

# Integrating LIDAR Range Scans and Photographs with Temporal Changes

Brittany Morago\*

Giang Bui\*

Ye Duan

University of Missouri-Columbia  
209 Engineering Building West  
Columbia, MO 65211

duanye@missouri.edu

## Abstract

*Registering 2D and 3D data is a rapidly growing research area. Motivating much of this work is the fact that 3D range scans and 2D imagery provide different, but complementing information about the same subject. Combining these two perspectives leads to the creation of accurate 3D models that are texture mapped with high resolution color information. Imagery can even be obtained on different days and in different seasons and registered together to show how a scene has changed with time. Finding correspondences among data captured with different cameras and containing content and temporal changes can be a challenging task. We address these difficulties by presenting a contextual approach for finding 2D matches, performing 2D-3D fusion by solving the projection matrix of a camera directly from its relationship to highly accurate range scan points, and minimizing an energy function based on gradient information in a 3D depth image.*

## 1. Introduction

Registering 2D and 3D imagery can create an enlightening venue for visualizing photographs with complete depth information and conveying spatial relationships among different images. 3D information in the form of LIDAR (Light Detection and Ranging) range scans shows users the full structure of a scene and allows them to navigate through a virtual environment to gain a new perspective. Photographs of the same location can reveal high resolution details on surfaces and specific data about a scene at one moment in time. When these two modalities are combined, 3D models are created that present all of this information at once. A series of photos taken over different seasons can be registered with a range scan and displayed together to show how the scene has changed over time. Combining data from different seasons can even be used to create 3D models of

a scene under conditions when it would not be feasible to collect a LIDAR scan, such as during a blizzard. The time consuming scan can be taken under more favorably conditions, photographs can quickly be captured in the snow, and they can be later registered to simulate a 3D representation of a winter scenario.

No matter what the purpose of the registration process may be, determining the 2D-3D correspondence can be described as a pose-estimation problem [11]. Solving for a camera's pose in relation to a 3D model has been explored in a variety of contexts with some using 3D range scan models [9, 15, 18, 21] and others using structure from motion (SfM) point clouds [4, 13, 19, 20, 24]. Every one of these methods requires defining and matching some type of features either exclusively in 2D or from 2D to 3D. Our work is focused on the approach of matching sets of 2D data, some of which are pre-registered with a 3D point cloud, and creating a 2D-3D link. Specifically, we work with photographs captured by both a LIDAR range scanner and various regular cameras. Images can be aligned for this task using information from locally defined feature descriptors [20], regional descriptors, or direct methods that rely on global pixel-pixel correspondences [27]. Locally defined keypoints use data such as gradient direction and magnitude from neighboring pixels within a relatively small window size whereas regional descriptors use information over a larger portion of the image to describe an area and potentially match it to corresponding regions in other images. SIFT [10] is a commonly used and robust local feature detector used for 2D matching that assigns a scale and orientation to each interest point. A descriptor is constructed based on local image gradients that is generally unique to a single area in the image. The neighborhood size used to construct the descriptor is usually around 16x16 pixels at the appropriate image scale.

The relatively small window size used by locally defined methods to create keypoint descriptors can be beneficial in many cases. They will be able to identify locally distinct areas accurately. However, cases where image pairs have

\*The first two authors contributed equally to this work.

large scale, lighting, and/or content changes may require a larger window size to correctly pinpoint very distinct key-point matches since the images will have so many visual differences. This is often the case when we start matching imagery obtained by very different cameras and in different seasons. We encounter these difficulties when working with LIDAR camera images that have large focal lengths and limited exposure values. This combined with our desire to match outdoor images obtained over the span of an entire year has motivated us to develop a more robust and contextual approach to image matching and 2D-3D alignment.

### 1.1. Related Work

Many groups have focused on 2D-3D registration and have generally followed the common framework of taking existing 3D information, such as a range scan, and fusing images and videos with it [9, 15, 18, 21]. This body of work can be broken down into two main modes of performing 2D-3D registration. One common way is to directly match 2D photographs with 3D range scans, generally matching large structural features such as lines or circles [21]. These methods assume that certain features are present in the scene which is reasonable since many groups focus on urban data that mainly contains regularized architecture. They also often require some sort of user guidance and usually work exclusively with images and range scans that are collected at the same time. Another approach is to assume that there is a set of photographs that is pre-registered with the range scan because they were taken by either by a camera built into the scanner or a camera mounted on the scanner with a known configuration. Our work falls under this second category.

