

# Alignment of 3D Building Models with Satellite Images Using Extended Chamfer Matching

Xi Zhang, Gady Agam

Illinois Institute of Technology

Chicago, IL 60616

xzhang22@hawk.iit.edu, agam@iit.edu

Xin Chen

HERE.com

Chicago, IL 60606

xin.5.chen@here.com

## Abstract

Large scale alignment of 3D building models and satellite images has many applications ranging from realistic 3D city modeling to urban planning. In this paper, we address this problem by matching the 2D projection of the building roofs and detected edges of satellite images. To better handle noise and occlusions in alignment, the proposed approach seeks an optimal matching location using an extended Chamfer matching algorithm. In addition the proposed approach attempt to optimize the alignment within large region using a global constraint. We show that the proposed approach can estimate the alignment of matching parts and produce robust result under occlusion. We test the proposed algorithm on two different datasets that covers the downtown areas of San Francisco and Chicago. The results show that the proposed algorithm significantly improves the registration accuracy while maintaining consistent performance.

## 1. Introduction

Alignment of different types of geospatial data has attracted numerous research efforts in recent years. Fusing geospatial data of various types provides a more comprehensive understanding of the data, Alignment can also benefit a large number of applications such as realistic city rendering and urban planning. In this paper, we address a 2D-3D alignment problem between 3D coarse building models and 2D satellite images.

This paper differs from previous work[13, 20, 11, 16] by avoiding the assumption of a global transformation between the registered models. A global transformation model is inaccurate in large scale registration problems. Instead, we treat the alignment as a shape matching problem. Because, in our dataset, both 3D buildings and the satellite images are

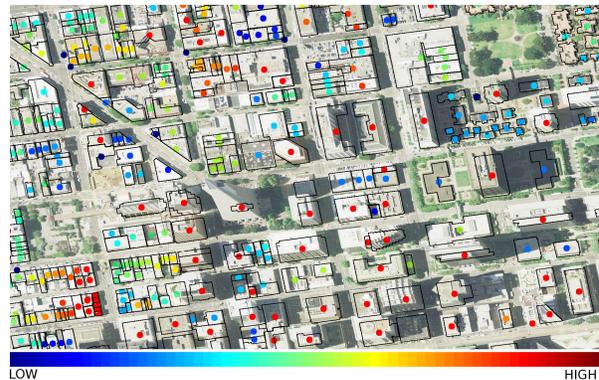


Figure 1. The initial data. We map the 2D projections of 3D buildings onto the satellite images using the geo-referenced data. The colored circle inside each building indicates the height of the building.

geo-referenced, pixels in the satellite images and vertices in the 3D buildings are referenced with respect to each other in the same coordinate system. Since satellite images contain obliquely projections, building roofs are shifted with respect to their base position. The alignment is reduced to a problem of 2D shape matching by using 2D projection of 3D building models. An example of the initial geo-referenced data used in this work is shown in Figure 1.

Shape matching is a fundamental problem in computer vision and has been used in various applications such as information retrieval and object tracking. Similarity measurement between a template and target objects is at the core of shape matching and has been studied extensively. For the purpose of shape matching, sketch and silhouette, have been shown to be a dominant element in human perception and cognition [10, 23]. These provide a compact representation of the object which has been shown to be very useful for matching. Because of these considerations, a large number of methods for matching objects using sketch

edges have been proposed. In most methods, a metric based on the distance measurement between corresponding pixels is used to evaluate the similarity among edge images. Depending on whether the correspondence is built by matching pixels or features, matching methods can be categorized into two groups: 1) feature-dependent[2, 3, 19], which solve the correspondence problem by minimizing distance of feature between corresponding points, and 2) feature-independent[12, 8, 18, 1], in which the distance is defined in a more general way over all of the points. Feature-independent based methods are preferred when speed is considered. However, the performance of such methods may be affected when the scene becomes more complicated and includes more noise. By using high dimensional spatial features, feature-dependent methods get better performance, and are more robust in the presence of clutter and occlusions. However this increased accuracy results in increased computation cost.

Chamfer matching methods which are based on integrating edge orientation[12, 18] are used when speed and accuracy are concerned. However, such methods suffer from large errors when objects are occluded or when the scene is noisy. In this paper, we use a robust Chamfer matching algorithm which takes into account the neighborhood context of pieces in the Chamfer distance and retains well matched pixels in the error computation. We show that we improve both the robustness and accuracy of the Chamfer matching when performing registration between satellite images and 3D building models. An example of the results obtained by the proposed approach using Chamfer matching[4] is shown in Figure 2.

