LETTER

New POI Construction with Street-Level Imagery*

Chillo GA[†], ***Member***, **Jeongho LEE**[†], **Won Hee LEE**[††], ***and*** **Kiyun YU**[†a)], ***Nonmembers***

SUMMARY We present a novel point of interest (POI) construction approach based on street-level imagery (SLI) such as Google StreetView. Our method consists of: (1) the creation of a conflation map between an SLI trace and a vector map; (2) the detection of the corresponding buildings between the SLI scene and the conflation map; and (3) POI name extraction from a signboard in the SLI scene by user-interactive text recognition. Finally, a POI is generated through a combination of the POI name and attributes of the building object on a vector map. The proposed method showed recall of 92.99% and precision of 97.10% for real-world POIs.

key words: *point of interest, street-level imagery, conflation, text extraction*

1. Introduction

In location-based services, a query based on a given location, such as “Where is the nearest OOO?”, is a fundamental function. A point of interest (POI), which is a specific geographic location closely related to people’s lives, such as a restaurant or a bank [1], makes such queries possible. Whether or not users are able to accurately find the POI they want depends on the quantity and quality of a POI database.

A POI database has generally been constructed by carrying out field surveys or integrating multiple third-party sources such as the Yellow Pages. A field survey is the most accurate method for constructing POIs, but it is time-consuming and labor-intensive. As the third-party sources include store name, address, and so on, constructing POIs based on them is relatively inexpensive and easy. However, it has some problems of positional error introduced in the geocoding process.

Recently, POI construction based on the Web 2.0 paradigm has been studied. Goodchild [2] highlighted the potential use of citizens around the world as voluntary sensors to create a global patchwork of geographic information. Some researchers have tried to extract new and meaningful POIs automatically based on geotagging information registered by users [3], [4]. This can be a very cost-effective way to construct POIs. It is not possible, however, to ensure the quality of the user-created data because of misspellings, positional errors, or biased information. This requires a review process before the distribution of the POIs.

We propose a new method of POI construction based on street-level imagery (SLI). An SLI is a panoramic image taken by a side-looking camera at street level, and Web portals offer free SLI services such as Google StreetView. SLI has a distinctive strength in POI construction because it gives us the same view as we would see on the street in the real world. In particular, a signboard in an SLI scene implies the existence of a POI, and the text on the signboard can be utilized as a POI name. We construct a POI by combining the POI name extracted from the SLI scene and attribute information, such as the location, address, and zip code, of the corresponding building object from a vector map conflated with the SLI trace. This method can be used to fill the gaps in previous methods.

2. SLI-Based POI Construction Method

Our method includes the creation of a conflation map between an SLI trace and a vector map, the detection of the corresponding buildings between the SLI scene and the conflation map, and POI name extraction from a signboard in the SLI scene by user-interactive text recognition (Fig. 1).

2.1 Conflating SLI Trace with Building Layer

While a vector map is digitized from aerial images or satellite images, an SLI is acquired by taking pictures and pairing with the camera position measured by global positioning system (GPS) and inertial navigation system (INS). When two datasets built by different methods are simply overlaid, spatial inconsistencies may occur [5] (Fig. 3 (a)).

To remove spatial inconsistencies, we create a conflation map between the SLI trace and the vector map. Through the conflation process, the camera position of an SLI is relocated correctly on the vector map, and a building shown in a certain scene can correspond to its counterpart on the vector map. The conflation uses three subprocesses: road intersection matching, control point pair (CPP) filtering, and alignment.

The SLI trace reflects the shape of the road because it is taken while driving or walking on the road. Road intersections capture the main features of the road network, and they are distributed evenly. Thus, we utilize the road intersections of the two datasets as candidates for control points in conflation maps. One geometric condition and two topological conditions shown in Eq. (1) are used for searching for the

Manuscript received June 12, 2012.

Manuscript revised August 27, 2012.

[†] The authors are with Seoul National University, Gwanak-gu, Seoul, 151–744, Republic of Korea.

[††] The author is with Chosun University, Dong-gu, Gwangju, 501–759, Republic of Korea.

