Geo-semantic Segmentation

Shervin Ardeshir[1] Kofi Malcolm Collins-Sibley[2] Mubarak Shah[1]
[1] Center for Research in Computer Vision, University of Central Florida
[2] Northeastern University
ardeshir@cs.ucf.edu, collins-sibley.k@husky.neu.edu, shah@crcv.ucf.edu

Abstract

The availability of GIS (Geographical Information System) databases for many urban areas, provides a valuable source of information for improving the performance of many computer vision tasks. In this paper, we propose a method which leverages information acquired from GIS databases to perform semantic segmentation of the image alongside with geo-referencing each semantic segment with its address and geo-location. First, the image is segmented into a set of initial super-pixels. Then, by projecting the information from GIS databases, a set of priors are obtained about the approximate location of the semantic entities such as buildings and streets in the image plane. However, there are significant inaccuracies (misalignments) in the projections, mainly due to inaccurate GPS-tags and camera parameters. In order to address this misalignment issue, we perform data fusion such that it improves the segmentation and GIS projections accuracy simultaneously with an iterative approach. At each iteration, the projections are evaluated and weighted in terms of reliability, and then fused with the super-pixel segmentations. First segmentation is performed using random walks, based on the GIS projections. Then the global transformation which best aligns the projections to their corresponding semantic entities is computed and applied to the projections to further align them to the content of the image. The iterative approach continues until the projections and segments are well aligned.

1. Introduction

Segmentation of an in image into coherent and semantically meaningful regions is a fundamental problem in computer vision. Many methods employing image content have been proposed in the literature during the last few decades. In particular, semantic segmentation of images taken from urban areas has been studied in the past few years [10, 14, 2, 4]. However, to the best of our knowledge, the problem of segmenting images into exact geo-referenced semantic labels has yet to be studied. Given the availability of a plethora of geographical information, geo-semantic segmentation could be a relevant topic of research in computer vision. In particular, GIS databases containing information about the exact boundaries (2D footprints in long-lat domain) of semantic regions such as buildings, roads, etc. and are available for many urban areas, making it a suitable source of information for performing geo-semantic segmentation. In this work, we propose a method to leverage the information extracted from GIS, to perform geo-semantic segmentation of the image content, and simultaneously refine the misalignment of the projections. First, the image is segmented into a set of initial super-pixels. Also, camera matrix is formed using the provided camera parameters such as sensor size and focal length, alongside with the GPS location of the camera using which the image was captured. Using the camera matrix, 3D points in the GIS (which belong to regions of interest, in our case buildings and streets) are projected onto the 2D image plane. In many cases, due to the inaccuracies in the meta-data and mosty due to the GPS location (EXIF Tag), the projections are far from perfect and usually not aligned with the objects of interest. This misalignment is more intense for regions which are at larger distances from the camera and/or have smaller projection sizes in the field of view of the camera. We show that these misaligned projections turn into misleading priors for the semantic segmentation step. Therefore, we evaluate each of the projections in terms of reliabil-

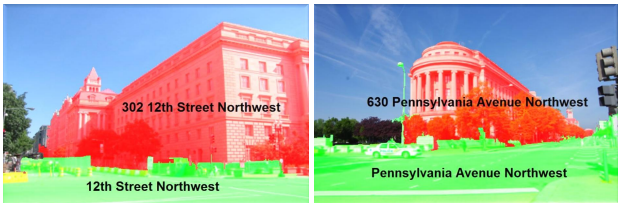


Figure 1: Labeling semantic entities in the image such as buildings (red) and streets (green) alongside with their geo-semantic labels (addresses).

ity, and weight their contribution before using them in the segmentation process. Once the projections are weighted, we get some priors for the initial super-pixels, and combining them with the spatial-visual consistency of super-pixels we obtain some initial semantic segmentation results. Given the fact that each semantic segment is computed using a projection, the reliability score of the projections can be propagated to the semantic segments. As a result, semantic segments corresponding to the most reliable projections (which usually belong to large buildings covering considerable proportion of the content of the image) will get significant weights. Using the projected GIS segments, and their corresponding semantic segments in the content of the image, the misalignment of the projections could be computed with respect to their location in the image content. Our experiments indicate that the aforementioned alignment can be used to obtain a new set of updated projections, with less misalignment. In other words, the results of the segmentation can be used for refining the projections and vice versa. This iterative process continues until our defined convergence criteria is met. We evaluated the authenticity of this refinement procedure by observing the alignment of the projections with the annotated content of the image.