The novel contributions of this paper are in several aspects. First, we define a generalization of Chamfer matching which retains best matched parts and thus is capable of matching occluded objects. Second, we introduce a new distance metric that improves the Chamfer matching. By using this metric, our approach is more robust to noise and generate more accurate results. Third, we propose an approach to better handle the alignment under circumstances of extreme occlusion or noise by optimizing the final building alignment using a consistency constraint imposed by the alignment of neighbouring buildings. Fourth, we show how the proposed approach can be applied to the registration of satellite images and 3D building models. We show that by using the proposed algorithm, the accuracy of matching building roof projections with satellite images is considerably better compared with existing approaches.

## 2. Related Work

Aligning of different types of geospatial data is a common task. In [11], an objective function is designed to reduce the matching error between extracted edges of a building image and a point cloud. By using this objective func-

tion Kaminsky et al. demonstrate a system for both outdoor and indoor LiDAR and image alignment. A similar problem is addressed in [20] where the structure from a point cloud is first roughly aligned with a surface model using GPS information. The alignment is further refined by maximizing the correlation between the height information of the two sources. In [16, 13], mutual information between optical images and LiDAR point cloud is maximized to generate an optimal alignment.

The idea of using Chamfer matching for image registration was first introduced by Barrow *et al.* [1] where they try to find the model of a coastline in a segmented satellite image. Since then many variations of the Chamfer matching have been introduced.

An important variation of the Chamfer matching by Borgefors [5] uses a hierarchical Chamfer matching algorithm (HCMA). This algorithm uses an image pyramid in searching for the optimal position of a template. The search for an optimal position is made in different resolution levels of the pyramid by using a representation of the distance image. The optimization of the objective function is done by discretizing the transformation parameters and stepping through them in each pyramid level. The speed of HCMA can be improved by modifying the computation of the distance transform image and by selecting the starting search position [22]. An HCMA algorithm based on interesting points has been used in [21] where a parallel computation scheme of Chamfer matching is discussed. The selection of interesting points in this work is done through a dynamic threshold scheme guided by a histogram.

A Chamfer matching algorithm which is based on multiple features was introduced by Gavrilu et al.[8],[9], where it is proposed to use edge orientation as a feature. An orientation channel is created for each feature and a distance transform image is generated for edges in the channel. This method uses a hierarchical scheme for matching multiple templates with an image in which similar templates are grouped at different levels.

Shotton et al.[18] use the Chamfer distance with an additional cost which measures the mismatch of edge orientations given by the average difference in orientation between template edges and the closest edges in target image. Instead of explicitly formulating a separate term of orientation mismatch, the orientation difference is generalized in the computation of the Chamfer distance[12]. A comparison between shape context matching and Chamfer matching is conducted in [19], where results show that using the Chamfer matching is faster than matching using shape context and that global matching using Chamfer matching is better than using shape context.

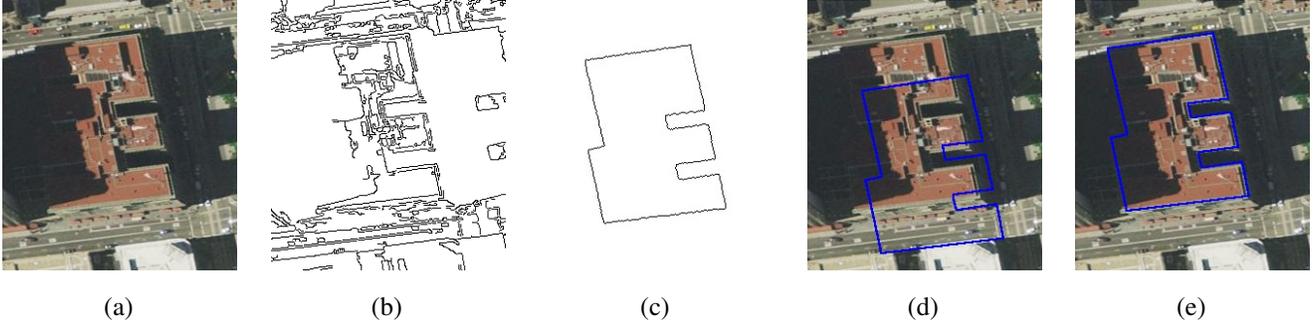


Figure 2. (a) A target building which is partially occluded by a shadow. (b) Edge information showing no detected edges within the shadowed area. (c) a template. (d) Basic Chamfer matching failed to find optimal location in the target image. (e) The proposed approach successfully matches the template and the target.

