Identifying Maps on the World Wide Web

Matthew Michelson, Aman Goel, and Craig A. Knoblock*

University of Southern California,
Information Sciences Institute,
Marina del Rey, CA 90292, USA,
{michelso,amangoel,knoblock}@isi.edu,
WWW home page: http://www.isi.edu/integration

Abstract. This paper presents an automatic approach to mining collections of maps from the Web. Our method harvests images from the Web and then classifies them as maps or non-maps by comparing them to previously classified map and non-map images using methods from Content-Based Image Retrieval (CBIR). Our approach outperforms the accuracy of the previous approach by 20% in F_1-measure. Further, our method is more scalable and less costly than previous approaches that rely on more traditional machine learning techniques.

1 Introduction

As more and more information comes online, disciplines ranging from medicine to geography benefit from the proliferation of freely available data. For example, in the field of geography, huge repositories of maps can be built by harvesting maps from all of the images on the Web. Once harvested, these maps can then be aligned to known geographic sources in a process known as "georeferencing" [1–3]. Previous work georeferences maps to satellite images by first automatically extracting the roads and intersections from the map image [4–6], and then using those roads and intersections to automatically align the map to the satellite image in a process known as conflation [1, 2]. Once georeferenced, the map collection can be queried using the same spatial queries used on the satellite images, such as queries by region, latitude/longitude, street name, etc. Further, not only are the maps returned for the given query, but they can be put in context by overlaying the map images on top of the satellite image. An example of a conflated map with a satellite image is given in Figure 1.

* This research is based upon work supported in part by the National Science Foundation under award number IIS-0324955, in part by the United States Air Force under contract number FA9550-08-C-0010, and in part by a gift from Microsoft. The U.S. Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

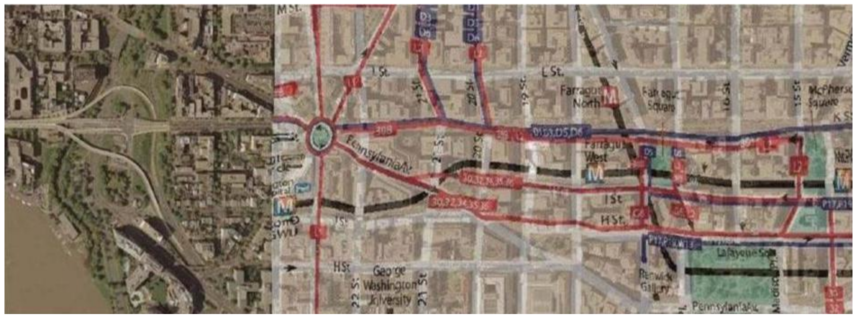

Fig. 1. A bus map for Washington, DC conflated with a satellite image of the city.

In this paper we focus on the first of these problems, automatically harvesting maps from the freely available images on the Web. However, given that there is no filter for publishing on the Web, a major challenge exists in separating the map images from the other types of images. Manually sifting through the images is both tedious and costly, especially given the huge amounts of data that must be examined. In this paper we address the problem of *automatically* separating geographic maps from other images collected on the World Wide Web. An autonomous approach not only eases the cost associated with identifying maps, but it also provides an easy and scalable approach to growing a collection of maps over time. Once collected, such maps are useful for data integration, intelligence analysis and geographic information systems. While we focus on the particular task of identifying maps, we believe the approach is general enough to work for other specific image types, such as medical images.

More specifically, we start with a collection of images harvested from the Web and our goal is to sort the images into maps and other images. Rather than collecting the images ourselves, which is costly, we leverage the image-search databases of large companies such as Microsoft, Yahoo!, and Google who index and collect millions of images from all over the Web. Since these databases are refreshed frequently with new content, this allows us to collect new maps over time without having to spider the Web itself. Using such image databases, we can collect the search results and classify each returned image as a map or non-map, storing the newly classified maps in a map server. These stored maps can then be georeferenced for future querying. Figure 2 shows this process.