In the context of semantic segmentation of urban area images, Teboul et al. [10] use procedure shape priors for segmenting building facades. Xiao et al. [13] perform semantic segmentation and model building facades and authors in [8, 9] perform procedural modeling for buildings. Authors in [3, 16, 5] train generative or discriminative models for different categories to perform semantic segmentation. As opposed to the aforementioned works which had the objective of segmenting images into different general categories, in our work we pursue the idea of labeling segments with their geo-semantic labels. Xiao et al [14] and Brostow et al [2] utilize 3D information to perform semantic segmentation, and [17] also perform image based modeling of urban scenes. In the context of utilizing GIS information, different computer vision applications have been proposed. Authors in [11, 12] used GIS for registering aerial images using semantic segments and [1] leveraged GIS to improve the performance of object detection; however, to the best of our knowledge, leveraging GIS for the purpose of performing semantic/geo-semantic segmentation has not been pursued.

2. Framework

The block diagram of the proposed method is shown in figure 2. Given a GPS-tagged image and the GIS database containing the outlines of the buildings, we project the points on the outlines and a set of points representing the streets to the 2D image plane in order to obtain some priors about the semantic segments present in the image. In addition, we perform some initial super-pixel segmentation

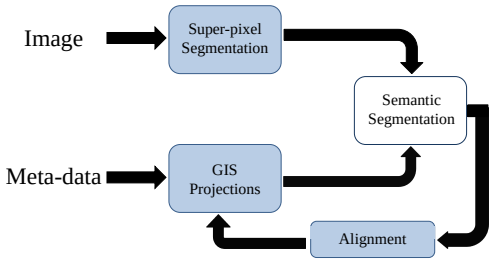


Figure 2: The block diagram of the proposed method. GIS projections are used for labeling the super-pixels, and the resulting semantic segments are being used for updating the quality of the priors in an iterative manner.

(a) (b)

Figure 3: Segmenting the image into an initial set of super-pixels. (a) shows the image to be segmented, and (b) shows its superpixel segmentation randomly colored.

of the image using a typical segmentation method. Finally, we construct a graph over the super-pixels and use the GIS-obtained priors, and the pairwise visual similarity among the superpixels for labeling the super-pixels with at most one of the geo-semantic labels which are supposed to be visible in the image. The labeling is done in an iterative manner and at each iteration, the quality of projections improve in addition to the segmentation accuracy. The details of each step are provided in the following sections.

2.1. Initial Super-pixel Segmentation

First we oversegment the image into n super-pixels. Any segmentation method could be used for this purpose. In our experiments we apply the entropy rate super-pixel segmentation method introduced in [7]. This method does a decent job in capturing the boundaries, and producing super-pixels with similar sizes. A sample image, and it's super-pixels can be seen in figure 3.

2.2. Projecting GIS Segments

The goal of this section is to project the 3D world coordinates to the 2D image coordinates to obtain the initial set of projections. We use the meta-data provided in the

EXIF tag such as GPS-tag, focal length and sensor size to form the initial camera matrix(P). In addition to the meta-data we assume that the buildings and camera have fixed heights (15 and 1.7 meters respectively), and camera has zero roll and tilt which is a reasonable initialization assumption for most of the images. Camera matrix has the standard form of: $\vec{P_0} = \vec{C}[\vec{R} \mid \vec{T}]$, where R, C, and T are the rotation, calibration, and translation matrices respectively. R is a multiplication of R_{roll}, R_{tilt} and R_{yaw} capturing the roll, tilt (both assumed to be zero), and yaw (extracted from compass) of the camera. T is the translation of the camera from the origin (a function of the GPS-tag extracted from the EXIF tag). C is the intrinsic camera matrix which is formed based on sensor size and the focal length of the camera (again, extracted from the EXIF tag). Once the camera matrix (P) is formed, we obtain the initial set of projections by projecting points from GIS using the following:

$$\begin{pmatrix} g \\ 1 \end{pmatrix} = PG, \tag{1}$$

where G denotes a 3D point in the GIS database which is in the field of view of the camera and g is its corresponding 2D projection in the image plane. We can define $\mathbf{G}_i$ as the set of 3D GPS coordinates belonging to the i^{th} semantic segment in the GIS database which is in the field of view of the camera (in our case either a specific building or street)[1], and $\mathbf{g}_i$ as the two dimensional locations of the i^{th} set of point in the image plane. A point from GIS will be projected as a vertical line on the image plane(as shown in figure 4). However, assuming zero tilt and a fixed height for buildings, we use a limited vertical line representing that point, and a polygon-shaped segment representing each set of connected points on an outline. This process is shown in figure 4. (a) shows the position and the viewpoint of the camera, in addition to the GIS outlines. Figure 4 (b) shows the projections, i.e. $\mathbf{g_i}$s. However, many of the GIS points will be occluded by other buildings, so we need a mechanism to eliminate the occluded GIS points from our projections. For that purpose, we make sure the ray connecting the location of the camera to the point on GIS does not intersect with the lines formed by building outlines.

2.3. Iterative Segmentation-Projection Data Fusion

As mentioned earlier, the main challenge for obtaining accurate geo-semantic segments, is the misalignment of GIS projections which is mostly due to inaccurate GPS-tags and camera parameters (example shown in figure 7). These misalignments are more severe when the projections are small and/or they belong to entities which are at a large distance to the camera. In those cases, slight inaccuracies might cause drastic displacements. In many cases, the inaccuracy is so drastic that there is no overlap between the

[1] all GPS positions are converted to ENU Cartesian coordinates system.

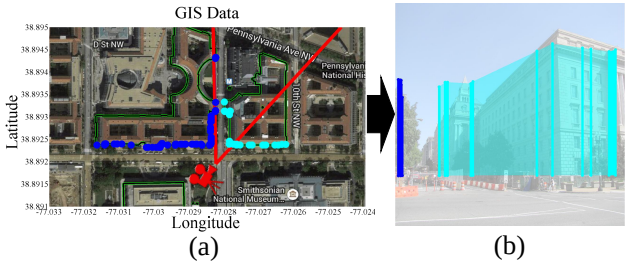


Figure 4: (a) shows the GIS database containing the building outlines. The red lines show the field of view of the camera and cyan points represent the outline of the main building visible in the image. The blue points represent the outline of another building which is actually not visible in the image, but due to inaccuracies in the camera parameters is mistakenly included in the field of view of the camera. (b) shows the projections of the two buildings (cyan and blue) from the GIS.

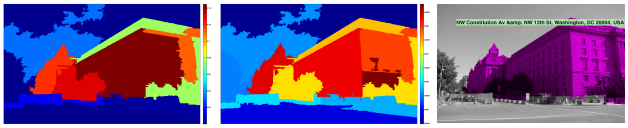


Figure 5: The smoothing process using Random Walks: Left shows the initial scores (of belonging to the semantic segment), computed for each of the super-pixels. Middle image shows the scores after smoothing (color coded), and right shows the resulting semantic segment as a result of thresholding the scores. Right image also shows the geo-semantic label (address) of the building superimposed on the image.

projection and the actual semantic segment in the image, which prevents our method from discovering it. In order to improve the performance of our method for such cases, we perform fusion of segmentation and GIS projections using an iterative method in which, segmentation and projection accuracy improve one another at each iteration. First, projections are generated as explained in section 2.2. The projections are evaluated and weighted based on their reliability, which is defined based on the size of the projection (area of the image covered by the prior), and the color consistency of the overlapping pixels. Using the smoothing step, the corresponding image segments for each of the projections are found. The accuracy of segmentation for a projection with a high reliability score is usually high. Therefore, the obtained segment is a good approximation for the actual location of the objects of interest in the image content. Therefore, we align the projections with their obtained corresponding segments and estimate the error caused by

