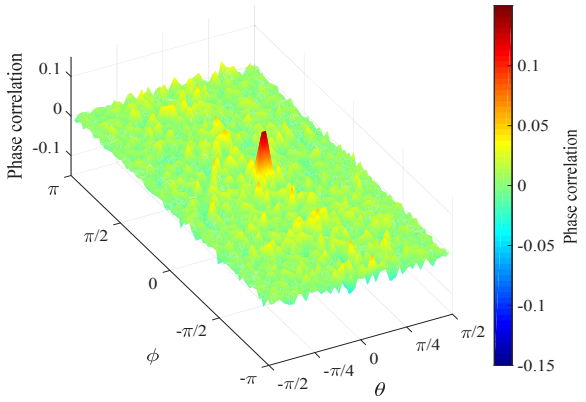

Figure 4. Phase correlation between the two panoramic range-images from Fig. 3. Despite the dissimilarities between the two images, the peak phase correlation, is easily discernible. The peak's position at $(\theta, \phi) = (0.07, 0)$ reflects that the images are quite well aligned.

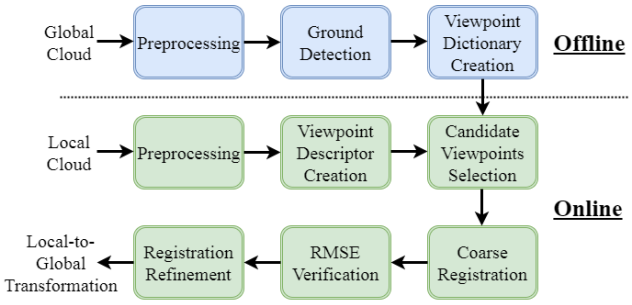

Figure 5. Proposed method block diagram.

4.1. Preprocessing and ground detection

Preprocessing of the global cloud consists here of downsampling, which is done using a voxel grid. We define a regular 3-dimensional, cubic grid over the entire global cloud, with voxel size d^G_{voxel}. In voxels that contain more than a single point, we select one at random, and the rest are removed. Depending on voxel size, it allows for strong downsampling in areas where the global cloud is dense (such as roads or rooftops), and weak downsampling, where it is sparse (*e.g.*, building walls). Preprocessing of the local clouds may involve denoising, if necessary, according to the type of data used. Downsampling is done similarly to the global cloud, using a voxel grid with a voxel size d^L_{voxel}.

Ground detection in the global cloud is done via a region

growing algorithm (such as available in PCL [18]). This allows rejecting viewpoints that are not near the ground (*e.g.*, inside buildings or on rooftops), as described in section 4.2.

4.2. Viewpoint dictionary creation

The first step of dictionary creation is the definition of a viewpoint grid over the global cloud. We create a 2D regular grid of points in x and y, with grid distance d_{grid}. For each point, we find the nearest-neighbor in the global cloud (only in x,y), using a k-d tree [2, 6]. The z coordinate of the nearest-neighbor temporarily defines the z coordinate of the corresponding viewpoint. The height above ground of the local-cloud acquisition sensor is notated as d_{sensor}. Grid viewpoints whose distance to the nearest ground point is larger than $2d_{sensor}$ are removed. For each viewpoint, an upward-facing normal vector of the ground near it is estimated. This is done by cropping an r-neighborhood around the viewpoint from the ground point cloud found in 4.1 (we use $r = 3m$), and fitting it with a plane using the MLESAC algorithm [22]. The normals are oriented upwards such that their z coordinate is positive. Then, each grid viewpoint is moved along its normal vector by this value.

Once the viewpoint grid is defined, the viewpoint descriptors are created. For each grid viewpoint, a panoramic range-image is created as described in section 3.1. The set of these descriptors constitutes the viewpoint dictionary, replacing the dictionary clouds defined in [1]. We note that computing phase-correlation requires only the DFT coefficients of the images, and not the images themselves. Hence, the viewpoint dictionary may contain *only* the DFT coefficients, where their computation is done offline as well.

To demonstrate the advantages of the proposed approach over the method in [1], we compare the proposed dictionary, which contains range-images DFT coefficients, to the cloud dictionary used in [1]. The comparison is done in terms of memory requirements, and in terms of dictionary search time-complexity. It is assumed that each coordinate (x,y, or z) is represented by 4 bytes and each complex DFT coefficient is represented by 8 bytes. We also assume that each dictionary or local cloud contains N_{pts} points. In [1], each viewpoint corresponds to N_{dir} dictionary clouds.