As shown in Figure 2, we exploit two repositories, one of maps and one of non-maps, to do our classification. While the specifics of our classifier are given in Section 2, intuitively, if an image to be classified is more similar to the maps in our repository than the non-maps, then it is more likely a map itself. That is, using techniques from Content-Based Image Retrieval (CBIR), we select the most similar images from the repositories for a given map and use those returned images to decide if the input image is a map or not. We find that our CBIR method is both more scalable than machine learning methods (as justified in

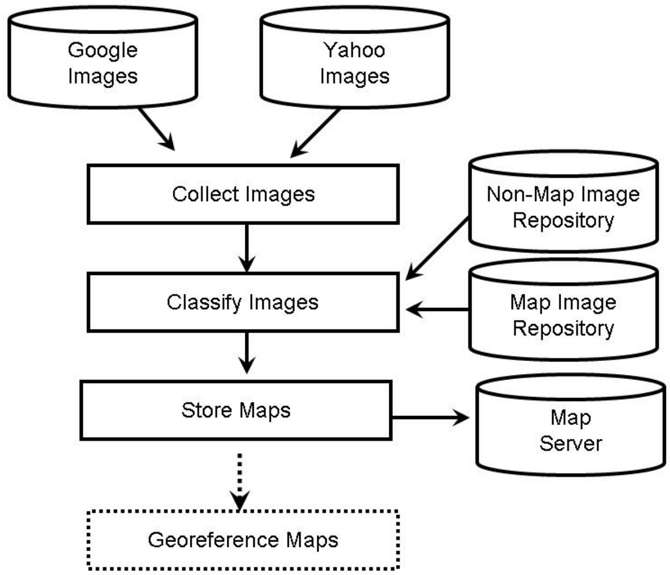

Fig. 2. A software system for automatically collecting maps from the Web.

Section 2), and it is also more accurate (as shown in the experiments of Section 3). This is the basis of our contribution: a method to sort images into maps that is scalable, robust, autonomous and accurate.

The rest of this paper is organized as follows. We describe the details of our CBIR map classifier in Section 2. We then describe our experiments and justify our approach in Section 3. We next present related work in Section 4, and we finish with our conclusions and future research in Section 5.

2 Automatically Classifying Maps

The focus of this paper is the classification of maps from images harvested from the Web. In this paper we broadly define a map as any digital image which contains "map like" characteristics such as lines, intersections and labels. The important aspect for our task is that we remain consistent in our definition of maps versus non-maps.

Our map classifier exploits techniques from Content-Based Image Retrieval (CBIR).[1] CBIR is essentially the image equivalent to traditional text based Web search. Instead of finding the most similar Web pages for a given set of query terms, in CBIR the most similar images from a set are returned for a given query

[1] See [7] for an excellent survey of the topic

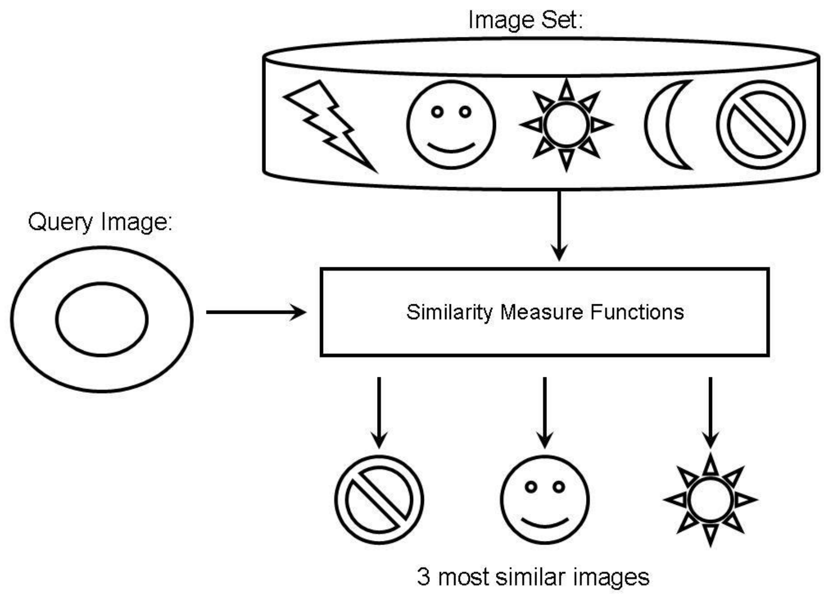

Fig. 3. An example of Content-Based Image Retrieval

image. For example, consider Figure 3. In this figure, the query image comes into the system and the top three most similar images are returned. This is the basis of our classifier.