[*] This work was supported by a grant from High-tech Urban Development Program (11 High-Tech Urban G10).

a) E-mail: kiyun@snu.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E96.D.129

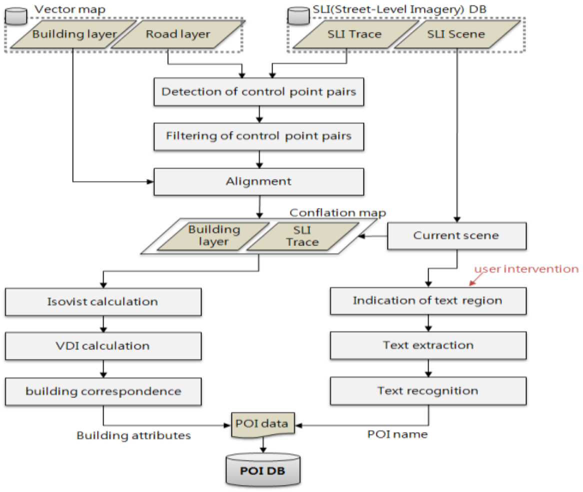
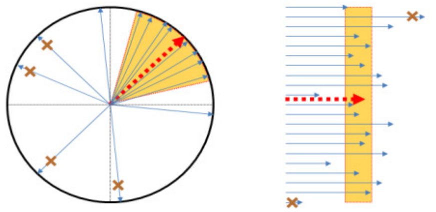
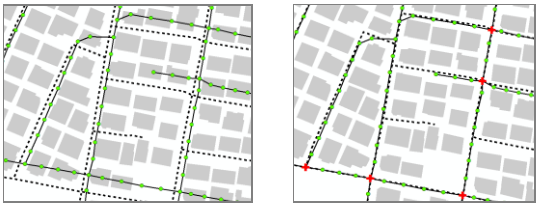


Fig. 1 Overall structure of proposed method.



(a) Median angle filter (b) Median length filter

Fig. 2 Filtering of inaccurate CPP vectors.



a) Simple overlay (b) Conflation map

Fig. 3 Creation of a conflation map.

same intersections in the two datasets. First, we find candidate intersections ($\mathbf{I}_j^R$) on the road layer of the vector map only within the threshold distance (Δd) from the SLI trace intersection ($\mathbf{I}_i^S$). We set the threshold distance to twice the average distance between pairs that were prematched with the initial large-enough value. Then, two topological conditions are applied for more robust matching; one is the degree of intersection (D), which is the number of road segments connected to the intersection, and the other is the minimum of the sum of angle differences between the road segments ($\mathbf{IS}_{im}^S$ and $\mathbf{IS}_{jn}^R$). Only intersections that meet all the conditions are detected as CPPs:

$$CPPs = \left\{ (\mathbf{I}_i^S, \mathbf{I}_j^R) \;\middle|\; \begin{array}{l} \|\mathbf{I}_i^S - \mathbf{I}_j^R\| < \Delta d \\ D(\mathbf{I}_i^S) = D(\mathbf{I}_j^R) \\ j = argmin\left(\sum \cos\left(\dfrac{\mathbf{IS}_{im}^S \bullet \mathbf{IS}_{jn}^R}{\|\mathbf{IS}_{im}^S\| \, \|\mathbf{IS}_{jn}^R\|} \right)^{-1} \right) \end{array} \right\} \tag{1}$$

Some detected CPPs may be mismatched. Because inaccurate CPPs reduce the accuracy of conflation between two datasets, mismatched CPPs must be removed before the alignment process. Most of the detected CPPs show similar patterns locally in directions and magnitudes. Thus, we filter out mismatched CPPs and out-of-pattern CPPs by analyzing the trend of CPPs for more accurate alignment.

For filtering, we applied a median flow filter (MFF) [6] consisting of a median angle filter (MAF) and a median length filter (MLF). First, each CPP forms a vector from a road intersection to an SLI trace intersection. The MAF begins by shifting all the vectors so that their origins are all at the same point, as shown in Fig. 2 (a). Then, the median angle vector is calculated as representative of the CPP vector trend after removing outliers, which are points outside the 1.5 interquartile (Q3–Q1) range. Only CPPs with deviations within the permissible range ($\mu \pm 2\sigma$) remain for the MLF, where μ and σ are the mean and standard deviation, respectively, calculated using the differences from the median angle vector. In the MLF, CPP vectors out of the trend are removed in the same way as MAF, but using lengths instead of angles. Finally, only CPPs that have similar patterns remain, and they are used in the alignment process as accurate CPPs.