The memory required per viewpoint is $3N_{dir}N_{pts}\times 4$ bytes in [1], in comparison to $2MN\times 4$ bytes in the proposed method. The dictionary search in [1] is based on computing the RMSE over nearest-neighbor distances between the local cloud and each dictionary cloud. Comparing the local cloud to a single dictionary viewpoint consists of N_{dir} RMSE computations. Using a k-d tree, the time complexity of N_{dir} RMSE computations is $O(N_{dir}N_{pts}logN_{pts})$, on average. In the proposed method, the time complexity of comparing a local range-image to a dictionary viewpoint range-image, using phase correlation, is $O(\hat{M}\hat{N}log\hat{M}\hat{N})$ as mentioned in section 3.2. We note that phase correlations between a local range-image and a set of dictionary range-images are independent. Thus, they can be computed in parallel. In Table 1, we summarize the comparison above and substitute the relevant parameters with typical values for demonstration. The overhead of creating the local range-image ($O(N_{pts})$, see section 3.1), is negligible in comparison to computing phase-correlations with the range-images of $N_{gridSearch}$ grid viewpoints within the search area (*e.g.*, $N_{gridSearch} = 200$). It can be seen that the proposed method requires less memory by almost two orders of magnitude and its time complexity is lower by more than an order of magnitude.

Table 1. Comparison between using dictionary clouds ([1]) and the proposed method in terms of memory requirements per viewpoint and of time complexity of checking a match between a local cloud and a dictionary viewpoint. Examples below are shown for typical parameters values: $N_{dir} = 24, N_{pts} = 10^4, M = 45, N = 90, \hat{M} = 64, \hat{N} = 128$.

	Dictionary contents	
	Point clouds [1]	Range-images (proposed)
Memory per viewpoint in bytes	$3N_{dir}N_{pts}\times 4$ $\{2880K\}$	$2MN\times 4$ $\{32.4K\}$
Time complexity	$O(N_{dir}N_{pts}log_2N_{pts})$ $\{N_{dir}N_{pts}log_2N_{pts} \approx 3.19\cdot 10^6\}$	$O(\hat{M}\hat{N}log_2\hat{M}\hat{N})$ $\{\hat{M}\hat{N}log_2\hat{M}\hat{N} \approx 0.11\cdot 10^6\}$

The following steps are done online (see lower part of Fig. 5), for each acquired local cloud, after it is preprocessed, as described in section 4.1, and its viewpoint descriptor is computed, as in section 3.1.

4.3. Candidate viewpoint selection

In this step, we identify candidate dictionary viewpoints, whose descriptors match the viewpoint descriptor of a given local cloud. If a GPS reading is available, we limit the search to grid viewpoints whose distance from the GPS reading is smaller than R_{search}. If a GPS reading, or any other kind of additional information, is unavailable, the entire viewpoint grid is considered. Candidate selection is done by computing phase correlations between the local viewpoint descriptor and each of the dictionary viewpoint descriptors, within the search area. Then, $N_{initial}$ dictionary viewpoints, with the largest phase-correlation peaks are selected as initial candidates.

4.4. Coarse registration and RMSE verification

For each of the initial candidates, we compute a local-to-global coarse registration. This is done by first translating the local cloud such that its viewpoint is moved to the corresponding candidate grid viewpoint. Then, the local cloud

is rotated, using the following rotation matrix:

$$R_{coarse} = R_v^D \cdot R_{PhC}(\Delta\phi) \cdot \left(R_v^L\right)^T, \tag{8}$$

where the columns of R_v^L correspond to the axes of the local viewpoint reference frame, given in the local reference frame, and R_v^D corresponds to the axes of the dictionary viewpoint reference frame, given in the global reference frame. $R_{PhC}(\Delta\phi)$ represents a rotation around the z axes, by angle $\Delta\phi$, derived from the location of the phase-correlation peak. When the local and dictionary viewpoints are the same and their normal vectors are aligned, this rotation is exact. The farther the two viewpoints are, the rotation becomes less accurate. In addition, if the phase-correlation peak is shifted from 0 in the elevation angle (θ) direction, this suggests the normal vectors are misaligned, which may cause an incorrect rotation. Since we next use RMSE verification, such incorrect rotations are likely to be rejected.

Next, we compute the Root Mean Square Error (RMSE), over the nearest-neighbor distances from each of the local cloud points to the global cloud. Only N_{final} candidates, with the lowest RMSE scores are considered in the following registration refinement step. This selection is done as a candidate filtering step, in order to reject unlikely candidates as well as to reduce the computational effort needed in the registration refinement step.

4.5. Registration refinement

Finally, the N_{final} coarse registrations, corresponding to the final candidates, are refined using ICP [3], while rejecting nearest-neighbor pairs whose distance is larger than d_{max}. Alternatively, one of the more recent variations of ICP (*e.g.*, [4]), may be used instead. The final local-to-global registration is selected based on the lowest RMSE score achieved after the refinement.

5. Experimental setup

We tested the performance of the proposed registration method on a dataset of LiDAR-scanned point clouds in a large-scale urban environment. The dataset includes a large-scale global cloud and 108 local clouds. The point cloud data and ground truth transformations were provided by GeoSim [23].

