# Fine scale image registration in large-scale urban LIDAR point sets

Maximilien Guislain, Julie Digne, Raphaëlle Chaine, Gilles Monnier

# Fine Scale Image Registration in Large-Scale Urban LIDAR Point Sets

Maximilien Guislain[a,b,*], Julie Digne[a], Raphaëlle Chaine[a], Gilles Monnier[b]

[a] *LIRIS, CNRS UMR5205, Université Claude Bernard Lyon 1, Lyon - France*
[b] *Technodigit, Hexagon Group, 444 rue des Jonchères, 69730 GENAY - France*

**Abstract**

Urban scenes acquisition is very often performed using laser scanners onboard a vehicle. In parallel, color information is also acquired through a set of coarsely aligned camera pictures. The question of combining both measures naturally arises for adding color to the 3D points or enhancing the geometry, but it faces important challenges. Indeed, 3D geometry acquisition is highly accurate while the images suffer from distortion and are only coarsely registered to the geometry. In this paper, we introduce a two-step method to register images to large-scale complex point clouds. Our method performs the image-to-geometry registration by iteratively registering the real image to a synthetic image obtained from the estimated camera pose and the point cloud, using either reflectance or normal information. First a coarse registration is performed by generating a wide-angle synthetic image and considering that small pitch and yaw rotations can be estimated as translations in the image plane. Then a fine registration is performed using a new image metric which is adapted to the difference of modality between the real and synthetic images. This new image metric is more resilient to missing data and large transformations than standard Mutual Information. In the process, we also introduce a method to generate synthetic images from a 3D point cloud that is adapted to large-scale urban scenes with occlusions and sparse areas. The efficiency of our algorithm is demonstrated both qualitatively and quantitatively on datasets of urban scans and associated images.

*Keywords:* Large Scale Point Sets, Image to Geometry Registration, Image Comparison Metric
*2010 MSC:* 68U05, 65D18

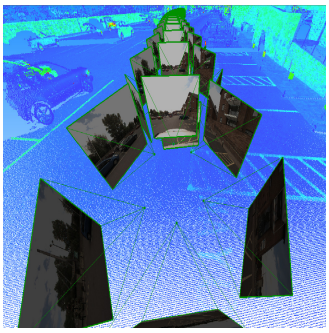
## 1. Introduction

Recent years have seen a fast development of acquisition technologies for acquiring urban scenes. Among all techniques, terrestrial laser scanners have gathered an important research interest. Acquisition campaigns covering whole cities have been led using LiDAR (Light Detection And Ranging) scanners onboard moving vehicles. The output of these campaigns consists in large, potentially unorganized, point clouds representing the buildings measured by the laser. These campaigns are often not limited to acquiring the geometry as a point cloud, but also embed other devices to measure various data. For example, this additional data can be a set of pictures taken at the same time as the points were measured. The set of pictures and the point clouds are usually aligned using onboard information such as GPS information or accelerometers. However, this initial alignment is often flawed which can lead to wrong interpretations in further processings using both points and pictures. This may be due to various factors such as sensor drift and uncertainty or even stability problems in the way the cameras are fixed to the vehicle. This misalignment can be corrected interactively but it is time-consuming and intractable for large point sets and picture sets. It is therefore necessary to devise an automatic way to correct the registration starting from the approximate camera pose given by the acquisition device.

*Data.* In this paper we consider data consisting in urban scenes point sets and associated pictures with approximate camera pose, which we propose to refine to obtain a precise camera position. We assume that all the camera intrinsic parameters are known. Our main test-case is a dataset containing point clouds and associated images acquired in the city of Shrewsbury, UK. It offers a large variety of architecture style (modern or more traditional buildings) and environments (high buildings or less dense areas with parking places) which makes it a good proof-of-concept for our algorithm. The point cloud itself is composed of 260 million points corresponding to the scans of several streets in the city center, as shown on Figure 1a. The whole point cloud itself is correctly registered and coherent along the 2.5km path taken by a moving LiDAR mounted on a vehicle equipped with 7 cameras. The pictures were taken at a regular distance from each other, in 6 directions (see Figure 1b). Along the path, 2452 pictures per direction were taken, yielding a total of 14712 images. The pictures have a resolution of $2046 \times 2046$ and are encoded using

*Corresponding author
*Email address:* maximilien.guislain@liris.cnrs.fr (Maximilien Guislain )

(a) Point cloud



(b) Initial image alignment

Figure 1: Example data, taken using a mobile LiDAR with mounted camera in the town of Shrewsbury (UK)

JPEG.

For the sake of clarity, the term *Real image* will refer to any real world image obtained using a camera. The term *Synthetic image* will refer to the projection of a point cloud using a pinhole camera model.

These real images will be converted to grayscale values in order to be compared to synthetic images. To generate these synthetic images, we will consider either laser intensities when available, or more simply normals of the point cloud. While most registration methods focus on using laser intensities, those are not always available. Indeed, depending on the acquisition device, the output can be limited to a set of unorganized 3D positions with no additional information. In that case, to be able to perform image to geometry registration, one must rely on purely geometric information such as the normals, estimated through Principal Component Analysis. As shown by our experiments, existing methods fail when using this normal information. On the contrary, our method is able to perform the registration using either the laser intensity if it is provided or the normals computed from the points.

The remainder of this paper is divided as follows: first, we review existing methods for image to geometry registration in section 2. Then, an overview of our method is given in section 3, followed by the synthetic image generation procedure (section 4), the comparison metric details (section 5), and our two-step registration method (section 6). Finally, we discuss the results and comparisons in section 7 before concluding (section 8) and exposing our future works.

## 2. Related work

Image to point cloud registration is a domain that was extensively explored in the past few years. Existing approaches can be divided into four categories: *2D feature-based methods*, *statistical methods*, *3D based methods* and *skyline based methods*. The two first categories are the most explored in the literature. They share the common approach to cast the problem of image to point cloud registration as an iterative process of image to image registration. This implies that the point cloud should be turned into a synthetic picture on which the real picture can be registered. Using both images a first camera pose is estimated, a synthetic image is regenerated using this new pose and the process is iterated. The synthetic image can either be obtained directly from the LiDAR scanner which sometimes provide a spherical image, or by projecting the point cloud on the image plane and giving each point a color corresponding to some geometry properties (estimated normals for example).

*2D Feature-based methods.* Feature based registration methods rely on establishing correspondences between feature points obtained using methods such as SIFT [1], or SURF [2] on real and synthetic 2D images. These methods need to be applied on point clouds that already possess either a reliable color information, or a reflectance value as presented in [3], where a complete reflectance image is used to perform a SURF detection and matching.

Similarly, Moussa *et al.* [4] rely on RGB coloration of the point cloud given by the laser to make comparisons between a synthetic image and a real image, using ASIFT [5] descriptors. Inconsistent correspondences are then removed by applying RANSAC [6]. Using the correspondences between 2D and 3D points, the camera pose is finally obtained by solving a Perspective-n-Point (PnP) problem using the EPnP algorithm [7]. Yang *et al.*[8] propose a method to register an image on a shape using another image with a perfectly defined pose. First, SIFT descriptors are computed on the image with a known pose and associated with the point cloud by backprojecting them on the geometry. Real image SIFT keypoints are then computed and compared to the point cloud descriptors and the best matches are kept. Finally a two-step refinement is performed to obtain the camera position. This method does not require any prior estimation of the camera pose, but still needs a real image that has its pose perfectly defined relatively to the point cloud. Gonzalez *et al.*[9] propose a methodology that registers LiDAR range images generated from Terrestrial Laser Scans (TLS) and digital camera images using image descriptors. The real image is preprocessed to remove the distortions and its contrast is increased, followed by radiometric equalization and bilinear interpolation. Then a manual resizing operation is performed on the synthetic image to fit the real image as well as possible. Feature points are detected and matched by combining cross-correlation, least squares matching and epipolar constraints. Finally the camera position and orientation is obtained using RANSAC. However it relies on high definition images to detect common features between LiDAR scans and photos. Furthermore, manual interaction is not possible for large datasets. A method guaranteeing the global optimality of the registration in case of points and lines within indoor scenes has also been proposed [10].

Plötz *et al.* [11] recently described a feature based registration method using the average shading gradients to

successfully register an image onto an untextured mesh object without any prior pose information.

*Statistical methods.* Methods using statistical analysis are widespread for aligning image to image (and thus image to geometry). Among all statistical methods, the most common metric is Mutual Information (MI). Proposed by Viola and Wells [12], MI is a measure of the mutual dependencies between two random variables based on the Shannon entropy. For image registration, the two variables are the pixel intensities of both images. MI measures the similarity between two images based on the level of dependency of the intensity distributions. Thus in order to align an image to another, a good strategy is to find the pose that maximizes their Mutual Information ([13],[14]). Considering image to geometry registration as an iterative image to image registration, Mutual Information can once again be used to measure the quality of the registration. Variations on the original metric have been later proposed: using normalized values (Normalized Mutual Information [15]) or adding SIFT information [13]. Several works have investigated Mutual Information in the context of comparing an image with some geometric information. Corsini *et al.*[16] presented an in-depth discussion about which combination of geometric properties should be used to achieve the best results, e.g. normal maps, intensities or even a mixture of several modalities. Mastin *et al.* [17] successfully used Mutual Information to correct small rotational errors in the registration of urban aerial images on the corresponding aerial Lidar data using elevation and reflectance data of the Lidar. In the case of data acquired by a mobile LiDAR acquisition system, MI has been used to obtain the position and orientation of an image relatively to the point set, using the similarities between images and scanner intensities [18]. Taylor *et al.* [19] propose a calibration framework that estimates the camera pose with no other information than the normal of the scanned points. To do so, a modified form of the Mutual Information is maximized using particle swarm optimization. Although this method gives good results, it suffers from several drawbacks. First, its high memory demand makes it impractical for large complex point clouds such as the ones we consider in the paper. Furthermore, it does not propose a way to cope with multiple depths layers that can be seen from a same viewpoint when the sampling is not dense enough. This problem, discussed in section 4.2, can lead to bad registration results as shown in our experiments (see Figure 15 for example). Another drawback lies in the dependency on panoramic spectral photography, a type of photography which provides a larger area of common information between the point cloud and the image leading to a more robust registration. When applying this method to regular images, as the ones available in our datasets, it does not work as well since regular images have less overlapping information. Taylor *et al.* [20] further improved their method by introducing a gradient based metric called Gradient Orientation Measure (GOM) instead of Mutual Information. GOM computes the difference of the gradient orientation angle between the synthetic image and the real image. This method improves the accuracy of the result compared to Normalized Mutual Information (NMI). To alleviate the computation cost of the synthetic image after each particle motion, another improvement is to use spherical images. However we will show that using a single metric of comparison is generally not discriminative enough and that better results can be obtained by combining several measures to highlight the differences at different scales.

