FEATURE EXTRACTION, CORRESPONDENCE AND STITCHING OF AERIAL IMAGES FOR GENERATING MAP OF DISASTER AFFECTED AREA

[1]Ankit Dwivedi, [2]Kumar Pratik Raj, [3]Neha R, [4]Komal Sharma, [5]Rajesh N
Dept of E&CE, Nitte Meenakshi Institute of Technology, Bangalore,India
Email:[1]avadraankit@gmail.com, [2]pratik8punked@gmail.com,
[3]neharavindra2408@gmail.com, [4]komalsharma.ks2310@gmail.com, [5]nrajesh7@gmail.com

Abstract— This paper presents a robust method for map generation of area affected by natural disaster. For minimising the damage, a disaster recovery plan is required to be developed which demands faster assessment and continuous collection of information. It requires full aerial view of disaster affected geographical region. Unmanned aerial vehicles (UAVs) serves as first responders to provide bird's eye view images of disaster affected areas and also used in various applications such as environmental monitoring, aerial imaging or surveillance

This paper proposes a technique to design a robust feature extractor and descriptor for construction of map. The extracted features are required to be invariant to different transformation of image like image rotation, scale change and illumination. We have adapted the combination of Scale Invariant Features Transform (SIFT) and Binary Robust Independent Elementary Features (BRIEF) algorithm for feature detection and descriptor development for Map Building applications. Here we develop a method to select the most stable features in order to improve the performance of map building.

Index Terms—BRIEF, Feature correspondence, SIFT, Image Stitching.

I. INTRODUCTION

In present scenario Computer Vision is playing a vital role in many engineering applications. The market requirements encourage the development of high performance computer vision algorithms. A lot of different feature detectors and descriptors have successfully been applied in many computer vision tasks. However, these methods require high computation time and memory storage. Aerial video, collected by visible video cameras deployed on small UAV platforms, is rapidly becoming low-cost and up-to-date source of imagery and greatly useful in natural disaster management[1].

Feature points extraction, descriptor construction and finding their correspondences are one of the foundations of computer vision for a variety of applications [2]. The features detected should be invariant to viewing conditions, such as rotation, translation, scaling and blur (affine transformation) [5]. Different approaches have been developed to extract invariant and distinctive features. SIFT is the most popular amongst all the methods. However due to huge computational requirements, SIFT doesn't caters to the real time requirements [4].

Developing an effective small UAV system by the integration of a GPS, an inertial sensor, and an image sensor will help in creating the map and collection of crucial information in the event of disasters [3]

II. IMAGE STITCHING PROCEDURE

The steps involved in the feature-based image stitching process are as follows: Pre-processing, Feature extraction, Descriptor development, Feature matching, Transformation Model Estimation, Image re-sampling and Image stitching.

A. Image Pre-Processing

The images to be processed often have scale differences and contain noise, motion blur, haze, and sensor nonlinearities. To facilitate feature selection, it may be necessary to enhance image intensities using smoothing or de-blurring operations.

B. Feature Extraction

There are different types of features. In this step we are finding the key points or feature points in an image which is nothing but the points in an image which have high frequencies and convey more information about the image than other points. It is the basis of feature detection as only after knowing the key points we can move ahead to building descriptors for the key points.

For extracting the point feature we have focused on SIFT method [4]. This is the stage where the interest points, which are called key points in the SIFT framework, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-smoothed images is taken. Key points are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales. Specifically, a DoG image is given by

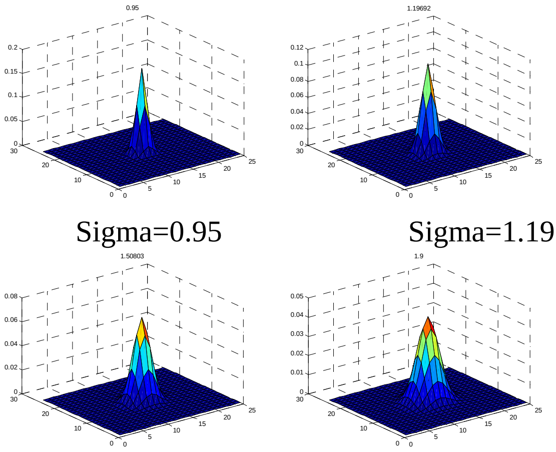
$$D(x,y,\sigma) = L(x,y,k_i\sigma) - L(x,y,k_j\sigma)$$
(1)