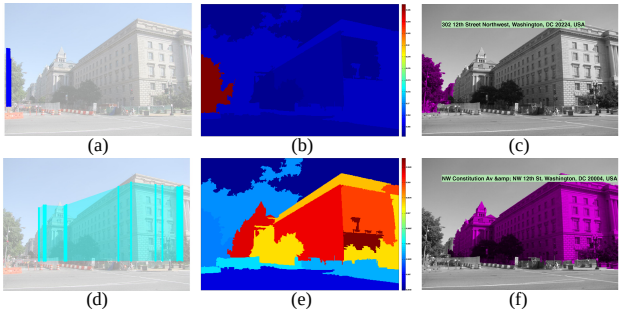


Figure 6: (a) shows a projection with low reliability score. (b) shows the final probability score of the superpixels belonging to that projection (color coded), and (c) shows its resulting semantic segment. Generally, the first row illustrates how an inaccurate projection, can lead to an inaccurate segmentation result. On the other hand (d), (e), and (f) show the same figures but for a more reliable projection. Our method, uses such highly reliable projection-segment pairs to improve the performance for the less reliable ones in an iterative fashion.

the camera location, giving higher weight to reliable projections and lower weight to less reliable ones. By applying that transformation to the projections, we are in fact acquiring a new set of projections which are more aligned with the content of the image. We continue going back and forth to projection and segmentation steps until they agree to a stable set of projections and stable geo-semantic labeling.

2.3.1 Geo-Semantic Segmentation using GIS Projections

We use GIS projections as priors to perform semantic segmentation. We perform multiple random walks in parallel on a fully connected graph which can be solved in linear time and has a closed form solution.

Parallel Random Walks on a fully connected Graph.

Given a set of projections $g_1^k, g_2^k, ..., g_m^k$ at iteration k, we compute x_{ij}^k, a score for super-pixel j belonging to the i^{th} geo-semantic entity. We associate a score to each super-pixel using the following:

$$x_{ij}^k = \frac{\frac{|sp_j \bigcap g_i^k|}{|sp_j|}}{\sum_{j=1}^{n} \frac{|sp_j \bigcap g_i^k|}{|sp_j|}}, \tag{2}$$

where sp_j is the set of pixels belonging to the i^{th} super-pixel. Here, the numerator is defined by the percentage of the pixels which the super-pixel shares with the semantic entity, and the denominator is a normalization factor, to make sure the summation of scores for each semantic entity is equal to 1. This score is stored in matrix $X_{m\times n}$ where m is the number of semantic entities and n is the number of super-pixels. We form a graph $G(V, E)$, in which V is the set of nodes and each node represents one of the super-pixels. E is the set of edges which captures pairwise similarity (transition probability) among neighboring super-pixels. For defining the transition probability we use the color consistency between the super-pixels and their spatial distance to encourage closer super-pixels to have similar labels. We define the pairwise similarity between super-pixel i and j as:

$$s_{ij} = \frac{e^{-\gamma|h_i-h_j|-\eta|l_i-l_j|}}{\sum_{j=1}^{n} e^{-\gamma|h_i-h_j|-\eta|l_i-l_j|}}, \tag{3}$$

η and γ are constants, and h_i, h_j, l_i, and l_j are color histograms, and spatial location of the center of the superpixels i and j respectively (we define the center of the superpixels as the median of all the points that it contains). The numerator captures color similarity and spatial closeness between the two super-pixels, and the denominator is a normalization term, to make sure that the transition probability from node i to all the other nodes sums up to 1.

We perform random walks on the constructed graph and update the scores of the nodes using their pairwise similarity. As a result, visually similar nodes are encouraged to obtain similar labels. Intuitively, if a random node mistakenly gets a high score due to high overlap with a misaligned projection, its score will decrease because of its color inconsistency with its nearby superpixels. On the other hand, if a super-pixel is visually consistent with some highly scored and spatially close superpixels, its score will increase after refinement.

Very similar to [6] and [15], each random walk iteration will update the scores matrix using the following equation:

$$X_{t+1} = \alpha S X_t + (1-\alpha) X_0, \tag{4}$$

in which α is a constant between zero and one, and is set to specify the contribution of the initial score versus the pairwise similarity, X_t is a $m \times n$ matrix containing the scores at iteration t, and S is the similarity matrix whose elements were defined in equation 3. As long as $0 < \alpha < 1$, the converged X_π should satisfy the following equation.

