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. As existing techniques detect changes based on pixel level comparison [] without index support or query optimization, the time cost for image comparison is high. When applying them to large scale social images, the efficiency issue becomes even unacceptable. Another challenge is the unavailability of some special features, like building shady, used in traditional change detection approaches []. In sharing communities, most of social images donot have shady, thus the shady-based matching can not be conducted. Forcing the existing techniques on the social images will cause low detection quality. Finally, traditional change detection over arial images suppose the image pairs to the same location points are known, which is not true in media sharing communities.

To address these issues, we propose a framework for change detections in sharing communities. First, we represent the metadata of each image as a set of weighted tags, and propose a social image similarity function called SIS over image metadata and location. Then, we extend the recursive 2-means clustering algorithm [] from vector space in $L_p - norm$ to key word set space, so Jaccard-based SIS measure can be applied. By this extended recursive 2-means clustering over the whole image dataset, a number of small clusters are generated. Images in the same cluster or neighboring clusters will have the high probability of being the image pairs of the sources. Following that, we conduct PCA-SIFT based matching, which determines if two images are really referring to the same source. Finally, we propose a robust boundary representation that is robust to the view point rotation and other global transformations of the same objects in different images. Based on this representation, the boundary matching between image pair candidates is performed, which decides if a change has happened after disaster.

The contributions of this study are as follows:

- We propose a new similarity function SIS over weighted tags and location of social images, and extend the recursive 2-means clustering for SIS similarity.
- We perform PCA-SIFT-based matching which identifies the image pairs from the same object sources.
- We propose a robust boundary representation model, based on which the boundary matching between images is conducted to identify the changes happened in disasters.
- Extensive experiments have been conducted over large real social image data collection to evaluate the effectiveness and efficiency of our change detection system.

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

In this section, we review the existing research closely related to this work, including the image copy detection and change detection.

2.1 Image Copy Detection

Image copy detection identifies the images of the same sources. Typically, image copy detection is done by first extracting the descriptors of local interest points in each image, and counting the number of matched descriptors between two compared ones. Examples on image descriptors include SIFT[2], PCA-SIFT[3], SURF[4], GLOH[5], and Eff^2[6] etc. In [2], Lowe invented SIFT descriptor to find the similarity between images. The SIFT descriptor is extracted by four steps: scale-space extrema selection, keypoint localization, orientation assignment, and keypoint descriptor computation. First, scale-space extrema selection finds the "interest points" in the image by using the Difference-of-Gussian Function(DoG). Then, by keypoint localization, the number of interest points is minimized and noise points are reduced. After that, the orientation assignment find the orientation of the images to ensure the invariance of descriptors with respect to image location, scale and rotation. Finally, a 128-dimensional descriptor vector is computed for each interest point. SIFT descriptor is invariant to the image translation, scaling, and rotation. However, the matching over SIFT can be expensive because of the high dimensionality of the descriptors.

To improve the efficiency of local descriptor matching, different variants of SIFT have been proposed [3–6]. In [3], PCA-SIFT was proposed to reduce the complexity of SIFT. It applies principal component analysis to the normalized gradient patches. PCA-SIFT conducts the operations same as the first three stages in SIFT, which accepts the sub-pixel location, scale and dominant orientations of each keypoint, and extracts a 41×41 patch centered over the keypoint at the given scale, and aligned its dominant orientation to a canonical direction. Different from the descriptor computation in SIFT, PCA-SIFT is obtained by first pre-computing an eigenspace to express the gradient images of local patches, and then projecting the gradient image vector computed for each patch into a 36-dimensional space with the support of the eigenspace. In [4], SURF descriptor was proposed based on the Hessian matrix to approximate the previous descriptor. SURF is a basic Laplacian-based detection, which exploits Gaussian scale-space analysis to localise interest points in the image and over scales. The SURF descriptor is extracted by first fixing a reproducible orientation based on a circular region around the interest point, constructing a square region aligned to the selected orientation, splitting each region into smaller 4×4 sub-regions and computing the sum of Haar wavelet responses and that of absolute response values vertically and horizontally. A descriptor vector for all 4×4 sub-regions of length 64 is obtained as a SURF descriptor. In [5], GLOH was proposed to extend the SIFT descriptor by changing the location grid and using PCA to reduce the size. In [6], Eff^2 was proposed by detecting the interest points of images using Difference of Gaussian over different scales just like SIFT descriptor, and then extracting the information of 8 orientation buckets over each of 3 grid cells around the point. This generates a 72-dimensional vector for each key point. Since PCA-SIFT has the stable performance in all situations as demonstrated in [7] and has lowest dimensionality, we select this descriptor in our image identification.

