Dense Point Cloud Extraction from UAV Captured Images in Forest Area

Wang Tao$^{\#1}$, Yan Lei$^{\#2}$, Peter Mooney^{*3}

$^{\#}$*Beijing Key lab of Spatial Information Integration and Its Applications (Peking University), China*
^{1}wangtao4321@yahoo.cn
^{2}Lyan@pku.edu.cn

**Department of Computer Science, National University of Ireland Maynooth (NUIM), Co. Kildare. Ireland.*
^{3}peter.mooney@nuim.ie

***Abstract— LIDAR (Light Detection And Ranging)* is widely used in forestry applications to obtain information about tree density, composition, change, etc. An advantage of LIDAR is its ability to get this information in a 3D structure. However, the density of LIDAR data is low, the acquisition of LIDAR data is often very expensive, and it is difficult to be utilised in small areas. In this article we present an alternative to LIDAR by using a UAV (Unmanned Aerial Vehicle) to acquire high resolution images of the forest. Using the dense match method a dense point cloud can be generated. Our analysis shows that this method can provide a good alternative to using LIDAR in situations such as these.**

***Keywords*— UAV, Dense Match, Point Cloud, Forest, SFM**

I. Introduction

Using point data one can extract an object's 3D information and structure. Point matching means matching corresponding points between two or more images of the same scene and this is an important feature of many computer vision and pattern recognition tasks, including object recognition and tracking and 3D scene reconstruction. This makes point data a very important source for data mapping purposes. Because of the importance of point cloud data for many applications LIDAR data is widely used in many projects. As Bartels and Wei [11] comment "is an important modality in terrain and land surveying for many environmental, engineering and civil applications". However, when using LIDAR to extract the parameters and characteristics of forest areas there are a number of problems with the approach namely: data resolution, cost, and data processing requirements. The Unmanned Aerial Vehicle (UAV) has many advantages. A key feature of the UAV is that it is especially applicable to capturing high resolution images in small areas. Overall, it is a low-cost system when compared to LIDAR systems. However, the images captured by the UVA are of low resolution. The UAV only possess a regular GPS receiver and a standard digital photogrammetry system on board.

When a collection of images are captured by the UAV they must be processed using computer vision techniques to proceed to the stage of point cloud extraction. Some software options are already available for use. Baltsavias et al [2] and Waser et al [3] developed a new image matching software package. They demonstrated its application in 3D tree modelling by comparing this to data obtained by the airborne laser. It showed that photogrammetric DSM (Digital Surface Models) can be denser than a DSM generated by LIDAR. Leberl et al [4] compared point clouds from aerial and street-side LIDAR systems with those created from images. They show that the photogrammetric accuracy compares very well with the LIDAR method. However the key advantage of the photogrammetric approach is that the density of surface points is much higher from the images than from the LIDAR method. The authors conclude that "throughput is commensurate with a fully automated all-digital approach". When image capture has been completed the next step is to manage and process the collection of images. Snavely [6] developed some new 3D reconstruction algorithms for his PhD thesis. These algorithms operate on large, diverse, image collections. Microsoft Live Labs have recently developed a commercial software package called Photosynth [12]. Photosynth works by taking a collection of digital photographs, mashing them together, and recreating a 3D scene from them which has 360^{o} providing users with a photo-realistic experience. Yasutaka [8] uses a simple but effective method for turning a patch model into a mesh suitable for image-based modelling. In this article we describe an approach to process the images captured by our UAV using computer vision algorithms and techniques to generate point cloud data. We show that a good quality point cloud can be generated from the UAV captured images.

The remainder of our paper is organised as follows. In section II we outlined the algorithm in detail providing information about its configuration and mathematical basis. In Section III we present an experimental example. The final section, section IV, we provide some conclusions.

II. ALGORITHM Description

In this section we will provide the details of our algorithm for extracting dense point clouds from UAV captured images. In this work we are using the Scale-Invariant Feature Transform (SIFT) feature extraction algorithm to extract feature points. SIFT is invariant to image scaling and rotation and partially invariant to change in illumination and 3D camera viewpoint. There are four components of SIFT: (1) Scale-space extrema detection, (2) keypoint localization, (3) orientation assignment, and (4) computation of the keypoint descriptors. We will now discuss each of these in detail in the remainder of this section.

A. Scale-space extrema detection

978-1-4244-8351-8/11/$26.00 ©2011 IEEE

The first stage of computation searches over all scales and image locations. This is implemented efficiently by using a Difference-of-Gaussian (DoG) function to identify potential interest points. These points will be invariant to scale and orientation.