$$X_\pi = \alpha S X_\pi + (1-\alpha) X_0, \tag{5}$$

$$\Rightarrow X_\pi - \alpha S X_\pi = (1-\alpha) X_0, \tag{6}$$

$$\Rightarrow (I - \alpha S) X_\pi = (1-\alpha) X_0, \tag{7}$$

therefore, the updated scores can be obtained using the following closed form solution:

$$X_\pi = (1-\alpha)(I-\alpha S)^{-1} X_0. \tag{8}$$

Since in our alignment method, each alignment iteration is based on a random walk convergence, we can write the same equation for iteration k to $k+1$ as the following:

$$X_{k+1} = (1-\alpha)(I-\alpha S)^{-1} X_k. \quad (9)$$

In other words, X_{k+1} is computed using X_k according to the equation above. The new matrix X_{k+1}, contains the new segmentation scores. By thresholding the updated scores, we assign each super-pixels to at most one label (the label with the highest score). If none of the classes had a high score for a specific super-pixel, it will be labeled as void. Since there is no distinctive cue for distinguishing two merging streets, we merge all of the street projections into one semantic entity in the iterative fusion step and compute the final segmentation for each street based on its overlap with its corresponding final projections. Therefore, if our projections contain 2 buildings (building A and building B) and 2 streets (street C and street D), we solve the problem assuming we have three semantic entities: building A, building B, and street. In which street is the union of the two projections (street C and street D).

2.3.2 Alignment and Updating Projections

Given a set of projections and their corresponding semantic segments, we aim to find a global transformation, mapping the projections to their corresponding semantic segments. As mentioned before, each of the projections are evaluated in terms of reliability using the following:

$$r_i = \mathrm{e}^{\zeta \frac{A_{g_i}}{A_{G_i}}}. \quad (10)$$

Here, r_i is the reliability score of the i^{th} projection, A_{g_i}, is the area of the image covered by its 2D projection on the image in terms of number of pixels, and A_{G_i} is the area covered by the corresponding building facades in real world (estimated by extracting its width from GIS data and a pre-assuming a fixed height), and ζ is a constant number which we set empirically. Intuitively, we want to assign high reliability scores to buildings which have large projections (large A_{g_i}) and are close to the camera (large $\frac{A_{g_i}}{A_{G_i}}$ ratio). Basically, between two buildings with the same size of projection, we assign higher reliability score to the building which is closer to the camera, since it's probability of being occluded in the image content (by vehicles, humans, trees etc.) is less and it is more likely to be completely visible.

Similarly, we evaluate each of the semantic segments based on it's visual consistency. Using the reliability score:

$$r_{seg_i} = \frac{1}{Z} \sum_{\forall p,q \in Y_i} \mathrm{e}^{-\psi |h_p - h_q|}. \quad (11)$$

Y_i is the set of super-pixels associated with the i^{th} entity, and $Z = \binom{|Y_i|}{2}$ is the number of possible pairs of super-pixels.

Finally, we combine these two reliability scores for each pair of projection-segment as:

$$w_i = \frac{r_i r_{seg_i}}{\sum_{i=1}^{m} r_i r_{seg_i}}. \quad (12)$$

We incorporate above in computing the transformation. As a result of this weighting, the prominent and reliable projection-segment correspondences will be highly penalized in case of not being consistent with the transformation. On the other hand, the less reliable pairs will cause smaller penalty and therefore will not have high contribution in the approximation of transformation. For computing the transformation, we generate two points per building: the rightmost, and leftmost point (minimum and maximum x value among the pixels covered by the projection/segment). We generate these points from each segment and each projection and set them as a corresponding pairs of points. The reliability score of each projection-segment is associated to that pair of points. We computed the transformation using weighted least square, which has the standard form of:

$$\mathbf{g^{k+1}} = ((\mathbf{WQ})^T(\mathbf{WQ}))^{-1}\mathbf{Q}^T\mathbf{WY}\mathbf{g^k}. \quad (13)$$

Here W is a diagonal Matrix containing the squared roots of the weights of the corresponding pairs (computed in equation 12), Q contains the points extracted from the projections, and Y contains the corresponding points from the segmentations. We compute the updated matrix obtained in the k^{th} iteration. We apply the computed global transformation to the projections and obtain an updated set of projections.

The iterative process terminates, if either the maximum number of iterations is reached, or the misalignment becomes less than a threshold (when it is considered as converged).

$$\sum_{i=1}^{m} |g_i^{k+1} - g_i^k| < \epsilon. \quad (14)$$

3. Experimental Results

We performed our evaluations using a GIS database of over 10 Sq. kilometers area of Washington DC.[2] Our dataset includes 200 consumer images downloaded from Panoramio and Flickr.