The global cloud was acquired in Vancouver using a Leica ALS80 airborne LiDAR sensor, covering an area of $\sim 0.93 km^2$ (see Fig. 1). After downsampling in the preprocessing step ($d_{voxel}^G = 0.5m$), the global cloud contains about $5.8M$ points. The global cloud is given in a standard global coordinate system (WGS84).

The local clouds were acquired using a Z+F IMAGER 5010 3D Laser scanner. The maximal range of the scanner is $187m$ (we limit local cloud range to $r_{max} = 100m$). The raw data is treated as a panoramic range-image, whose size is 2222×5002. Each raw local clouds contain $2222 \times 5002 \approx 11M$ points. The sensor was mounted on a vehicle at a height d_{sensor} of approximately $2m$ above the ground. For denoising, the median range was computed using a 7×7 filter. A pixel whose range is different from its 7×7 neighborhood median range by more than $0.03m$ is considered invalid and its corresponding cloud point is removed. The local clouds are then downsampled, using a voxel grid ($d_{voxel}^L = 0.25m$), as described in section 4.1. After denoising and downsampling, each local cloud contains, on average, around $80K$ points. Each local clouds is given in its own reference frame, where the sensor is located at the origin. Fig. 2 displays an example of a local cloud.

The provided ground truth transformations between the local reference frames and the global one are found based on differential GPS measurements, refined by a Waypoint software solution [24], using ground control points. For additional details, see [19].

6. Results

We evaluate the performance of the proposed algorithm in terms of registration accuracy and runtime. Registration accuracy is measured by localization and rotation errors. Localization error is defined as the Euclidean distance between the location of the local sensor according to the ground truth, and its location according to the estimated local-to-global transformation. Rotation error is measured by the Relative Rotation Error (RRE) criterion, defined in [11] as the sum of absolute Euler angle errors between the Ground truth and estimated rotation. The time measurements are for a MATLAB® implementation, run on a PC (i7-5820K CPU @ 3.30 GHz, RAM: 64GB). The following reported runtime measurements of registration time per local cloud do not include local cloud preprocessing (denoising and downsampling), since it may vary significantly for different sensors used to acquire the local clouds. The mean denoising time per local cloud, averaged over the 108 local clouds, is $4.9sec$ ($STD = 0.2sec$), and the mean downsampling time is $1.1sec$ ($STD = 0.2sec$).

We have found that it is beneficial to further downsample both the local and global clouds, prior to the RMSE verification and registration refinement steps. Further downsampling, using a voxel size of $d_{voxel} = 2m$, has led to a reduction in registration time per local cloud, with no decrease in registration accuracy.

The following registration results, were obtained while using a fixed set of reference parameters, shown in Table 2. The mean localization error, for the reference set of parameters, is $0.43m$ ($STD = 0.27m$), and the mean RRE is 0.76^o ($STD = 0.37^o$) (see registration in Fig. 1). The maximal localization error and RRE are $1.84m$ and 1.85^o, respectively. The mean registration time per local cloud is $2sec$ ($STD = 0.4sec$) (not including preprocessing time).

Table 2. Reference set of main parameters.

Parameter	Value	Description
d_{grid}	3m	Viewpoint grid distance
α	4^o	range-images angular resolution
R_{search}	30m	Initial candidate search radius around GPS reading
$N_{initial}$	10	Number of initial candidates
N_{final}	3	Number of final candidates
d_{max}	7m	Max. valid nearest-neighbor distance in ICP

For the reference set of parameters, the typical number of viewpoints in a search area is 200.

We have also tested the influence of changing the grid distance d_{grid}, or the angular resolution α, while the other parameters remain fixed. The mean localization error and RRE (as well as their corresponding STDs), remained unchanged for $d_{grid} \in [2m, 5m]$ and $\alpha \in [2^o, 8^o]$. This is the consequence of using ICP, which converges to a good solution from a range of possible coarse registrations.

We evaluated the algorithm's performance, when no GPS reading is available, such that the search for candidates is done over the entire viewpoint grid, containing approximately $43K$ viewpoints. This simulates a one-time scenario where a "blind" initialization is necessary. In order to obtain registration accuracy close to the reference, the number of initial candidates $N_{initial}$ was increased from 10 to 40, and the number of final candidates N_{final} was increased from 3 to 5. The mean localization error, in this case is $0.44m$ $(STD = 0.27m)$, and the mean RRE is 0.78^o $(STD = 0.39^o)$. The maximal localization error and RRE are $1.94m$ and 1.96^o, respectively. The mean registration time per local cloud is $15.4sec$ $(STD = 0.7sec)$.

