Change Detection From Media Sharing Community

Naoki Kito, Dr.Xiangmin Zhou, and Assoc Prof. James Thom

No Institute Given

Abstract. From ancient time, the damages or the destructions to the countries caused by the natural disasters were the major issues. Recently, through the improvement of image change detection technologies, social media and the high-resolution images, the damages caused by natural disasters can be analysed in more details to identify the situations in the cities or towns. Many researchers approached to analyse the damage by using the aerial images and the satellites images, but these images are often published to the public after the things settle down. However, when the disasters happen, people want the information of disasters as soon as possible. This research proposes to investigate how the social media images and the image change detection techniques can be used to identify the damages caused by the natural disasters. We first propose a framework that takes advantages of fast clustering and image near duplicate identification for the change detection in disasters. Then we model the social images by exploiting the image tags and location information. Following that, we propose a recursive 2 means algorithm over the new data model. Finally, we refine the changes by local interest point-based similarity matching. Extensive experiments have been done to evaluate the high effectiveness and efficiency of our approach.

1 Introduction

Before, during and after the natural disasters, information about the damages and the current situations are vital for people to make decisions for their next actions. For example, the Nepal earthquake in 2015 did a huge damage to everything, including buildings, roads, infrastructure, and people, which resulted in 8,019 people died and 17,866 people injured; in a Tokyo earthquake, people are still able to walk back home since the earthquake damage the infrastructure, but not the buildings and paths. A recent study [1] found that people would like to know earthquake size and epicentre. Moreover, knowing these information could prevent the secondary and tertiary disasters. It is also found that in Tokyo, only 67.8% of people managed to get back their house on the day of an earthquake, and the rest 32.2% had to become "*homeless*", among them 2% failed to going back home only because they could not find the safe path. People want the information about earthquakes, but there is always the question "How should the people get the information of earthquakes?".

Recently, there are lots of researchers had approached to detect the damages caused by the natural disasters by using the change detection techniques with the aerial images. However, the aerial images consume longer times retrieve and harder to get compare to the other images. On the other hand, social media is pervasive, and updates very quickly especially on large events, e.g. natural disasters, by millions of people all the time. Thus, we investigate the problem of change detection from social media images, so as to let the public aware of the latest situations of natural disasters on the spot. One of the challenges here is the large-scale of the social media images, which makes current change detection techniques infeasible if not impossible. One of the limitation of using the large-scale images with current change detection techniques is the time cost.

YL: Explain why time cost is a issue for them?

Another limitation is the unstable/unrelated images. The social media image features are made by the up-loader and this human inputs make the unstable/unrelated images.

YL: are we solving this?

The image rotation, resolution and the quality of the images also the concern of the limitation.

To solve the limitations, we deploy existing clustering technique before using the change detection methods to detect the changes. By applying the clustering methods to the large-scale image datasets, we are able to retrieve the related images from the clustered images for the change detection. Only applying the change detection method to the related images should increase the time performance and the accuracy of the change detection. We use the Recursive 2 Means clustering and the new feature, Social Image Similarity is proposed for the data sets of the clustering. The SIS is the total similarity between the images and it is the combination of the Tags Jaccard Similarity and the Locations Similarity. The tags are the texture data which present what the images contain and the locations are the place where the images are upload to the social media. Finally, we are using the PCA-SIFT algorithm to find the image similarities.

YL: It is still too detailed. I think a highlevel summarization about your technique would be better. After that, maybe one/two sentences for the details, but not too detailed.

The contributions of this study are as follows:

- New Feature Social Image Similarity
- Weighted Tags
- ...

YL: Please list the contributions here

The rest of the paper is structured as follows: Section 2 reviews the related work; Section 3 presents the framework proposed in this study; Section 4 details the modelling of the social media data; Section 5 presents the proposed change detection algorithm; Section 6 includes the experiment evaluation; Section 7 concludes the paper.

