

Change Detection From Media Sharing Community

No Author Given

No Institute Given

5 Change Detection Over Large Data Sets

In this section, we present the change detection algorithm from the large-scale social images, including how to identify the image pairs from the same source and how to detect the changes in images.

5.1 Image Copy Detection

This section presents how to effectively and efficiently identify the image pairs of the same source (same building or infrastructures etc). Many image copy detection approaches have been proposed and applied for various applications. However, since each image may contain up to thousands of 36-dimensional local descriptors, directly deploying existing technique is obviously not suitable for large image databases due to the high time cost. To improve the efficiency of image copy identification, we propose a process of *SIS*-based clustering, then deploy a PCA-SIFT-based matching over images in the same or neighboring clusters.

SIS-based clustering Intuitively, images from the same source usually have similarity over the concept level. Thus it is reasonable to cluster social images based on their tag and location information, such that those in a single cluster or neighboring clusters will have high possibility of being from the same source. In this way, the identification of image copy pairs can be simplified as the comparison between images from same or neighboring clusters, which avoids the costly pair-wise image matching over the whole data collection. To do this, we need to find an efficient clustering technique that can be extended to our *SIS* social image similarity as well. As we operate on large scale image database, the efficiency of clustering process is extremely important. Moreover, we only care about the similarity between images in each cluster, thus permit the overlaps between multiple ones. Thus, we extend the 2-means clustering algorithm that was initially proposed for L_p distance for our *SIS* similarity considering its low time cost as stated in [?,?].

Given an image collection, we conduct the clustering by three steps: 1) for each image, we model its metadata as tag set and its location as a pair of latitude and longitude values; 2) we select two images with the lowest similarity as the cluster centres of two initial cluster from the current data collection. Each of

the remaining images are assigned to the cluster that has the higher similarity with it. The centre points are recursively recalculated based on Equations ?? and ??, together with the images allocation to clusters based on their similarity, until the new generated centre points are stable; 3) finally, we recursively select a bigger cluster on which the second step is conducted, until the number of clusters reaches to a threshold κ .

In the social media images often there are more than 10 tags on each image and comparing these tags between the images one by one is time-consuming. String hashing has been applied in many applications for improving the system efficiency [?]. Thus we improve our change detection efficiency by selecting a “good” hashing function class. To reduce the time expenses on tags comparison, we use the *djb2* hash function techniques [?], which one of the best hash function techniques for the string data. In *djb2* the hash values are populated by $hash * 33 + c$, where the hash is the long data and c is the string character. Daniel J. Bernstein uses the magic number 33 to times the hash data, however, he did not explain why it works better than any other constants. In our system, the *djb2* is applied to all the images tags and the hash value is created for each tag. These hash values are used when the J_t similarity is calculated. The first step is making the tags on the image A and the image B to the hash values, and then it compares the hash values of A with those of B . Secondly, if the hash value of A does not exist in that of B , it is considered as unmatched. As such the system does not need to compare all the string tags. This hashing reduces the total number of string comparison in the J_t similarity.

PCA-SIFT-based matching PCA-SIFT-based matching will be used for deciding if two image candidates are referring to the same objects. It will find the similarity between the objects. The main objects are the buildings, statues and other objects which are not moving. Also, the unrelated images selected by the PCA-SIFT algorithm will be removed. The removing occur when the images have the high *SIS* similarity, but the images are not what the user want. The kept image pairs are passed to change detection stage for assessing if any damages had happened in a natural disaster.

We apply the OOS to the similarity calculation between the local descriptor sets of two images [?]. Given two local interest descriptors, the similarity between them is measured by the *Cosine* similarity between these two vectors. For two local interest points from two images, they are match pair candidate if the similarity between them is bigger than a threshold value. OOS further check if any one of these two points is the nearest neighbor of the other among all the descriptors of its image. If they are nearest neighbors of each other, they are a real matched pair. The final similarity between two images is calculated by the average similarity of all their matched pairs. To improve the local interest points matching, we use the LIP-IS index proposed in [?] as well. All these ensure that effective and efficient PCA-SIFT-based matching is performed.

5.2 Change Detection

Change detection assesses the damages caused during disasters. Intuitively, two pictures to the same location point contains same buildings or other objects, each is described as a boundary. If there is no damage happened at this place, the object boundaries in two images match. Otherwise, if there exists boundary missing or unmatched from the before image to the after one, the damages could have be caused in the disaster. Therefore, in this section, we propose a new image object boundary modelling together with a novel boundary matching method, which are robust to different image transformations, rotations or editing, for effective change detection.

Model image boundary Existing works model building shadow area boundaries [?], boundary shapes [?] using the coordinates of each pixel falling on the boundary of objects in an image. However, these boundary modelling incurs low effectiveness of detection for social images because of the possible object changes with respect to the viewpoints, rotations, space shift etc. To address this issue, we propose a robust boundary representation, called *relative position annulus* (RPA), which describe each boundary as an annulus of the difference of neighboring edge lengths. Specifically, we exploit sober edge detector to detect a number of boundaries in an image, since it can detect the emphasising edges while reduce the effect of noise edges. Given a boundary consisting of m vertexes $\{v_1, \dots, v_m\}$, we represent the boundary as $\{d(v_1, v_2) - d(v_2, v_3), \dots, d(v_{m-2}, v_{m-1}) - d(v_{m-1} - v_m), d(v_{m-1}, v_m) - d(v_m, v_1)\}$, where any element can be the start point while other points are ordered clockwise. As such, the RPA representation will be robust to object rotation, viewpoint change and space shift in social images.