To match the local descriptor sets of two images, there are mainly two approaches: one-many matching and one to one symmetric matching (OOS). In [2], the similarity between two local descriptor sets is measured by identifying the nearest neighbor of each local descriptor based on Euclidean distance, and calculating the number of their matched key point pairs. Using this approach, multiple key pints in a query image can be matched with a single point of an image data, thus matches over noise key points can be introduced. In [8], Zhao et al. proposed OOS matching based on a cosine distance based partial similarity matching. Using this approach, one key point in a query image can only be matched with a single key point of an image data. As such the matches caused by noise can be excluded. The similarity between two images is determined by the number of matched interest point pairs. As OOS matching achieves better performance in image copy detection, we choose it for our local descriptor similarity measure in this work.

2.2 Change Detection

Approaches have been proposed to detect changes happened at a location during a natural disasters. Traditionally, changes are detected by using pixel-based techniques. Typical pixel-based approaches include *image differencing*, *image regression*, *image rationing*, *vegetation index differencing*, *change vector analysis*, *background subtraction* and *pixelwise fuzzy XOR operator*. All these methods are based on pairwise pixel comparison, which is not robust to the image content shift or noises. Moreover, the pixel-level comparison suffers from high time cost.

Existing literatures have proposed object-based change detection(OBCD) techniques for the geographic data. OBCD compares and detects the changes by using the objects, each of which is the group of pixels contains meaningful data. Normally, OBCD algorithms are applied to satellites, remote sensing and the Synthetic aperture radar (SAR) images and detect the damage and geographical changes from them. Recently, 3D GIS model and Terra-SAR-X are evolved from the GIS and the SAR for change detection. With the improvement of object segmentation, object-based damage detection were well studied for the urban area damage detection. In [9], H.Murakami et al. proposed the simple damage detection method, which identifies changes by subtracting the digital surface model(DSM) from another DSM. In [10], M.Turker et al. used the watershed segmentation to create the segmented building vectors and calculate the shadow area of the segmented building to detect the damages. In [11], L.Matikainen et al. use object-based GIS model data with the overlap analysis algorithm change detection. Recently, L.Gong et al.[12] used VHR Terra SAR-X for finding the changes during earthquake. In [13], J.Tu et al. use the 3D GIS model image to detect the building damages during Beichuan earthquake. The 3D GIS model is exploited to extract the vectors of building images, and the height of a building is estimated using the shadow detection. Different building damage types are detected based on the changes detection of the building area, the height between the pre- and post-disaster, and the building rooftops texture information. Using the satellite, remote and SAR images with damage detection algorithms can achieve high accuracy. Although these images can cover the urban area, there is an issue of retrieving pre-event data sets. Taking all the imagery data for whole country or land is always hard to achieve. There are some areas which are not covered by the images. Media sharing communities provide great sources for capturing disasters during crisis. Thus, it is demanded to conduct the single pre- and post-event image detection in social communities for disaster management. However, existing techniques use typical features for satellite images such as shadow can not be obtained in images from public uploading in media sharing communities. New techniques should be developed for identifying the detection of damages over social images.

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. 1, 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.

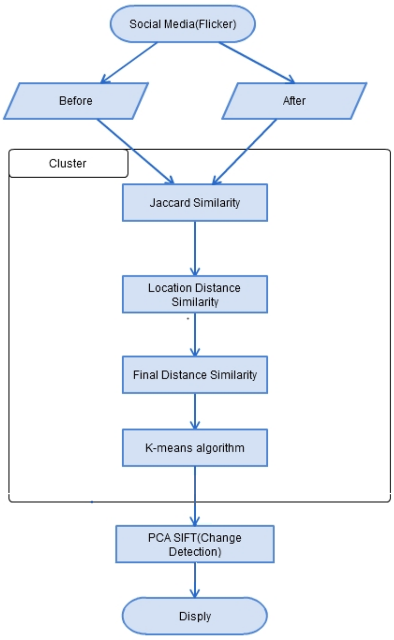

Fig. 1. The Framework

– **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 change detection from social media images. Specifically, we model 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, which contain their semantic meanings. Intuitively, the images with similar tag sets have high probability of coming from the same source. Thus, it is necessary to design a similarity function over tag sets of images, based on which the possible image pairs from the same source can be identified. Jaccard similarity has been successfully used in existing literatures for set matching. Thus, in this work, we exploit Jaccard similarity-based measure for tag set measure, and focus on how to construct virtual tag set for a group of images, and how to update it in dynamic environment. For single images, its tag set can be modelled as a set of single tags with weight 1. Given a group of images, its virtual tag set is constructed by averaging the weights of each tag appearing in all its images. If a specific tag only appears in part of images, the weight of this tag in the other images of the group will be 0. Given N images, let $\{K_1, K_2, ...K_n\}$ be the tags appearing in the image tag sets, and w_{ij} the weight of tag K_i in image j. This image set can be modelled as a set of weighted tags as below:

$$CentroidKeyword = \{W_1K_1, W_2K_2...W_nK_n\} \tag{1}$$

where W_i is computed by:

$$W_i = \frac{\sum_{j=1}^{N} w_{ij}}{N} \tag{2}$$

Suppose we have 2 images, I_1 and I_2, containing 3 and 5 keywords, where I_1: {"*earthquake*", "*apple*", "*happy*"} and I_2: {"*earthquake*", "*natural*", "*disaster*", "*apple*", "*Nepal*"}. Then, the virtual tag set of this group is {(1)earthquake, (1)apple, (0.5)natural, (0.5)disaster, (0.5)Nepal}. Figure 2 shows an sample from the change detection application[1].

```
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. 2. Centroids Keywords from the appreciation

[1] 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.

Given two weighted tag sets A and B, their Jaccard-based tag similarity is defined as:

$$J_t(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} = \frac{\|A \cap B\|}{\|A\| + \|B\| - \|A \cap B\|} \tag{3}$$

where the $\|A\|$ is the weighted number of tags in set A.

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 [14] 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.

$$a = sin(Lat/2) + cosLat1cosLat2sin(Lon/2) \tag{4}$$

$$c = 2atan2(\sqrt{a}, \sqrt{(1-a)}) \tag{5}$$

$$d = Rc \tag{6}$$

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 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 [15], the location distances are normalised to removed the potential biases among different features. Specifically, Johor et al. 2013 [15] 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:

$$D' = \frac{D - min(D)}{max(D) - min(D)} \tag{7}$$

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):

$$S_{SocSim} = \frac{1}{D+1} * S_{tag}, \tag{8}$$

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 [16].

Moreover, Cheung [17] 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.

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,

[1] Layer is the complete learning processes(step 1-5) for the K-MEANS
[2] The process at updating the centroid(step 2-3)

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.

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:

$$CentroidKeyword = K_1(\frac{P_0 + P_1..P_n}{n}), K_2(\frac{P_0 + P_1..P_n}{n})...K_n(\frac{P_n}{n}), \tag{9}$$

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. 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"}

$$\begin{gathered} CentroidKeyword = earthqauake(\tfrac{P_1+P_2}{2}), apple(\tfrac{P_1+P_2}{2}), natural(\tfrac{P_1}{2}), \\ disaster(\tfrac{P_1}{2}), Nepal(\tfrac{P_1}{2}) = earthqauake(\tfrac{1+1}{2}), apple(\tfrac{1+1}{2}), \\ natural(\tfrac{1}{2}), disaster(\tfrac{1}{2}), Nepal(\tfrac{1}{2}). \end{gathered} \tag{10}$$

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.

6 Evaluation

6.1 Experimental set up

probability of missed detection = missed detections / targets

-number of missed detections = number of right image pairs that have not been detected -number of targets = the image manually i choose

probability of false alarm = false alarms / non targets -number of false alarms = number of detected image pairs that are not correct (not in manually labelled ground truth). -number of non targets = auto gathered targets (by using tags) except the right detections

6.2 methodology

6.3 Effectiveness

Effect of clustering probability of missed detection by cluster number changing probability of false alarm by cluster number changing

Comparing with existing technique 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 You do clustering using two approaches (Recursive 2-Means and K-Means) over the whole dataset, and donot use hashing. test the time cost by changing the cluster number.

Effect of hashing index comparing clustering with and without HASH

Efficiency comparison Detection efficiency by varying data size

this is only for comparing our boundary-based approach and existing shape-based approach.

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. 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.
3. 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.
4. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, pp. 404–417, Springer, 2006.
5. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. on PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.
6. H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, and L. Amsaleg, "Scalability of local image descriptors: a comparative study," in *MM*, pp. 589–598, 2006.
7. 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.
8. W.-L. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning.," *IEEE Trans. on MM*, pp. 1037–1048, 2007.
9. H. Murakami, K. Nakagawa, H. Hasegawa, T. Shibata, and E. Iwanami, "Change detection of buildings using an airborne laser scanner," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2, pp. 148–152, 1999.
10. M. Turker and E. Sumer, "Building-based damage detection due to earthquake using the watershed segmentation of the post-event aerial images," *International Journal of Remote Sensing*, vol. 29, no. 11, pp. 3073–3089, 2008.
11. L. Matikainen, J. Hyypp, E. Ahokas, L. Markelin, and H. Kaartinen, "Automatic detection of buildings and changes in buildings for updating of maps," *Remote Sensing*, vol. 2, no. 5, pp. 1217–1248, 2010.
12. L. Gong, C. Wang, F. Wu, J. Zhang, H. Zhang, and Q. Li, "Earthquake-induced building damage detection with post-event sub-meter vhr terrasar-x staring spotlight imagery," *Remote Sensing*, vol. 8, no. 11, p. 887, 2016.
13. J. Tu, H. Sui, W. Feng, and Z. Song, "Automatic building damage detection method using high-resolution remote sensing images and 3d gis model.," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 43–50, 2016.
14. C. C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.
15. 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.
16. P. Berkhin, *A Survey of Clustering Data Mining Techniques*, pp. 25–71. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
17. Y.-M. Cheung, "k-means: A new generalized k-means clustering algorithm," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2883 – 2893, 2003.