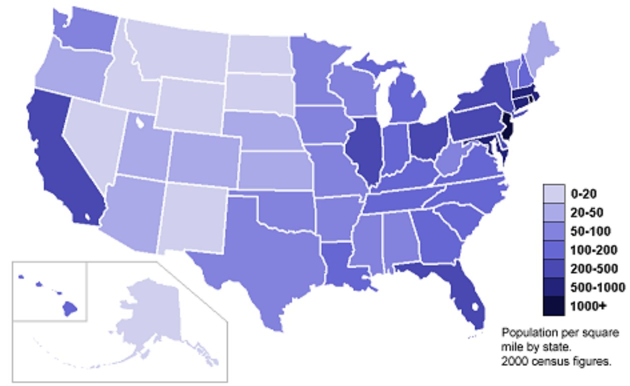
Identifying Maps on the World Wide Web

Matthew Michelson, Aman Goel and Craig A. Knoblock
Information Sciences Institute
University of Southern California
2008

Motivation - Leveraging existing maps

Earthquake map

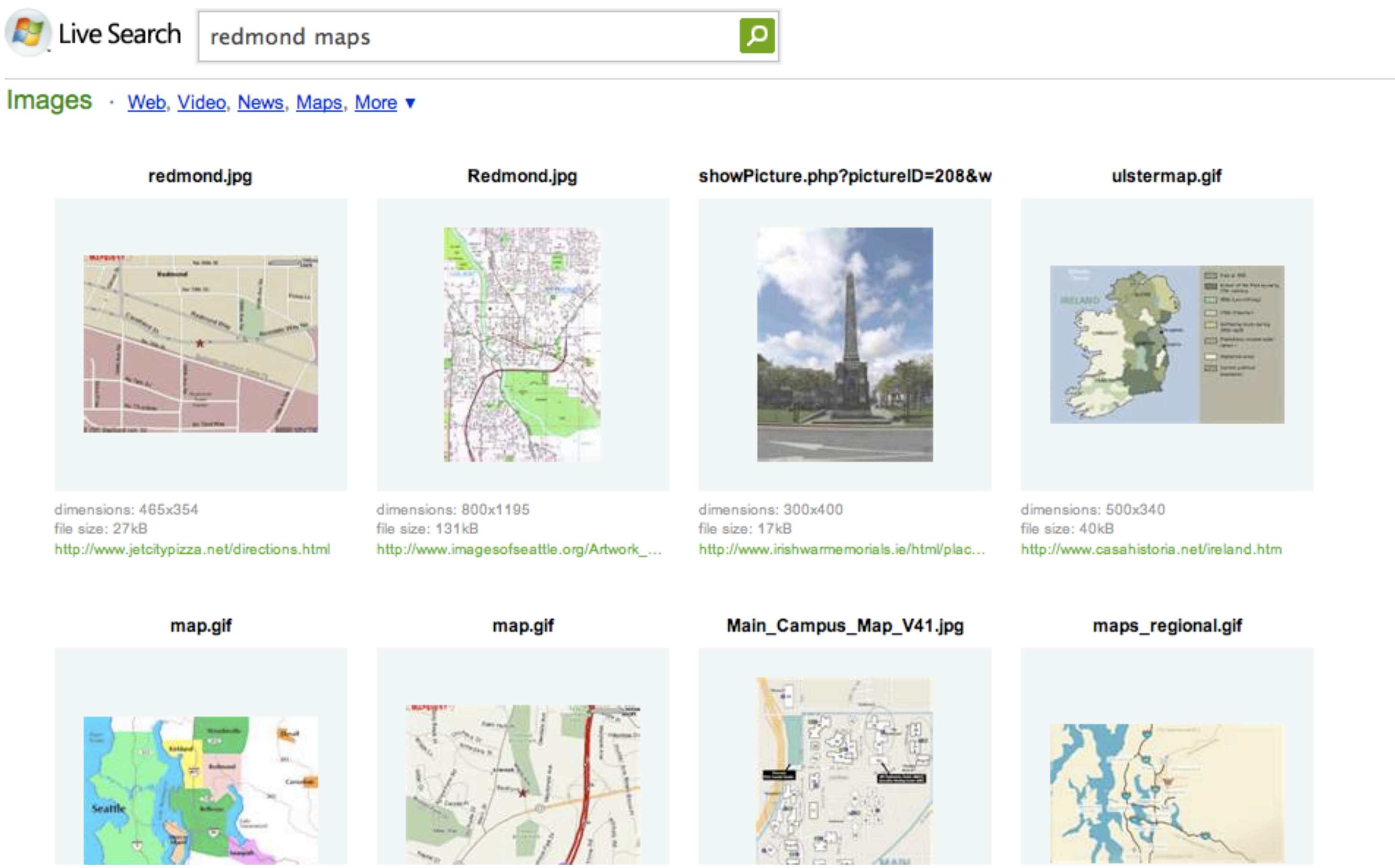


Population density

Alignment

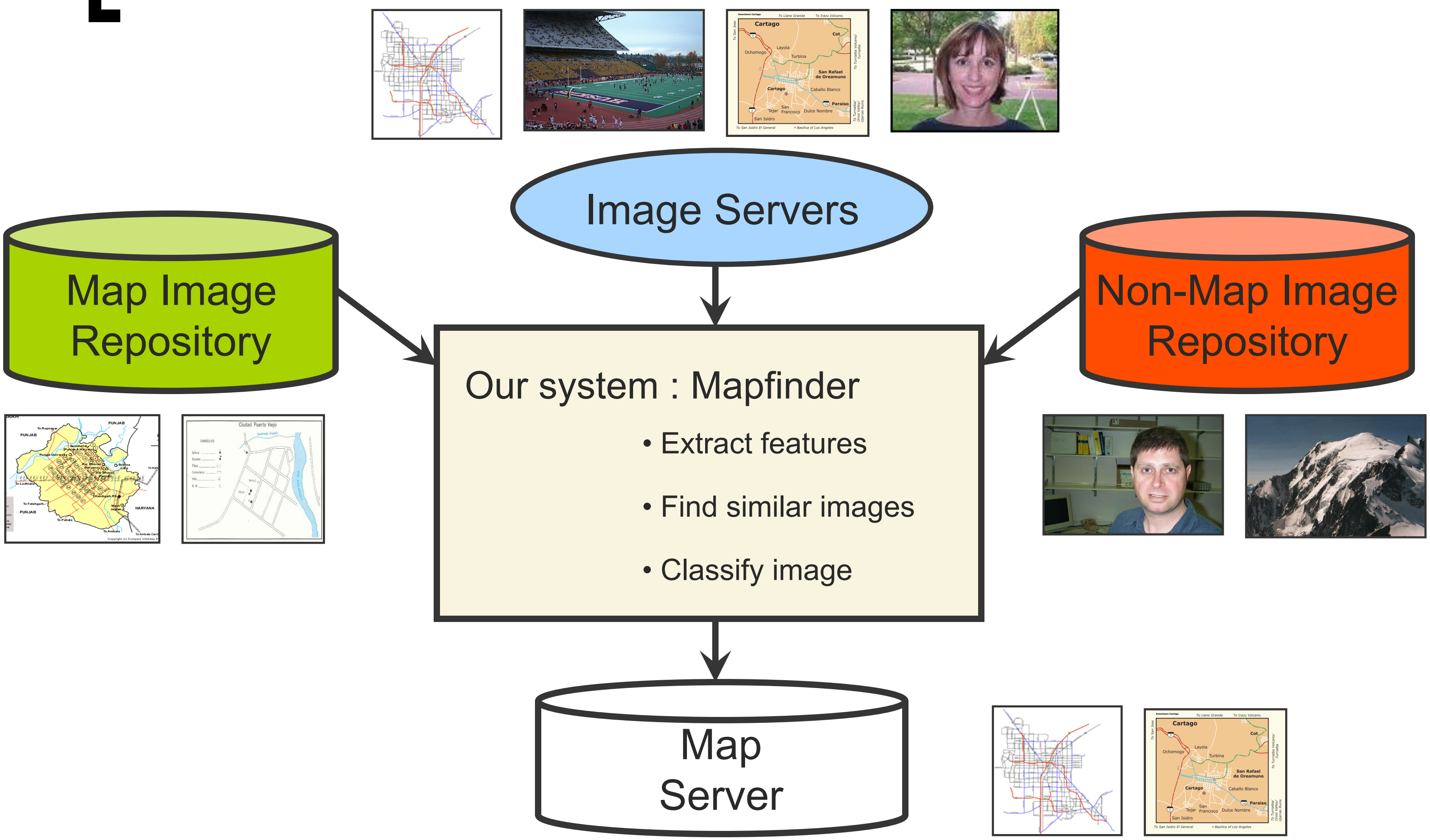
Estimate of potential damage

The problem



Result of search for maps on internet

Identifying maps among images



