A Training-by-Example Approach for Symbol Spotting from Raster Maps

Yao-Yi Chiang, Phokgoan Chioh, Sima Moghaddam

University of Southern California, Spatial Sciences Institute, 3616 Trousdale Parkway, AHF B55
Los Angeles, CA 90089-0374
Email: {yaoyic; chioh; khashkha}@usc.edu

1. Introduction

Graphic symbols in maps depict important and interesting geographic phenomena, such as wetlands (Figure 1). The descriptive metadata of these symbols can be found in map labels or keys; however, labels are only capable of displaying limited information (e.g., place names) and keys provide categorical information. For example, Figure 2 shows a group of unique buildings in a U.S. Geological Survey (USGS) topographic map but the map does not provide any information about these buildings (e.g., names). Figure 3 shows a scanned map of Baghdad, Iraq where most symbols are labeled with place names but retrieving and integrating further information (e.g., addresses) of these places from other sources requires additional efforts such as using the place names and locations to search on Wikipedia or DBpedia (a structured version of Wikipedia).

In this paper, we present a training-by-example approach for spotting graphic symbols in raster maps. We demonstrate that our approach efficiently enables automatic linkages between DBpedia records and locations in a map. Traditional document analysis techniques for spotting map symbols generally require a large amount of training datasets, the presence of map keys (e.g., Samet and Soffer, 1998), or ad-hoc preprocessing steps (e.g., image thresholding) (Chiang et al, 2014; Lladós et al., 2002). In contrast, our approach takes only one user-selected symbol example to extract the locations of all symbols that have similar graphical appearance to the example.

Figure 1: Wetlands in a historical USGS topographic map (Miami, Florida, circa 1958).

Figure 2: Buildings of the Park La Brea Apartment in a historical USGS topographic map (Hollywood, California, circa 1953) (left) and Google Earth imagery (right).

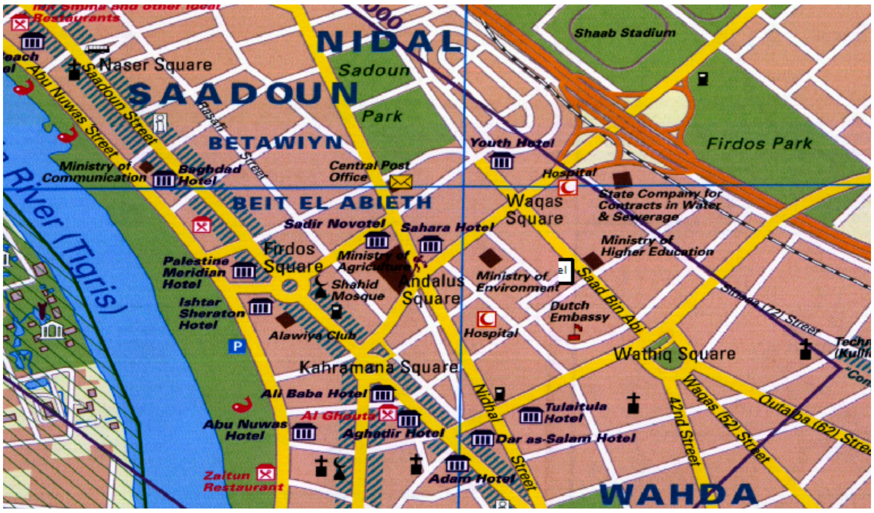


Figure 3: Symbols labeled with place names in a scanned Baghdad map.

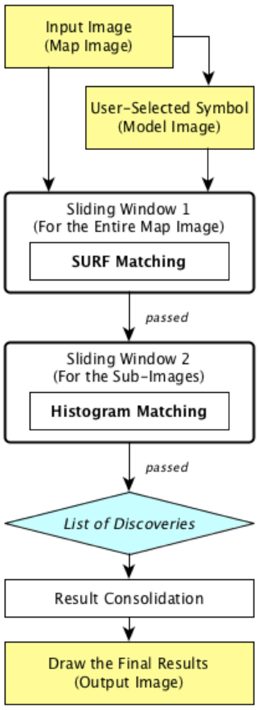


Figure 4: The SymbolRecognizer framework.