2 Related Work

Lu et al., 2004[2] state that there are seven categories for the change detection algorithms, which are: 1.Algebra 2.Transformation 3.Classification 4.Advance models 5.Geographical Information System (GIS) approaches 6.visual analysis 7.Other. [2]. Lu et al.,2004 [2] also state that these algorithms have their own unique characteristic, advantages, and disadvantages. These various change detections are commonly used in remote sensing, which often use to find the changes of the land surface, nature, and cities/towns. For another example, change detection in video surveillance often uses for the security purpose.

2.1 K-Means

The Recursive 2 Means algorithm are used. After the clustering, the PCA-SIFT is used to find the changes between the before and the after images as shown in figure 2. The Recursive 2 means Clustering has the similar process with the standard K-MEAN Algorithm, but instead of having several K clusters(more than 2), the Recursive 2 means clustering only has two clusters for each layer[1]. The Recursive 2 means clustering steps:

- 1: Create 2 random centroids(can be pre-made)
- 2: For each N, find the closest centroid and assign N to the K
- 3: Update the K centroids to the average of current assign N
- 4: Repeat the 2 and 3 until the K stop moving
- 5: Repeat 1 to 4 within the clustered data

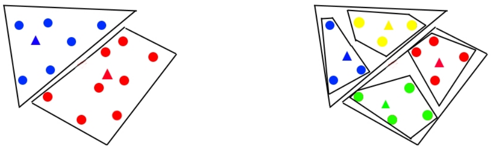


Fig. 1. Recursive 2 Means Cluster Sample process
The left figure shows the first cluster(step 1 to 4) result. The right figure shows the second clustering process(step 5).

As figure 1 shows, the Recursive 2 means cluster will always keep the number of K(cluster) to 2 for each layer. However, the number of total K(clusters) at the end of the process will have more than 2. For example, there are 4 clusters

[3] One of the type of Hash Index in Java programming

in figure 1. In H Gifford [3] experiment, it shows that the Recursive K-Means is around 10 times faster than the standard K-Means algorithm. However, Kohei et al., 2007 [4] had done several k-means algorithms experiments, which were using the real world data sets and they found that the Hierarchical K-Means Algorithm's error rate was higher compared to the K-Means Algorithm, which was using CCIA("Cluster Center Initialization Algorithm" proposed by Shehroz et al., 2004 [5]) for the Wine data set. They stated that this because the raw data had the far difference scales, and the error rate reduced after the normalization. They conclude that the Hierarchical K-Means has the speed benefit compared to other k-means algorithms.

2.2 PCA-SIFT

YL: This belongs to related work.

After the SIFT algorithm proposed by Lowe., 1999 [6], several image change detection algorithms developed based on the SIFT algorithm. For example, Speeded Up Robust Features(SURF)[7] and PCA-SIFT[8] is one of the improved version of the SIFT algorithm. Juan et al.,2009[9] had done the performance experiment for SIFT, PCA-SIFT, and SURF. They evaluated the three algorithms based on the "time","scale","Rotation","Blue","Illumination" and "Affine". In their experiments, it shows that the PCA-SIFT didn't have any standout performance. However, when they summarise all of the results, even though the PCA-SIFT did not have the best performance in each experiment, it has the stable performance in all situations. Which made Juan et al.,2009[9] to conclude that the PCA-SIFT has the best performance in overall. Therefore, in this thesis, PCA-SIFT is chosen to be the main change detection algorithm.
The PCA-SIFT algorithm was proposed by Yan et al.,2004[8] and it is a combination of two algorithms "PCA" and "SIFT".

Principal Component Analysis(PCA) The Principal Component Analysis(PCA) was introduced by Pearson in 1901 and developed by Hotelling 1933. In 2002, Ian, 2002 [10] published the second edition book, which talked about the PCA. In the book, it mentions that the main purpose of PCA is to reduce the dimensionality of the large datasets, and identify the uncorrelated variables "Principle Component". This dimensionality reduction has the advantages of multicollinearity and data sets visualisation.