Our method

1. Extract features from query image
   - Water-filling features
2. Find images similar to query image from repository
   - Content based image retrieval (CBIR)
3. Classify query image
   - k - Nearest neighbor classification (k-NN)

Our method

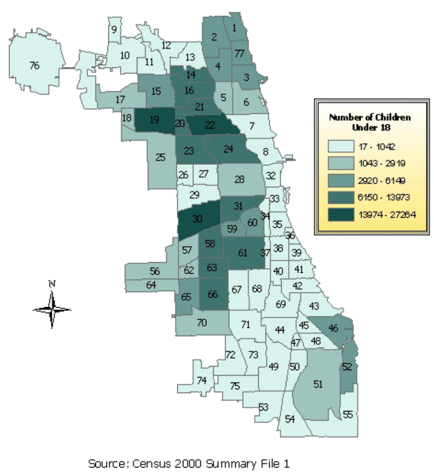
1. Extract features from query image ✔
   - Water-filling features

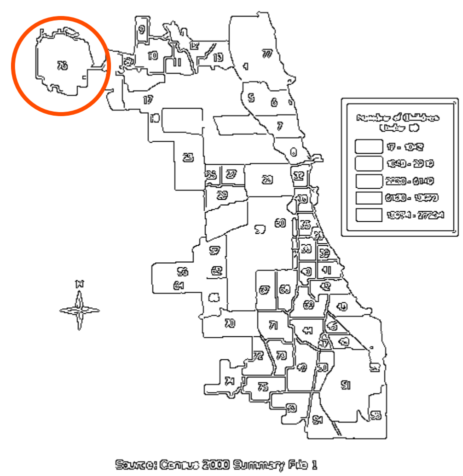
2. Find images similar to query image from repository.