Statistical methods for image to geometry registration are an active field of research. For example, Pascoe [21] recently introduced a Normalized Information Distance metric, based on Mutual Information and entropy variation, to retrieve the camera position in an urban environment.

*3D based methods.* In sharp contrast with the two first categories, some methods propose to use 3D reconstruction and then 3D matching to achieve proper registration. For example, Corsini *et al.* [22] start by performing a Structure From Motion (SFM) reconstruction from an image dataset. SFM is a powerful and widely used method that reconstructs a set of 3D points using a set of images capturing the scene with a small variation in position and orientation. Usually, common features are identified on this set of images using descriptors such as SIFT or SURF. The variations of these identified descriptors allow to reconstruct the descriptor 3D position and thus camera positions. This type of reconstruction is powerful but only gives a small amount of 3D points. The reconstructed points and the relative pose of the images are then fitted to an existing, denser point cloud using a scale independent version of 4-Points Congruent Sets [23]. After merging the sparse point cloud with the complete point cloud, the camera pose is estimated and then refined using Mutual Information maximization. Moussa *et al.* [24] also use Structure From Motion to perform image to geometry registration, but instead of doing a full 3D registration, they register images to the polar laser intensity images that are generated during the scanning process. Correct matches can be obtained from real and synthetic images using standard image descriptors. Thus, the polar image pose is obtained through SFM which finally yields the actual image pose estimate.

*Skyline based methods.* Skyline registration methods aim at retrieving the camera pose by analyzing the uniqueness of the skyline in urban environments. The color differences between the sky and the buildings are used to register single camera images on a corresponding point cloud, starting from an estimated pose. Although this approach is less spread, Hofmann *et al.*[25] proposed to rely on the skyline of the buildings. The sky is first automatically extracted in the real images and independently in the synthetic image using pixel intensity thresholding. The outline is then computed and refined. Extracted skylines in both images

are then merged using a modified ICP method in the image plane. After this fusion, a better camera pose is estimated. These methods are adapted to large scale cities but unfortunately suffer from relying only on the skyline. Hence, every problem on the skyline such as missing data, jagged skyline, or even too much vegetation, is likely to affect the results, or not give any result at all if no building skyline is visible in the images.

Our work focuses on the case of complex, large scale urban scenes. The data is acquired using a mobile LiDAR system, which gives as output clouds of several million points with coarsely registered corresponding images. The images are given by *in situ* CCD perspective cameras, that can be considered as standard digital cameras with a narrow field of view. Despite the fact that some scanners provide interesting properties such as the reflectance for each scanned point, we develop a more general framework that is able to perform the whole registration process using only the geometry data, *i.e.*, the spatial positions. This is interesting when the laser intensity information is not embedded in the data format, or with a view to extending the approach to other acquisition devices. When the synthetic images are based only on the geometry, the descriptor-based methods fail to align our rendered images. Besides, the geometry of urban scenes make it difficult to use 3D based methods since the presence of vegetation, moving vehicles and pedestrians make Structure From Motion methods fail. This would yield a flawed reconstructed point cloud leading to a wrong registration. These methods also require a large number of images taken with a small variation in space. Such data may not always be available. Similarly, Skyline based registration is not always applicable in urban contexts due to the presence of either jagged or partial skyline or even a total lack of skyline in some images. Thus, statistical methods are a better choice, but in our urban context the Mutual Information objective exhibits a highly non convex profile because of the sparsity of the synthetic images. To be able to minimize this objective, one could resort to the proposed solution of [20] to perform particle swarm optimization. Unfortunately this method does not provide a way to cope with occlusions arising from multiple scenes layers, such as the ones described in section 4.2.

## 3. Overview

Our method, summarized in Figure 2, takes as input a point cloud of a urban scene and a corresponding picture with initial pose estimate and known intrinsic parameters. Such an image can be acquired during the scanning process by a camera mounted on a vehicle or from a standard digital camera handled independently. We propose a two-step registration method to refine the camera pose, knowing the camera intrinsic parameters and a reasonable initial pose estimation. First a wide angle synthetic image is generated and used to optimize the camera pose with 3 degrees of freedom (optimizing only for a rotation). This rotation

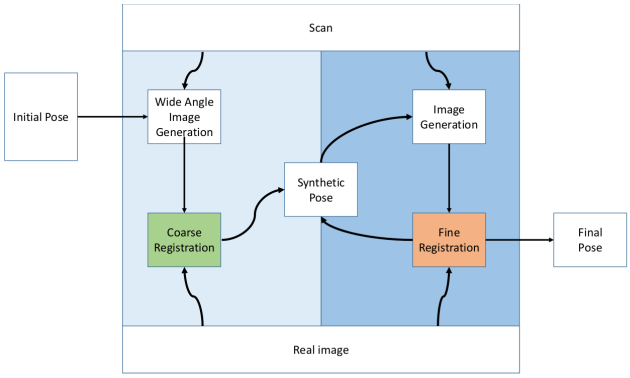



Figure 2: Overview of our method

estimation is performed, in a multiscale fashion thanks to a new metric, called MIDHOG, that combines Normalized Mutual Information and Histogram of Oriented Gradients (HOG) descriptors to measure the consistency of the real image with a part of the wide angle synthetic image.

Starting from the refined orientation, the fine registration step gradually performs a full 6 degrees of freedom pose estimation. During this second step, two strategies are available: either using the MIDHOG metric, to ensure the accuracy, or to replace it only with DHOG, to improve the computation time at the cost of losing some precision.

To summarize, our contributions are:

- A process to generate synthetic images from a point cloud and a camera pose, adapted to large scale urban scenes, even when no reflectance information is available at each point.
- A new image comparison metric, more robust to incomplete image data and large pose transformation.
- An efficient iterative pose estimation method using our new metric to obtain a good estimation of the rotation, alleviating the registration problem for important rotations.

## 4. Synthetic image generation for large scale urban scenes

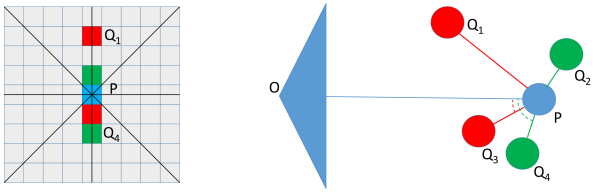
When casting the image to geometry registration problem to an iterative image-to-image registration problem, one must generate synthetic images of the geometry as seen by the camera at the given pose. If the camera pose corresponds to the pose that yields the real image, the metric comparing both images should be minimum. Here we address the synthetic image generation problem when dealing with a large and complex point cloud.

### *4.1. Projection model and color information*

To generate an image from the point cloud, a pinhole camera model is adopted. Assuming that the intrinsic parameters of the camera are known, as well as the distor-

(a) Point cloud projection (b) Occlusion estimation (c) Interpolation

Figure 3: Different steps to generate the synthetic image. The pixel intensities are computed using the points normals.




(a) Image view (b) Side view

Figure 4: Projection of 5 points onto an image. The considered point and its projection on a pixel are depicted in blue. The visibility angle is the angle $< \vec{OP}, \vec{QP} >$. An horizon pixel is a pixel corresponding to the point with the smallest angle in a sector. These pixels are depicted in red, they are the points that best occlude the central point. Other pixels are in green.

tion parameters (3 radial and 2 tangential) of the Brown-Conrady camera distortion model [26], the 2D coordinates of a point that is projected from the ambient space onto the image can be obtained.

Yet this step only yields a binary image: pixels are lit if there is a projected information and off if there is none. Therefore, a color should be assigned to the projected points. In our work, we either use the laser intensity provided by the scanning device or a grayscale information which is deduced solely from the geometry and more precisely from the point cloud normals that are estimated using PCA [27]. In case the normals are used, they must be turned into grayscale values. To do so several options are available. Taylor *et al.*[19] propose to use the scalar product between the normal and the up axis to reflect the fact that most of the luminosity comes down from the sky. However such lighting conditions, although they may be adapted to natural environment, give bad results in urban environment where the walls, which are usually perpendicular to the ground, offer a large variety of details that should be taken into account. Instead of using a direction coming from the sky, one can use any other arbitrary direction and a natural choice would be one that enhances the details. In our experiments, the best results are achieved using a lighting direction from the center of the camera. In that case, the computation of each point grayscale intensity breaks down to using the absolute value of the cosine angle between the camera direction and the point normal. This choice enhances interesting details on the geometry that would not appear as efficiently using other coloration methods. Besides, it can happen that several points coming from different surfaces project onto the same pixel. It can be due either to a sampling density higher than the pixel size or to the fact that points from several surfaces (eg. buildings in different streets) can be seen through the foreground surface, since there is no watertight model and since we are dealing with large scale aggregated scans. In that case a choice should be performed to decide which piece of surface occludes the other by keeping the point that is closest to the camera position.

