CLOUD BASE HEIGHT ESTIMATION USING HIGH-RESOLUTION WHOLE SKY IMAGERS

Florian M. Savoy,[1] *Joseph C. Lemaitre,*[1,2] *Soumyabrata Dev,*[3] *Yee Hui Lee,*[3] *Stefan Winkler*[1]

[1] Advanced Digital Sciences Center (ADSC), University of Illinois at Urbana-Champaign, Singapore
[2] School of Basic Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[3] School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore

ABSTRACT

Fine scale cloud monitoring using ground-based imagers is becoming popular for a variety of applications and domains. We present a framework for cloud base height estimation using two such imagers; our method is based on stereoscopic scene flow. We demonstrate the feasibility of our approach and use computer-generated images with controlled cloud height to validate the accuracy of our method.

***Index Terms*—** WAHRSIS, ground-based sky camera, cloud altitude, cloud motion, scene flow

1. INTRODUCTION

A precise localization of clouds in the atmosphere is required for several applications. The energy production of solar panels greatly depends on solar irradiance: due to its intermittency, operators need a precise short-term forecast of the cloud coverage above solar plants to take preventive actions before a ramp-down. Similarly, air-to-ground or air-to-air communications through the atmosphere suffer from attenuation due to rain, clouds, atmospheric particles, and water vapor [1]. Accurate information about cloud formations along the signal path is key to better understanding these phenomena.

Localizing clouds in 3D space is a challenging task. Clouds usually do not exhibit precise boundaries and may not have distinctive textures, which makes matching two images difficult. However, clouds evolve slowly, and successive frames look similar.

We present a new method to estimate the cloud base height from a succession of images captured by ground-based imagers with a hemispherical view of the sky. We use stereoscopic scene flow, which is defined as the three-dimensional motion field of points in the world [2]. This allows us to retrieve the 3D location as well as the motion of the clouds, by incorporating past frames in the computation. Although we focus on the 3D location (i.e. cloud base height) in this work, the motion information can be used as input for cloud movement prediction.

We propose a complete workflow to apply a three-dimensional scene flow algorithm in the context of cloud base height estimation. We apply our method on high-resolution images captured by WAHRSIS, our custom made whole sky imagers [3, 4], and on computer-generated images with user defined cloud parameters. The latter constitute a precise ground truth to evaluate the accuracy of our approach.

Section 2 presents related work on cloud base height estimation and scene flow. Section 3 details the workflow of our method. The validation using computer-generated images is described in Section 4. Conclusions and future work are presented in Section 5.

2. RELATED WORK

Allmen and Kegelmeyer [5] compute cloud base height from a pair of whole sky imagers using correlation of images rectified by a pseudo-Cartesian transformation and optical flow fields along successive images. Seiz et al. [6] match points extracted with Förstner and Harris operators from two different cameras with fish-eye lenses. They use least-squares matching and introduce a hierarchical pyramid-based approach based. Kassianov et al. [7] compute the overlapping area between two images from a pair of ground-based imagers using a merit function. They derive the height of the clouds from the size of this overlapping area, which results in one value per image pair. Peng et al. [8] use three total sky imagers for 3D cloud detection and tracking. They assume clouds have only planar motion vectors and define an objective function to be maximized based on both motion tracking and height estimation.

The concept of 3D scene flow was first introduced by Vedula et al. [2]. Popular methods for computing scene flow include Huguet and Devernay [9], who couple optical flow and dense stereo matching via partial differential equations; Li and Sclaroff [10], who introduce probability distributions for optical flow and disparity; and Cech et al. [11], who reduce the computational complexity using seed growing.

This research is funded by the Defence Science and Technology Agency (DSTA), Singapore.

Send correspondence to F. M. Savoy, E-mail: f.savoy@adsc.com.sg.

3. WORKFLOW

This section describes the workflow of our proposed approach. It takes as input a sequence of images taken simultaneously by two sky imagers, such as those shown in Fig. 1. They were taken by a pair of our custom-designed whole sky imagers [3, 4], which we call Wide-Angle High-Resolution Sky Imaging System (WAHRSIS), and are part of a sequence of 3.5 minutes, with an image taken every 10 seconds. The two imagers are placed approximately 95 meters apart on rooftops of the Nanyang Technological University in Singapore.

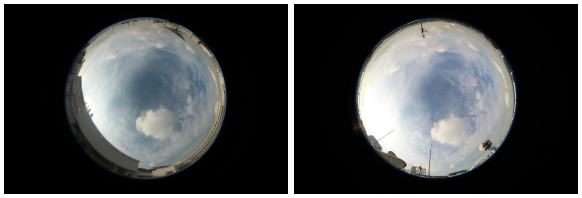


Fig. 1: Images captured by the pair of sky imagers.