### 3. Chamfer Matching

Given  $U$  and  $V$ , two binary edge images, where  $U$  is the target image and  $V$  is the template image, we use  $\{u_j\}_{j=1}^m$  and  $\{v_i\}_{i=1}^n$  to denote the edge pixels in these images respectively. Let  $U(u_j)$  denote the value of image  $U$  at location  $u_j$ .

In Chamfer matching, we seek a correspondence between  $\{u_j\}$  and  $\{v_i\}$  under transformation  $W$ . Assuming that the transformation  $W$  between the template and target images is rigid with translation  $T$  and rotation  $R$ , a template edge pixel  $v_i$  is transformed into the target image by using following expression:

$$W(v_i; R, T) = R \cdot v_i + T \equiv v_i^u \quad (1)$$

Given a distance metric  $d(\cdot)$ , we can solve for the transformation parameters  $T$  and  $R$  by minimizing the total distance:

$$D(U, V; R, T) = \frac{1}{n} \sum_{1 \leq i \leq n} d(v_i^u, u_j) \quad (2)$$

where  $u_j$  is the closest target edge pixel to  $v_i^u$  in the sense of the distance metric  $d(\cdot)$ .

The computation of  $d(v_i^u, u_j)$  can be done in linear time using the *distance transform* of the target image[17]. Denoting the distance transform image of  $U$  by  $U_{DT}$ , we can write Equation (2) as:

$$D(U, V; R, T) = \frac{1}{n} \sum_{1 \leq i \leq n} U_{DT}(v_i^u). \quad (3)$$

Various types of distance transforms can be produced using different distance metrics  $d(\cdot)$ . In [17], city-block distances are used and a two-step linear algorithm is proposed to compute the distances. Euclidean distances approximations are provided by Borgefors[4], Montanari[14], and Danielsson[6]. An efficient squared Euclidean distance computation algorithm is described by Felzenszwalb[7], Felzenszwalb's algorithm can be generalized to compute other distances.

### 4. The Proposed Approach

#### 4.1. Data Preparation

Given geo-referenced 3D building models and satellite images, the proposed approach aims to minimize the matching error between roofs in these two datasets. Building roofs are shifted in the image in proportion to their height. Let  $r$  be the resolution of a single pixel in meters in the satellite image. Let  $h$  be the height of a building given by a 3D model. Assuming an oblique projection not greater than  $45^\circ$ , we can compute the radius of the search window as  $w = h \cdot \cos(45^\circ)/r$ . We limit the search for optimal alignment of each building to a search window as defined above.

The Chamfer matching is based on edges extracted from the satellite image. To better handle noise and low resolution issues in the edge detection, a mean-shift filter is first used to locally homogenize small differences in the image. Edges are extracted by a canny edge detector and are further refined by removing short edges. We use  $U$  in Equation (3) to denote the edge image of a satellite image within the search window.

In Chamfer matching, the counterpart  $V$  in Equation (3) is extracted as the boundary of a 3D building projection onto 2D. Since roofs are the most visible part of buildings in satellite images, we use the projection of the roof of each building. The extraction of 3D building roofs is done by computing the surface normal of the 3D mesh and comparing it with the vertical direction. To avoid the case where the extracted roof surface of a building is a small region, we group the extracted roofs within highest 15 meters into one.

#### 4.2. Extended Chamfer Matching

Object matching is a fundamental problem in computer vision. Robust matching should be resistant to noise and be able to deal with cases where objects are partially occluded. The basic Chamfer matching is sensitive to noise and can not reliably handle occlusion. To address these shortcomings, we extend the Chamfer matching in several

ways. First we propose a general way of computing edge orientation in both the target and template images, and use it in a new distance metric. Second, we propose a method using the context of pixels to rule out outliers when computing the distance error. Third, we provide a strategy for matching incomplete targets.

### Distance Metric with Edge Orientation

We extend the basic Chamfer matching and obtain additional robustness by jointly minimizing the spatial distance between pixels and the misalignment of edge orientations. A similar distance metric is used in [12] where a linear representation of edges is generated to model the edge orientation. The proposed approach avoids the need for a linear representation of edges by computing the edge orientation directly from the images.

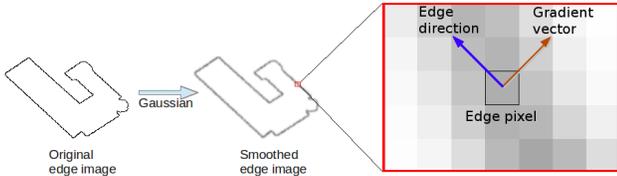


Figure 3. Computing edge orientation in the binary template image.