SIFT In 1999, Lowe, 1999 [6] invented SIFT algorithm and it widely used to find the similarity between the images. Lowe, 1999 [6] approached to make the local feature vectors, which were invariant to the image translation, scaling, and rotation. Vectors were also partially invariant to some other changes. There are four main approaches to make these invariant feature vectors for the images. In Lowe, 2004 [11] paper, there are 4 approaches for the SIFT:

- Scale-space peak selection
- Keypoint localization
- Orientation assignment
- Keypoint descriptor

3 Framework

In this section, we first define two terms, *change* and *change detection*; then presents the overview of the change detection framework from social media images.

Definition 1. *In image processing,* change *is defined as the difference between the two pixels or the objects in the different images. Note, the* difference *varies in different situations. In this paper, the* difference *is limited to the damages to the buildings, the roads or the infrastructures, which caused by the natural disasters. Such as, when the earthquake happens and a bridge breaks down, the damage to the bridge will be the* change *in this situation.*

Definition 2. Detection *is defined as* Find. *As defined in Definition 1,* change *refers to the damages to the buildings, roads and infrastructures. Thus,* change detection *is defined as "Finding the damages which cause by the earthquake or natural disasters".*

As shown in the Fig. 2, the framework includes the following two component:

- **Retrieving relevant Images** The proposed change detection framework accepts a set of tags as input, which are used to retrieve images from the social media community (e.g. Flickr). While the images are retrieving, the images are separated into the before and the after natural disasters.
- **Data Modelling** The information about retrieved images include the photo ID, tags, and location features. Firstly, we model them to numeric data, then deploy two metrics to measure the similarity between tags and locations, separately. Finally we fuse these two similarities to obtain the final similarity between two social images, which is terms *Social Image Similarity* here. Then, the recursive 2 means algorithm is applied to cluster similar images into groups, which significantly reduce the complexity of the problem of change detection based on large-scale social images. This will be detailed in Section 4.
- **Detecting Changes** After the similar images are grouped together, a PCA-SIFT algorithm is applied here to find the similarities between the before and the after images and detect the damages caused by the natural disasters.

4 Data modelling

In this section, we present how to model images' tags and location data for the purposed of change detection from social media images. Specifically, we model

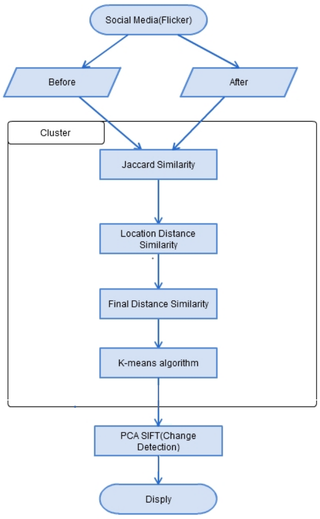


Fig. 2. The Framework

them to numeric data first, then measure tags and locations by using two different similarity metrics, respectively. Finally, we fuse them together to obtain the *difference* of social media images.

4.1 Tag-based Similarity

In social media, the posted images normally have some tags on it, which contain semantic meanings of the images. The tags are attached to the images by the uploaders and they may vary. We first weight the tags on images and then deploy a weighted version of Jaccard metric to measure the similarity between images. Specifically, the tags assign with the value of 1 as a default, but when the centroids are calculated, the tags weight will be the average of the tags appear in the images. We will discuss in more details about the weighted tags in section 5.

YL: I suggest to explain the weights here clearly, as it is very relevant here. In addition, it is your contribution, and should be emphasised as early as possible.