#### 1.1.1 Direct 2D-3D Registration

The photo-realistic urban modeling system produced by Stamos et al. [21] identifies edges and vanishing points in the 2D imagery and matching 3D linear features to estimate the camera parameters used in the 2D-3D registration. Schindler et al. [18] register 2D and 3D urban data by identifying patterns in the structural features in both the 2D and 3D data and then matching them. Wan et al. [24] also take advantage of the linear features inherent in urban scenes to compute camera locations from vanishing lines. The work of Li et al. [9] identifies repetitive structures as well and utilizes vanishing lines to rectify photographs before registering them with the LIDAR range scans. Another approach to direct registration is to create a structure from motion point cloud from a set of photographs and match the 3D output with the range scan [21, 28].

Mutual information can also be used for direct 2D-3D registration. 2D images can be constructed from a LIDAR scan that visualize various properties of the scan such as

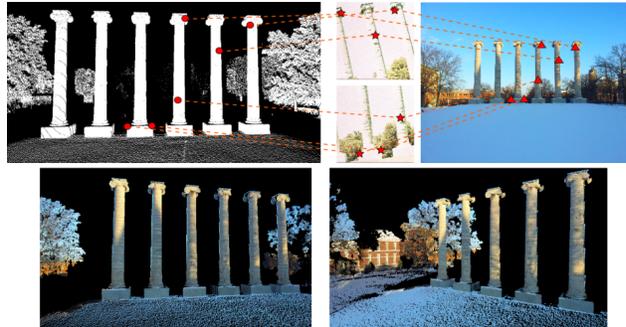


Figure 1. *Top*: Diagram showing 3D-2D link from LIDAR scan (circles) to LIDAR images (stars) to regular photograph (triangles). *Bottom*: 3D registration result shown from two perspectives.

the reflectivity of the laser [16], normals [3], or the relative height throughout a point cloud [1, 11]. The entropy between these types of images and regular photographs is minimized to uncover the relationship between the two.

#### 1.1.2 Using Pre-registered Images for Registration

Using pre-registered photographs to guide the 2D-3D registration maybe a more tractable approach since data is available that is similar to the photographs being registered. However, there are still many difficulties to handle especially when the goal is to register 2D and 3D data that are obtained on different days as the scene may have changed or if different modalities are used to depict a scene such as infrared images or paintings. Yang et al. [26] match difficult image pairs by identifying one stable matching feature between an image with unknown location and a pre-registered image and use a region growing method to find more correspondences. As the feature search space increases, they consider matches between corners, edges, and normals. They feel that their iterative approach to searching for and checking feature correspondences helps to overcome some of the inaccuracies encountered when matching keypoints between images that differ in qualities such as illumination and viewpoint. Weinmann et al. [25] handle the challenge of aligning regular and infrared photographs by matching general shapes of the images' gradient fields across planar areas. Structurally accurate paintings of scenes have even been registered with 3D scans using histogram of gradient information of distinctive regions and by matching contours between 2D and 3D imagery [17, 2].

### 1.2. Contributions

For the 2D-2D matching stage of our pipeline, we present a unique contextual approach for finding image correspondences that builds on the SIFT feature descriptor. Once we have an initial estimate of the 2D-3D correspondence for a camera and a range scan, we refine our estima-

tion using the 3D depth image to directly determine the best alignment of the 2D photograph and the 3D point cloud.

We show that our approach is general enough to work on imagery containing different types of architecture and foliage. Much of the current work in the field will face difficulties in complex scenes, including those heavy in trees, as they rely on extracting straight lines or planes or computing the normals of 3D points. We do not determine correspondences between data by assuming a specific type of structural feature is abundant and identifiable. We use both local and regional feature descriptors to match images that will strongly respond to different image properties which we believe makes our pipeline more flexible when using a variety of data sets.