In terms of constant parameters used in our formulation, we set η, γ, α, ψ, and ζ to 0.9, 1.3, 0.85, 5, and 10^-2 respectively. Also, for the convergence criteria we set ϵ to 20.

[2]Dataset available at: http://crcv.ucf.edu/projects/Geosemantic/

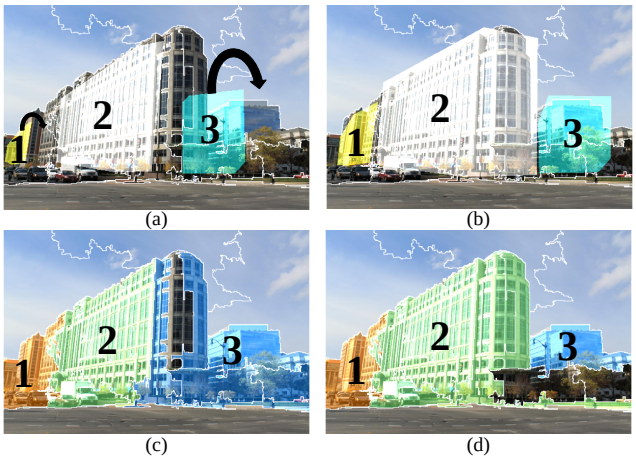


Figure 7: (a) shows the initial projections obtained using the original meta-data provided in the EXIF-tag of the image. (b) shows the projections after the iterative refinement. As illustrated, projection 1 is an example of a misaligned projection and therefore its corresponding segmentation result shown in (c) is inaccurate. On the other hand, 2 is a relatively reliable projection-segmentation pair. (d) shows segmentation results obtained using the updated projections in (b). Comparing the segmentation results for segment 3 before and after alignment, it is easy to see the effect of our updating step. Projection-segment pair 1 and mostly 2 are being used as an anchor point for aligning 3 which is the least accurate.

3.1. Evaluating Semantic Segmentation Accuracy

For evaluating the performance of our semantic segmentation approach, we measured their accuracy based on the standard measures of intersection over union with the groundtruth, and pixel classification accuracy. As it can be observed in table 1. We evaluated the accuracy of semantic and geo-semantic segmentation for two classes: buildings and streets. We evaluated the accuracy for semantic segmentation by classifying super-pixels as either of the three categories of building, street, or void. In addition, we evaluated labeling accuracy in terms of geo-semantic segmentation, which means each segment should be labeled with it's GPS location and address. Therefore, incorrectly classifying building A as building B will have zero geo-semantic accuracy, even if the segments perfectly represent buildings(semantic segmentation has 100% accuracy). It is clear that the iterative refinement method improves the geo-semantic segmentation accuracy significantly. Note that the difference between segmentation and geo-semantic segmentation accuracy is significant before our iterative alignment. However, after alignment, the geo-semantic segmentation has accuracy almost similar to segmentation accuracy. This shows that our refinement scheme is having a notable effect on aligning the projections to their true correspondences in the image content. In our experiments, since there is no distinctive cue for visually distinguishing two merging streets, we merge all of the street projections into one semantic entity in the iterative fusion step and compute the final segmentation for each street based on its overlap with its corresponding final projections (using the score defined in section 2.3.1).

	Intersection/Union		Pixel Classification Accuracy	
	First Itr	Last Itr	First Itr	Last Itr
Buildings (Semantic)	39	40	40	44
Buildings (Geo-Semantic)	31	38	**35**	**44**
Streets (Semantic)	45	45.5	49	50.2
Streets (Geo-Semantic)	36	39	44	47
Total (Semantic)	41	42.3	43	45
Total (Geo-Semantic)	33	38	**38**	**46**

Table 1: Evaluation of the semantic and geo-semantic segmentation accuracy in terms of pixel classification accuracy and intersection over union. First Itr is the segmentation results employing the initial projections. Last Itr is the results after the iterative process and convergence.