   - Content based image retrieval (CBIR)

3. Classify query image
   - k - Nearest neighbor classification

Extract features

- Water-filling features
  - Zhou, X.S. et al. - Water-filling: A novel way for image structure feature extraction, 1999, Intl. conference on Image Processing
  - Works well on images with strong edges

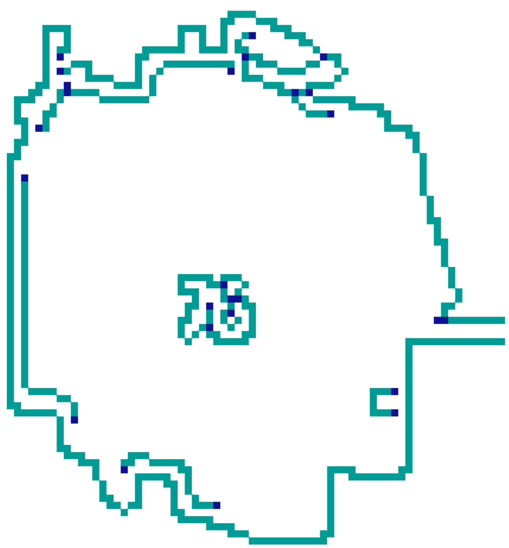




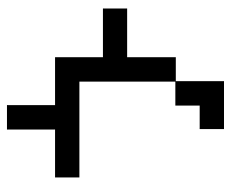
  - Works on standard Canny edge maps of original images
    - Color invariant