Because each point of an accurate CPP indicates the same intersection in the two datasets, spatial inconsistencies can be removed by performing local alignments. The space is first partitioned into small pieces with the CPPs by Delaunay triangulation, which is appropriate for local adjustments [7]. Then, alignment is performed by rubber-sheeting between the corresponding triangles. As shown in Fig. 3 (b), a spatial inconsistency-removed conflation map of the SLI trace and the building layer is finally produced by alignment of the SLI trace and the road layer.

2.2 Correspondence of Building Object

We can find which building object in the conflation map corresponds to the building shown in an SLI scene by utilizing parameters such as viewing position, viewing direction, and field of view. In this study, we propose a method of finding a building object in a conflation map corresponding to the most visually dominant building in the current SLI scene. A visually dominant building is one close to the viewing position and similar to the viewing direction, which has the least distortion and occupies the largest visible area in the scene. Human beings are easily able to identify the visually dominant building in an SLI scene. We apply the isovist technique [8] to search for a building object automatically from the conflation map corresponding to the building in the SLI scene. The isovist is a two-dimensional visible area (the white area in Fig. 4) composed of the set of all points visible from a single viewpoint. We propose a visual dominance

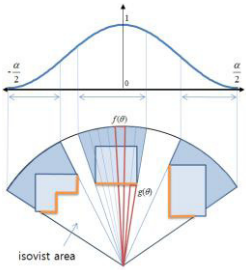


Fig. 4 Concept of VDI calculation.

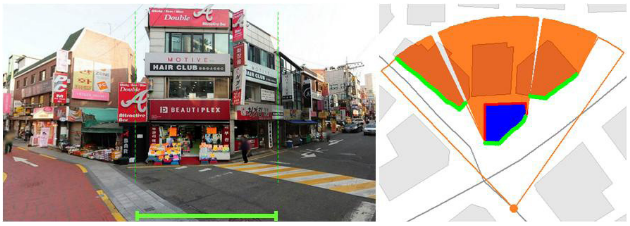


Fig. 5 Example of building correspondence.

index (VDI) based on the area occluded by building objects, which is a complementary set to the isovist area. The visually dominant building object is selected as the counterpart of a building in the current scene by the VDI.

The first step of the VDI calculation is to explore the directional isovist area by utilizing the viewing position, viewing direction, and field of view of an SLI scene, as shown in Fig. 4. The line where the isovist boundary meets with a building boundary is a facade line of the building seen from the viewing position. The size of the area occluded by the facade line is proportional to the building size seen from the viewing position. As shown in Fig. 4, the VDI is calculated by applying the cosine weight function (Eq. (2)), which has a weight value from 0 to 1 depending on the similarity to the viewing direction, to the occluded area. In Eq. (2), α is the field of view, θ is the sampling interval, ω is the weight for the viewing direction, and $f(\theta)$ and $g(\theta)$ are the areas of the sector to the region boundary and to the facade line, respectively. As shown in Fig. 5, when a user sees an SLI scene, the building object with the largest VDI is intuitively located at the center of the scene. In addition, that building has the least distortion and the largest visible area in the scene, and is therefore selected as a target building for extracting a POI.

$$
\begin{aligned}
&VDI_i = \sum_{j=-\alpha/2}^{\alpha/2} [(f_i(\theta_j) - g_i(\theta_j)) \cdot \omega_j] \\
&\omega = \frac{1}{2}\cos\left(\frac{2\pi}{\alpha}\theta\right) + \frac{1}{2} \\
&buildig\ correspondence = MAX|VDI_i| \qquad (2)
\end{aligned}
$$

2.3 User-Interactive POI Name Extraction

After the previous processes, the system can identify the building shown in an SLI scene and the corresponding building object in the conflation map. Then the POI is created by combining the POI name extracted from a signboard in an SLI scene and attributes of the corresponding building object in the conflation map.