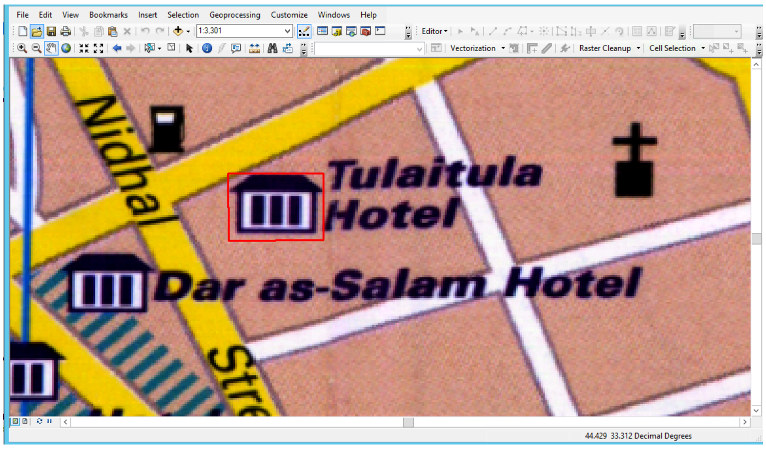


Figure 5: A user-selected symbol example.

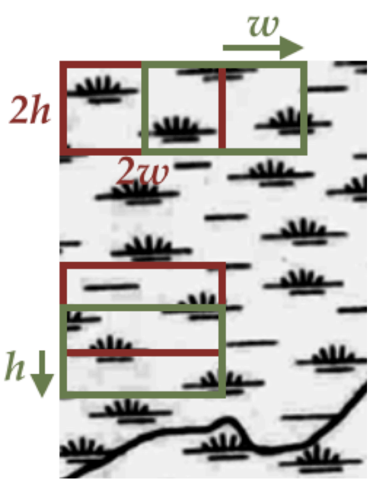


Figure 6: The SURF matching sliding window.

2. Symbol Spotting

This section presents our symbol spotting approach called SymbolRecognizer (Figure 4). A model image is an image that covers a user-selected example in the input map (the red rectangle in Figure 5). The recognition task is to search the map for symbols that matches the model (i.e., target symbols). SymbolRecognizer utilizes a two-phase process: (1) Using the SURF (Speeded Up Robust Features) matching (Lowe, 1999; Bay et al., 2006) to efficiently identify the local regions (sub-images) where a target symbol might present and (2) Using pixel intensity distribution (with histogram matching) to verify the presence of a target symbol in each sub-image.

2.2 SURF (Speeded Up Robust Features) Matching

Considering a model image with width and height equal to w and h pixels, in the first phase, SymbolRecognizer uses a sliding window of the size equal to $2w$ and $2h$ pixels and moves w or h pixels in the horizontal or vertical direction to scan through the entire input map (Figure 6). The size of the sliding window guarantees that every target symbol is covered completely in at least one window (a sub-image). At each position of the sliding window, SymbolRecognizer detects the SURF features from the sub-image and compares the detected features with the SURF features of the model image. If the comparison result contains a high number of matched features, the sub-image is highly likely to contain a target symbol (see Lowe (1999) for details of this object recognition procedure) and is passed to the next phase.

2.2 Histogram Matching

The SURF matching is efficient and widely used to recognize real world objects in photography or videos, but map symbols have simpler shapes (than real world objects) and are relatively small, which can cause frequent false positives in the matching results. Therefore, SymbolRecognizer compares the pixel intensity distributions of the model image and each sub-image that passes the SURF matching to determine whether or not a target symbol presents and to extract the symbol location.