Water-filling algorithm

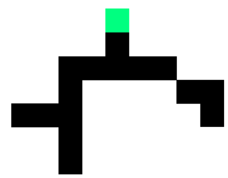
- Edge map has disjoint segments.
- Simulates flow of water through each segment



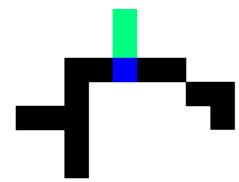
Simulation on one segment



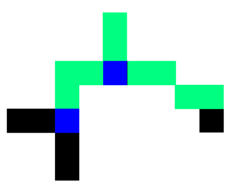
- FC : 0
- FT : 0
- WA : 0



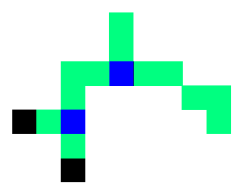
- FC : 0
- FT : 1
- WA : 1



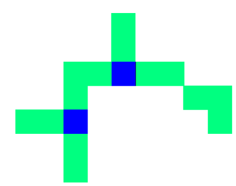
- FC : 1
- FT : 3
- WA : 3



- FC : 2
- FT : 7
- WA : 11



- FC : 2
- FT : 8
- WA : 14



- FC : 2
- FT : 9
- WA : 16

FC: Fork Count

FT: Filling Time

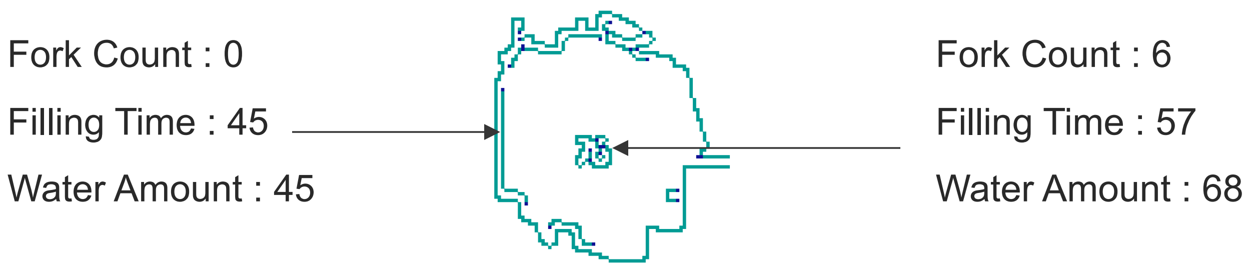
WA:Water Amount

Relevance of features

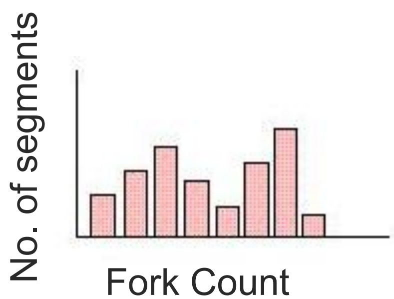
- Fork count (FC)
  - Complexity of segment
- Filling time (FT)
  - Length of segment
- Water amount (WA)
  - Size of segment