### 1.3. Data Acquisition

We use a Leica C10 HDS LIDAR scanner which provides a high resolution point cloud of a scene and 2D images of the scanned subject using a built-in camera. These images tend to have small fields of view and exposure inconsistencies making them difficult to match to typical images people take with regular cameras. The output data from the scanner also consists of files containing the internal and external camera parameters for each image. From this, we are able to establish the link between the 2D LIDAR images and the LIDAR range scan as shown in Figure 1. The photographs and videos were taken with a variety of cameras including a Nikon D80 and several smartphones.

## 2. Methodology

In order to register an image with the LIDAR range scan, we must calculate the camera pose for the photograph in relation to the 3D point cloud. This entails matching an image of interest to other images whose 3D correspondences are known, such as our LIDAR camera images, and solving for the camera’s projection matrix. Our 2D-3D registration method can be divided into two main phases. First, we perform 2D image matching between LIDAR photographs and new photographs to be registered and try to obtain as many keypoints matches as possible. We then move onto our 2D-3D phase in which we solve each camera’s projection matrix to align it with the range scan and use depth information to refine our 3D pose estimate.

### 2.1. 2D Contextual Feature Matching

For simplicity in our discussion, we will refer to the 2D images of a pair as  $I_1$  and  $I_2$  which have matching keypoints  $A$  and  $B$  respectively. We obtain an initial set of matches for an image pair using the SIFT feature detector and descriptor which is very robust and describes the rich gradient information surrounding an interest point. Traditionally, for SIFT to accept that  $A$  and  $B$  are a correct match their feature descriptors must pass a distinctiveness ratio test such as

$$\frac{\|A_{des} - B_{des}\|}{\|A_{des} - C_{des}\|} < 0.7, \text{ where } C \text{ is the second best match to } A \text{ in } I_2.$$

In order to identify correct point matches that may not pass the ratio test (and increase our pool of matches), we also take into account contextual information such as line segments and histogram of gradients (HOG) descriptors. We look at the similarities of the lines and HOG’s in the larger neighborhoods surrounding a potential keypoint match to determine if the overall regions agree with each other. These two different features each have their strong points. Our SIFT + line segment method is robust to lighting and content changes and helps by identifying salient linear features. Our SIFT + HOG method makes no assumptions about the existing structure and is useful for matching regions lacking dominant lines but that are rich in texture. Both methods are used to clear up some of the ambiguity between possible keypoint matches. If the neighborhoods of the matches with relatively higher distinctiveness ratios ( $0.7 < ratio < 0.9$ ) match well using one of our measures, we include them in our list of putative correspondences that are used for future registration.

In order to use either similarity measure, we must first define corresponding neighborhoods in the image pair. We use the scale and dominant orientation provided by the SIFT detector to identify these regions that contain the same content and are based in the same coordinate system. The neighborhood in  $I_1$  is centered at keypoint  $A$  with a width of  $k * A_{scale}$  where  $A_{scale}$  is  $A$ ’s SIFT descriptor scale and  $k = 50$ .  $k$  is application dependent and should be chosen so that neighborhoods encompass relatively large and interesting areas. We have experimentally determined one value for  $k$  to use on our dataset that is very stable.

To incorporate linear features into our matching criteria, we identify stable line segments within the defined neighborhood using the Line Segment Detector method [6]. All the line segments in this neighborhood are transformed to a polar coordinate system determined by the SIFT feature’s orientation and normalized by the SIFT feature’s scale and are represented as  $(\rho, \theta)$ . We then match groupings of line segments in corresponding neighborhoods using Hungarian graph matching [8]. Figure 2 outlines this process. The similarity measure for a pair of line segments  $(l_A, l_B)$  is shown in Equation 1.

$$sim(l_A, l_B) = (\rho_A - \rho_B)^2 + \omega(\theta_A - \theta_B)^2 \quad (1)$$

$\omega$  is a weight parameter that normalizes the range of  $\theta$  to that of  $\rho$ . We have determined experimentally that  $\omega$  is stable over a range of [3, 4]. This similarity measure is only calculated if at least three lines are identified in the neighborhood of the SIFT keypoint. If at least half of the lines match according to Equation 1 ( $sim(l_A, l_B) < 2.0$  for our experiments), the match is saved.