For each sub-image that passes the first phase, SymbolRecognizer uses the model image to scan from the top-left corner and moves *one* pixel in the horizontal or vertical directions (i.e., Sliding Window 2 in Figure 4). Each scanning position records a similarity score calculated using the correlation of the grayscale histogram of the model image (H^{model}) and the grayscale histogram of the overlapping image patch (the overlapping area between the model image and the sub-image) (H^{patch}). The correlation is defined as follows:

$$Similarity\ Score = \frac{\sum_{i=0}^{255}(H_i^{model} - \overline{H^{model}})(H_i^{patch} - \overline{H^{patch}})}{\sqrt{\sum_{i=0}^{255}(H_i^{model} - \overline{H^{model}})^2 \sum_{i=0}^{255}(H_i^{patch} - \overline{H^{patch}})^2}}$$

SymbolRecognizer uses an empirically set threshold of 90% on the similarity score to filter out the sub-images that do not contain a target symbol and to locate the symbol location. If none of the scanning positions in a sub-image has a similarity score higher than 90%, the sub-image does not contain a target symbol; otherwise, the scanning position that has the highest similarity score (in a sub-image) is the detected location of a target symbol.

2.3 Result Consolidation

A target symbol can be detected in overlapping sub-images during the SURF matching since the sliding window can cover a symbol more than once (Figure 7(a)). To consolidate the results, if overlapping sub-images contain multiple target symbols, SymbolRecognizer keeps only the target symbol with the highest histogram matching score (Figure 7(b)).

(a) Before consolidation. (b) After consolidation.

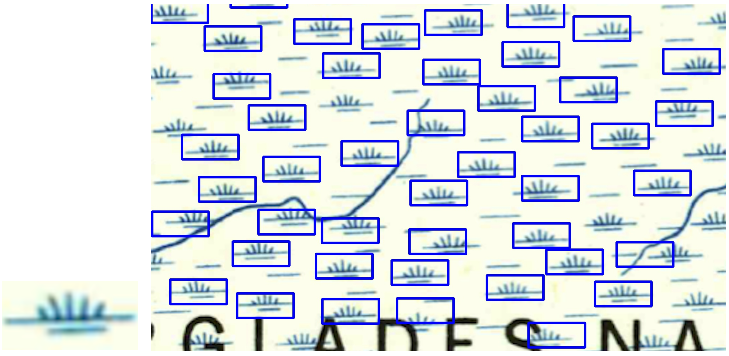
Figure 7: Result consolidation for overlapping sub-images.

3. Preliminary Results and Discussion

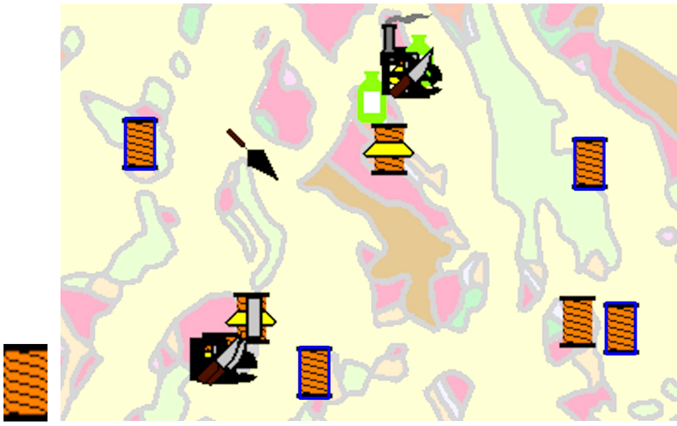
We implemented SymbolRecognizer in our map processing system, Strabo, as an Esri ArcMap plugin and tested the plugin with maps from four sources (Figure 3, Figure 8, and Table 1). For each test map, the user selected one sample symbol and Strabo automatically processed the sample to find other symbols in the map.

The results showed promising extraction precision (with only a few false positives). The USGS Hollywood map had the lowest extraction precision since the target symbols (the Park La Brea apartment buildings) are in different orientations. Although the SURF matching is rotation invariant, the histogram matching results could be compromised if the image patch did not cover the entire symbol of different orientations in the sub-image. All other test maps that contain symbols in the same orientation had more than 97% extraction precision.