Extracting features to build vectors

- Features computed for each segment



- Normalized histogram - size invariant



- 3 features x 8 buckets = 24 element feature vector

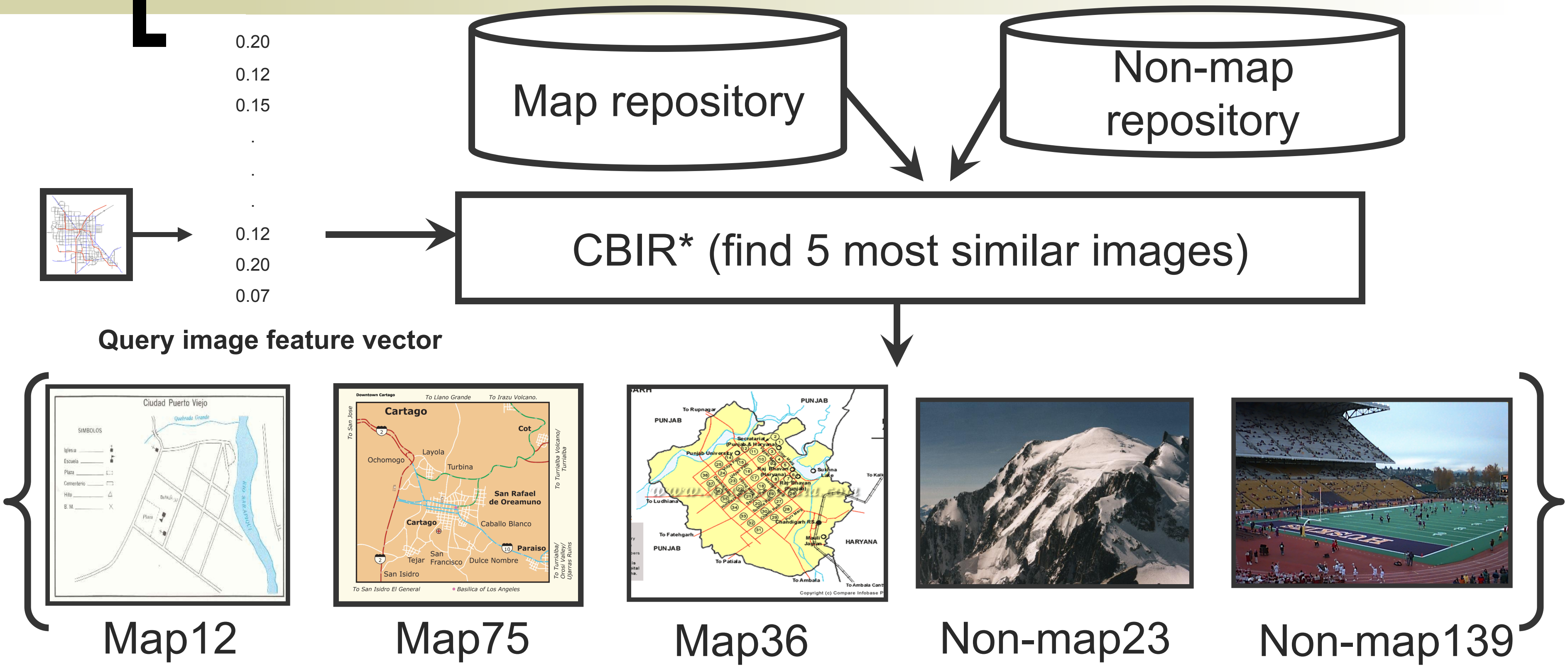
Our method

1. Extract features from query image
   - Water-filling features

2. Find images similar to query image from repository ✔
   - Content based image retrieval (CBIR)

3. Classify query image
   - k - Nearest neighbor classification

Content Based Image Retrieval (CBIR)



- Built on top of Lire system (http://www.semanticmetadata.net/lire/)

* In our experiment we used 9 similar images

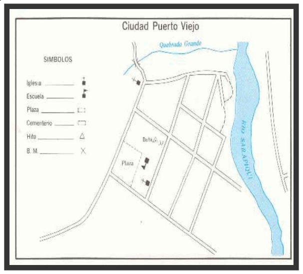
Our method

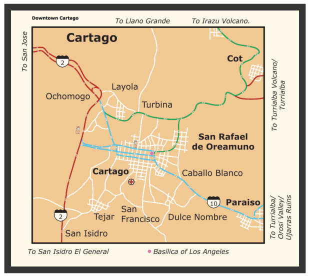
1. Extract features from query image
   - Water-filling features

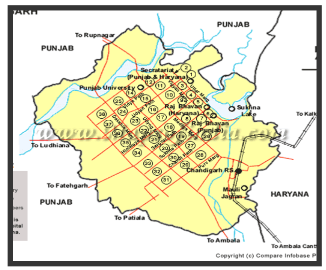
2. Find images similar to query image from repository
   - Content based image retrieval (CBIR)