Matching boundaries As each image may contain multiple objects, each image is described as a set of relative position annulus of multiple boundaries. To assess the damages in a disaster, we need to do two steps matching: (1) the measure between two RPAs; (2) the measure between two images. To further reduce the influence of small noise objects, we only use the top κ biggest boundaries for boundary comparison between two images.

Given two RPAs, $\mathcal{Q} :< q_1, q_2, \dots, q_m >$ and $\mathcal{D} :< v_1, v_2, \dots, v_n >$, we measure the similarity between them by extending the DTW [?] for our annulus representation. In the matching, we consider \mathcal{Q} as a series, and \mathcal{D} as a set of n series, where each series in the set takes v_i ($i = 1, \dots, n$) as the start point of the boundary and the remaining ones are ordered clockwise. Denote the series to v_i as \mathcal{D}_i , and its elements as $< v_1^i, \dots, v_n^i >$, where $v_j^i = v_{((i+j-1) \pmod n)}$. Then the similarity between \mathcal{Q} and \mathcal{D}_i is measured by:

$$SRPA_i(\mathcal{Q}, \mathcal{D}_i) = \begin{cases} 0 & m = m_1 - 1 \text{ or } n = n_1 - 1 \\ \max\{SRPA_i(\mathcal{Q}_{m-1}, \mathcal{D}_{n-1}) + Sim(q_m, v_n^i), \\ SRPA_i(\mathcal{Q}_m, \mathcal{D}_{n-1}), \\ SRPA_i(\mathcal{D}_{m-1}, \mathcal{D}_n)\} & otherwise \end{cases} \quad (1)$$

where Sim is the similarity between q_m and v_n^i computed based on L_1 distance:

$$Sim(q_m, v_n^i) = \frac{1}{1 + |q_m - v_n^i|}. \quad (2)$$

The final boundary distance is decided by finding the maximal DTW between Q and \mathcal{D}_i

$$SRPA = \max_{i=1}^n SRPA_i. \quad (3)$$

6 Experiment

In this section, we examine the performance of the proposed method, focusing on its effectiveness and efficiency. Specifically, we answer:

- How does the proposed Social Image Similarity function perform? We answer this by investigate the efficiency of the SIS-based clustering models, the effect of cluster numbers with different types of clustering techniques, and the effect of hash index.
- How effective and efficient of the proposed boundary-based change detection method (*BBCD*)? We compare *BBCD* with existing state-of-the-art algorithms, the shape base change detection(*SBCD*) [?].

6.1 Experimental Setup

The dataset we experimented with is collected from Flickr by focusing on images relevant to the *Nepal earthquake*, which is also known as *Gorkha earthquake*, in 2015. Finally, 10,000 images are collected, which include images *before* and *after* the earthquake. The ground-truth is manually identified by the authors of this paper via careful comparison of all *before* and *after* images. Specifically, for a *ground-truth*, both *before* and *after* image have to contain at least one same building, but the corresponding image contents, angles, resolutions, colours and light effects could be different. This would allow the proposed method to analyse whether the before and the after images are taken in the same location and building.

We conduct the efficiency experiments on 10000 after and before the earthquake images and their features. 4000 after and before the earthquake images are collected for the effectiveness experiment.

YL: this is confusing. How many images you collected? 10000 + 4000? or the 4000 is in the 10000 actually? why do effectiveness only on 4000? for time reason?

6.2 Measure metrics

To evaluate the effectiveness of algorithms, we used two metrics in [?], the probability of miss detection and false alarm (P_{miss} and P_{fa}). Specifically, the *missed*

detection mean that the algorithm fails to detect the ground-truth, and the *false alarm* means the detection of non-target pairs. P_{miss} and P_{fa} are defined as follows:

$$P_{miss} = \frac{\text{number of missed detections}}{\text{number of ground truth}} \quad (4)$$

$$P_{fa} = \frac{\text{false alarms}}{\text{nontargets}} \quad (5)$$

a small value of P_{miss} and P_{fa} means better effectiveness.

The evaluation of efficiency includes two parts: 1) number of clusters, and 2) comparison with the existing damage detection algorithms. Specifically, for number of clusters part test, it is expected to obtain the best cluster numbers to achieve the smallest P_{miss} and P_{fa} . We evaluate the efficiency of the proposed approach in terms of the overall time cost of clustering and hash index. We also evaluate the time cost of *BBCD* and *SBCD*, and to make the comparison more precise, we only compare the after the image matching time cost for change detection. Experiment are conducted on [machine information]...

YL: The above paragraph is very confusing, and please double check I rewrite what you want to present correctly.

6.3 Effectiveness

We first examine the effect of number of cluster in *SIS-based clustering* and *OOS+LIP-IS*. After that, we compare the proposed approach with *SBCD* by performing the change detection over the collected Flickr image datasets.

Effect of clustering

Effectiveness comparasion Comparing with existing technique We compare our *DTW + change detection* with *SBCD* used in [?]. We assume 20% of the building collapse are significant damage, we set the threshold of the unchanged building for the *SBCD* as 80%. In *SBCD* they did not perform the actual building matching. Therefore, in this test we selected 10 ground truth pairs(having similar angle) and 5 non ground truth pairs, which are paired by the *OOS+LIP-IS*. comparing with shape-based approach. (in paper: **automatic detection of buildings and changes in buildings for updating of maps**)

by varying the dataset size from small to big, test the probability of missed detection and probability of false alarm at each dataset size point

6.4 Efficiency

Effect of different clustering techniques

Effect of hashing index

Efficiency comparison Detection efficiency by varying data size
this is only for comparing our boundary-based approach and existing shape-based approach.