The basic equation for Jaccard Similarity from [12] is:

YL: can you provide a weighted version of Jaccard by using your weighted tags? this is something interesting, not the standard Jaccard equation.

$$S_{tag} = \ldots \tag{1}$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2}$$

In equation 2, the ‘A’ is the user input and the ‘B’ is the tags. This equation calculates the distances between the user inputs and all the images tags. After the distance calculation, the average of the *JDS* is calculated to find the centroid *JDS*.

$$0 < J(A,B) < 1 : 0 = notmatch, 1 = match \tag{3}$$

Examples:

- User Input = “Happy”, “cake”
- Tags = “Happy”, “fire”,“rain”,“party”,“apple”,“cake”

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{4}$$

$$= \frac{2}{|2| + |6| - |2|} = \frac{2}{6} = 0.33 \tag{5}$$

The equation 4 is the example of two user inputs and six image tags, and both two inputs matched with the tags. In this case, the result is ‘0.33’, which means the image is partially related to what the user want. However, only using the Jaccard Similarity distances will cause the human error issues when the k-means algorithm use them as the data sets. This because the image tags could contain the unrelated tags. For example, the image of the car is tagged with the building. Therefore, to increase the accuracy of clustering we use the Location Distance Similarity.

4.2 Location-based similarity(LDS)

To detect changes from images, it is important to prevent gathering the different location images, but having the similar tags. Thus, location information is a necessary.

We deploy the “Great Circle Distance(GCD)” to calculate the distance between the images. The Haversine Formula [13] is used to calculate GCD. However at this point, the bearing and the midpoint where not important values for the Location similarity, so it only calculates the distance between 2 points.

$$a = sin(Lat/2) + cosLat1cosLat2sin(Lon/2) \tag{6}$$

$$c = 2atan2(\sqrt{a}, \sqrt{(1-a)}) \tag{7}$$

$$d = Rc \tag{8}$$

Where a denotes the Haversine, c denotes the great circle distance in radians and d denotes the location distance. The Haversine is using 3 points/locations to

[1] http://bit.ly/1XdoJ36

[2] http://bit.ly/1Uka1Fa

calculate the Great Circle Distance, and the above formula is using the earth's radian for the third point. As suggested by Johor et al. 2013 [14], the location distances are normalised to removed the potential biases among different features. Specifically, Johor et al. 2013 [14] found the " Min-Max" standardization had the best/lowest error rate to k-means algorithms. Thus, we choose to use the "Min-Max" as a normalization technique:

$$D' = \frac{D - min(D)}{max(D) - min(D)} \tag{9}$$

where D denotes the calculate location between images.

4.3 Social Similarity (SocSim)

We finally define the similarity between two social images by fusing the tag-based distance and location-based distance, which is termed as *Social Similarity* (SocSim):

$$S_{SocSim} = \frac{1}{D+1} * S_{tag}, \tag{10}$$

where D is the location-based distance and S_{tag} is the tag-based similarity. The S_{SocSim} will be the range from 0 to 1. A larger value means the images closer to each other.

5 Change Detection Over Large Data Sets

In this section, we present the change detection algorithm from the large-scale social media images.

5.1 Clustering Techniques and Improvement

The idea of the clustering algorithms is to group the large datasets into the smaller groups. Always the data within the same group have the similar data than the data in other clusters. The clustering allows the system to use the centroids as the references of the cluster, and by referencing the centroids the system does not have to analyse all the data in the data sets. By knowing which cluster should be analysed, it increases the speed performance. However, there are some issues or limits on the clustering methods, such as the clustering methods could not handle some attribute types, time complexity and etc [15].

Moreover, Cheung [16] found there are 3 drawbacks for the K-MEANS algorithm. One of the drawbacks is that when the initialized(random) point are far away from the other points, it will remove immediately without learning within the learning process[2]. This means if the centroid is far from most of the points, the centroid's dimension is different from the major points' dimensions. In this research, there is one approach done with using this learning attitude. The approach is to pre-make the second centroid at the farthest point from the first centroid. The steps are followed by initialize the first random centroid, calculate the distance between the centroid and the points, and choose the farthest point as the second centroid. This approach is done to reduce the cycle of learning process(updating centroids) so that the process will improve the speed performance. This can be done in this research because, the main purpose of using Recursive 2 MEANS algorithm is to find the related images in the data sets and remove the unrelated images.