To exploit CBIR for classification, we use a voting, k-Nearest Neighbor classifier [8]. We first use CBIR to find the nine most similar images (neighbors) from the combined set of images in the map and non-map repositories. The similarity measure in our CBIR method is based on Water-Filling features [9]. Water-filling uses the edge maps of an image to quantify measures such as edge complexity (based on forking), edge lengths, etc. These Water-Filling features are well suited for images with clear edge structures [9], such as maps. Further, by using edge-based features we make our classifier color invariant.

We then employ a simple majority voting mechanism [8]. If the majority of returned images are maps, we classify the query image as a map. If the majority of returned images are non-maps, we classify the query image as a non-map. This simple algorithm is shown in Figure 4. Therefore, although it may be the case that other images on the Web will have clear edge structures (such as diagrams), since our technique relies not only on the edge features themselves, but also on the similarity to the edge features of the images in our map repository, such images will be filtered out. The accuracy of our experimental results indeed show this to be the case.

We choose a CBIR based k-Nearest Neighbor classifier over more traditional machine learning methods for the following reasons. First, CBIR similarity meth-

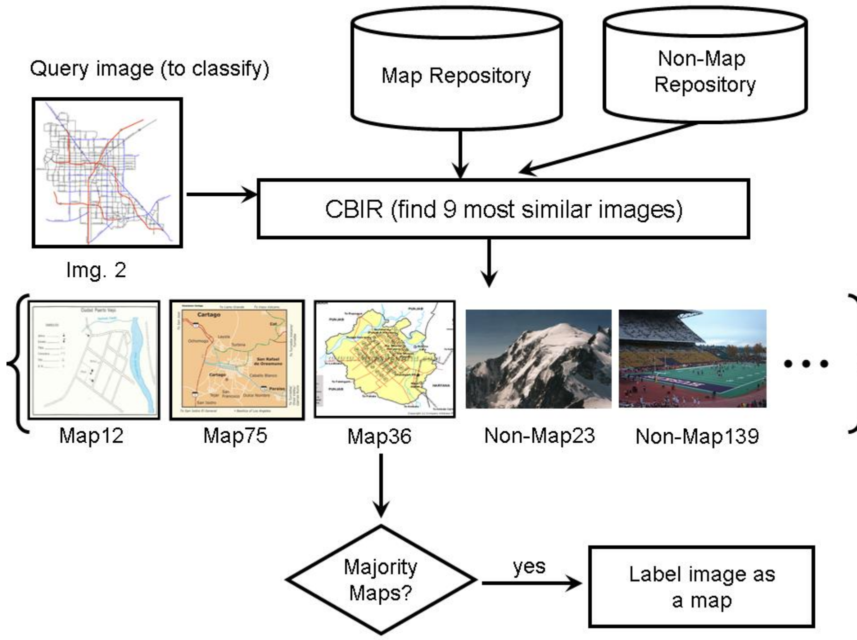

Fig. 4. A k-Nearest Neighbor map classifier using CBIR.

ods allow us to exploit image similarities without explicitly modeling them. For instance, hydrography maps are similar to other hydrography maps and urban city maps are more similar to urban city maps but these types of maps may be quite different from each other. By using CBIR methods, these similarities can be exploited without modeling them explicitly because the returned images encompass the image similarities implicitly (that is why they are returned as similar). If we use traditional machine learning methods, such as Support Vector Machines, we have two options to capture these different types of image similarity. On the one hand, we can train a model for each class of map, then if an incoming image matches any of these map classes, we know it is a map (since each class composes the image similarities). That is, we can take the hydography maps and learn a model of what a hydrography map should be. We can then take the urban city map and learn an urban city map model.

There are several problems with trying to learn a model for each type of map. For one, the number of such classes is not known in advance. Therefore, a user will have to make a decision as to which maps constitute a class and hope he or she made the correct choices to lead to accurate classification. Along these lines, the user must make decisions as to the granularity of the classes (is an urban-hydrographical map its own class?), which can further complicate the creation of classes. Also, learning a new model may be a non-trivial process both in the work and time required. So, if a new class is discovered or needed, this can become a prohibitively costly problem.