However, if there is not enough points on the closest surface, sparse sampling artifacts will appear as shown in Figure 3a. The next steps focuses on overcoming this limitation.

### *4.2. Reducing the sparse sampling artifacts*

Since the method operates on large point clouds created from multiple scans, a simple projection produces visual artifacts on sparsely sampled areas that occlude each other. It is possible to get rid of these artifacts in pinhole images, following the method proposed by Pintus *et al.*[28] which we summarize briefly:

For each pixel $p$ corresponding to a 3D point $P$, the region in a $l \times l$ neighborhood around it is divided into 8 sectors (usually $l = 9$). In a sector, for each 3D point $Q$ corresponding to a pixel $q$ of this sector, we define the visibility angle relative to the pixel $q$ as the angle between the vector $\vec{OP}$ and the vector $\vec{QP}$ (see Figure 4). The sum of the smallest angles for each sector is considered as the solid angle of visibility for the pixel $p$.

If the solid angle is larger than a threshold value $\psi$ the central pixel is classified as being visible. The value $\psi = 2.0sr$ is used in the remainder of this paper. This method successfully removes points that should not have been visible, as shown in Figure 3b.

It can be noted that points may remain visible through large holes such as windows, as shown in Figure 3b. Although these points have no real geometric meaning, they still induce important visual changes in most cases leading to a correct registration.

### *4.3. Interpolation*

If the image is taken too close to the surface, a lot (if not most) of pixels do not have any information available, even in the case of a dense point cloud. To deal with missing information in the generated image, one has to retrieve information for undefined pixels. A simple bilinear interpolation on the available data, as proposed in [9], leads to a widening of the edges (Figure 5), which could cause registration inaccuracy. To avoid that widening we propose to divide the neighborhood of the considered pixel into 4 sectors, and the interpolation is performed only if at least 3 of these 4 sectors contain pixels with data. In some cases this may leave a lot of undefined pixels, but this is a good trade-off between edge location preservation

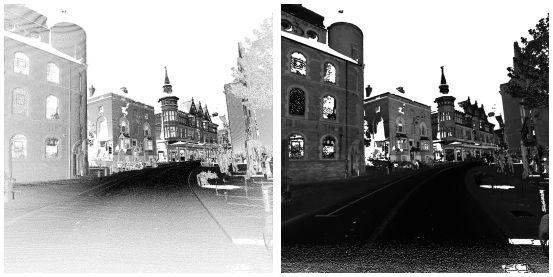
| | No interpolation | Splatting | Bilinear | Ours |
|---|---|---|---|---|
| Accuracy (pixels) | 41.89 | 23.25 | 21.49 | 20.05 |
| Standard deviation | 109.63 | 16.55 | 7.57 | 6.98 |
| Success ratio (%) | 73 | 82 | 82 | 89 |
| Time (s) | 40.61 | 263.46 | 58.65 | 56.11 |

Table 1: Average error in pixels observed after different kinds of interpolation. These results were obtained using coarse registration only from 45 images. See section 7 for evaluation methodology details

and information addition. In our implementation we used a neighborhood of $5 \times 5$ pixels, a choice that reveals much needed details in the generated images, without affecting the edges of the scene (Figures 3c, 5 and table 1).

The influence of our edge-preserving interpolation scheme on the whole registration process is evaluated in Table 1. Performing a registration without any interpolation produces a significantly higher residual error. The gain of using a border preserving interpolation method further improves the accuracy at a very small additional computation cost. Even if this improvement is not as big in average as one could expect, it has proven to be useful in some cases (as illustrated in Figure 6).

Surface splatting is an alternative to produce synthetic images which is robust with respect to sparse sampling [29]. However, using a simple surfel rendering, as described in [30], also yields a widening of the edges (Figure 5c). Furthermore it generates higher levels of noise when the surfel orientation is not well defined (such as in the trees).



(a) No interpolation (b) Bilinear interpolation

(c) Splat based rendering (d) Edge preserving interpolation (ours)

(e) Bilinear (detail) (f) Splatting (detail) (g) Edge preserving (detail) (ours)

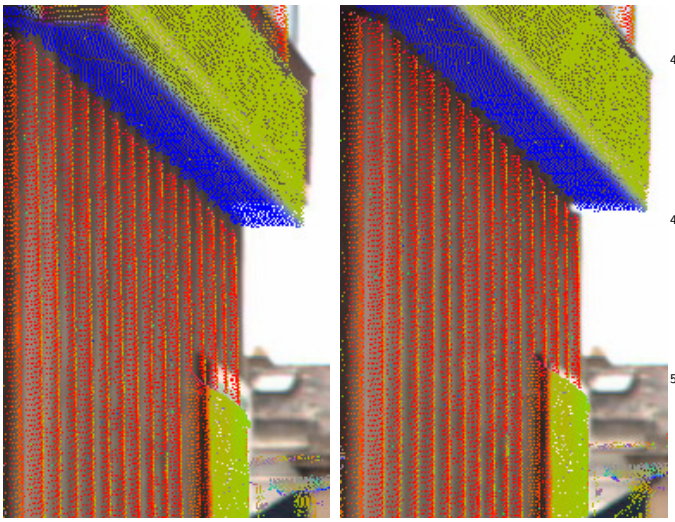
Figure 5: Effect of the bilinear interpolation versus an edge preserving bilinear interpolation. As can be seen in these images, our proposed edge preserving interpolation improves the data density sufficiently to give an idea of the image entropy, while preserving the details localization compared to a classical bilinear interpolation, whereas other methods give dense images but at the cost of an edge dilatation. The original data without interpolation is shown in magenta.

## 5. Robust comparison of synthetic and real images

Our method heavily relies on 2D images comparison. We must therefore define an image comparison metric, to determine if two images represent the same scene. In our case, this metric should be resilient to noise and incomplete data. This is even more important since the modality is not the same in both images. Indeed, in our case the real image encodes the color information while the synthetic image encodes an intensity derived from point normals or point reflectance. In this section we present a new way to compare images, building on two existing approaches, Normalized Mutual Information (NMI) and Histogram of Oriented Gradients (HOG), which are detailed below.

### *5.1. Normalized Mutual Information*

MI, as introduced in section 2, is widely used for image registration based on image comparison even in the case of different modalities, as stated by Kim *et al.* [14]. We focus here on Normalized Mutual Information (NMI), a modified version of MI proposed by Studholme [15]. This normalized version ensures that the MI values are bounded. This



(a) Bilinear interpolation (b) Proposed bilinear interpolation

Figure 6: Details of a registration of the same region, using bilinear interpolation, or our edge preserving bilinear interpolation.

also allows to compare images when the amount of known data varies.

The NMI between two images $I_1$ and $I_2$ is defined as:

$$\mathrm{NMI}(I_1, I_2) = \frac{H(I_1) + H(I_2)}{H(I_1, I_2)} \quad (1)$$

where $H(I)$ is the image entropy defined as

$$H(I) = \sum_i p_i log(\frac{1}{p_i}) = \sum_i -p_i log(p_i) \quad (2)$$

and $p_i$ is the probability for a pixel $k$ of image $I$ to be of intensity $i$ (*i.e.* $p(I(k) = i)$). Hereafter, $N$ is the total number of pixels in the image $I$ and $T(\cdot = \cdot)$ equals 1 if $\cdot = \cdot$ is true, and 0 otherwise. Undefined pixels in the synthetic images are not considered in the NMI calculation. The probability $p_i$ is defined as follows:

$$p_i = \frac{1}{N} \sum_k T(I(k) = i). \quad (3)$$

Let $p_{(m,n)}$ be the joint probability that pixel $k$ has an intensity $m$ in image $I_1$ and $n$ in image $I_2$. The joint entropy $H(I_1, I_2)$ is defined as

$$H(I_1, I_2) = \sum_m \sum_n -p_{(m,n)} log(p_{(m,n)}) \quad (4)$$

Image to image registration using NMI is efficient in most cases, giving a measure varying between 1.0 for no mutual information to 2.0 for two identical images. However, due to the amount of missing data in our synthetic images, NMI sometimes exhibits important flaws. Among others, non-convex profile of this measure might lead to an error in the maximization process yielding a wrong registration, or the global maximum might not correspond to the actual image superposition (see Figure 8). This can be explained by the fact that NMI, and MI in general, take the whole image into consideration. We propose to re-introduce some spatial locality in the analysis instead. Interestingly, another attempt at localizing MI was proposed in pixel-wise mutual information [13], but in case of images using only normal information this metric exhibits a highly nonconvex profile making it impossible to recover a good registration. Our proposed approach works differently as it combines NMI with local gradient histograms.

### *5.2. Distance between Histogram of Oriented Gradients*

In this section, we introduce a metric based on the spatial distribution of intensity gradients called *Distance between Histogram of Oriented Gradients* (DHOG). It corresponds to a localized integration of distances between local Histogram of Oriented Gradients (HOG) that we briefly describe.

#### *5.2.1. HOG*

HOG, introduced in [31], is a feature descriptor characterizing image areas using their gradient information. HOG is widely used to compare and match images [32], or to detect objects in images.