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{x^2+y^2\big/2\sigma^2}$$

To perform this the image is convolved with Gaussian filters at different scales. After this the difference of successive Gaussian-blurred images are computed. Keypoints are then extracted as maxima/minima of the difference of Gaussians which occur at multiple scales. Specifically, a DoG image is given by

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

Where $L(x,y,\sigma)$ is the convolution of the original image

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

The process is explained in Figure 1 below. For scale-space extrema detection using the SIFT algorithm the image is first convolved with Gaussian-blurs at different scales. Then the difference-of-Gaussian images are taken from adjacent Gaussian-blurred images on a per octave basis.

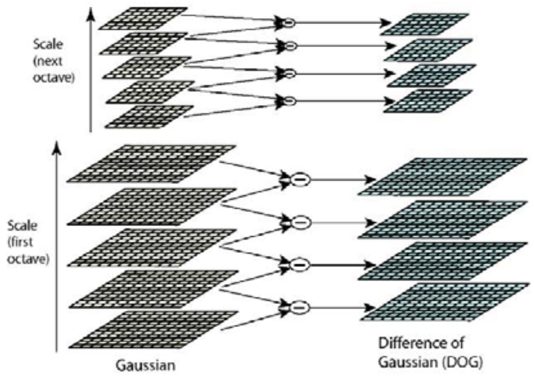

Fig.1.Diagram showing the blurred images at different scales, and the computation of the difference-of-Gaussian images

In the discrete case, the algorithm will compare the nearest 26 neighbours in a discretized scale-space volume, as shown in Figure 2. The 26 neighbors are coloured green.

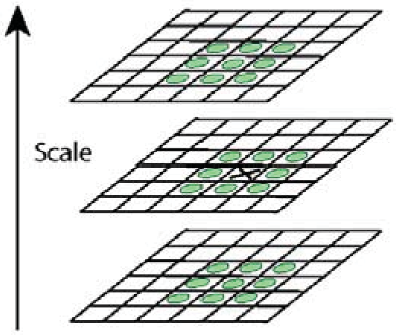

Fig.2.Local extrema detection, the pixel marked x is compared against its 26 neighbors in a 3*3*3 neighborhood that spans adjacent DoG images

B. Keypoint localization

The next component in SIFT is keypoint localization. At each keypoint candidate location a model must be fitted to determine location and scale. Keypoints are selected based on measurement of their stability. The interpolation is performed using the quadratic Taylor expansion of the Difference-of-Gaussian (DoG) function scale-space function.
This Taylor expansion is given by:

$$D(X)=D+\frac{\partial D^T}{\partial X}X+\frac{1}{2}X^T\frac{\partial D^T}{\partial X}X$$

Where x=(x,y,) is the offset from this point.

C. Orientation assignment

One or more orientations are assigned to each keypoint location based on the local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation scale and location for each feature thereby providing invariance to these transformations. First, the Gaussian-smoothed image $L(x,y,\sigma)$ at the keypoint's scale σ is taken so that all computations are performed in a scale-invariant manner. For a sample image $L(x,y)$ at scale σ, the gradient magnitude, $m(x,y)$, and orientation, $\theta(x,y)$, are precomputed using pixel differences. The equations for $\theta(x,y)$ and $m(x,y)$ are given as follows:

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

$$\theta(x,y)=\arctan\frac{L(x+1,y)-L(x-1,y)}{L(x,y-1)-L(x,y-1)}$$

D. Keypoint descriptor

The final component of SIFT involves the computation of the keypoint descriptors. The local image gradients are measured at the selected scale in the region around each keypoint, as shown in Figure 3. These local image gradients are transformed into a representation that allows for significant levels of local shape distortion and change illumination.

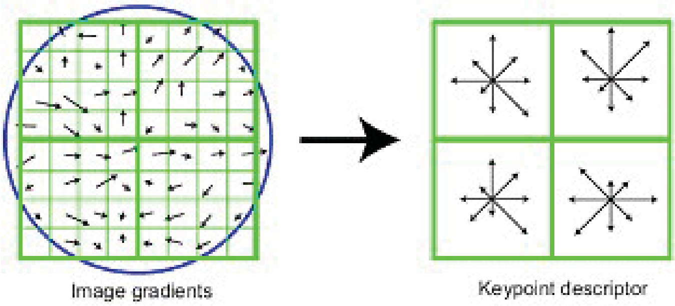

Fig.3.Sift feature descriptor

SIFT also computes a vector describing the local image appearance around the location of that feature. One simple example of a descriptor is a window of color (or grayscale) values around the detected point. The descriptor used by SIFT considers image gradients, rather than intensity values, as image derivatives are invariant to adding a constant value to the intensity of each pixel. In fact, SIFT looks at the directions of these gradients, rather than their raw magnitude, as gradient directions are even more invariant to variations in brightness and contrast across images. In particular, SIFT computes histograms of local image gradient directions. It creates a 4 × 4 grid of histograms around a feature point, where each histogram contains eight bins for gradient directions, resulting in a 4 × 4 × 8 = 128-dimensional descriptor. Thus, each feature f consists of a 2D location (fx,fy), and a descriptor vector fd. The canonical scale and orientation of a feature are not used in the remainder of the pipeline.