YL: I suggest to move all Recursive 2 means related stuff to Section 2, as this is not your contribution. And focus on the Hash index one, and emphasis why it works, and how it works, and how much improvement it achieves, while you are using it with the k-means method.

The approach is to use a Hash Index technique to improve the data accessing speed. 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 are time-consuming. To reduce the time expenses on tags comparison, we used the *djb2* hash function techniques. The *djb2* is known as best hash function technique for the string data compares to other hash function. 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 Jaccard Distance Similarity is calculated. The first step is making the tags on the ImageA and the ImageB to the hash values, and then it compares the ImageA hash values with ImageB hash values. Secondly, if the ImageA hash value does not exist in the ImageB hash value, it considers as a not match. So that the system does not need to compare all the tags values in the string. This hashing reduce the total number of String comparison in the Jaccard Distance Similarity.

5.2 Using PCA-SIFT

In this research, PCA-SIFT will use for two processes. Firstly, 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 choose by the PCA-SIFT algorithm will remove. The removing occur when the images have the Final Distance same or close, but the images are not what the user want. Those removed images are used to calculate the error rates. The second process is to detect the changes between the images if the images contain the similar objects. These process will be explained in detail at the “Experiment” section.

YL: Please detail the two runs of PCA-SIFT, which is your work. ;)

[1] Layer is the complete learning processes(step 1-5) for the K-MEANS

[2] The process at updating the centroid(step 2-3)

5.3 Datasets Issues and approaches

As mentioned in Section 4, the Recursive 2 MEANS algorithm uses the numeric data. So, the image features are modelled to numeric to measure the similarities between images. However, the modelled data do not show the tags information and locations information which causes a difficulty when the PCA-SIFT try to find the before and after images. Therefore, it is needed to find the tags and the weighted tags in the centroids to see the texture differences between the centroids and the distances. The tags in the centroids are the average of the tags within a cluster, which can be calculated with:

$$CentroidKeyword = K_1(\frac{P_0 + P_1..P_n}{n}), K_2(\frac{P_0 + P_1..P_n}{n})...K_n(\frac{P_n}{n}), \tag{11}$$

where P_n denotes the points, K_n are the keywords and n denotes the number of modelled data. If all the points in the cluster have the same keyword, the Keyword weight will be assign to 1.

YL: what does 'the data' refer to?

For example, assume there are 2 points where the point 1 contains 3 keywords and point 2 contain 5 keywords.
Point 1 Keywords = {“earthquake”,“apple”,“happy”}
Point 2 Keywords = {“earthquake”,“natural”,“disaster”,“apple”,“Nepal”}

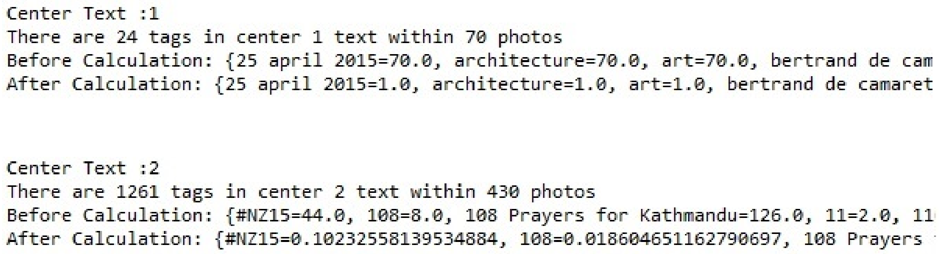
$$\begin{gathered} CentroidKeyword = earthqauake(\tfrac{P_1+P_2}{2}), apple(\tfrac{P_1+P_2}{2}), natural(\tfrac{P_1}{2}), \\ disaster(\tfrac{P_1}{2}), Nepal(\tfrac{P_1}{2}) = earthqauake(\tfrac{1+1}{2}), apple(\tfrac{1+1}{2}), \\ natural(\tfrac{1}{2}), disaster(\tfrac{1}{2}), Nepal(\tfrac{1}{2}). \end{gathered} \tag{12}$$

YL: This equation is confusing. what is V?