The Histograms of Oriented Gradients of an image can be obtained by computing the gradients on the whole image. Then the image is divided into regularly sized patches called cells. Orientation-based histograms are then computed within each cell. Each pixel in each cell contributes to one bin of the histogram with a weight depending on the magnitude of the gradient. Once a histogram has been obtained within each cell, the cells are grouped by blocks. For each of those blocks, a normalization factor is computed. The blocks overlap to produce resilience to illumination and contrast change. Therefore we have $n_{blocks} \times n_{cells\,per\,blocks}$ normalized histograms.

#### *5.2.2. DHOG*

HOG is usually computed on sliding windows, to detect known size patterns in an image. However, here we consider whole images, with fixed cells position. Let us consider two images $I_1$ and $I_2$ on which we compute the cells histograms of oriented gradients as described previously. To quantify the similarity between these two images, we integrate the square distance between each corresponding pair of histograms.

Since we operate on an inaccurate projection model, it is better to favor image similarity in areas that are less subject to distortion. In pinhole camera models, radial distortions affect the borders of the image, rather than the image center. Using a weighted sum of squared differences between HOG will give more importance to the registration error near the center of the image, and help the registration even when the image distortions are not well defined.

Besides alleviating the bad calibration, this weighting scheme also increases the registration accuracy: on the 45 images groundtruth, it improved the registration accuracy by 3 pixels and the registration success ratio by 4% in average.

Denoting $wb$ and $hb$ the image width and height in numbers of blocks, we define the weight $w_{B_{ij}}$ of a block $B_{ij}$ centered at coordinates $(i, j)$ as:

$$w_{B_{ij}} = \exp - \frac{(i - \frac{wb}{2})^2 + (j - \frac{hb}{2})^2}{(\frac{wb}{2})^2 + (\frac{hb}{2})^2} \tag{5}$$

This weight will be close to 1 around the image center, and will decay as the considered blocks are closer to the picture's border.

Due to the particular structure of the HOG data, given a real image $I_r$ and a synthetic image $I_s$ this integration is a function of the values $v^{I_r}_{cbij}$ and $v^{I_s}_{cbij}$ of a HOG bin $b$ in a cell $c$ belonging to a block $B_{ij}$ (located at coordinates $i$, $j$):

$$\text{DHOG} = \frac{\sum_{i,j} \sum_c \sum_b (v^{I_r}_{cbij} - v^{I_s}_{cbij})^2 \times w_{B_{ij}}}{\sum_{i,j} w_{B_{ij}}}. \tag{6}$$

When applied on images with only a few texture, such as normal based synthetic images, DHOG performs much better than NMI. However on images with a lot of textures, NMI gives more accurate results. Thus, by combining NMI and DHOG, we are able to overcome the failure cases of both as illustrated in Figure 8.

### *5.3. MIDHOG*

As shown in Figure 8, the metric variation of NMI and DHOG are different for the same transformation. Interestingly their defects appear in different cases. Based on this observation we combine NMI and DHOG so that a proper rough registration can be achieved where either one of the metric would tend to drift to a wrong position. MIDHOG is based on the dissimilarity of the images, a value of 1.0 represents two images where gradients are opposite from one another, which is rather uncommon. On a set of 20 images we observed that NMI values usually varies between $[0.87, 0.96]$ whereas DHOG varies between $[0.033, 0.058]$. Combination gives best results using a simple addition of the components (see equation 7), DHOG being weighted by a parameter $\alpha$.

$$\text{MIDHOG} = (2.0 - \text{NMI}) + \alpha \cdot \text{DHOG} \tag{7}$$

MIDHOG inherits the properties of both MI and DHOG, it is zero when the two compared images are totally identical and it is symmetrical.

An error study of the coarse registration error compared to our ground truth shows that an $\alpha$ value between 5 and 20 gives similar and satisfactory results (see Figure 7) whereas relying too much on NMI fails to properly register the images. On the other hand, increasing the weight of DHOG too much produces unsatisfactory registration in some images. A good trade-off was obtained using $\alpha = 10$ which is used in the remainder of this paper.

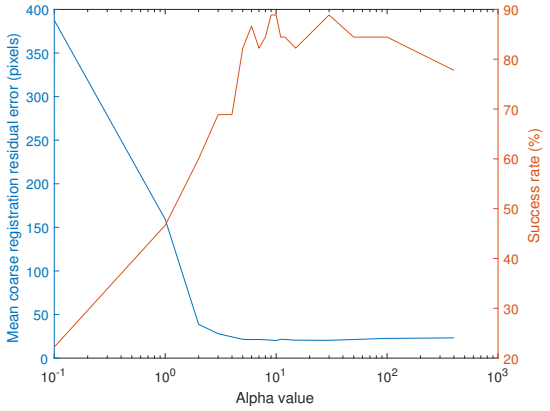
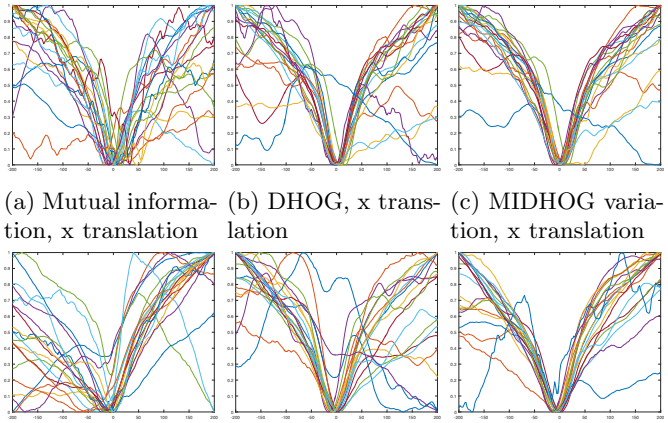



Figure 7: Average error after a coarse registration for different choices of $\alpha$ values for 45 groundtruth images with random initial disruptions.



(a) Mutual information, x translation (b) DHOG, x translation (c) MIDHOG variation, x translation

(d) Mutual information, y translation (e) DHOG, y translation (f) MIDHOG variation, y translation

Figure 8: Variation of three image comparison metrics: NMI, DHOG and MIDHOG. The top row corresponds to a per pixel translation on the horizontal axis of the subimage in a wide angle image (see Figure 9), and the bottom row corresponds to a translation in the vertical axis.

## 6. Method

Given an input point cloud data and a real image with initial pose estimate $\Omega$, our goal is to find a refined pose estimate $\Omega'$. We introduce a two step registration consisting in a fast but coarse registration, optimizing only for the rotation, followed by a slower fine-scale registration optimizing for both rotation and translation.

### *6.1. Wide angle image registration*

A first and fast registration step is performed by generating a *wide angle* image of the point cloud from the initial camera pose and by refining 3 degrees of freedom on the location of this pose. For the sake of clarity, an overview of this coarse registration step is given in Algorithm 1.

**Algorithm 1:** Coarse image registration step

**Data:** $\Omega$: initial pose
$\mathcal{P}$: camera intrinsic parameters
$I_R$: real image
**Result:** $\Omega_1$ a coarsely estimated camera pose
$scale = 1/4$ ;
PixelMotion = 0;
$\Omega_1 = \Omega$ ;
**while** $scale \leq 1/2$ **do**
  $I_S$ = Generate synthetic wide image using $\mathcal{P}$, $\Omega_1$ and $scale$;
  $(\delta x, \delta y, \theta)$ = Minimize MIDHOG between $I_R$ and $I_S$ using an initial step of 1/20 of the image pixel size;
  $\Omega_1$ = Obtain 3D pose from triplet $(\delta x, \delta y, \theta)$;
  PixelMotion = $\max(\delta x, \delta y)$;
  **if** *PixelMotion* $\leq 10$ **then**
    $scale = 2 \times scale$;

The rationale behind this first step is that a single wide-angle image can be substituted to several steps of regular-size synthetic image generation. A small pitch and yaw rotation or translation of the pose will only marginally distort the pixels but will affect the position of the image center in the image plane. Thus a small pitch and yaw rotation or translation of the pose can be approximated by a small translation in the wide-angle image plane as depicted in Figure 9. As far as the Roll Rotation is concerned it corresponds to a rotation around its center in the wide-angle image plane. Instead of generating a new synthetic image after each small motion, a single wide-image is thus generated and its sub-images are considered as good approximates of smaller images after a *small* viewpoint change. As a side-effect, it will also produce smoother metric variations than the one observed when performing a 3D rotation of the camera.

This approximation can hold for both small rotations and small translations, however we observed experimentally, during manual image registration, that the error in the image registration are mostly due to rotations. Therefore we omit the translation in this coarse step and optimize only for the rotation (3 degrees of freedom). To be even faster, we do not optimize for all rotations as real rotations but rather approximate the Yaw rotation by a translation along the $x$ axis in the image plane and the Pitch rotation by a translation along the $y$ axis in the image plane. The last rotation, the Roll, is computed as a rotation in the image plane around the image center.

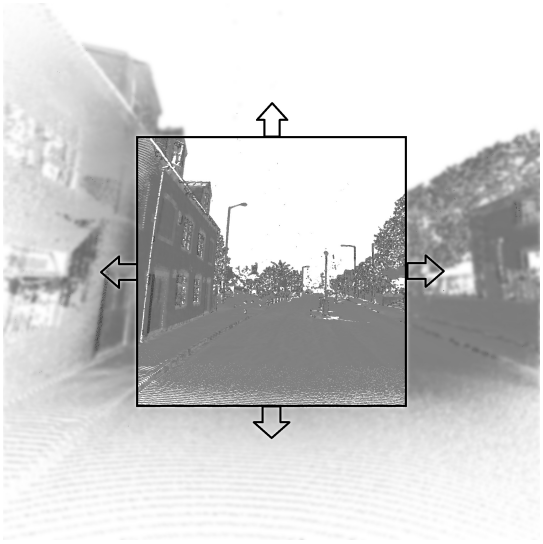


Figure 9: A small pitch and yaw rotation or translation of the pose can be approximated by a small translation in the wide-angle image plan.