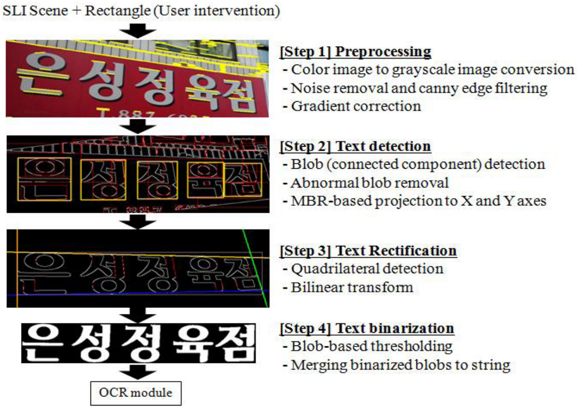

SLI Scene + Rectangle (User intervention)
[Step 1] Preprocessing
- Color image to grayscale image conversion
- Noise removal and canny edge filtering
- Gradient correction
[Step 2] Text detection
- Blob (connected component) detection
- Abnormal blob removal
- MBR-based projection to X and Y axes
[Step 3] Text Rectification
- Quadrilateral detection
- Bilinear transform
[Step 4] Text binarization
- Blob-based thresholding
- Merging binarized blobs to string
OCR module
은성정육점

Fig. 6 Process of user-interactive POI name extraction.

Despite many attempts at extracting text automatically from natural scene images like SLI, the extraction rate is still not satisfactory. The reason is that scene images include very complex backgrounds and have many noisy elements [9].

In particular, text candidate region localization is the most challenging step in scene text extraction process, and errors in this step are propagated throughout all other processes. If a text candidate region is known, it is possible to effectively reduce the region for searching for text. In addition, extraction accuracy can be increased because the text color is homogeneous but significantly different from the nontext color within the text candidate region. Therefore, we proposed a user-interactive POI name extraction method in which the user approximately indicates the rectangular text region by simple mouse dragging. We improved the extraction method to be robust to distortion within the SLI by applying the edge-based method of Milevskiy and Ha [10], which utilizes the characteristics of signboard text.

If an extracted text is recognized correctly, it is utilized directly as the POI name. If not, the POI name must be input manually. More detailed steps of POI name extraction are summarized below, and the major result of each step is shown in Fig. 6.

Step 1 (Preprocessing): We perform some preprocessing such as grayscale conversion, noise removal, and edge filtering. Then, the tiled image taken from an arbitrary angle is rotated to be corrected depending on the average direction of straight line components (±30° from the horizontal line).

Step 2 (Text detection): Blobs (connected components) are detected in the rotated edge image. Because each blob we want to detect is a character, the abnormally shaped blobs are excluded. To extract the main string and character-based blobs, a list of minimum bounding rectangles (MBRs) of the remaining blobs is projected to the horizontal and vertical axes, and each profile is segmented.

Step 3 (Text rectification): Perspective distortion of SLI makes it more difficult to recognize text accurately. To correct the distortion, we search the quadrilateral region based on the upper and lower lines of character-based blobs and vertical peripheral straight lines of character components in character-based blobs. Then we rectify the quadrilateral to a rectangle by approximating three-dimensional nonlinear distortions to two-dimensional rectangles using a bilinear transform.

Step 4 (Text binarization): We can segment character and background using a global thresholding method because the color variance of character and background is distinctive in each MBR of the character-based blobs.

Finally, the segmented text string becomes the input data of the optical character recognition module.

3. Evaluation

We implemented a prototype system to test the usability of our proposed method. Daum Road View (Open API)[†] and KLIS-rn[††] were used as the SLI and vector map, respectively. The Road View covers almost the entire territory of South Korea, and its trace is a series of points acquired at about every 10 m. The KLIS-rn accommodates the new address system based on street names and has a multilayered data structure with point, polyline, and polygon primitives; only the road (polyline) and building (polygon) layers were used. We constructed POIs (SLI–POIs) in sample areas by the prototype system and then assessed the results for 757 reference POIs constructed by field surveys. Recall, precision, and F-measure were used as evaluation measures.

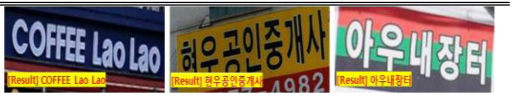
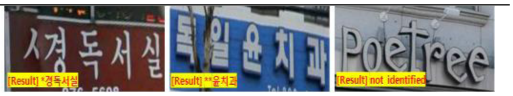
Table 1 summarizes the results of POI construction including the automatic and manual input of POI names. SLI–POIs showed recall of 92.99% and precision of 97.10% for the reference POIs. The results show that our method based on SLI and vector maps can reflect most real-world POIs with very high accuracy. We could not, however, extract about 32 POIs (about 4%); their signboards were not identified because of serious distortion and occlusion in the SLI. The precision error of SLI–POIs is caused by a correspondence error between the visually dominant building in an SLI scene and its counterpart in the conflation map. In such a case, the created POI is tagged as another building object.