For the above sample, the centroid between two points is

$$CentroidKeyword = earthquake(1), apple(1), natural(0.5), disaster(0.5), Nepal(0.5).$$

By finding the centroid tags, we are able to know the main tags tagged within the cluster. Comparing the before and after images clusters with the weighted centroids tags, we are able to reduce the searching loop for before and after image clusters. The figure 3, is the sample from the change detection application.



```
Center Text :1
There are 24 tags in center 1 text within 70 photos
Before Calculation: {25 april 2015=70.0, architecture=70.0, art=70.0, bertrand de cam
After Calculation: {25 april 2015=1.0, architecture=1.0, art=1.0, bertrand de camaret


Center Text :2
There are 1261 tags in center 2 text within 430 photos
Before Calculation: {#NZ15=44.0, 108=8.0, 108 Prayers for Kathmandu=126.0, 11=2.0, 11
After Calculation: {#NZ15=0.10232558139534884, 108=0.018604651162790697, 108 Prayers
```

Fig. 3. Centroids Keywords from the appreciation
Sample clusters tags data and weighted tags from application. Where the *Center Text :1* is the first cluster and *Center Text :2* is the second cluster. The line *Before Calculation* and *After Calculation* show the before and after the average calculation.

6 Evaluation

6.1 Experimental set up

6.2 methodology

6.3 Effective

Effect of number tags : Effect of T

Effect of Cluster : Effect of K

comparison of EoT and EoK

6.4 Efficiency

Clustering with and without HASH

Data size comparison

Recursive 2 Means and K-Means

7 Conclusion

References

1. H. U, S. Naoya, N. Ryota, W. Shuntaro, and H. Hidenori, "Questionnaire survey concerning stranded commuters in metropolitan area in the east japan great earthquake," *Journal of social safety science*, pp. 343–353, nov 2011.

2. D. Lu, P. Mausel, E. Brondizio, and E. Moran, “Change detection techniques,” *International journal of remote sensing*, vol. 25, no. 12, pp. 2365–2401, 2004.
3. H. Gifford, “Hierarchical k-means for unsupervised learning.”
4. K. Arai and A. R. Barakbah, “Hierarchical k-means: an algorithm for centroids initialization for k-means,” *Reports of the Faculty of Science and Engineering*, vol. 36, no. 1, pp. 25–31, 2007.
5. S. S. Khan and A. Ahmad, “Cluster center initialization algorithm for k-means clustering,” *Pattern recognition letters*, vol. 25, no. 11, pp. 1293–1302, 2004.
6. D. G. Lowe, “Object recognition from local scale-invariant features,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150–1157 vol.2, 1999.
7. H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer vision–ECCV 2006*, pp. 404–417, Springer, 2006.
8. Y. Ke and R. Sukthankar, “Pca-sift: a more distinctive representation for local image descriptors,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–506–II–513 Vol.2, June 2004.
9. L. Juan and O. Gwun, “A comparison of sift, pca-sift and surf,” *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.
10. I. Jolliffe, *Principal component analysis.* Wiley Online Library, 2002.
11. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
12. E. N. Suphakit Niwattanakul, Jatsada Singthongchai and S. Wanapu, “Using of jaccard coefficient for keywords similarity,” *Proceedings of The International Multi-Conference of Engineers and Computer Scientists 2013*, vol. 1, pp. 380–384, March 2013.
13. C. C. Robusto, “The cosine-haversine formula,” *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.
14. D. U. Ismail Bin Mohamad, “Standardization and its effects on k-means clustering algorithm,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 17, pp. 3299–3303, 2013.
15. P. Berkhin, *A Survey of Clustering Data Mining Techniques*, pp. 25–71. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
16. Y.-M. Cheung, “k-means: A new generalized k-means clustering algorithm,” *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2883 – 2893, 2003.