The edge orientation at each location is computed as a vector perpendicular to the gradient vector at that location. In the target image, the distance transform image  $U_{DT}$  is used to compute the edge orientation at each location. Specifically, given a transformed location  $v_i^u$ , its edge orientation  $\vec{v}_i^u$  is computed as a vector perpendicular to the gradient vector  $\nabla U_{DT}(v_i^u)$ . To compute the edge orientation in the binary template image  $V$ , we create a gradient vector field around edges by smoothing  $V$  using a standard Gaussian kernel. The edge orientation vectors  $\vec{v}_i$  are then calculated as vectors perpendicular to the gradient vectors obtained from the gradient vector field. An example of the edge orientation computation in  $V$  is shown in Figure 3.

Having the edge orientation computed in both the target and template images, the distance between pixels  $v_i^u$  and  $u_j$  is computed by:

$$d(v_i^u, u_j) = \lambda U_{DT}(v_i^u) + (1 - \lambda)(1 - |\cos(\alpha_{v_i^u})|) \quad (4)$$

where  $\lambda$  is a weight factor that controls the importance of orientation mismatch,  $\cos(\alpha_{v_i^u}) = \langle \vec{v}_i^u, \vec{v}_i \rangle$  measures the orientation mismatch, and it is assumed that  $\vec{v}_i^u$  and  $\vec{v}_i$  are normalized.

A discussion of the selection of  $\lambda$  is given in section 6. A squared Euclidean norm is used in the first term of Equation (4), which gives a larger penalty to mismatched pixels. A method to generate the squared Euclidean norm distance transform in linear time is described in [7].



Figure 4. Example of the modified distance measure. In both cases we use a neighborhood of size  $p = 13$ , and select the lowest  $q = 5$  neighbors. While the distance at the pixel is the same (1.5), the modified distance measure in the bottom example is higher due to the larger distance to neighbors.

### Edge Distance Variance

To have a robust matching result, the matching error of a pixel  $v_i$  should not solely depend on  $d(v_i^u, u_j)$ . This is because it is possible that  $v_i^u$  will be an incorrect edge pixel in the target image that happened to have a small error. Based on this consideration, we argue that to get a confident measurement of the matching error it is necessary to take into account the matching error of neighbouring pixels.

Given the transformed template edge pixels  $\{v_i^u\}_{i=1}^n$ , we find for each pixel  $v_i^u$  matching scores for the  $p$  closest transformed template edge pixels. To estimate contextual matching error of  $d(v_i^u, u_j)$  at location  $v_i^u$ , the  $q$  pixels with lowest matching error, where  $q < p$ , are selected. We denoted these  $q$  pixels as  $\{v_k^u\}_{k=1}^q$ . We then calculate the variance of the distance  $d(v_i^u, u_j)$  of the  $q$  selected pixels:  $\varphi(v_i^u) = \frac{1}{q} \sum_{k=1}^q (d(v_k^u, u_j) - \bar{d})^2$ , where  $\bar{d}$  is the average matching error of the  $q$  neighbors.

The distance variance  $\varphi(v_i^u)$  provides a more stable assessment of the matching result at  $v_i$ . The parameters  $p$  and  $q$  control the size of the contextual information used in the computation. A larger  $p$  leads to more contextual information included while  $q$  helps in excluding outliers. In our experiments, we set  $p = 13$  and  $q = 5$ . Since  $p$  and  $q$  are constants, the asymptotic time complexity of the algorithm is not affected by them. From a practical point of view, to save time when searching for the  $p$  closest neighbors at each  $v_i$ , we generate and maintain a list of  $p$  closest neighbors for each  $v_i$  before the matching begins.

As we would like to give preference to distance measures with small distance variance, we modify the distance metric in Equation (4) by multiplying it by a factor which is proportional to the distance variance:

$$d_{\varphi}(v_i^u, u_j) = d(v_i^u, u_j) \times (1 + \varphi(v_i^u)) \quad (5)$$

Figure 4 shows an example of the modified distance com-

putation.

The significance of multiplying the distance variance in Equation (5) lies in several aspects. First, the distance measure is no longer solely dependent on each pixel’s individual matching distance as the distance now considers the relationship between a pixel and its best matched neighbours. Consequently, we expect each pixel and its connected neighbours to have a small distance. Second, the variance in Equation (5) makes it much easier to separate well matched pixels from mismatched ones by increasing the width of boundary that separates the well matched pixels and mismatched ones.