3.2. Accuracy vs. Distance and Size

In order to study the suitability of our reliability score, we evaluated the segmentation results for different examples and observed the change in their accuracy versus their size and closeness to the camera. Intuitively, a segment being close to the camera and having a large projection, will prevent it from being hugely misaligned and therefore gives us better accuracy. Figure 8 shows the segmentation accuracy and its changes versus distance and projection size. It can be observed that smaller distance and larger projection size usually lead to higher accuracy. Comparing the changes from (a) to (b), as expected, we can observe that projections with lower reliability score (at larger distance and with smaller projection size), gain notable improvement after the iterative alignment procedure.

3.3. Evaluating Projection Accuracy

Similar to segmentation evaluation, we evaluated the initial projections in terms of alignment. We measured their intersection over union and pixel classification accuracy as if they are results of segmentation. The quantitative results are presented in the table 2, and it can be observed that there is a significant improvement in the alignment accuracy after our iterative refinement procedure. Similar to the segmentation, the improvement is more notable in the exact labeling task. Comparing the numerical results in tables 1 and 2, we can observe the improvement of the segmentation results compared to the projections.

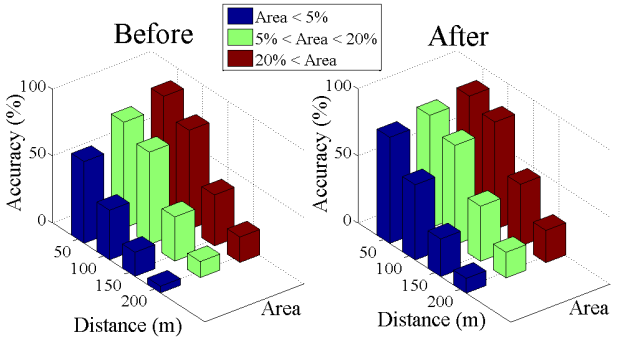


Figure 8: Left shows the initial semantic segmentation accuracy for semantic entities vs their distances to the camera and their projection sizes on the image plane. Right, shows the same but after applying our proposed iterative segmentation-projection refinement method. As it can be seen, our alignment step improves the accuracy of the less confident segments due to properly aligning them to their corresponding entities in the image content, while maintaining similar accuracy in confident semantic entities.

Figure 9 (a) shows the scatter plot of the input vs output accuracy of the projections. It can be seen that once the projections have some meaningful accuracy in the initial iteration, the refinement results can further improve the alignment. However, having a minimum reliability is required, since having completely misaligned projections, might cause completely wrong correspondences in the first iteration, resulting into a wrong transformation and propagating the error toward less accuracy. This issue is discussed more in depth in the section related to failure cases. Figure 9 (b), shows the segmentation accuracy improvement over different iterations. As expected, by comparing the three curves it can be observed that at each step, the higher confident projection-segment pairs are used as an anchor point for aligning the projections with slightly less confidence. For instance, comparing results of the initial segmentation vs the results after two iterations, we can observe that projection-segment pairs with confidence score more than 0.9 are being used for computing the transformation and therefore aligning pairs with confidence within 0.5 and 0.9. Also, doing the same comparison between results of the second iteration versus the final results, we can observe that most of the improvement is done by aligning the least reliable projections (confidence score less than 0.4), using the higher confident projections aligned in previous iterations (score between 0.4 and 1).

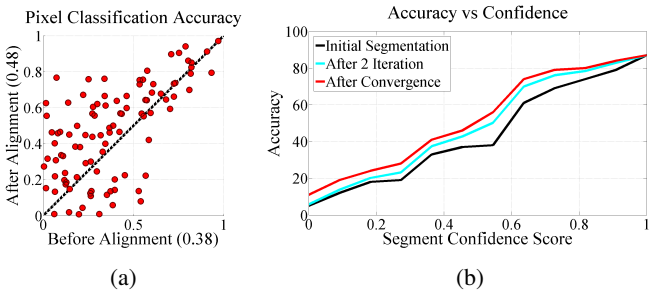


Figure 9: The scatter plot in (a) shows the accuracy improvement due to our updated projections. Each point represents the overall segmentation accuracy in one of the images before and after our iterative method. (b) shows the amount of improvement gained in each iteration vs the reliability of the semantic objects.