Instead, rather than modeling all of the different types of image similarities in distinct classes, one could try to learn a single model that encompasses *all* of these different types of similarities. However, since different types of map images can vary wildly, trying to generalize to cover all of these different similarities leads to a learned model that does not discriminate well (as shown in our experiments).

The other reason we chose CBIR instead of machine learning has to do with robustness and scalability. We can exploit huge repositories in our method, which is not the case using machine learning. In machine learning, learning models from massive amounts of data can take so long that it is not practical. CBIR techniques, however, are built on methods from information retrieval which the major search engines have shown to be fast and robust in very large, practical settings. Further, we can freely tweak the size and composition of our repository to test the effect (something we do in the experiments to test this idea). Using machine learning, we have to retrain a model each time we tweak the repository (training data). In situations where training is costly, this is not a good solution. Therefore, by using CBIR methods we can grow the repository over time without retraining which allows for a scalable and autonomous solution to classifying maps and building good map repositories. In our experiments, we show the effects of growing the repository on the accuracy of classification.

3 Experiments

We collected 1,704 map images and 3,462 non-map images for our experiment. We used Yahoo Image Search[2] to gather all of the map images and some of the non-map images. Most of the non-map images come from the CALTECH 101 data set [10], which is a set of pictures of objects belonging to 101 different categories. Table 1 presents the distribution of images by source. Note that we labeled all images as maps or non-maps manually.

Our experimental collection includes not only cities in the United States, but also international cities. Specifically, we included maps from New Dehli, and the maps retrieved for the keywords "city maps" include maps from China, Hungary, Israel, Ireland, France, Scotland and many other countries. Further, the maps in our collection retrieved by city keywords (such as "Pittsburgh maps") include not only urban street level maps, but also county maps, highway maps, weather maps, "metro area" maps, tourist maps (such as parks), and other types. Therefore, our map collection is diverse and interesting.

Our CBIR-based classifier builds upon LIRE[3], an open source CBIR library written in Java,[4] which we augmented to use Water-Filling features[9]. Previous work demonstrated that Water-Filling features increase performance in the retrieval of images with clear edge structures [9], a condition that applies well to

[2] images.search.yahoo.com
[3] http://www.semanticmetadata.net/lire/
[4] LIRE is part of the Calif&Emir project [11].

Table 1. Distribution of images by source

Source of image (Keyword used)	**Number of images**	**Number of map images**	**Number of non-map images**
Los Angeles Maps	378	327	51
Seattle Maps	132	87	45
Chicago Maps	480	376	104
Pittsburgh Maps	139	92	47
New York Maps	143	87	56
New Delhi Maps	188	124	64
City maps	624	611	13
NonMap (CALTECH 101)	3,082	0	3,082
ALL	*5,166*	*1,704*	*3,462*

maps. Since Water-Filling uses edge maps, we pre-process each image using the Canny edge detector[12] to create the edge maps.

For our experiments, we randomly select 800 maps and 800 non-maps from the entire set of images to build the repository for the CBIR method. Since the repository acts as the set of labeled samples for the CBIR method, we used this same set to train the SVM. Then we test each method on 800 randomly chosen maps and 800 randomly chosen non-maps to test the method's effectiveness. We repeated this process over 10 trials.

Our experiments test two hypotheses. First, we test whether CBIR is a more accurate classifier than the SVM by comparing both methods, using the Water-Filling features. Second, we also test whether Water-Filling features are more suited to map classification by comparing an SVM using Water-Filling features to one that uses Law's Textures [3]. By comparing an SVM using Law's Textures, which was used in a machine learning approach to the same problem [3], to our CBIR-based method using Water-Filling, we can also show that not only is our method more scalable and robust (since it does not require a training phase), it also outperforms the previous approach. Table 2 presents our experimental results as average precision, recall and F_1-measure (the first harmonic mean between precision and recall), the standard metrics for classification tasks.