---

**Algorithm 1** Computation of  $D(U, V; R, T)$ .

---

**Input:**

- Images  $U, V$  and their edge pixels.
- Current transformation  $R, T$ .
- Parameters  $p, q, \lambda, \theta, t_s, t_\alpha$  and  $t_\varphi$ .

**Output:**

- $D(U, V; R, T)$
1. Compute  $\Phi$  using Equation (6).
  2. Compute  $d_\varphi(v_i^u, u_j)$  for each  $v_i^u$  using Equation (5).
  3. Create the set  $S = \{(v_i^u, u_j) \mid d_\varphi(v_i^u, u_j) < \Phi\}$ .
  4. if  $\#S < \theta \cdot n$ , add to  $S$ ,  $(\theta \cdot n - \#S)$  pairs of  $(v_i^u, u_j)$  not already in it with the smallest  $d_\varphi(v_i^u, u_j)$ .
  5. Compute  $D(U, V; R, T)$  using  $S$ .
- return**  $D(U, V; R, T)$ .
- 

### Matching Incomplete Targets

The basic Chamfer matching does not handle well occlusions in the target. This is because distance transform values change and become unpredictable in parts which are occluded. Including the matching distance in occluded parts will result in a shift from the true optimal location. Therefore, we aim at separating well matched pixels from pixels in occluded areas, and retain only pixels with small distances for error computation. By using the proposed distance measure of Equation (5), the proposed approach generates a much more pronounced boundary between low error and high error regions, thus making it easier to separate them.

Let the matching distance of each template edge pixel be denoted by  $\{d_\varphi(v_i^u, u_j)\}_{i=1}^n$ . The goal in matching is to get as many pixels with acceptable matching distance as possible in computing the final matching error. Thus, we set two parameters. The first parameter  $\theta$  is the fraction of template pixels that have to be used in the distance computation. The second parameter  $\Phi$  is a tolerance of distance error. The distance metric we propose contains three components: spatial distance ( $t_s$ ), angular difference ( $t_\alpha$ , in degrees), and a distance variance ( $t_\varphi$ ), by setting thresholds for these three components, the distance tolerance  $\Phi$  can be

computed using Equation (5) as:

$$\Phi = (\lambda t_s^2 + (1 - \lambda)(1 - |\cos(t_\alpha)|)) \times (1 + t_\varphi) \quad (6)$$

Given the parameters  $\theta$  and  $\Phi$ , the computation of the total distance  $D(U, V; R, T)$  as defined in Equation (2) is performed using pixels with acceptable distance. This total distance is then used to drive the registration by attempting to minimize it. A summary of the necessary steps for computing  $D(U, V; R, T)$  is provided in Algorithm 1.

### 5. Alignment Using a Global Constraint

The optimal transformation parameters  $R$  and  $T$  are obtained by minimizing  $D(U, V; R, T)$  in the search window. In the proposed approach, we ignore the rotation  $R$  in the transformation since this is resolved by the geo-referencing of the data. We represent the translation  $T$  of building as a shift vector pointing from the initial geo-referenced location to the location returned by the alignment process. Examples of the shift vectors returned by the proposed Chamfer matching are shown in Figure 5.

The proposed algorithm handles building roof alignment well and achieves high accuracy in general. However, in some cases, the proposed approach returns incorrect alignment result. Such incorrect alignments are caused by several factors. First, the proposed approach may fail to detect the target if the image quality is low due to large occlusions or strong shadows in the image. Second, the proposed approach may fail in situations where multiple similar targets are present in the search region. Finally, coarse or inaccurate modeling of 3D building in our dataset may cause the proposed approach to fail.



Figure 5. Illustration of the alignment using global constraint. Blue bars represent the shift vectors computed by our approach. The red disk shows the nearest building search area aligning using the global constraint.

To handle possible failures in the alignment computation, we introduce a global constraint which takes into account the alignments of neighboring buildings. Assuming

that the shift vector direction of buildings in a region are close to each other, we require every shift vector to be consistent with its neighbors during the alignment.