3. Classify query image ✔
   - k - Nearest neighbor classification (k-NN)

k - Nearest neighbor classification







Map12	Map75	Map36	Non-map23	Non-map139
0.80	0.75	0.50	0.35	0.30

0.80 + 0.75 + 0.50 = 2.05

0.35 + 0.30 = 0.65

Majority Maps? → yes → Label image as a map

- Votes weighted proportional to similarity

Previous work on map identification

- SVM using Law's Textures (Desai, et. al. 2005)
  - Support Vector Machine
    - ✓ Trained on labeled examples
    - ✓ Learns a model of the class
  - Law's Texture
    - ✓ Convolution of gray-scale image with 5 texture masks
    - ✓ Distribution of intensity values on resulting images

Comparison of experiment parameters

- Claim 1:
  - CBIR better than SVM
    - Compare methods when both use Water-Filling
      - 1600 training images (repository)
        - 800 maps/ 800 non-maps
      - 1600 testing images
        - 800 maps/ 800 non-maps
- Claim 2:
  - Water-Filling better than Law's Textures
    - Compare features when both use SVM

Experiments

- Given: collection of images
- Task: separate maps/non-maps

Source of image (Keyword used)	**Total 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
N/A (CALTECH 101)	3,082	0	3,082
ALL	**5,166**	***1,704***	***3,462***

Results

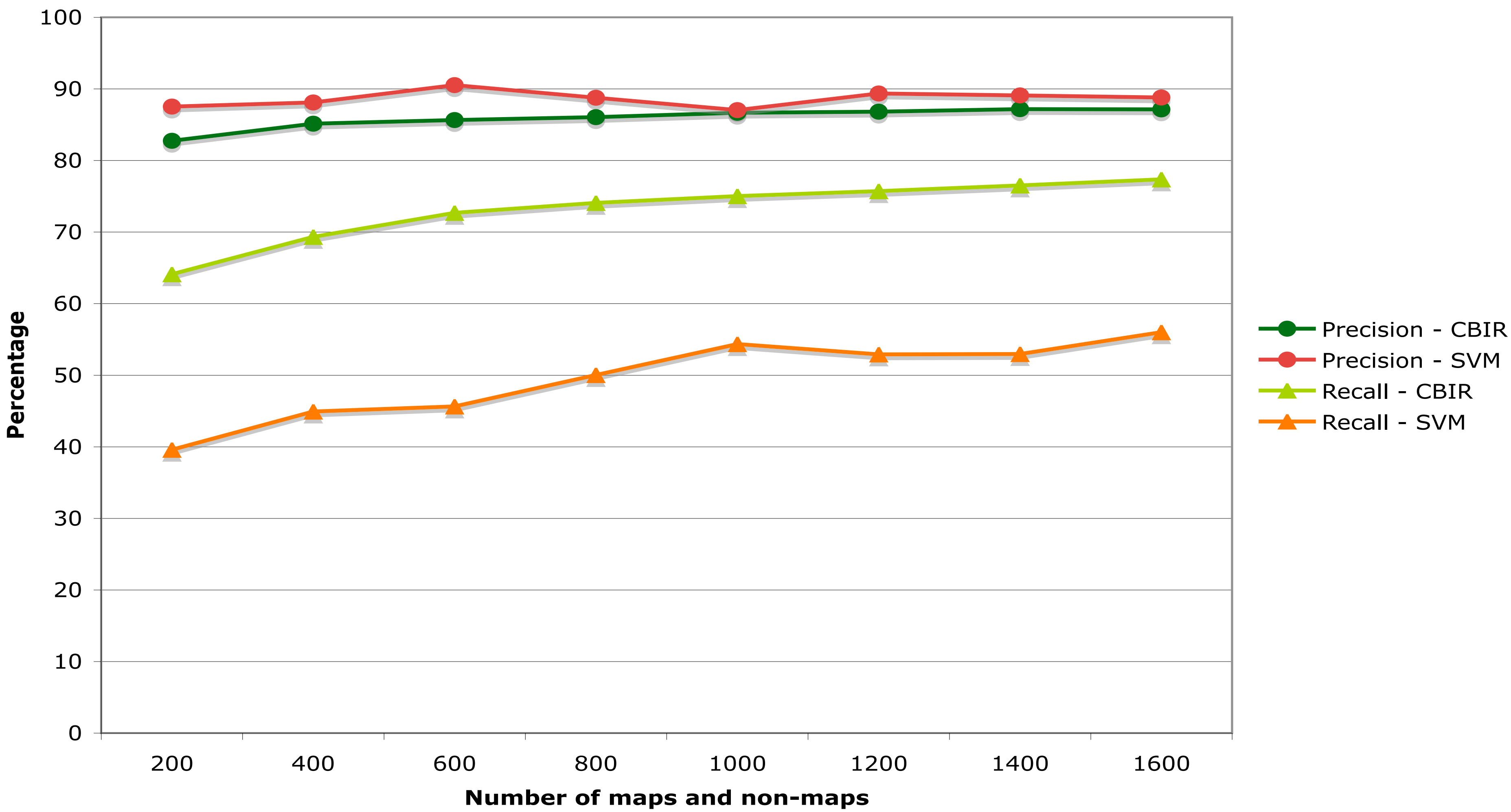
Method	Precision	Recall	F_1-Measure
CBIR w/ Water-Filling	87.14	77.36	**81.96**
SVM w/ Water-Filling	88.80	56.00	68.69
SVM w/ Law's Textures	69.50	47.43	56.38