Table 2. Performance of the CBIR and SVM methods

Method	**Precision**	**Recall**	**F_1-measure**
CBIR with WaterFilling	87.14	77.36	81.96
SVM with WaterFilling	88.80	56.00	68.69
SVM with Laws' Texture	69.50	47.43	56.38

The first result to notice is that the CBIR with WaterFilling method greatly outperforms the other two methods in F_1-measure, which means it has the superior map classification performance. Note that all differences, except for one, are statistically significant, using a two-tailed t-test with α set to 0.05. The only difference that is not statistically significant is the precision between the CBIR with WaterFilling and the SVM with WaterFilling methods.

Note that while the precision using WaterFilling is similar (not statistically significant) using either the SVM or the CBIR method, the recall is much improved using the CBIR method, yielding the best F_1-measure. This supports our notion that CBIR generalizes to many map types better than SVM does. In order to increase the SVM's recall, we would need to train it for the various map classes. As it stands, training it to learn a single "meta-map" class only results in its classification of roughly half of the maps correctly. Meanwhile, because the CBIR method returns the nearest neighbors, which may be of arbitrary map types, it is able to cover many more map classes, and return a higher proportion of the maps. It does this while maintaining a high level of precision, and is therefore a more accurate classifier.

The next result to notice is that WaterFilling features are much better suited for the map classification task. This is clear by examining the difference in precision between the SVM method using WaterFilling features and the SVM method using Law's Textures. Both methods have very similar recall values, generally capturing half of the maps in the testing data, but the Law's Textures method does a far worse job at distinguishing noise from the true maps. As stated, since maps have a strong-edge structure, Water-Filling can do a much better job at distinguishing true maps from false positives.

We also analyze the effect of the size of the repository on the CBIR method's classification performance. Since the repository is equivalent to the training data for the SVM method, this is analogous to studying the effect of the amount of training data on the SVM. To do this, we repeated the above classification experiment, this time varying the repository size (training data) by 200 maps and 200 non-maps, starting with 200 maps and 200 non-maps up to 1,400 maps and 1,400 non-maps. Figure 5 again compares the F_1-measures for the three methods, CBIR with WaterFilling, SVM with WaterFilling, and SVM with Law's Textures.

Figure 5 suggests that even with a small repository, the CBIR method outperforms the SVM methods. Moreover, as we add more images to the repository, the CBIR method's F_1-measure improves steadily. Again, the primary cause for both of the SVM methods' low F_1-measure is their low recall. As we argue above, the primary reason for the low recall is that maps vary dramatically in their shapes and density, such that when the SVM tries to learn one model for all of these types as maps, it converges to a set of feature values which are an average over these maps. As a result it defines a hyperplane approximately in the middle of the feature space resulting in lots of maps falling on the non-map side of the plane. In fact the recall hovers around the 50% mark reaching 56% for the most training. Note that the way to overcome this would be train an SVM for each map type, but this is infeasible given that we do not know the map types on the Web ahead on time.