Where $L(x,y,k\sigma)$ is the convolution of the original image $I(x,y)$ with the Gaussian filter $G(x,y,k\sigma)$ at scale $k\sigma$, i.e.

$$L(x,y,k\sigma) = G(x,y,k\sigma) * I(x,y)$$
(2)

For scale space extrema detection, the convolved images are grouped by quad (a quad corresponds to using four values of σ), and the value of k_i is selected so that we obtain a fixed number of convolved images per quad. Then the Difference-of-Gaussian images are taken from adjacent Gaussian-smoothed images per quad.

Once DoG images have been obtained, key points are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate key point.



Sigma=0.95 Sigma=1.196

Sigma=1.508 Sigma=1.9

Fig. 1: Gaussian filter with different variance

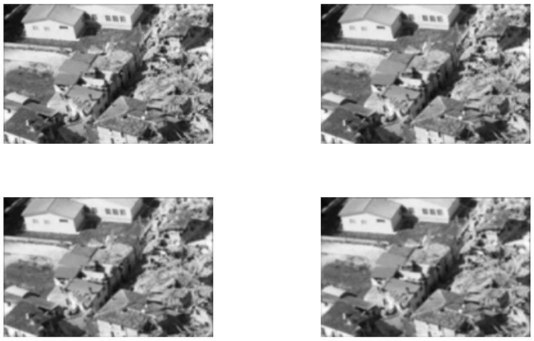


Fig.2: Gaussian filtered images with 4 different variances

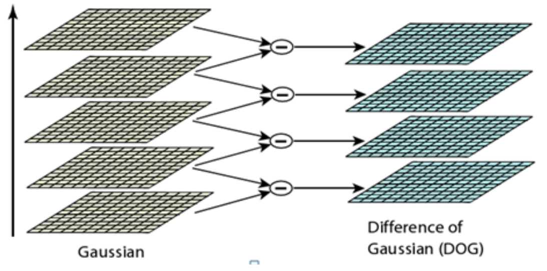


Fig. 3: Difference of Gaussian

Fig. 4: Three sets of DOG

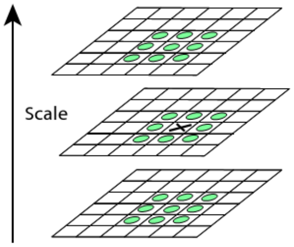


Fig. 5: Extrema detection

C. Descriptor Building

In this step, each key point is assigned one or more orientations based on local image gradient directions. This is the major step in achieving invariance to rotation as the key point descriptor can be represented with respect to this orientation compared with its neighbors and therefore it achieves invariance to image rotation, translation, etc.

First, we select Gaussian-smoothed image $L(x,y,\sigma)$ at the key point's scaleσ, so that all computations are performed in a scale-invariant manner. For an image $L(x,y)$ at scale σ, the gradient magnitude, $m(x,y)$, and orientation, $\theta(x,y)$, are pre computed using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$
(3)

$$\theta(x,y) = \mathrm{atan2}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y))$$
(4)

The magnitude and orientation calculations for the gradient are done for each pixel in the neighborhood of the key point in the Gaussian smoothed image L. An histogram of the relative orientations with 36 bins is formed, with each bin having a range of 10 degrees. The peak present in this histogram corresponds to the dominant orientations. Once the histogram is tabulated, the orientations corresponding to the highest peak and local peaks that are within 80% of the highest peaks are assigned to the key point.

The steps explained above steps finds key point location at particular scale and assigns orientations for them. This factor ensures invariance to image location, scale and rotation. The next step is to compute a descriptor vector for each strong key point such that the descriptor is highly redundant and invariant to the variations such as illumination, rotation, translation, different viewpoints , etc. This step is performed on the Gaussian image which is closest in scale to the key point's scale.

First we take the 16 x 16 region around the key point and then we split that region into the sub regions of dimensions 4 x 4. Now for each 4 x 4 sub region we define a histogram of 8 bins with each bin having a range of 45 degrees. This histogram is then filled using the magnitude and orientation values. The magnitudes are again weighted by a Gaussian function with the sigma value equal to one half the width of the descriptor window size. The descriptor then is finally computed using the values of histogram. Since we have 16 sub regions of 4 x 4 dimensions and hence we have 16 histograms with each having 8 bins and hence the vector becomes of size 16 x 8 = 128.