To generate a wide angle image, the internal parameters of the camera are kept identical to those given as input, but the size of the captor and the resolution of the image are increased. In the following explanations, images were generated using a double sensor size and resolution. Starting from the wide image, the sub-image that would correspond to an image generated with the standard parameters is extracted. Applying a translation in the $X$ and $Y$ direction of the image is, if the variation is small enough, very close to doing a real world rotation (see Figure 9). This way, we approximate the image synthesis without having to regenerate a synthetic view of the point cloud. By performing a match limited to 3 degrees of freedom ($x$-$y$ translation and a $\theta$ rotation around the central pixel), an approximation of the image with a *pseudo-pose*, i.e. a pose in the image plane, can be obtained quickly.

Unfortunately, for larger camera motions the hypothesis does not hold. To cope with this problem, we iterate this step several times, regenerating a wide angle image after each step, or if the estimated change is superior to 10 pixels. This leads to images with less deformation after each iteration, allowing for a real convergence toward the metric minimum. To further improve both the computation time and the convergence of the method, we perform this wide angle image registration at different scale levels.

A Gaussian pyramid is first built from the image. Then, the smoothest image is considered. Indeed, the details of the image are smoothed out as is the noise, leading to a smoother cost function easier to minimize. Once this first minimum is found, the next level of the pyramid is considered and the objective is once again minimized starting from the pose found at the previous level. Each iteration provides thus a better accuracy by increasing the resolution of the image and re-generating the view while taking into account the pose robustly estimated from the previous iteration. If the resulting estimated plane translation is too large, the assumption that the translation in the panoramic image plane approximates a real-world rotation is not valid anymore. Therefore the image is re-generated at the same scale and the process is repeated.

The quality of the registration is evaluated using our MIDHOG metric. To find the camera pose minimizing MIDHOG (Eq. 7), we use the BOBYQA algorithm [33] and stop when the pose variation is small enough (2 pixels in our implementation). BOBYQA is a deterministic, derivative-free optimization algorithm that relies on an iteratively constructed quadratic approximation. It shows the same kind of flaws as the method presented by [34], such as the difficulty to overcome local minimums, as pointed out by Taylor *et al.*. A better alternative would be to use Particle Swarm Optimization with a meaningful number of particles but it would become computationally intractable. Fortunately, in our case the local minimum problem that prevents the use of BOBYQA, is smoothed by the multi-scale approach proposed in this coarse registration.

For each iteration, we consider the two vector $\vec{dir}(0,0,f)$ and $n\vec{dir}(\delta_x,\delta_y,f)$, the original view direction and the modified view direction respectively, with $\delta_x$ and $\delta_y$ the translation found in pixels and $f$ the focal length in pixels. We define $\vec{d_{yaw}}$ and $\vec{d_{pitch}}$ projections of $n\vec{dir}$ on the image horizontal and vertical plane passing through the image optical center. Using these vectors we can determine the rotations $\omega$ (yaw) and $\phi$ (pitch) corresponding to the pseudo-pose estimation using the equations 8 and 9. The roll is itself not considered as a translation in pixels, but is estimated by performing a rotation of the pixels in the image plane around the transformed image central axis.

$$\omega = -sgn(\delta x)atan(\|\vec{dir} \times \vec{d_{yaw}}\|_2, \vec{dir} \cdot \vec{d_{yaw}}) \quad (8)$$

$$\phi = -sgn(\delta y)atan(\|\vec{dir} \times \vec{d_{pitch}}\|_2, \vec{dir} \cdot \vec{d_{pitch}}) \quad (9)$$

The result of this coarse step is a modified pose $\Omega_1$, that will be further refined in the following fine registration step.

### *6.2. Image to geometry fine registration*

Having obtained a first estimation of the pose efficiently, the fine scale registration consists in estimating the real pose $\Omega'$ not far from the coarse estimation $\Omega_1$ by considering the full 6 degrees of freedom. Despite the non convex form of the similarity metric with respect to the pose, we can find a satisfactory local minimum since $\Omega_1$ is close to $\Omega'$. For that, we rely once again on the BOBYQA algorithm [33] to perform the derivative free minimization. Similarly to the coarse step, the MIDHOG metric defined in equation 7 allows for a better camera pose estimation, especially if the synthetic image is sparse. However, if the priority is given to the computation speed at the risk of losing some precision, it is safe to drop the NMI component of MIDHOG to rely solely on DHOG. This increases drastically the processing speed. Interestingly, this substitution can be done relatively safely only in the fine registration step since the search is limited to a narrow band around the pose $\Omega_1$ found in the coarse registration step.

Contrarily to Taylor *et al.* [20], we do not need to perform particle swarm optimization since the first step has given an approximation *close to the global minimum*, where the metric behaves like a smooth convex function. This leads to a much lower computation time.

To recap our two step method, Table 2 lists the differences between the coarse and the fine registration steps.

| | Coarse Registration | Fine Registration |
|---|---|---|
| Cost function | MIDHOG | MIDHOG (precision) / DHOG (speed) |
| Multi-resolution | Yes | No |
| Parameters | Rotation (3DoF) | Rotation + Translation |
| Parallax | No | Yes |

Table 2: Comparison of the differences and similarity between the coarse and the fine step of the presented method. We do not consider any parallax in the coarse registration step, as we do not modify the camera position, but first try to determine the best viewing angle of the scene.

## 7. Results and Discussions

Our method was tested on the Shrewsbury dataset described in the introduction. This set contains both point clouds and associated images. Several places of the city with different architecture style and environment were selected to observe the performance of the method. For each of these places we singled out an image and run the two-step registration. The point cloud was automatically cut using a bounding box in a large area around the selected camera initial pose to limit the memory impact. Our tests were run on a laptop (Intel Core i7 2.7GHz CPU, NVIDIA Quadro K3100M), with approximately 100 million points processed at a time. Of course, larger point clouds can be loaded at once if enough memory is available. A first preprocessing step was performed on the real images to convert them from RGB to grayscale images using the stan-

dard Luma rec 601 conversion. The whole method was implemented in C++ using the NLOPT [35] library for the BOBYQA algorithm to minimize MIDHOG.

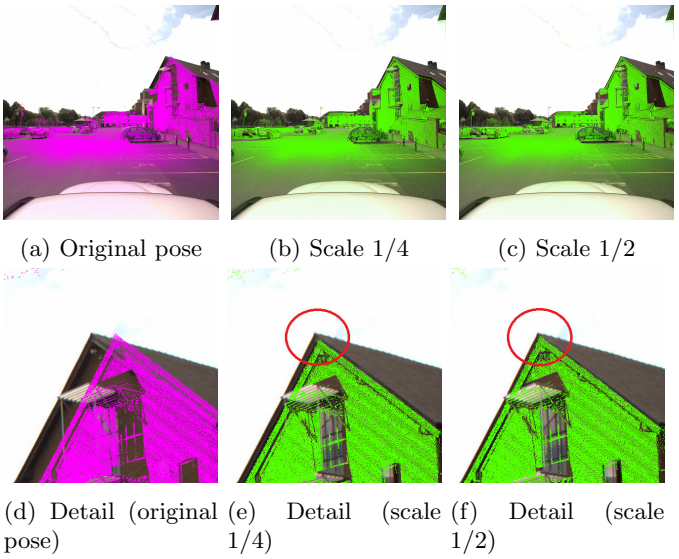
To evaluate the performances of the algorithm, we built a special groundtruth dataset using manual registration. First a series of 45 images was randomly selected among all pictures taken along the path of the vehicle mounted LiDAR. These images were then rectified using the given intrinsic parameters and manually registered to the point cloud by selecting 20 to 40 corresponding points both in the images and on the geometry. This dataset of 45 registered images are considered as an acceptable ground truth. However it should be noted that even if a special care was taken to select relevant common points, the registration in some images is still imperfect and may eventually lead to small disturbance. Besides the rectified images are only an approximation of the undistorded reality and may contain residual errors that might impact the quality of the registration. To evaluate quantitatively our proposed method, a perturbation was applied to each groundtruth camera pose, consisting of a random uniform variation up to 5° in both yaw and pitch, up to 2° in roll and up to $10cm$ in $X$, $Y$ and $Z$. This yields a set of images with perturbed camera pose that can be registered to the pointset using our method. Since the perturbation is known, the quality of the registration can now be evaluated with respect to the manual groundtruth. Setting up this groundtruth led us to the observation that the average registration error is around $6cm$ with errors up to $30cm$ in the estimation position and around 1° in yaw and pitch with peaks up to 4°. Errors in roll are always smaller than 1°.

### 7.1. Coarse image to geometry registration

The panoramic image registration is an important step since it generates an excellent rotation approximation for a very small computation time. To compute DHOG, we used square cells of constant $32 \times 32$ pixels size, with unsigned gradients and 9 bins in the histograms. Blocks were composed of $4 \times 4$ cells and their $\ell^2$-norm was used as the normalization factor. For this coarse registration step, we found that scales 1/4 and 1/2 are enough to get a good registration approximation. For MIDHOG minimization, we used a $x$ and $y$ step starting at 1/20 of the image size in pixels at the finest scale. For the roll $\theta$, since the initial error is in general much lower (less than 1°), a small step of 0.6° is used.