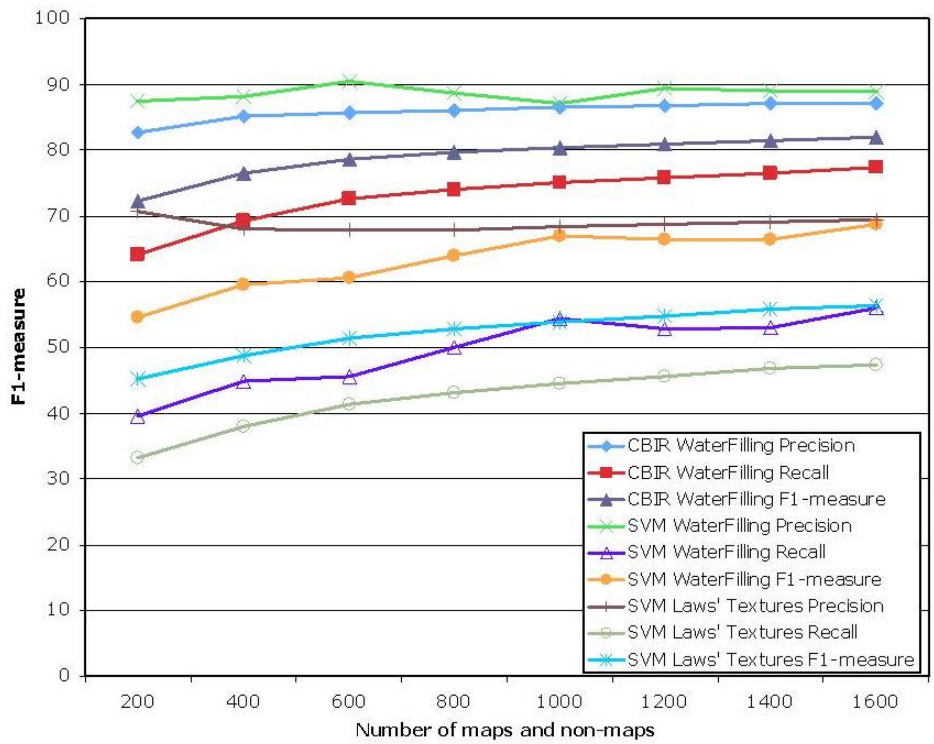

Fig. 5. Comparison of CBIR and SVM varying the repository size

Lastly, we examine the maps misclassified by the CBIR method. Since the precision is high, we focus on issues dealing with recall. Specifically, we delve more deeply into false-negatives (maps that are errantly classified as non-maps), which are the determinant for recall misses. We realized that the majority of false-negative cases are caused when there is one more non-map than map in the returned neighbors, which leads the system to classify the input image as a non-map (since it has a five to four majority). In most of these cases, it is a small set of non-map images that repeatedly get included as these neighbors which sway the vote to an errant classification. We call these “culprit” images. Figure 6 shows the edge-map of an example culprit image. This image is a person in front of a stack of books. However, the edge map for the stack of books in the background looks quite similar to an urban map (it has lots of squares and intersections). To deal with “culprit images” we plan to use relevance feedback techniques to find this small set of non-map images and remove them from the repository.

(a)

(b)

Fig. 6. An image where the edge map looks like a city map; Figure (a) shows the original image, Figure (b) shows the edge map

4 Related Work

CBIR methods have been applied to various scientific disciplines ranging from astronomy[13] to botany[14]. Much attention has been given to CBIR methods in medicine, where they can have tremendous impact [15]. For example, in one medical system the authors use a CBIR-based, k-Nearest Neighbor approach to classify medical images [16]. This work also includes Water-Filling features for the CBIR component. However, their work (and most of those above) differs from ours in the context in which it is applied. Our system is geared toward automatically harvesting maps from the Web, while their system is used to classify images so that images can be queried categorically. More specifically, these authors use many different types of features to classify their images, while we use only water-filling in the interest of accurate classification. Further, we are the first to use CBIR methods to automatically harvest maps from the Web.

Although we are the first to propose CBIR-based classification for map harvesting, other work has been proposed to automatically classify (and harvest) maps from the Web [3]. As stated in our experiments, Desai, et. al. [3] propose machine learning methods to classify maps versus non-maps. In our experiments we find this method is less effective than our CBIR based classifier. More specifically, not only is the machine learning component not as accurate as the CBIR-based classifier, but a machine learning method requires different models for each map type, since our experiments show that learning a single model to cover all maps leads to inaccurate classification.

Our experiments point to the necessity of using the correct features for CBIR. While we choose Water-Filling, which are good for images such as maps with strong edge maps, other methods could perhaps work as well. For instance, authors have proposed "salient point" features based on wavelets [17]. Another set of features based on shape similarity [18] could be well suited for our task as well, since maps seems to share certain shapes within them. Lastly, methods have been proposed to more efficiently store color information [19], which makes retrieval more efficient. Although we use textures based on edge maps to make our method color invariant, color information might help in discriminating maps. Our CBIR-method is not tied to Water-Filling, though those features perform well. It will be interesting future work to compare the various features and their efficiency and accuracy for map classification.

5 Conclusion

In this paper we present an autonomous, robust and accurate method for classifying maps from images collected on the World Wide Web. We find that a CBIR based classification method outperforms a machine learning based method, largely because we do not have to explicitly model the different types of maps that should be covered by the classifier. That is, by leveraging CBIR we can classify a variety of maps without having to explicitly train a classification model for each one. Further, we find that Water-Filling features, which are shown to work well on images with clear edge structures, work well for maps in our classifier.