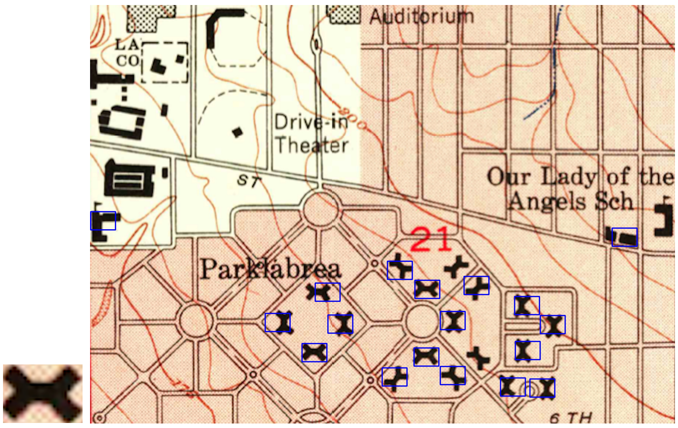
Considering the extraction recall, significantly overlapped features were the main cause of true negatives. Figure 9 shows two examples of overlapping symbols in the USGS Mine and Mineral map. The overlapping symbol to the right was detected because only a small portion of the symbol was overlapped by another symbol. The symbol to the left (Figure 9) was not detected since the entire symbol was almost covered by other symbols. The USGS Mine and Mineral map contains 12 (out of 25) significantly overlapped symbols and hence the extraction recall was the lowest among the test maps. All other test maps had more than 83% extraction recall.



(a) A historical USGS topographic map (Miami, Florida, 1958).



(b) The USGS Mine and Mineral Processing Plant Locations map.



(c) A historical USGS topographic map (Hollywood, California, circa 1953).

Figure 8: Model images (left) and sample results where blue rectangles are the recognized locations (right).

Table 1. Recognition Results.

Source	Image Size (pixels)	# of Target Symbols	Precision	Recall
USGS Miami (1958)	409x438	87	97.33%	83.91%
USGS Mine and Mineral	2465x2150	25	100%	48%
USGS Hollywood (1953)	554x396	18	88.89%	88.89%
Gecko Maps, Baghdad	5104x2616	17	100%	88.23%

Figure 9: Examples of overlapping symbols.

Figure 10 shows the Baghdad map with the identified symbols linked with DBpedia URIs. Once the symbols were identified, Strabo queried the DBpedia SPARQL endpoint to retrieve the nearest DBpedia entries to individual symbol locations. These entries had various DBpedia types such as Museum, Embassy, School, and Hotel. Since the identified symbols represented places in the same category, Strabo first detected the most popular category among the retrieved entries and only linked a symbol to DBpedia if the closest entry of the symbol was in the popular category. In this test area, the most popular category is Hotel and there were only four hotel entries on DBpedia.