Figures 10 and 11 present registration results obtained solely using the coarse registration method. It appears clearly on all these figures that the coarse registration improves the camera pose estimation compared to the original registration. Detailed analyses are provided for each of these figures in the next paragraphs.

*Council street.* The first example set is acquired near the town council and offers generic city geometrical properties, with square shaped buildings, and low angle rooftops. Figure 10 shows the evolution of the registration through the mutiscale registration steps. Our method performs also well in areas with missing building pieces or jagged skyline (Figure 10a). A different view of the same location (Figure 11a) offers different challenges such as missing rooftops, missing wall parts and occluding shadows around the foreground elements such as the fence. While the details in Figure 11b show an improvement after the coarse registration step, the registration is only roughly accurate.



(a) Original pose (b) Scale 1/4 (c) Scale 1/2

(d) Detail (original pose) (e) Detail (scale 1/4) (f) Detail (scale 1/2)

Figure 10: Coarse Registration comparison on Council street data, with initial registration error of 2.4° (yaw) and 0.5° (pitch). Magenta color is the original registration and green color is the coarse registration results at different scale. The computation took around $60s$.

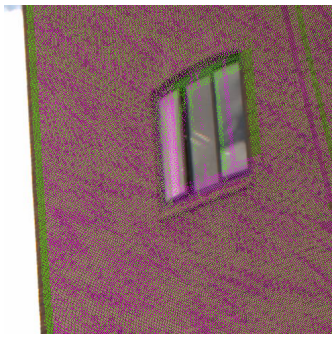
*Castle street.* This is a complex geometry area with high angle rooftops, two moving vehicles and some pedestrians away from the acquisition vehicle. Results of the coarse registration are less accurate than on council street (Figure 11b), however, the pose estimation has clearly improved compared to the initial pose, as visible in Figure 11e, and in Figure 11f for larger input errors.

*Shopping street.* The third example is a narrow city street environment with high buildings around the street and no access to any skyline (Figure 11c). Since the surfaces are close to the camera position, the point cloud density in the image plane in this area is low. We can observe here that the improvement provided by the coarse registration is marginal (Figure 11c and Figure 11d), mostly due to the fact that we are already close to the best possible solution. The coarse step provides thus a fast pose approximation, but there is still a residual error (Figure 11e and Figure 11f) that will be reduced in the next step.

### 7.2. Fine image to geometry registration

Although the coarse registration might look visually satisfactory (figures 10, 11a and 11c), a closer inspection

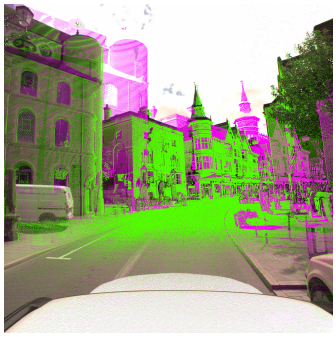
(a) Coarse Registration - Council Street



(b) Detail - Council Street

(c) Coarse Registration - Shopping Street

(d) Detail - Shopping Street



(e) Coarse Registration - Castle Street

(f) Detail - Castle Street

Figure 11: Different effects of the coarse registration method using an input image taken from a lateral camera. Original registration (magenta) and coarse registration (green).

(a) Coarse

(b) Coarse+fine

(c) Coarse (detail)

(d) Coarse+Fine (detail)

Figure 12: Details of the registration on a part of Castle street, for coarse registration only, or for coarse and fine registration. The improvement of the registration with the fine method is clearly visible around the street lights.

on the details of the image/point cloud superposition reveals that the coarse registration still produces important errors. These errors are particularly visible in figures 11b and 11d. In those cases the fine registration step improves drastically the registration, as shown on Figure 12. When comparing the results of the coarse only registration against the coarse plus fine registration, one can clearly see an improvement in the fitting of the image to the point cloud.

As explained in section 6, the fine scale registration step should be performed with MIDHOG leading to a precise registration. However it comes at the cost of higher computation times. In order to alleviate this drawback it is possible to drop the NMI part of MIDHOG and thus to use solely DHOG, possibly yielding larger residual errors but faster computations. Table 3 compares the accuracy, computation times and successful registration ratio of both alternatives. The success ratio is obtained by considering the registration to be successful when the registration error is below a threshold (25 in our case). The remaining error is the error computed on the images considered as correctly registered. As expected the computation time is improved by using only DHOG with a gain of about $40s$ in average. However, when relying only on normals, the overall registration quality suffers from the unique use of DHOG, whereas using MIDHOG as the image comparison metric leads to an average registration error 4 pixels lower

| | Error | Std | Time | Ratio | Remaining Error |
|---|---|---|---|---|---|
| Original disruption | 123.57 | 40.82 | N/A | N/A | N/A |
| NMI (normals) | 448.94 | 633.3 | 302s | 31% | 20.05 |
| NMI (reflectance) | 24.25 | 77.23 | 585s | 91% | 8.56 |
| DHOG (normals) | 20.74 | 20.35 | 498s | 89% | 15.34 |
| DHOG (reflectance) | 15.25 | 21.05 | 572s | 98% | 12.28 |
| MIDHOG (normals) | 16.77 | 10.77 | 541s | 93% | 14.85 |
| MIDHOG (reflectance) | 15.25 | 21.05 | 597s | 98% | 12.28 |

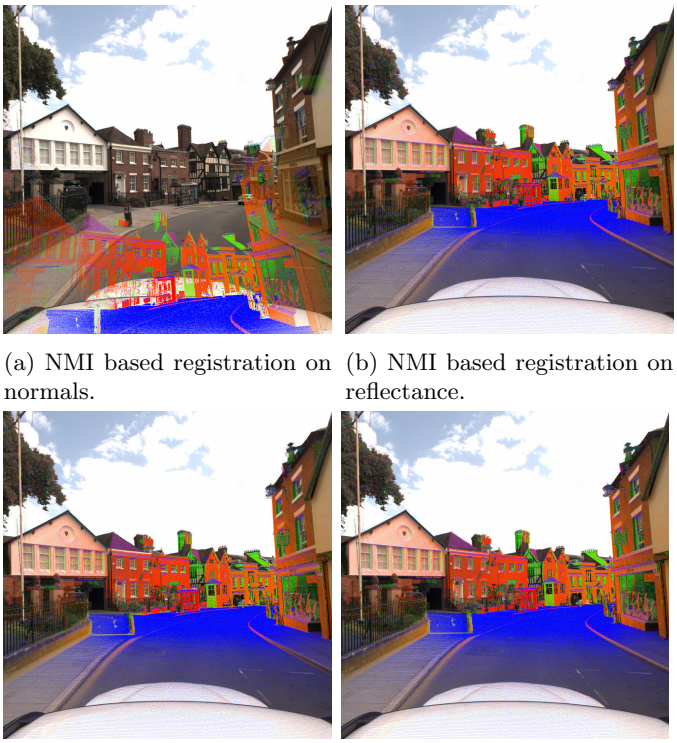
Table 3: Comparison of the average error in pixels, standard deviation, convergence time, successful registration ratio and remaining error on the successful registration case for 45 images using either NMI, DHOG or MIDHOG as image comparison metric. The remaining error is the error computed on the images considered as correctly registered. Lines containing (reflectance) mark are based on the reflectance values rather than on the normal value for the metric calculation. Using the reflectance values leads to a major improvement in the results quality.

and a higher success rate. When using reflectance information both MIDHOG and DHOG give similar results.

All previous experiments were run by considering synthetic images created from estimated normals or reflectance. In the difficult case of normal based synthetic images, we showed that our proposed metric was able to perform the registration while other methods failed. Yet LiDAR devices can provide an additional information: the reflectance of the laser. This information produces synthetic images that can be compared more easily with real images, and most methods work on this type of modality. Table 3 also shows that when using the reflectance information for synthetic image generation, our method outperforms state-of-the art methods in terms of successful registration.Using the groundtruth described in section 6, the registration efficiency was compared using NMI with either simple geometrical information or reflectance values and the same comparison was done using MIDHOG. As can be seen in table 3, using NMI without the reflectance values had a huge impact on the final registration. Mastin *et al.*[17] observed that for aerial scans and photo, the use of reflectance only marginally improved the registration. However in our case the reflectance values, if available, improve greatly the final registration. This statement is also true but less spectacular when using MIDHOG, in which case the reflectance values improve the final registration only slightly. The NMI registration based on the normals fails to properly register the image and the point cloud. On the other hand, MIDHOG and NMI based on the reflectance give similar and satisfactory results. MIDHOG based solely on the estimated normals also gives satisfactory results, but is sightly less accurate than using the reflectance.

| | $x(m)$ | $y(m)$ | $z(m)$ | $\omega(^\circ)$ | $\phi(^\circ)$ | $\kappa(^\circ)$ | pixels |
|---|---|---|---|---|---|---|---|
| Normals | 0.051 | 0.061 | 0.058 | 0.516 | 0.745 | 0.458 | 16.6 |
| Reflectance | 0.056 | 0.060 | 0.058 | 0.344 | 0.401 | 0.401 | 9.74 |
| KITTI | 0.082 | 0.055 | 0.057 | 1.031 | 0.115 | 0.458 | 14.29 |

Table 4: Average registration error of our two-step method compared to the ground truth for 45 random images and for the KITTI dataset. $x, y, z$: translation, $\omega, \phi, \kappa$: Euler angles; and error measured in the image (in pixels). N is a Normal based registration, R is a reflectance intensity based registration, KITTI is the result on the KITTI dataset (normals based).