Descriptors which are having dimensions lesser than 128 bits don't perform very good for a lot of matching tasks and hence computational cost remains low due to the fact that we use approximate method for finding the nearest neighbor. Longer descriptors are better but they have the problems of taking longer computational time and also that there is an additional danger to increased distortion and occlusion. It is shown that feature matching has a good accuracy. Therefore SIFT descriptors are invariant to a lot of minor affine changes.

BRIEF KEY POINT DESCRIPTOR: the image patches can be effectively classified on the basis of comparatively small number of pair wise intensity comparisons. Here, we simply create a bit vector out of the test responses, which we compute after having smoothed the image patch. More specifically, we define test on patch p of size S × S as

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

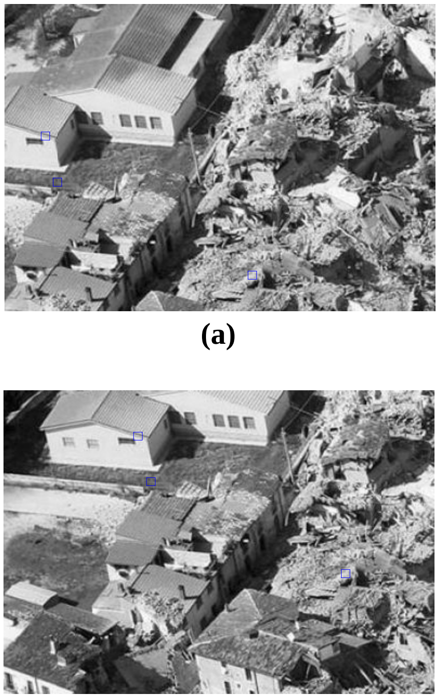
Where p(x) is the pixel intensity in a smoothed version of p at x = (u, v). Choosing a set of n (x, y)-location pairs uniquely defines a set of binary tests.

We take our BRIEF descriptor to be the n-dimensional bit string

$$f_{n_d}(\mathbf{p}) := \sum_{1 \le i \le n_d} 2^{i-1}\, \tau(\mathbf{p};\mathbf{x}_i,\mathbf{y}_i) \tag{6}$$

D. *Feature matching*

Here descriptors form both the sets are compared and based upon the matching amount of the descriptors we can find the matching percentage between the two image features.



(a)

(b)

Fig. 6: (a), (b): Three most stable matched feature points.

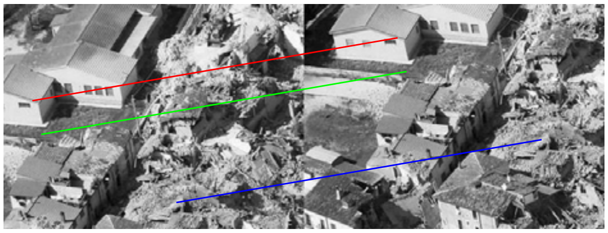


Fig. 7: Matching the feature points

E. *Transformation Model Estimation*

Geometric transformations modify the spatial relationship between pixels in an image. A geometric transformation consists of two basic operations: A spatial transformation of coordinates and intensity interpolation that assigns intensity values to the spatially transformed pixels.

The transformation of coordinates may be expressed as : (x,y)=T{(v,w)}, Where (v,w) are pixel coordinates in the original image and (x,y) are the corresponding pixel coordinates in the transformed image.

After the feature correspondence is established the mapping function is constructed and it should transform the target image to be aligned on the referenced one.

Knowing a set of corresponding control points in two images, many transformation functions can be used to accurately map the control points to each other. Since a transformation is computed from the control point correspondences, error in the correspondences will carry over to the transformation function. It is desired for a transformation to smooth the noise and small inaccuracies in the correspondences. Therefore, in image registration when noise and inaccuracies are present, approximation methods are preferred over interpolating methods.