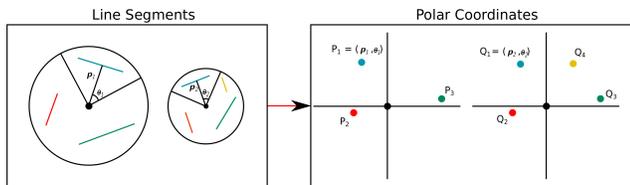


Figure 2. Matching lines in the neighborhood of a SIFT feature point. *Left:* Line segments are identified within neighborhoods centering a SIFT match. The SIFT point is shown in black in both neighborhoods and the SIFT scale is represented by the black circle. *Right:* Line segments are converted to polar coordinates based on their distance from and orientation in relation to the SIFT point and matched.

Our second approach for contextual matching takes advantage of every pixel in corresponding neighborhoods surrounding potential keypoint matches. To measure how well the two neighborhoods match, we begin by computing HOG's for each image patch. A vector is constructed for each HOG cell by concatenating its nine bin values and normalizing using the vector's magnitude. We calculate the L2 distance between the vectors for corresponding cells in the matching neighborhoods. If the average of all these distances is under a defined threshold ( $threshold = 0.4$  for our experiments) we say that the neighborhoods surrounding the match have a strong correspondence and save the keypoint match.

All 2D matches are geometrically verified by calculating the homography relating  $I_1$  and  $I_2$  and removing outlying matches.

## 2.2. 2D-3D Registration and Optimization

Using the set of 2D matches we have found, we proceed to our 2D-3D registration stage. We know from the pre-processing stage what 3D point matches to what 2D LIDAR image pixel. By matching our regular photograph to the LIDAR images, we determine what 3D points match to the feature points of our photograph as is shown in Figure 1. A single list of 2D-3D correspondences is created that is based on the feature matches from the full set of photographs with which an image shares correspondences. We consider the 3D points to be very reliable and treat them as the absolute locations for the feature points. This set of 2D-3D correspondences is then used to calculate the projection matrix of the camera.

We carry out the six-point algorithm with DLT (Direct Linear Transform) [7] and RANSAC (Random Sample Consensus) [5] to find the set of matches that calculate the most accurate projection matrix  $P$ . We refine  $P$  using the set of inlying matches with Levenberg-Marquardt optimization. At this point we have a fairly strong estimate of the projection matrix as we have searched extensively for 2D keypoint matches and have calculated both 2D and

3D camera pose relationships to guide our registration. We refine our 3D pose estimate one more time using 2D-3D information directly to make sure we have a really clean alignment.

### 2.2.1 Depth Imaged-Based Guided Matching and Optimization

Once we have a strong estimate of  $P$ , we refine the registration one last time using a depth image computed using  $P$  and the 2D photograph being registered,  $I$ . We create the grayscale depth image  $D$  by projecting the 3D point cloud onto  $I$ 's 2D plane using  $P$ . This depth image is a valuable source of information because it contains the dominant edges of the 3D point cloud as they are viewed from our estimated camera pose and are in the same 2D coordinate system as  $I$ .

Using the alignment of the depth image and the regular photograph, we perform guided matching along the depth image's edges to directly obtain a relatively denser set of 2D-3D matches that cover the image plane. These matches are used to refine the projection matrix estimate.  $D$ 's edges ideally should align with the strong gradient responses in  $I$ . To ensure that this is the case we incorporate gradient descent in a hierarchical fashion. We first look for this alignment globally, finding one transformation to project  $D$  to  $I$ . We then divide  $D$  and  $I$  into smaller and smaller patches, calculating the best alignment for each patch pair. For every edge pixel in  $D$ , we save the alignment that gave us the smallest energy in an effort to minimize Equation 2, where  $(e_x, e_y, e_z)$  is the 3D point corresponding to an edge pixel in  $D$ .

$$E = \sum_{j=0}^{\# \text{ of edges in } D} [1 - \nabla I(P * (e_{j_x}, e_{j_y}, e_{j_z}, 1)^T)]^2 \quad (2)$$

We take this multiple patch approach because varying amounts of image information will help us find the best alignment for different areas. During the gradient descent process, very accurately matched sections of the depth image along the 2D plane may be shifted away from their optimal alignment due to noise and inaccuracies in surrounding regions. In these cases, it is expected that a smaller window size, looking at a subset of the image, will help maintain the best alignment for the edges in these sections. Less-well matched sections, however, may need to rely on the stable regions elsewhere in the image to help guide them to their best alignment, thus requiring a larger window.