Given a set of  $n$  target images and  $n$  template images denoted by  $\{(U_b, V_b)\}_{b=1}^n$  for buildings in a city scale region, we look for a corresponding set of shift vectors  $\mathcal{T} = \{T_b\}_{b=1}^n$  which minimizes the global alignment error shown below:

$$E(\mathcal{T}) = E(\{T_b\}_{b=1}^n) = \sum_{b=1}^n E(U_b, V_b; T_b) \quad (7)$$

where  $E(U_b, V_b; T_b)$  is an error term for building  $b$ . In the computation of  $E(U_b, V_b; T_b)$ , we take into account the consistency among the alignments in the neighborhood which is measured by the difference between  $T_b$  and the dominant shift vector of the neighboring buildings. Thus  $E(U_b, V_b; T_b)$  is computed as:

$$E(U_b, V_b; T_b) = \beta D_b + \frac{(1 - \beta)}{2} \left(1 - \frac{T_b}{\|T_b\|} \cdot \frac{T'_b}{\|T'_b\|}\right) \quad (8)$$

where  $D_b$  is an abbreviation for  $D_b(U_b, V_b; R_b \equiv I, T_b)$ , the Chamfer matching distance with the translation  $T_b$  and no rotation, and  $T'_b$  is the dominant shift vector in the neighborhood of building  $b$ . We use  $\beta$  to balance the importance between the distance and angular terms ( $\beta = 0.4$  is used in all our tests). To ensure that the ranges of the distance term and the angular term are the same, we normalize  $D_b$  to a value between 0 and 1. The 30 closest buildings in the neighborhood are considered when computing dominant shift vector and we use RANSAC and principal component analysis (PCA) to better handle outliers in the computation of the dominant shift vector. To efficiently minimize  $E(U_b, V_b; T_b)$ , we keep track of a set of local minimum values of  $D_b$  during the Chamfer matching stage which is denoted by  $\mathcal{S}_b$ . We further use FLANN[15] to accelerate the neighborhood search.

The final optimized shift vectors are computed as the ones that minimize  $E(\mathcal{T})$ .  $E(\mathcal{T})$  is optimized iteratively and the optimization process is shown in Algorithm 2.

## 6. Experimental Results

We tested the proposed approach on two datasets, where each contains 1000 building models selected from a San Francisco (SF) and a Chicago (CHI) urban area.

The four parameters in Algorithm 1 are set as follows:  $\theta = 50\%$ ,  $t_s = 5$ ,  $t_\alpha = 15$ , and  $t_\varphi = 0.8$ . By setting these parameters, we allow for at least 50% of template pixels to be inliers in the computation of  $D(U, V; R, T)$ ; the accepted matching error of obtained transformation has to be within error of 5 pixels in terms of spatial distance and

---

### Algorithm 2 Optimization of $E(\mathcal{T})$ .

---

**Input:**

- $\{\mathcal{S}_b\}_{b=1}^n$  for  $n$  buildings

**Output:**

- $\{T_b^*\}_{b=1}^n$  that minimize  $E(\mathcal{T})$
- while**  $E^k(\mathcal{T}) - E^{k-1}(\mathcal{T}) > \gamma$  **do**
- for**  $b = 1, \dots, n$  **do**
- $T_b^k = \operatorname{argmin}_{T_b^k \in \mathcal{S}_b} \operatorname{RANSAC}(E^k(U_b, V_b; T_b^k))$
- end for**
- $E^k(\mathcal{T}) = \sum_{b=1}^n E_b^k(U_b, V_b, T_b^k)$
- end while**
- for**  $b=1, \dots, n$  **do**
- $T_b^* = T_b^k$
- end for**
- return**  $\{T_b^*\}_{b=1}^n$ .
- 

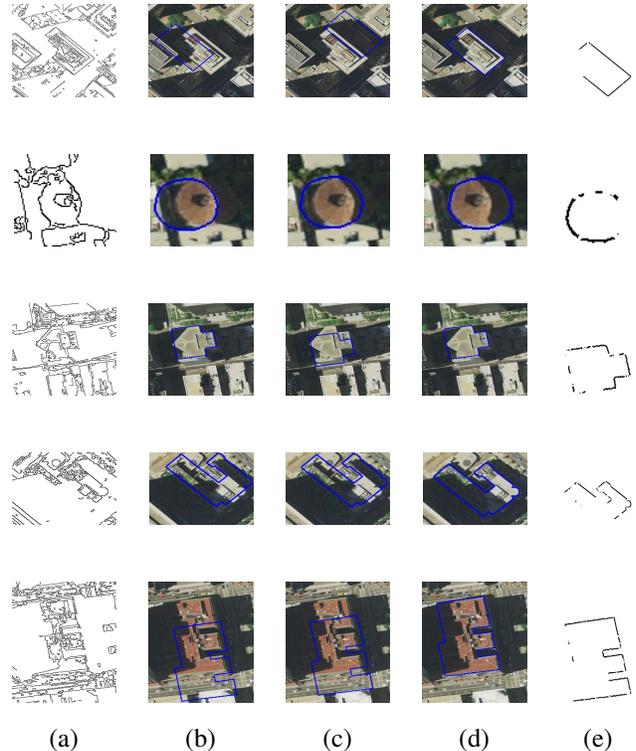


Figure 6. Results of matching partially occluded buildings. (a) Edge images detected from target image. (b)-(d) Results of the CM algorithm, the results of the DCM algorithm and the results of the proposed algorithm, respectively. (e) The pixels selected for computing the average error during the matching process.