Many simple spatial transformations can be expressed in terms of the general 3 x 3 transformation matrix shown below. It handles local scaling, overall scaling, shearing, rotation, reflection, translation, and perspective in 2-space without loss of generality. When given Matrix is multiplied with [x, y, w], a 2-space vector represented in homogeneous coordinates, it yields the 2-space vector [u, v, w'].

$$\begin{bmatrix} u & v & w' \end{bmatrix} = \begin{bmatrix} x & y & w \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{7}$$

Multiple transformations can be collapsed into a single composite transform. The transformations are combined by taking the product of the 3 x 3 matrices. For example, the composite transform representing a translation followed by a rotation and a scale change is given below.

$$[u \quad v \quad 1] = [x \quad y \quad 1]\, M_{comp} \tag{8}$$

Where,

$$M_{comp} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_{comp} = \left[\begin{matrix} S_x \cos\theta & S_y \sin\theta \\ -S_x \cos\theta & S_y \cos\theta \\ S_x(T_x \cos\theta - T_y \sin\theta) & S_y(T_x \sin\theta + T_y \cos\theta) \end{matrix}\right.$$

From the previous step, i.e. feature matching, we have obtained the corresponding points on the reference and target image. This means we have [u, v] and [x, y]. The task of transformation model estimation is to find the transformation matrix Mcomp. This can be done by following the steps given below:

- Accumulate the pairs of control points say, [u1, v1]-[x1, y1], [u2, v2]-[x2, y2], [u3, v3]-[x3, y3] and so on, where [u, v] are control points on target image and [x, y] are control points on the reference image.
- Substitute the accumulated points in the equation of affine transform.
- Perform the matrix multiplication and obtain the simultaneous equations.
- The transformation matrix can be obtained by solving these simultaneous equations.

The control points are given in the following form:

$$\begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \tag{9}$$

By dividing the u-v matrix by x-y matrix yields the transformation matrix which is the transformation model. The next step in image registration is image transformation, where the target image is reformed to match the reference image.

F. *Image re-sampling and Image stitching*

The mapping functions constructed during the previous step are used to transform the sensed image and thus to register the images. The transformation can be realized in a forward or backward manner. A forward method is complicated to implement, as it can produce holes and/or overlaps in the output image (due to the discretization and rounding). Hence, the backward approach is usually chosen. The registered image data from the sensed image are determined using the coordinates of the target pixel (the same coordinate system as of the reference image) and the inverse of the estimated mapping function. The image interpolation takes place in the sensed image on the regular grid. In this way neither holes nor overlaps can occur in the output image. Different images are stitched and shown in Fig. 8.

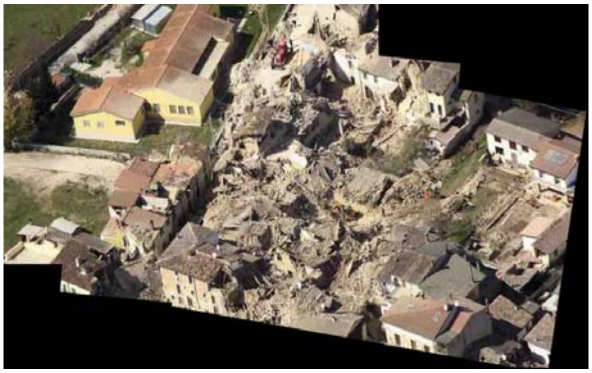


Fig. 8: Stitched Images.

III. Conclusion

Feature detection, correspondence and stitching method based on SIFT and BRIEF used here shows the good performance. Experiments were successfully carried out with real image data.

REFERENCES

[1] Jun Wu and Guoqing Zhou, “High-Resolution Planimetric Mapping From UAV Video For Quick-Response to Natural Disaster”, IEEE 2006

[2] Abdelkrim Nemra and Nabil Aouf “Robust Feature Extraction and Correspondence For UAV Map Building” , 17th Mediterranean Conference on Control & Automation Makedonia Palace, Thessaloniki, Greece June 24 - 26, 2009

[3] Taro Suzuki1, Daichi Miyoshi, Jun-ichi Meguro, Yoshiharu Amano, Takumi Hashizume, Koich Sato and Jun-ich Takiguchi, “ Real-time Hazard Map Generation Using Small Unmanned Aerial Vehicle”, SICE Annual Conference 2008

[4] David G. Lowe “Distinctive Image Features from Scale-Invariant keypoints” International Journal of Computer Vision, 60(2), p 91-110, 2004,.

[5] Krystian Mikolajczyk & Cordelia Schmid “Scale & affine invariant interest point detectors” International Journal of Computer Vision 60(1), p 63–86, 2004.