The new set of 2D-3D matches are searched for and extracted along the shifted edges of  $D$ . If  $D$  and  $I$  both have a strong edge at the same pixel location and the neighboring pixels have similar gradient orientations in both images we save the corresponding 2D and 3D points as a new match.

Image	# Matches	% Inliers	Avg. Error
Fig. 1	210	49.05%	2.82
Fig. 3 a. i.	1745	98.05%	4.3
Fig. 3 a. ii.	734	88.69%	3.32
Fig. 3 b. i.	207	92.27%	4.16
Fig. 3 b. ii.	2169	91.06%	3.63
Fig. 3 b. iii.	1004	91.11%	2.86

Table 1. Quantitative results for our 2D-3D registration. The resolution of the LIDAR images is 1920x1920 and that of the regular photographs is 3264x2448.



Figure 3. 2D-3D registration results. Fall and winter photographs registered with summer scan. *Top*: 3D environment after registration. *Bottom*: 2D images being registered.

We refine  $P$  using Levenberg-Marquardt optimization to minimize the reprojection error of this new set of matches. We are more confident in our final calculation of the projection matrix since it is based on a large set of matches located throughout a large percentage of the image plane that have been chosen using contextual information after multiple rounds of registration estimation and refinement.

The image can finally be registered with the scan by mapping every 3D point onto the image plane using the refined projection matrix. The color assigned to each 3D point is the color of the pixel it is closest to when projected onto the image plane. During this stage, the image’s 2D-3D correspondences are saved so that it can be used to guide the registration of new images.

### 3. Results and Discussion

We judge the accuracy of our results both qualitatively and quantitatively. When the 2D images are viewed next to the registration of the images and the range scan (Figure 3), one can easily visually judge the accuracy of the projected data. We also consider the number of matches and their reprojection error when 3D points are mapped to the image plane. These quantitative results are presented in Table 1 which shows the number of matches, the percentage of feature points identified as inliers, and the average reprojection error of the inliers.

### 3.1. Limitations and Future Work

Our 2D-3D registration relies on initially matching 2D images for which we use SIFT and our contextual matching methods. Since SIFT is not affine invariant, we are limited to matching photographs with relatively small baselines. We can expand our feature matching stage to also use an affine invariant feature descriptor such as ASIFT [14] or MSER [12] to make it more robust to affine transformations. In the cases where we still cannot find an initial set of matches because the visual conditions between images is just too great, we can allow the user to select a few initial correspondences and then continue with our registration.

At the moment, we also do not perform any explicit change detection between the 2D photographs and the 3D point cloud such as is discussed in [22]. If the content in these different types of imagery changes too much, our registration will not have a pleasing visual result. For instance, if a building is completely covered by trees in the summer that are rather far in front of it and then it becomes exposed in the winter, the image will not have any correct 3D information onto which it can be projected. In the future, we will need to update the 3D information that has large changes to produce a more visually accurate result. We can do this by creating a structure from motion point cloud from the new set of photographs and use it to alter the LIDAR point cloud.

The visual representation of our results could also be improved by taking into account lighting differences between images that overlap when registered with the LIDAR scan. At the moment we average the color values if multiple images project onto the same 3D point. While this does help to blend the images, seams are still visible if the lighting changes too much. A more sophisticated method such as that presented in the work of Troccoli et al. [23] could help improve this.

### 4. Conclusion

We have presented a general 2D-3D registration method that centers around solving a camera’s projection matrix in relation to an accurate LIDAR range scan and minimizing a gradient-based energy function using a 3D depth image. Included in this process is a contextual method for matching 2D images that takes into account stable linear features and HOG descriptors of unstructured, richly textured areas. We show that our method is robust enough to register data with highly varying visual properties. We work with images with drastically differing focal lengths and exposure values and changes of lighting conditions and seasons. Multiple photographs can be viewed simultaneously in a 3D environment to easily view the full context of a scene from a variety of perspectives. We have also shown that imagery captured on different days and registered with the same scan can quickly and clearly reveal to a user how a location has changed with

time.

## 5. Acknowledgements

This work is supported in part by the NSF CC-NIE award #1245795, NSF CMMI award #1039433, and the NSF Graduate Research Fellowship award #0943941.