	Intersection/Union		Pixel Classification Accuracy	
	First Itr	Last Itr	First Itr	Last Itr
Buildings (Semantic)	14	24	16	25
Buildings (Geo-Semantic)	6	11.6	**8**	**22**
Streets (Semantic)	22	23	23	25
Streets (Geo-Semantic)	18	22	19	22.6
Total (Semantic)	17	23	17	24
Total (Geo-Semantic)	10	16.2	**13**	**22.5**

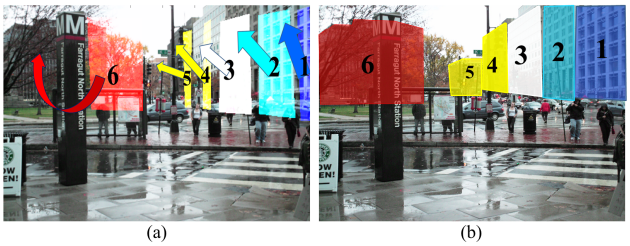
Table 2: Evaluation of the projection alignment by considering it as semantic segmentation results. First Itr column shows the results using the initial projections formed using the original meta-data(EXIF header). Last Itr column shows the results after the iterative process and convergence.

4. Failure Analysis

The projections for an example of the failure cases is illustrated in figure 10. The arrows show the correct location of the corresponding buildings in the image. It can be observed, that due to misalignment, the projection belonging to building number 2, is perfectly matching with building number 1, and the projection of building number 3 has a high overlap with building number 2. Due to this initial mismatching, all of our computations lead to a random result.

5. Conclusion

We proposed a method for performing semantic segmentation by leveraging the information in GIS databases. We segment the image into several super-pixels, project the GIS outlines to the image plane, and fuse two source of information to get an initial semantic segmentation. After that, we use our initial set of semantic segments for refining and updating the set of projections. Our experiments show that the iterative approach has a notable impact in improving the



(a) (b)

Figure 10: (a) shows a failure case where misalignment forms wrong correspondences. The true match for each of the buildings is shown by the arrow. (b) is the ideal location of each projection.

overall segmentation performance.

References

[1] S. Ardeshir, A. R. Zamir, A. Torroella, and M. Shah. Gis-assisted object detection and geospatial localization. In *European Conference on Computer Vision–ECCV 2014, pages 602-617. Springer.* 2

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Computer Vision–ECCV 2008*, pages 44–57. Springer, 2008. 1, 2

[3] X. He, R. S. Zemel, and M. Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, volume 2, pages II–695. IEEE, 2004. 2

[4] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up., 2005. 1

[5] C. C. Lerma and J. Kosecka. Semantic segmentation of urban environments into object and background categories. Technical report, DTIC Document, 2013. 2

[6] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *International Conference on Multimedia*, 2009. 4

[7] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011. 2

[8] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. *Procedural modeling of buildings*, volume 25. ACM, 2006. 2

[9] P. Musialski, M. Wimmer, and P. Wonka. Interactive coherence-based façade modeling. In *Computer Graphics Forum*, volume 31, pages 661–670. Wiley Online Library, 2012. 2

[10] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3105–3112. IEEE, 2010. 1, 2

[11] H. Uchiyama, H. Saito, M. Servieres, G. Moreau, and E. C. d. N.-C. IRSTV. Ar gis on a physical map based on map image retrieval using llah tracking. In *MVA*, pages 382–385, 2009. 2

[12] L. Wang and U. Neumann. A robust approach for automatic registration of aerial images with untextured aerial lidar data. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2623–2630, June 2009. 2

[13] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, and L. Quan. Image-based façade modeling. In *ACM Transactions on Graphics (TOG)*, volume 27, page 161. ACM, 2008. 2

[14] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, and L. Quan. Image-based street-side city modeling. *ACM Transactions on Graphics (TOG)*, 28(5):114, 2009. 1, 2

[15] A. R. Zamir, S. Ardeshir, and M. Shah. Gps-tag refinement using random walks with an adaptive damping factor. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[16] H. Zhang, J. Xiao, and L. Quan. Supervised label transfer for semantic segmentation of street scenes. In *Computer Vision–ECCV 2010*, pages 561–574. Springer, 2010. 2

[17] P. Zhao, T. Fang, J. Xiao, H. Zhang, Q. Zhao, and L. Quan. Rectilinear parsing of architecture in urban environment. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 342–349. IEEE, 2010. 2