The success rate of automatic POI name extraction was not high enough in our experiment. As shown in Table 2, about half of the recognition failures were caused by abnormal signboards: signboards with mosaic error; occluded by trees or other facilities; or with very complicated design. However, signboards with relatively normal shapes showed an extraction rate of 73%. This rate could be increased by further post-processing such as string matching of partially recognized text and a signboard name database built beforehand.

Table 1 Results of POI construction.

	Recall	Precision	F-measure
SLI–POIs	92.99%	97.10%	95.00%

Table 2 Results of user-interactive POI name extraction.

Normal signboards	 COFFEE Lao Lao [Result] COFFEE Lao Lao 현우공인중개사 4982 [Result] 현우공인중개사 아우내장터 [Result] 아우내장터 Extraction rate : 262/359
Abnormal signboards	 경독서실 [Result] *경독서실 독일윤치과 [Result] **윤치과 PoeTree [Result] not identified Extraction rate : 39/366

In addition, to evaluate the practicability of our approach, we compared our results with the POIs in a commercial navigation system. The comparison showed that the proposed method generated a more abundant and reliable POI database than the navigation system (recall of 77.41%, precision of 83.95%). Although the comparison against navigation POIs is not absolutely fair because of their different criteria, the test results reveal that our method is promising for constructing POIs effectively.

4. Conclusions

The main contribution of this paper is a new POI construction approach based on the SLI service offered by a portal. More precisely, we first created a conflation map of an SLI trace and the building layer with spatial inconsistencies removed to accurately locate the SLI-viewing position on the vector map. We proposed the VDI based on the area occluded by a building object using isovist for correspondence between a visually dominant building in the SLI and its counterpart in the conflation map. Finally, we improved the POI name-extraction method to be more robust to noise and distortion in an SLI by a simple user interaction.

The experimental results showed the potential of the method as an alternative way of constructing POIs effectively. This method offers several benefits to POI service providers, including an indoor, weatherproof process and relatively low cost by using existing SLI and vector maps. It can be also used when field exploration is not feasible.

[†] http://dna.daum.net/apis/maps/v3

[††] The dataset is available at http://juso.go.kr

References

[1] X. Zhu and C. Zhou, "POI inquiries and data update based on LBS," International Symposium on Information Engineering and Electronic Commerce, Ternopil, Ukraine, pp.730–734, May 2009.

[2] M.F. Goodchild, "Citizens as voluntary sensors: Spatial data infrastructure in the world of Web 2.0," Int. J. Spat. Data Infrastruct. Res., vol.2, no.1, pp.24–32, 2007.

[3] L. Mummidi and J. Krumm, "Discovering points of interest from users' map annotations," GeoJournal, vol.72, no.3, pp.215–227, 2008.

[4] S. Ahern, M. Naaman, R. Nair, and J. Yang, "World explorer: Visualizing aggregate data from unstructured text in geo-referenced collections," Proc. 7th ACM/IEEE-CS joint Conf. on Digital libraries, Vancouver, Canada, pp.1–10, June 2007.

[5] A. Samal, S.C. Seth, and K. Cueto, "A feature-based approach to conflation of geospatial sources," Int. J. Geogr. Inf. Sci., vol.18, no.5, pp.459–489, 2004.

[6] L. Chen, X. Wang, and X. Liang, "An effective video stitching method," International Conference on Computer Design and Applications, Qinhuangdao, China, pp.25–27, June 2010.

[7] A. Saalfeld, "Conflation: Automated map compilation," Int. J. Geogr. Inf. Sci., vol.2, no.3, pp.217–228, 1988.

[8] M. Benedikt, "To take hold of space: Isovist and isovist field," Environ. Plan. B, vol.6, pp.47–65, 1979.

[9] J. Jung, S. Lee, M. Cho, and J. Kim, "Touch TT: Scene text extractor using touchscreen interface," ETRI J., vol.33, no.1, pp.78–88, 2011.

[10] I. Milevskiy and J. Ha, "A fast algorithm for Korean text extraction and segmentation from subway signboard images utilizing smartphone sensors," J. Comput. Sci. Eng., vol.5, no.3, pp.161–166, 2011.