15 degrees in terms of orientation difference; finally 0.8 average distance variance is used to rule out outliers during Chamfer matching.

In our experiments, we test the accuracy of our algorithm in estimating the optimal transformation during alignment. We manually labelled the ground truth locations of the building roofs in satellite images. The accuracy of

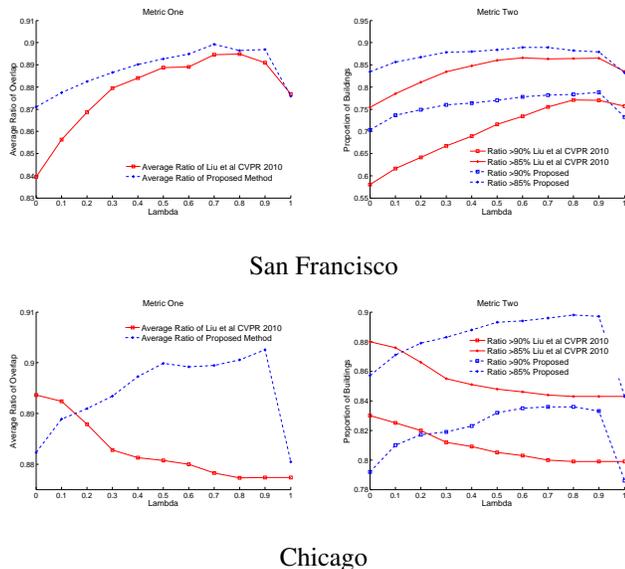


Figure 7. Experimental evaluation results on the San Francisco (upper row) and Chicago (lower row) datasets. The left and right columns show different evaluation metrics.

the result is measured by the ratio of overlap between the bounding box of the ground truth roof mask and the one transformed by the proposed algorithm. Note that simply measuring the Root Mean Square error (RMSE) between the aligned targets and templates is not accurate as it is sensitive to broken edges and incorrect alignments.

We first test the proposed Chamfer matching without using global constraint. We divide our evaluation into two parts. In the first part, the evaluation is done on all the buildings of both datasets using the same  $\lambda$ . We then change  $\lambda$  between 0 and 1 to compare the robustness of the proposed approach, and evaluate the relative importance of the spatial and angular terms. In the second part, to assess the performance of the proposed algorithm on matching occluded objects, we specifically run tests on building images in which the target buildings are partially occluded. In all the tests, we compare our results with that of the directional Chamfer matching (DCM) proposed in [12] and the basic Chamfer matching (CM). Note that the result of CM is a special case of the DCM algorithm when  $\lambda = 1.0$ .

Two kinds of metrics were used in our evaluation. In the first metric, we compute the average ratio of overlap area in the two datasets. In the second metric, we consider only results with area overlap above a certain rate (85% and 90%).

In the first test where all the buildings were included, it could be observed in Figure 7 that our algorithm generates better results compared with DCM and CM at almost every setting of  $\lambda$ . In both datasets, our algorithm maintains a similar and consistent performance, while DCM performs differently on the two datasets.

		Metric One	Metric Two $\geq 85\%$	Metric Two $\geq 90\%$
SF	Proposed	91%	0.94	0.90
	DCM	63%	0.33	0.28
	CM	35%	0.20	0.18
CHI	Proposed	80%	0.81	0.78
	DCM	51%	0.21	0.20
	CM	32%	0.16	0.13

Table 1. Accuracy results of aligning buildings partially occluded by shadow.

To test proposed algorithm’s strength in finding matching targets under partial occlusion, we specifically test and evaluate the algorithms’ performances on 53 buildings from San Francisco and 74 buildings from Chicago which are significantly occluded by shadows. We use a parameter of  $\lambda = 0.7$  in this test. The results are shown in Table 1. As can be observed, the proposed algorithm achieves 80% accuracy on both datasets compared with about 50% accuracy when using DCM and 30% accuracy when using the original Chamfer matching. Some examples of matching incomplete buildings are shown in Figure 6.

We finally tested the improvement obtained by the global constraint for  $\lambda = 0.7$ . For both data sets, we obtain at least 2% improvement in terms of accuracy. The comparisons of alignment results with and without global constraint alignment are given in Table 2.