However, although our method performs well, there are still areas for improvement. For example, some maps are misclassified when the majority vote is borderline (for example, one image sways the classification as a map or non-map). In this case, we can deal with the ambiguity by employing relevance feedback techniques from information retrieval. Such relevance feedback could help us to identify and prune away "culprit" images who consistently sway the vote in miss-classifications. Other future work involves exploring different features such as wavelets and shape similarities.

Nonetheless, despite the future work proposed above, our technique provides an automatic, accurate, practical and scalable solution to the problem of creating useful image repositories from the Web. More importantly, by plugging our method in with methods for aligning raster maps with satellite images we can create a harvesting framework for scouring the freely available images on the Web to build map collections for any given region in the world.

6 Acknowledgment

The authors thank Vipul Verma for helping implement the Water-Filling features and CBIR method.

References

1. Chen, C.C., Knoblock, C.A., Shahabi, C.: Automatically conflating road vector data with orthoimagery. Geoinformatica **10**(4) (2006) 495–530
2. Chen, C.C., Knoblock, C.A., Shahabi, C.: Automatically and accurately conflating raster maps with orthoimagery. GeoInformatica (in press, 2008)
3. Desai, S., Knoblock, C.A., Chiang, Y.Y., Desai, K., Chen, C.C.: Automatically identifying and georeferencing street maps on the web. In: Proceedings of the 2nd International Workshop on Geographic Information Retrieval. (2005)

4. Chiang, Y.Y., Knoblock, C.A., Shahabi, C., Chen, C.C.: Accurate and automatic extraction of road intersections from raster maps. Geoinformatica (in press, 2008)
5. Chiang, Y.Y., Knoblock, C.A.: Classification of line and character pixels on raster maps using discrete cosine transformation coefficients and support vector machines. In: Proceedings of the 18th International Conference on Pattern Recognition. (2006)
6. Chiang, Y.Y., Knoblock, C.A., Chen, C.C.: Automatic extraction of road intersections from raster maps. In: Proceedings of the 13th ACM International Symposium on Advances in Geographic Information Systems. (2005)
7. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1349–1380
8. Fix, E., Hodges, J.L.: Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical report 4, USAF School of Aviation Medicine, Randolph Field, TX (1951)
9. Zhou, X.S., Rui, Y., Huang, T.S.: Water-filling: A novel way for image structural feature extraction. In: Proceedings of the International Conference on Image Processing. (1999) 570–574
10. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceedings of IEEE CVPR Workshop on Generative-Model Based Vision. (2004)
11. Lux, M., Becker, J., Krottmaier, H.: Caliph&emir: Semantic annotation and retrieval in personal digital photo libraries. In: Proceedings of CAiSE '03 Forum at 15th Conference on Advanced Information Systems Engineering. (2003)
12. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence **8** (1986) 679–714
13. Csillaghy, A., Hinterberger, H., Benz, A.O.: Content based image retrieval in astronomy. Information Retrieval **3**(3) (2000) 229–241
14. Wang, Z., Chi, Z., Feng, D.: Fuzzy integral for leaf image retrieval. In: Proc. of IEEE Intl. Conference on Fuzzy Systems. (2002)
15. Müller, H., Michoux, N., Bandon, D., Geissbuhler, A.: A review of content-based image retrieval systems in medical applicationsclinical benefits and future directions. International Journal of Medical Informatics **73** (2004) 1–23
16. Lehmann, T.M., Güld, M.O., Deselaers, T., Keysers, D., Schubert, H., Spitzer, K., Ney, H., Wein, B.B.: Automatic categorization of medical images for content-based retrieval and data mining. Computerized Medical Imaging and Graphics **29** (2005) 143–155
17. Tian, Q., Sebe, N., Lew, M.S., Loupias, E., Huang, T.S.: Image retrieval using wavelet-based salient points. Journal of Electronic Imaging **10**(4) (2001) 835–849
18. Latecki, L.J., Lakamper, R.: Shape similarity measure based on correspondence of visual parts. IEEE Trans. Pattern Analysis and Machine Intelligence **22**(10) (2000) 1185–1190
19. Deng, Y., Manjunath, B.S., Kenney, C., Moore, M.S., Shin, H.: An efficient color representation for image retrieval. IEEE Trans. Image Processing **10**(1) (2001) 140–147