In comparison to [1], for a subset of 24 local clouds, we achieve similar registration accuracy (localization errors and RRE), while significantly reducing memory requirements and time complexity, as discussed in section 4.2.

We also compared the performance of the proposed local-to-global registration algorithm to that of FPFH [17], using the same subset of 24 local clouds. We selected persistent feature points in both local and global clouds, using multiscale FPFH features, as described in [17]. We used four neighborhood radiuses: 1m,2m,3m,4m. Configurations with doubled or halved radiuses were also tested. The parameter β, that controls the uniqueness of selected feature points in [17], was set to 1.5, which resulted in $\sim$1000 feature points per local cloud. Initial registration was found using the SAmple Consensus Initial Alignment (SAC-IA) algorithm, described in section IV of [17]. The number of iterations used was 2000 and the number of nearest-neighbors in FPFH feature-space (L1-metric), considered for each local cloud descriptor was 30. Finally, registration refinement was performed using point-to-point ICP, similarly to our proposed method.

Using these configurations, at most 6 out 24 local clouds had localization errors lower than $3m$. For the other 18 clouds the registration failed, due to the small number of correct correspondences established between the local and global clouds. We found that establishing correct correspondences between point clouds with very different properties, based on local features, such as FPFH, is unreliable. In contrast, using the proposed method, the maximal localization error over the same 24 local clouds, was $0.79m$).

7. Conclusion

We presented in this work a local-to-global point cloud registration method, based on a viewpoint dictionary which uses panoramic range-images as viewpoint descriptors. An efficient dictionary search method was proposed, using phase correlation between panoramic range-images. It was shown that the use of these viewpoint descriptors have resulted in substantial improvements over [1], that uses a dictionary of clouds. Dictionary memory requirements were reduced by almost two orders of magnitude, and dictionary search time was reduced by more than one order of magnitude, without reducing the registration accuracy. We have demonstrated the robustness of the method to significant difference in properties between local and global clouds, even without using GPS. In terms of runtime, we have shown that the method can achieve local-to-global registration, in a large-scale 3D environment, in a few seconds per local cloud (on a PC, running MATLAB®). We have also shown the advantages of the proposed method over a state-of-the-art feature-based point cloud registration method (FPFH), in a challenging local-to-global registration scenario.

We believe the proposed method could prove useful for other applications such as indoor/outdoor robot localization (*e.g.*, solving the "kidnapped robot" problem), and can be adapted for registration of sequences of local clouds, without a global cloud.

Acknowledgments: This work was supported by the Omek consortium, which is a program of the Israel Innovation Authority of the Israeli ministry of economy. We thank Viktor Shenkar, Yigal Eilam and Ramon Axelrod, at GeoSim, for helpful discussions and for providing us with the point cloud data used in this paper. We would also like to acknowledge the support of Erez Nur, technical manager of Omek. The support of Nimrod Peleg and Yair Moshe of the Signal and Image Processing Lab (SIPL), at the Technion, is greatly appreciated.

References

[1] D. Avidar, D. Malah, and M. Barzohar. Point cloud registration using a viewpoint dictionary, 2016. IEEE International Conference on the Science of Electrical Engineering (ICSEE), Eilat, Israel.

[2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[3] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[4] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. In *Computer graphics forum*, volume 32, pages 113–123. Wiley Online Library, 2013.

[5] H. Foroosh, J. B. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE transactions on image processing*, 11(3):188–200, 2002.

[6] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[7] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016.

[8] H. Hansen W. andGross and U. Thoennessen. Line-based registration of terrestrial and airborne lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B3a):161–166, 2008.

[9] B. K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984.

[10] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method, 1975. IEEE international Conference on Cybernetics and Society.

[11] Y. Ma, Y. Guo, J. Zhao, M. Lu, J. Zhang, and J. Wan. Fast and accurate registration of structured point clouds with small overlaps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.

[12] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3d point clouds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1297–1304. IEEE, 2006.

[13] Z. C. Marton, R. B. Rusu, and M. Beetz. On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[14] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.

[15] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. Fast registration based on noisy planes with unknown correspondences for 3-d mapping. *IEEE Transactions on Robotics*, 26(3):424–441, 2010.

[16] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[17] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[18] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[19] V. Shenkar and Y. Eilam. Point cloud fusion. *U.S. Patent 9,562,971, issued February 7, 2017*.

[20] T.-A. Teo and S.-H. Huang. Surface-based registration of airborne and terrestrial mobile lidar point clouds. *Remote Sensing*, 6(12):12686–12707, 2014.

[21] F. Tombari, S. Salti, and L. Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013.

[22] P. H. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[23] Webpage. Geosim, advanced 3d city modeling technologies, http://geosimcities.com (accessed on March 10, 2017).

[24] Webpage. Waypoint software for accurate gps postprocessing, http://www.novatel.com/products/software (accessed on March 10, 2017).