		Metric One	Metric Two $\geq 85\%$	Metric Two $\geq 90\%$
SF		89% $\rightarrow$ 91%	88% $\rightarrow$ 90%	78% $\rightarrow$ 79%
CHI		89% $\rightarrow$ 92%	89% $\rightarrow$ 91%	83% $\rightarrow$ 85%

Table 2. Results of alignment using global constraint. The number on the right side of the arrows show the results obtained by alignment using the global constraint.

## 7. Conclusion

In this paper we address the problem of the 2D and 3D alignment between 3D coarse building models and 2D satellite images. To solve this problem, we extend the Chamfer matching algorithm by measuring the distance error based on context information instead of individual pixel error, using edge orientation, and segmenting distance errors. This results in a new distance metric. We demonstrate that by using this metric, our extended Chamfer matching is more robust to noise as well as partial occlusion of the target image. The proposed algorithm was tested on two sets of satellite building images which contain many cases where buildings are occluded or covered by strong shadows. To better handle the alignment failures in several extreme cases, a strategy using a constrained imposed by the alignment of neighboring buildings are proposed. Experimental re-

sults show that, compared with known methods, we obtain increased accuracy over the entire dataset and get superior performance on a set of occluded building images.

## References

- [1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Artificial intelligence (IJCAI), 1977 Proceedings of the 5th international joint conference on*, pages 659–663, 1977. 2
- [2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002. 2
- [3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR’2005 Proceedings of the 2005 IEEE computer society conference on Computer vision and pattern recognition*, pages 26–33, 2005. 2
- [4] G. Borgefors. Distance transforms in digital images. In *Computer Vision, Graphics, and Image Processing*, volume 34, pages 344–371. Elsevier, 1986. 2, 3
- [5] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(6):849–865, 1988. 2
- [6] P.-E. Danielsson. Euclidean distance mapping. In *Computer Graphics and Image Processing*, pages 227–248, 1980. 3
- [7] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004. 3, 4
- [8] D. M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Pattern Recognition, 1998 Proceedings of 14th International Conference on*, pages 439–444, 1998. 2
- [9] D. M. Gavrila and V. Philomin. Real-time object detection for “smart” vehicles. In *Computer Vision, 1999 The Proceedings of the Seventh IEEE International Conference on*, pages 87–93, 1999. 2
- [10] D. D. Hoffman and M. Singh. Saliency of visual parts. *Cognition*, 63(1):29–78, 1997. 1
- [11] R. S. Kaminsky, N. Snavely, S. M. Seitz, and R. Szeliski. Alignment of 3d point clouds to overhead images. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 63–70. IEEE, 2009. 1, 2
- [12] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Computer Society Conference on*, pages 1696–1703, 2010. 2, 4, 7
- [13] A. Mastin, J. Kepner, and J. Fisher. Automatic registration of lidar and optical images of urban scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2639–2646. IEEE, 2009. 1, 2
- [14] U. Montanari. A method for obtaining skeletons using a quasi-euclidean distance. *Journal of the Association for Computing Machinery*, 15(4):600–624, 1968. 3
- [15] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP’09*, pages 331–340. INSTICC Press, 2009. 6
- [16] E. G. Parmehr, C. S. Fraser, C. Zhang, and J. Leach. Automatic registration of optical imagery with 3d lidar data using local combined mutual information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:28–40, 2014. 1, 2
- [17] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, 1966. 3
- [18] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1270–1281, 2008. 2
- [19] A. Thayananthan, B. Stenger, P. H. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Computer Vision and Pattern Recognition (CVPR), 2003 IEEE Computer Society Conference on*, pages 127–133, 2003. 2
- [20] A. Wendel, A. Irschara, and H. Bischof. Automatic alignment of 3d reconstructions using a digital surface model. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 29–36. IEEE, 2011. 1, 2
- [21] J. You, W. Zhu, E. Pissaloux, and H. Cohen. Hierarchical image matching: a chamfer matching algorithm using interesting points. In *Intelligent Information Systems, 1995 Proceedings of the 3rd Australian and New Zealand Conference on*, pages 70–75, 1995. 2
- [22] Q. Zhang, P. Xu, W. Li, Z. Wu, and M. Zhou. Efficient edge matching using improved hierarchical chamfer matching. In *Circuits and Systems. 2009 IEEE International Symposium on*, pages 1645–1648, 2009. 2
- [23] X. Zhang and G. Agam. A learning-based approach for automated quality assessment of computer-rendered images. In *Image Quality and System Performance IX, 2012 SPIE Proceedings on*, pages 68–75, 2012. 1