## References

- [1] H. Alismail, L. Baker, and B. Browning. Automatic calibration of a range sensor and camera system. In *3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 286–292, 2012. 2
- [2] M. Aubry, B. Russell, and J. Sivic. Painting-to-3d model alignment via discriminative visual elements. In *Transactions on Graphics*. ACM, 2013. 2
- [3] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno. Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. In *Computer Graphics Forum*, volume 28, pages 1755–1764. Wiley Online Library, 2009. 2
- [4] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Computer Vision and Pattern Recognition*, pages 3001–3008. IEEE, 2011. 1
- [5] M. A. Fischler and 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):381–395, 1981. 4
- [6] V. Gioi, R. Grompone, J. Jakubowicz, J. Morel, and G. Randall. Lsd: a line segment detector. *Image Processing On Line*, 2012. 3
- [7] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, Cambridge, United Kingdom, 2010. 4
- [8] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 3
- [9] Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N. Mitra. 2D-3D fusion for layer decomposition of urban facades. In *International Conference on Computer Vision*, pages 882–889. IEEE, 2011. 1, 2
- [10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1
- [11] A. Mastin, J. Kepner, and J. Fisher. Automatic registration of LIDAR and optical images of urban scenes. In *Computer Vision and Pattern Recognition*, pages 2639–2646. IEEE, 2009. 1, 2
- [12] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. 5
- [13] B. Micusik and J. Kosecka. Piecewise planar city 3D modeling from street view panoramic sequences. In *Computer Vision and Pattern Recognition*, pages 2906–2912. IEEE, 2009. 1
- [14] J. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *Journal on Imaging Sciences*, 2(2):438–469, 2009. 5
- [15] U. Neumann, S. You, J. Hu, B. Jiang, and J. Lee. Augmented virtual environments (ave): Dynamic fusion of imagery and 3d models. *Proc. Virtual Reality*, pages 61–67, 2003. 1, 2
- [16] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. *Proc. AAAI National Conference on Artificial Intelligence*, 2012. 2
- [17] B. Russell, J. Sivic, J. Ponce, and H. Dessales. Automatic alignment of paintings and photographs depicting a 3d scene. In *International Conference on Computer Vision Workshops*, pages 545–552. IEEE, 2011. 2
- [18] G. Schindler, P. Krishnamurthy, R. Lubliner, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008. 1, 2
- [19] S. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *Proc. International Conference on Computer Vision*, pages 1881–1888, 2009. 1
- [20] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2007. 1
- [21] I. Stamos, L. Liu, C. Chen, G. Wolberg, G. Yu, and S. Zokai. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *International Journal of Computer Vision*, 78(2):237–260, 2008. 1, 2
- [22] A. Taneja, L. Ballan, and M. Pollefeys. City-scale change detection in cadastral 3d models using images. In *Computer Vision and Pattern Recognition*, pages 113–120. IEEE, 2013. 5
- [23] A. Troccoli and P. Allen. Building illumination coherent 3d models of large-scale outdoor scenes. *International Journal of Computer Vision*, 78(2-3):261–280, 2008. 5
- [24] G. Wan, N. Snavely, D. Cohen-Or, Q. Zheng, B. Chen, and S. Li. Sorting unorganized photo sets for urban reconstruction. *Graphical Models*, 74(1):14–28, 2012. 1, 2
- [25] M. Weinmann, L. Hoegner, J. Leitloff, U. Stilla, S. Hinz, and B. Jutzi. Fusing passive and active sensed images to gain infrared-textured 3d models. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 39:71–76, 2012. 2
- [26] G. Yang, J. Becker, and C. Stewart. Estimating the location of a camera with respect to a 3d model. In *3-D Digital Imaging and Modeling*, pages 159–166. IEEE, 2007. 2
- [27] G. Yang, C. V. Stewart, M. Sofka, and C. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *Trans. on Pattern Analysis and Machine Intelligence*, 29(11):1973–1989, 2007. 1
- [28] W. Zhao, D. Nister, and S. Hsu. Alignment of continuous video onto 3D point clouds. *Trans. Pattern Analysis and Machine Intelligence*, 27(8):1305–1318, 2005. 2