(a) NMI based registration on normals. (b) NMI based registration on reflectance. (c) MIDHOG based registration on normals. (d) MIDHOG based registration on reflectance.

Figure 13: Different metrics used for the registration based either on the normals or on the reflectance values of the point cloud.

Table 4 gives the average errors on the groundtruth dataset for a complete MIDHOG registration using either the normals or using the laser reflectance. These results were computed with randomly generated errors, different than the ones present in table 3, which explain the slightly different residual error values.

We also applied our registration method to the KITTI dataset [36]. This dataset was obtained using a Velodyn LiDAR and co-registered camera. However, the Velodyn LiDAR outputs a point cloud that covers only a fraction of the space around the moving vehicle (see Figure 14). When projecting the point cloud on the image plane, the captured geometry covers only roughly half the image, and points located too far from the Lidar are also not acquired

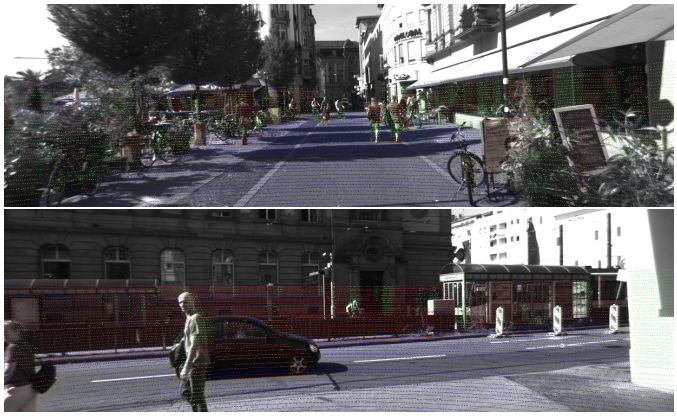


Figure 14: Projection of the point cloud on two of the images of the Kitti dataset containing image/scan pairs. The Velodyne LiDAR point cloud is sparse, and its scanning height is limited, which only offers a small amount of corresponding data.

at all. This is a huge difference with our data where point clouds have neither height nor depth limit, since the scans were previously merged and consolidated. To assess the registration quality of our approach, a process similar to the one described in the beginning of section 7 was applied to the KITTI data. Considering one image/scan pair at a time, a random error between $\pm 5°$ was applied in pitch and yaw, and a random error $\pm 0.1m$ was added to the position. Our two step registration was applied on drive set number 71. It appears that due to the nature of the data, the registration of a single image/scan pair does not give satisfactory results. In this particular case, the method proposed by [19] appears to work better, and is applicable due the low size of image and the low size of the point cloud. However, if we consider several image/scan pairs at once, in a similar way to [18] to compute MIDHOG, our method successfully registers the images to the scans as shown in table 4 (last row).

The two-step registration method exhibits several advantages compared to a direct 6 degrees of freedom registration. One of these advantages is its resilience to important rotations. Indeed, as can be seen in table 5 , applying directly a 6 degrees of freedom pose estimation in a non optimal environment yields far larger errors. This can be explained by the sparse nature of the images, errors in the point cloud and missing data. Clearly, the two-step registration outperforms a single step registration. As visible in tables 6 and 7, our method can handle rather large input errors with acceptable accuracy results. However errors above 15° tend to be too high to be reliably overcome.

### *7.3. Comparisons*

We compared our approach with two recent works for registering images on a point cloud. The first one is the original Taylor algorithm [19] based on Normalized Mutual Information and the second one is the GOM metric of the same author [20]. Comparisons were run on a subset of our real dataset, around Castle street, limited to 16

| | Error (pixels) | Std | Time | Ratio | Remaining Error |
|---|---|---|---|---|---|
| Original disruption | 123.57 | 40.82 | N/A | N/A | N/A |
| NMI (fine only) | 169.93 | 204.12 | 550s | 7% | 28.9 |
| NMI (coarse + fine) | 448.94 | 633.3 | 302s | 31% | 20.05 |
| DHOG (fine only) | 107.83 | 54.50 | 366s | 11% | 12.69 |
| DHOG (coarse + fine) | 20.74 | 20.35 | 498s | 89% | 15.34 |
| MIDHOG (fine only) | 104.78 | 56.99 | 435s | 17% | 12.36 |
| MIDHOG (coarse + fine) | 16.77 | 10.77 | 541s | 93% | 14.85 |

Table 5: Comparison of the average error, standard deviation, convergence time, successful registration ratio and remaining error for 45 images using either NMI, DHOG or MIDHOG as image comparison metric. The remaining error is the error computed on the images considered as correctly registered.. The results obtained using the fine step only have largely worse results than the one obtained by the full coarse plus fine registration. All these data were obtained using the normal information of the point to compute the metric.

| Angle (degrees) | 3 | 6 | 9 | 12 | 14 | 17 | 20 | 23 |
|---|---|---|---|---|---|---|---|---|
| Success ratio (%) | 100 | 72 | 65.4 | 47 | 46.6 | 27.3 | 30 | 0 |

Table 6: Registration success ratio when applying a random disruption around a random rotation axis for 12 images. the displayed angle is the angle between the viewing direction of the original and disturbed camera, therefore even a rather small angle can represent important pitch, yaw and roll disruption.

| Pitch Yaw | 5° | 10° | 15° | 20° |
|---|---|---|---|---|
| 5° | 100% | 66% | 50% | 25% |
| 10° | 83% | 66% | 50% | 33% |
| 15° | 66% | 58% | 33% | 25% |
| 20° | 41% | 16% | 16% | 16% |

Table 7: Registration success ratio when applying a yaw and pitch disruption for 12 images. Ratio obtained for original disruptions lesser than 10° are quite acceptable, however, original disruptions superior to 15° dramatically decrease the registration quality.

Million points due to the memory limitation of the Matlab implementation. First, the GOM method does not address point visibility problems which leads to areas blurred by inconsistent information superposition, as shown on Figure 15a, influencing the algorithm convergence. This test was run with particle swarm of $0.5m$ variation in translation, a $2.5°$ range in yaw, pitch and roll. Figure 15 shows that using the GOM metric does not lead to a proper registration.The GOM metric is not robust enough to resist to sparsity and missing parts of the synthetic images. The sparsity and the missing parts of the generated image disturbed the metric too much and actually prevented the registration. A similar test was run using the NMI metric (Figure 15b) and, once again, we can observe a failure to properly register the image on the point cloud.

Other tests were run on different scenes, applying the same $\pm 0.1m$ and $\pm 5°$ of search range. This is illustrated by Figure 16 where the original image appears in magenta and the point cloud projection appears in green.Results show once again that both NMI and GOM fail to properly register the image. Tests based solely on NMI failed to register properly (figures 16a and 16f), even when using the reflectance data (figures 16b and 16g). Tests using GOM metric also failed to register properly the image (figures 16c, 16h and 16i, except when using the reflectance value (Figure 16d), which yields an acceptable result, whereas our algorithm yields a good results in all cases (Figure 16e and 16j).

## 8. Conclusion

We presented a process to register efficiently and accurately an image in a large scale point set starting from an initial position. This method proceeds in two steps: a coarse registration step and a fine scale registration step. Both steps play an important role: the coarse registration step makes the process more resilient to a bad initial pose, while the fine scale registration step permits to obtain very low registration errors. Both steps rely on a synthetic image generation adapted to large urban point clouds. A key feature of our work is that we propose an alternative for the generation of synthetic images from the sole geometric information when no reflectance information is available. We also proposed a new robust image comparison metric adapted to the comparison of a real image with a synthetic image resilient to large pose transformation. As a future work, the method can be further improved by integrating the re-estimation of the image distortions during the fine registration step which would lead to a better distortion model and even more precise results.

## Acknowledgment

The authors acknowledges support from ANRT, PhD grant $n°2014/0319$.

## References

[1] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[2] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: Computer vision–ECCV 2006, Springer, 2006, pp. 404–417.

[3] S. Shahzad, M. Wiggenhagen, Co-registration of terrestrial laser scans and close range digital images using scale invariant features, Allgemeine Vermessungs-Nachrichten 117 (6) (2010) 208–212.

[4] W. Moussa, M. Abdel-Wahab, D. Fritsch, An automatic procedure for combining digital images and laser scanner data, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 39 (2012) B5.

[5] J.-M. Morel, G. Yu, Asift: A new framework for fully affine invariant image comparison, SIAM Journal on Imaging Sciences 2 (2) (2009) 438–469.

[6] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[7] V. Lepetit, F. Moreno-Noguer, P. Fua, Epnp: An accurate o (n) solution to the pnp problem, International journal of computer vision 81 (2) (2009) 155–166.

[8] G. Yang, J. Becker, C. V. Stewart, Estimating the location of a camera with respect to a 3d model, in: 3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on, IEEE, 2007, pp. 159–166.

[9] D. González-Aguilera, P. Rodríguez-Gonzálvez, J. Gómez-Lahoz, An automatic procedure for co-registration of terrestrial laser scanners and digital cameras, ISPRS Journal of Photogrammetry and Remote Sensing 64 (3) (2009) 308–316.

[10] M. Brown, D. Windridge, J.-Y. Guillemaut, Globally optimal 2d-3d registration from points or lines without correspondences, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2111–2119.

[11] T. Plotz, S. Roth, Registering images to untextured geometry using average shading gradients, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2030–2038.

[12] P. Viola, W. M. Wells III, Alignment by maximization of mutual information, International journal of computer vision 24 (2) (1997) 137–154.