RANSAC is an abbreviation for "random sample consensus"[14]. It is an algorithm to estimate parameters of a mathematical model from a set of observed data which contains outliers.

The input to the RANSAC algorithm is a set of observed data values, a parameterized model which can explain or be fitted to the observations, and some confidence parameters. RANSAC achieves its goal by iteratively selecting a random subset of the original data. These data are hypothetical inliers and this hypothesis is then tested as follows:

1. A model is fitted to the hypothetical inlier, i.e. all free parameters of the model are reconstructed from the data set.

2. All other data are then tested against the fitted model and, if a point fits well to the estimated model, also considered as a hypothetical inlier.

3. The estimated model is reasonably good if sufficiently many points have been classified as hypothetical inliers.

4. The model is re-estimated from all hypothetical inliers, because it has only been estimated from the initial set of hypothetical inliers.

5. Finally, the model is evaluated by estimating the error of the inliers relative to the model.

For each input photo, the pipeline determines the location from which the photo was taken and direction in which the camera was pointed, and recovers the 3D coordinates of a sparse set of points in the scene.

Refine matching using RANSAC +8-point algorithm to estimate fundamental matrices between pairs.

It is difficult to initialize all the cameras at once. Because structure from motion with two cameras is easy. Once we have an initial model, it's easy to add new cameras.

We start with a small seed reconstruction, and grow.

It is necessary to use a strong initial pair of images with many matches, but which has as large a baseline as possible. By chose two calibrated images with corresponding points, compute the camera and point positions. Use the 5-point method to find the essential matrix between the images. While there are connected images remaining, pick image that sees the most existing 3D points, estimate the pose of that camera, triangulate any new points, run bundle adjustment.

III. EXPERIMENT

Test site was located in the SiZhouTou of Xiangshan town, Zhejiang Province, China, as shown in Fig.4. The flying system has a size of 830 mm, payload of it is 200g. The weight of the system varies between 1 kg and 1.5kg.

The system is restricted to fly under wind speeds smaller than 6m/s. A maximum flight height of 4000m and an operation distance of up to 5km are possible. For take-off and landing a runway with a length of 5m to 25m is needed.

The sensor on the system is FUJIFILM-FinepixZ10fd digital camera, pixel is 720, focal length is 18.9mm.

According to the photogrammetry requirement, image along track overlap should be 60%, couldn't be small than 53%, image across track overlap should be 30%, couldn't be small than 15%. Considering the weather condition, the along track overlap is 80%, across track overlap is 60%.

The maximal offset can be expected in our experiment in the main flight direction X.

When hotshoe of camera triggers, the flight control system records the following data:

Image number, camera roll angle, camera pitch angle, camera yaw angle, UAV latitude, UAV longitude, UAV altitude