- Precision : percentage of images correctly classified as maps
- Recall : percentage of maps identified

- CBIR outperforms SVM
- Water-Filling is better than Law's Textures

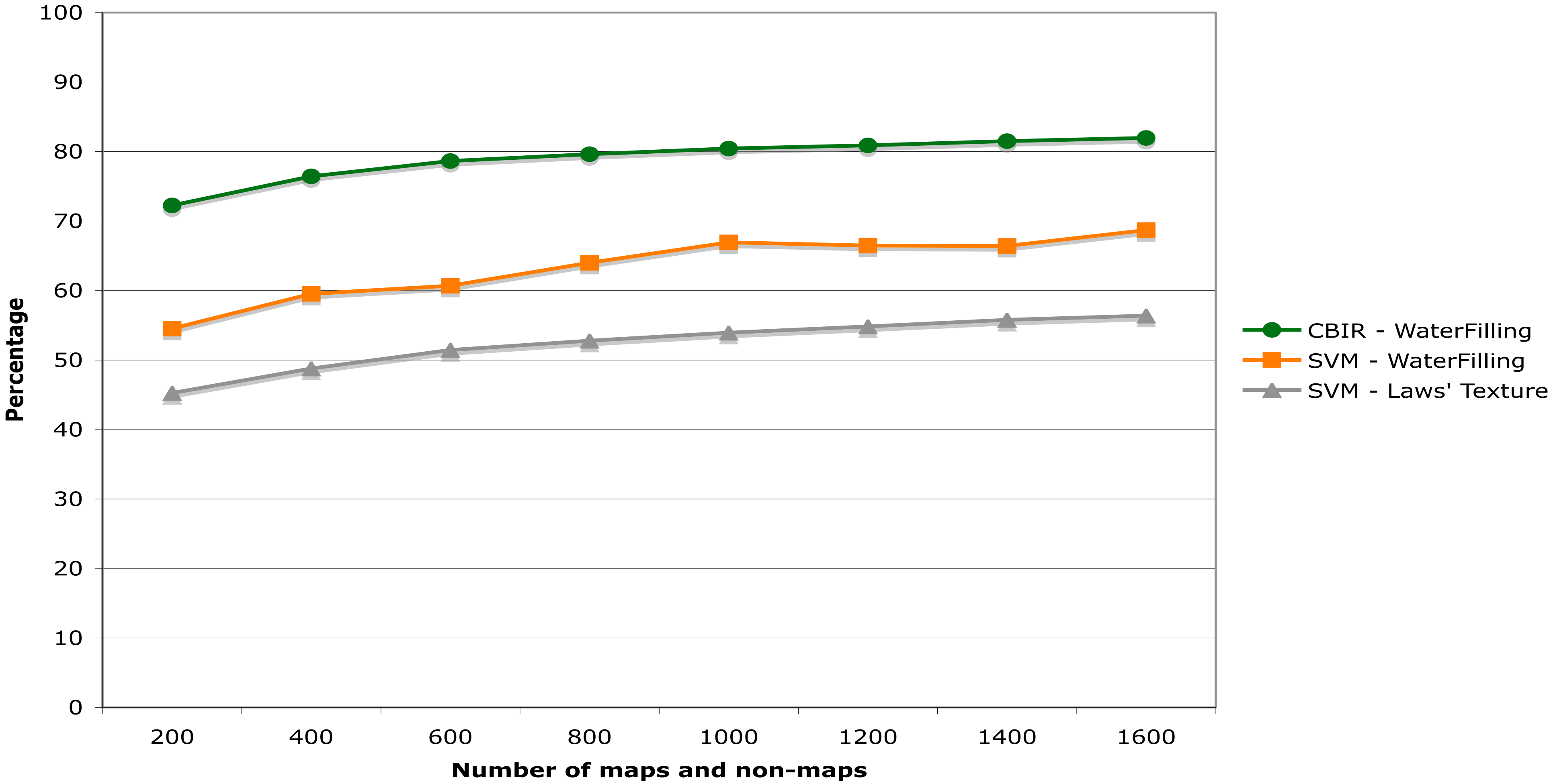
Results (2)