[13] M. Gong, S. Zhao, L. Jiao, D. Tian, S. Wang, A novel coarse-to-fine scheme for automatic image registration based on sift and mutual information, Geoscience and Remote Sensing, IEEE Transactions on 52 (7) (2014) 4328–4338.

[14] J. Kim, V. Kolmogorov, R. Zabih, Visual correspondence using energy minimization and mutual information, in: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE, 2003, pp. 1033–1040.

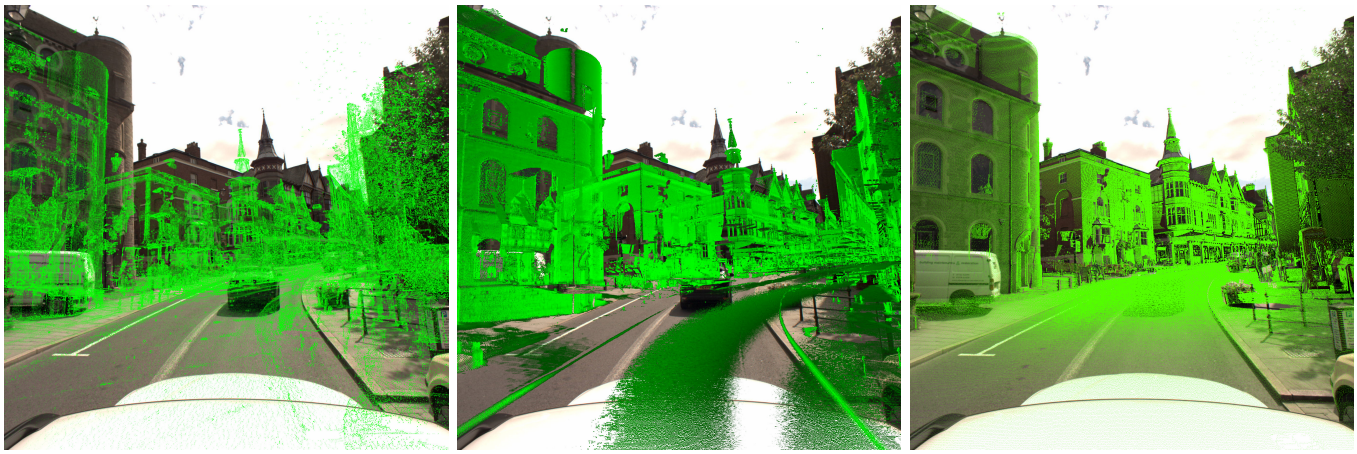
[15] C. Studholme, D. L. Hill, D. J. Hawkes, An overlap invariant entropy measure of 3d medical image alignment, Pattern recognition 32 (1) (1999) 71–86.

[16] M. Corsini, M. Dellepiane, F. Ponchio, R. Scopigno, Image-to-Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties, Computer Graphics Forum 28 (7) (2009) 1755–1764.

[17] A. Mastin, J. Kepner, J. Fisher, Automatic registration of lidar and optical images of urban scenes, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2639–2646.

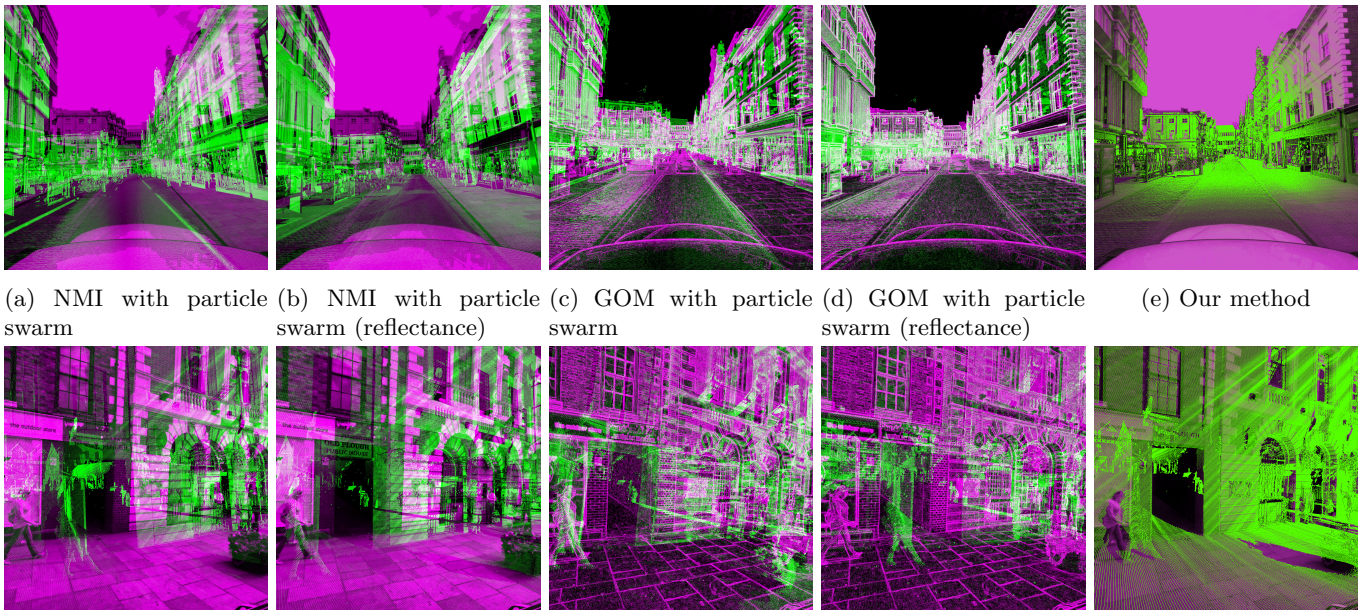
[18] G. Pandey, J. R. McBride, S. Savarese, R. Eustice, Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information., in: AAAI, 2012.

[19] Z. Taylor, J. Nieto, Automatic calibration of lidar and camera images using normalized mutual information, in: Robotics and Automation (ICRA), 2013 IEEE International Conference on, 2013.



(a) GOM method [20] (40 minutes) (b) NMI method [19] (62 minutes) (c) Our two-step method (10 minutes)

Figure 15: Different registration results using various techniques on a subset of the point cloud. Our method clearly leads to a good registration whereas other methods fail. The high amount of noise and artifacts, characteristic of complex urban scenes, coupled with the lack of occlusion may be the origin of this registration failure.



(a) NMI with particle swarm (b) NMI with particle swarm (reflectance) (c) GOM with particle swarm (d) GOM with particle swarm (reflectance) (e) Our method

(f) NMI with particle swarm (g) NMI with particle swarm (reflectance) (h) GOM with particle swarm (i) GOM with particle swarm (reflectance) (j) Our method

Figure 16: Different metrics from Taylor *et al.*[19, 20] used with a particle swarm optimization. Registration with NMI are clearly misaligned. GOM based on the estimated normals also lead to wrong registration. However using the reflectance values combined with GOM clearly gives acceptable results in one case (16d) whereas our algorithm yields a good results in all cases.

[20] Z. Taylor, J. Nieto, D. Johnson, Automatic calibration of multi-modal sensor systems using a gradient orientation measure, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, 2013, pp. 1293–1300.

[21] G. Pascoe, W. Maddern, P. Newman, Robust direct visual localisation using normalised information distance, in: British Machine Vision Conference (BMVC), Swansea, Wales, Vol. 3, 2015, p. 4.

[22] M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, A. Fusiello, R. Scopigno, Fully automatic registration of image sets on approximate geometry, International journal of computer vision 102 (1-3) (2013) 91–111.

[23] D. Aiger, N. J. Mitra, D. Cohen-Or, 4-points congruent sets for robust pairwise surface registration, in: ACM Transactions on Graphics (TOG), Vol. 27, ACM, 2008, p. 85.

[24] W. Moussa, Integration of digital photogrammetry and terrestrial laser scanning for cultural heritage data recording, Ph.D. thesis, University of Stuttgart (2014).

[25] S. Hofmann, D. Eggert, C. Brenner, Skyline matching based camera orientation from images and mobile mapping point clouds, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 1 (2014) 181–188.

[26] D. C. Brown, Decentering distortion of lenses, Photometric Engineering 32 (3) (1966) 444–462.

[27] K. Pearson, Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572.

[28] R. Pintus, E. Gobbetti, M. Agus, Real-time rendering of massive unstructured raw point clouds using screen-space operators, in: F. Niccolucci, M. Dellepiane, S. P. Serna, H. Rushmeier, L. V. Gool (Eds.), VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, The Eurographics Association, 2011.

[29] M. Zwicker, H. Pfister, J. Van Baar, M. Gross, Surface splatting, in: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, 2001, pp. 371–378.

[30] M. Botsch, A. Hornung, M. Zwicke, K. Kobbelt, High-quality surface splatting on today's gpus, in: Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings, IEEE, 2005, pp. 17–141.

[31] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 886–893.

[32] A. Shrivastava, T. Malisiewicz, A. Gupta, A. a. Efros, Data-driven visual similarity for cross-domain image matching, ACM Transactions on Graphics 30 (6) (2011) 1.

[33] M. J. Powell, The bobyqa algorithm for bound constrained optimization without derivatives, Tech. Rep. DAMTP 2009/NA06, University of Cambridge (2009).

[34] A. Mastin, J. Kepner, J. Fisher, Automatic registration of lidar and optical images of urban scenes, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2639–2646.

[35] S. G. Johnson, The nlopt nonlinear-optimization package, http://ab-initio.mit.edu/nlopt.

[36] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, International Journal of Robotics Research (IJRR).