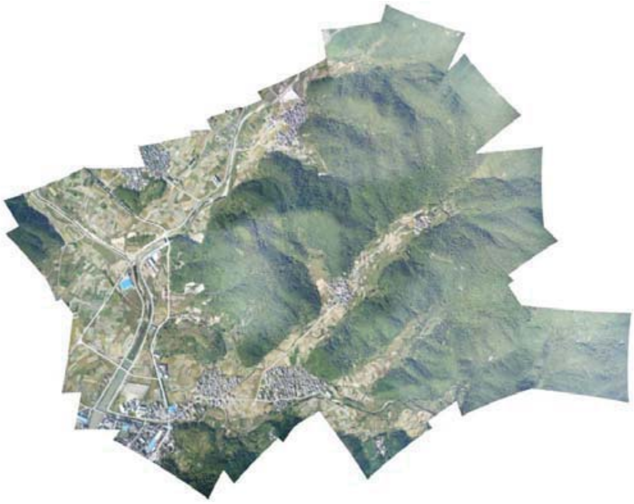

Fig. 4 Mosaic image of the whole forest area

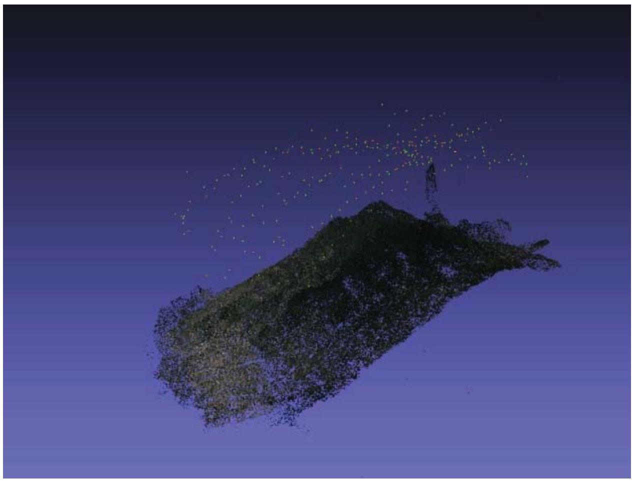

Fig. 5 Point cloud of the entire area

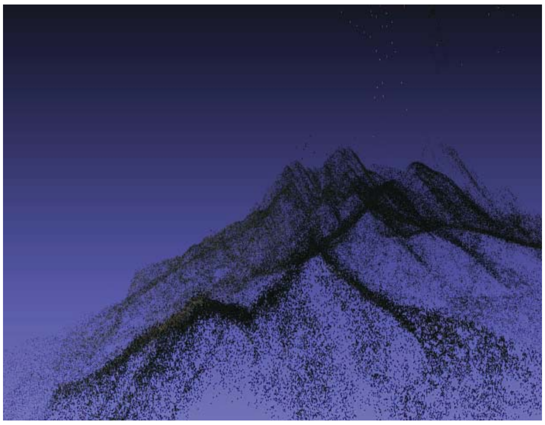

Fig. 6 Point cloud of the local area

IV. Conclusions

In this paper we have described a method for dense point cloud extraction from UAV captured imagery for the purposes of forestry analysis. We believe that this approach can provide a cost-effective alternative to LIDAR methods in performing the same task. Following from the experimental detail outlined in the previous section we show the point clouds generated for the forest area in Figure 4 in both Figure 5 and Figure 6. Figure 5 shows the point cloud for the entire area while Figure 6 shows the point cloud for one of the local areas with the test mosaic. Our approach generates a very good point cloud representation and reconstruction of the 3-D scene for the forest area. However, at this initial stage of the work we feel that the density of the point cloud is not sufficient to allow us to extract information about the tree parameters (accurately). In future work by using the navigation pose information we shall be able to generate cloud point data using geodetic coordinate more quickly. We will also investigate changing the density matching algorithm to allow for the generation of denser point clouds.

References

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] Baltsavias, E.,Gruen, A.,Eisenbeiss,H.,Zhang,L. and Waser,L. T. (2008) “High-quality imagematching and automated generation of 3D tree models,” International Journal of Remote Sensing, 29:5, pp.1243 – 1259, 2008

[3] Waser,L.T.,Baltsavias,E.,Ecker,K.,Eisenbeiss,H.,Ginzler,C.,Küchler,M ., Thee,P. and Zhang,L. “High-resolution digital surface models (DSMs) for modelling fractional shrub/tree cover in a mireenvironment,” International Journal of Remote Sensing,29:5, pp.1261 – 1276, 2008.

[4] F.Leberl, A.Irschara, T.Pock, P.Meixner, M.Gruber, S. Scholz,and A.Wiechert. “Point Clouds: Lidar versus 3D Vision”, Photogrammetric & Remote Sensing, 76:10, pp.1123-1134, 2010.

[5] R.I.Hartley and A.Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, 2004.

[6] Noah Snavely. "Scene Reconstruction and Visualization from Internet Photo Collections," Doctoral thesis, University of Washington, 2008.

[7] Li Zhang, “automatic digital surface model generation from linear array images,” Doctoral thesis, swiss federal institute of technology zurich, 2005.

[8] Yasutaka Furukawa, “High-Fidelity Image-Based Modeling”, Doctoral thesis, May 2008. University of Illinois at Urbana-Champaign. http://gradworks.umi.com/33/14/3314770.html

[9] Andrea Vedaldi, “Invariant Representation and Learning for Computer Vision,” Doctoral thesis, 2008.

[10] Yuan Xiuxiao, Modern Methodologies for Photogrammetric Point Determination,32(11),Nov.2007.

[11] Marc Bartels, Hong Wei, Threshold-free object and ground point separation in LIDAR data, Pattern Recognition Letters, Volume 31, Issue 10, July 2010, Pages 1089-1099,

[12] Microsoft Photosynth – http://photosynth.com/(2010)

[13] LeLu, Xiangtian Dai, Gregory Hager, Efficient particle filtering using RANSAC with application to 3D face tracking, Image and Vision Computing, Volume 24, Issue 6, Face Processing in Video Sequences, 1 June 2006, Pages 581-592

[14] Chia-Ming Cheng, Shang-Hong Lai, A consensus sampling technique for fast and robust model fitting, Pattern Recognition, Volume 42, Issue 7, July 2009, Pages 1318-1329,