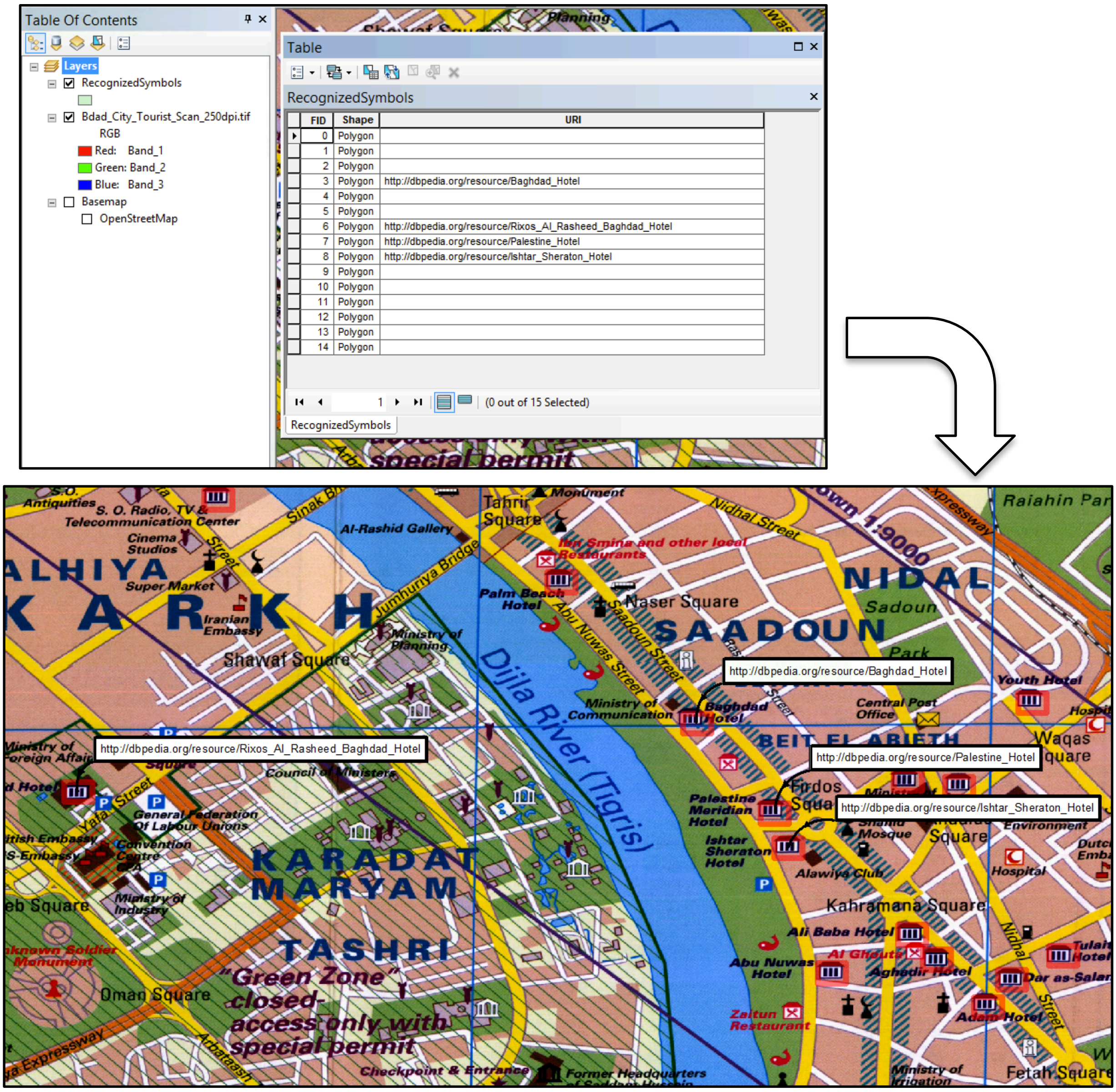


Figure 10: Automatic linkages between map locations and DBpedia records.

4. Summary and Outlook

We presented a training-by-example approach for symbol spotting from raster maps. Our approach requires very little user effort and can handle various types of maps and symbols. We plan to test on more symbol types and further investigate automatic methods to link the extracted symbol locations to other sources.

References

Bay, H., Tuytelaars, T., and Gool, L. V., 2006, SURF: Speeded up robust features. In the Proceedings of the 9th *ECCV*, pages 404–417.

Chiang, Y.-Y., Leyk, S., and Knoblock, C. A., 2014, A survey of digital map processing techniques. *ACM Computing Surveys*. doi: 10.1145/2557423, in press.

Lladós, J., Valveny, E., Sánchez, G., Martí, E., 2002, Symbol recognition: Current advances and perspectives. In *GREC*, pages 104–127.

Lowe, D. G., 1999, Object recognition from local scale-invariant features. In ICCV, vol. 2, pages 1150–1157,

Samet, H. and Soffer, A., 1998, Magellan: Map acquisition of geographic labels by legend analysis. *IJDAR*, 1(2): 89–101.