Varying the repository size (amount of training data) w/ Water-Filling features



Results (3)

Varying the repository size across all methods (**F_1-Measure**)



Reasons

- SVM class modeling issues
  - Learns 1 model for all maps
  - Needs to be trained for all distinct classes

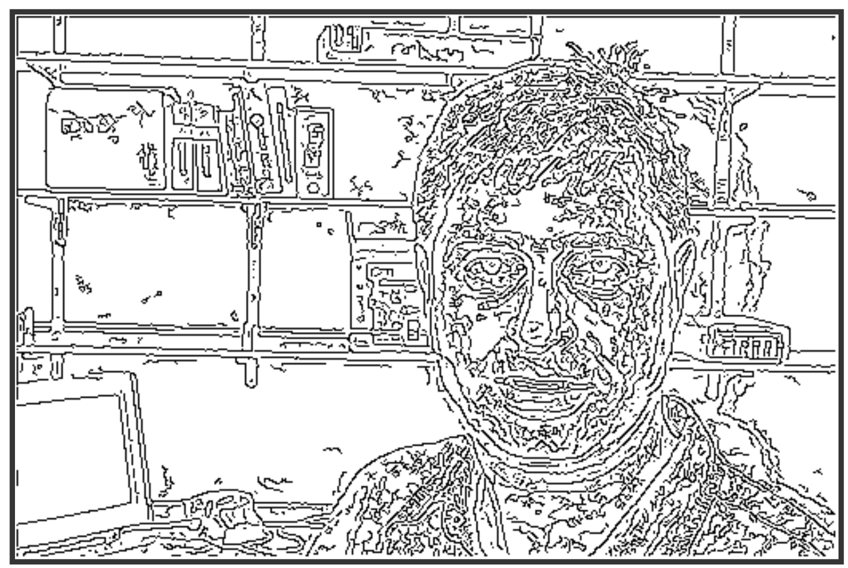
- More scalable
  - Addition to repository index; SVM needs to be re-trained
  - Law's Texture has many more features and takes more time to extract them per image

Related Work

- Classifying maps
  - SVM using Law’s Textures (Desai, et. al. 2005)
    - Law’s Textures: generates intensity maps based on textures
      - SVM Requires training, Law’s generates many, many features
      - Outperformed by our method
- CBIR-based k-NN
  - Classify images in the medical domain (Lehmann, et. al. 2005)
    - Used for classification/querying, not harvesting
- Other features for CBIR
  - salient points as features (based on wavelets) (Tian, et. al. 2001)
  - shape similarity features (Latecki & Lakamper, 2000)
    - Could plug-in to our method as future work

Scope for improvement

- Common classification error
  - A non-map gets repeatedly included in the set of similar images due to map-like features
  - Remove with relevance feedback from user



Conclusions

- Automatically harvest maps from the Web
  - Accurate
  - Fast
  - Scalable
  - Cost-effective
- Future work
  - Remove non-map images with map-like features
  - Explore other classifiers/features
  - Plug into georeferencing framework