3.1. Undistortion

The first step consists of removing the distortion due to the fish-eye lens in the images. The projection behavior of the lens used in our imagers follows the equisolid angle mapping function, $r = af\sin(\theta/2)$, where r is the distance from the image center, θ is the elevation angle of the incident light ray, f is the focal length, and a is a constant. We thus can retrieve the incident light ray of every pixel in the image. The position along that ray is the value which needs to be computed by triangulation to find the position of the cloud in 3D. Fig. 2 shows the original image and the pixel values projected on a hemisphere.

We use a ray-tracing approach to generate undistorted images. Considering the camera at the origin, we place a virtual plane at a user-defined height. We then compute the pixel values of the image by looking for the incident light ray that intersects the plane at the pixel location, as shown in Fig. 2. There is usually no ray going through an exact pixel location, so we interpolate using the nearest rays.

In order to map this image to real world coordinates, we give each pixel a width of one meter. We then define the height of the plane to be at 400 meters, which translates to a viewing angle of 103 degrees.

3.2. Rectification

We use the real world units defined above to place the image plane of the second imager with respect to the first one. In practice, the positions of the imagers can be retrieved by GPS.

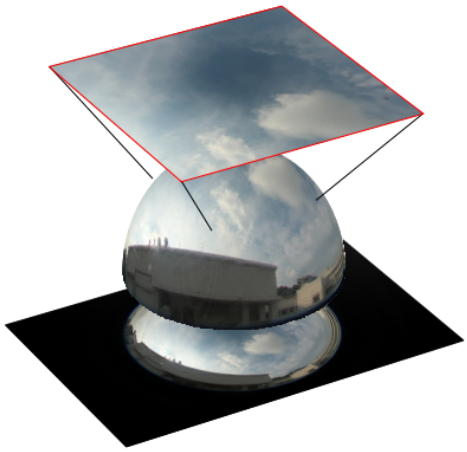
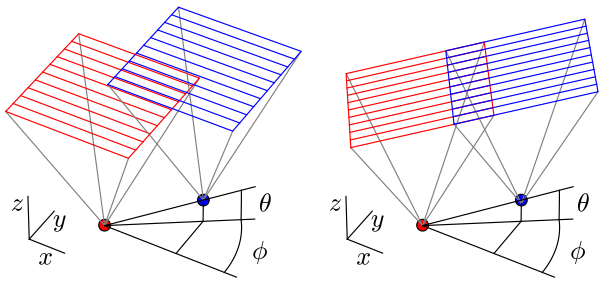


Fig. 2: Illustration of the undistortion process of an image taken with our fish-eye lens (bottom). The color value of each incident light ray is projected on a hemisphere (middle). The image plane (top) is computed as the value of each light ray going through the pixel at its location (as indicated at the corners of the image).

Stereo vision algorithms typically compute the disparity between the left and right images along one axis only in order to reduce the complexity. For this, the input images need to lie on a common image plane. The process of computing such images is known as rectification.

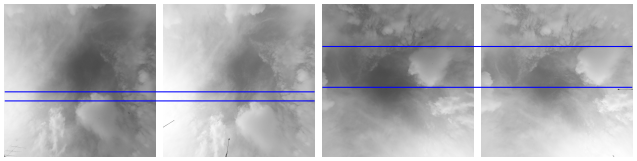
We incorporate this step into the undistortion process described above. We apply 3D rotations on the virtual image plane before computing the pixel values, as shown in Fig. 3. Two rotations are needed. The first one is applied around the z axis and relates the difference of longitude and latitude between the imagers, quantified with the azimuth angle ϕ. The second rotation is applied around the rotated y axis and relates the difference in altitude between the imagers, which is quantified with the elevation angle θ.



(a) Before rectification (b) After rectification

Fig. 3: Illustration of the rectification process. The two coloured dots represent the two imagers with their respective image planes. The disparity between the images only appears along one axis after rectification.

Fig. 4 shows both pairs of undistorted images before and after rectification. Notice how the disparity between the images is only present along the horizontal axis in Fig. 4b.



(a) Before rectification (b) After rectification

Fig. 4: Image pair before and after rectification.

3.3. Disparity Map

The next step is to compute the disparity map between the sequence of pairs of undistorted, rectified images. We use the three-dimensional scene flow algorithm proposed by Cech et al. [11], which jointly estimates a disparity map and optical flows between stereo image frames.

The strength of the scene flow approach is the inclusion of past frames into the computation of the disparity map. While clouds do not have many distinct features to track, they evolve slowly. Successive frames taken at appropriate intervals indeed do not exhibit large differences, and past frames can thus complement the information provided in the current frame.

Fig. 5 shows the disparity map computed from the pair of images shown in Fig. 4b. Each pixel value of the disparity map indicates the shift to add to the horizontal image coordinate to obtain the position of the pixel capturing the same physical point in the other image. Note that the algorithm does not define values over the whole image, as some parts do not exhibit sufficient texture or do not look similar in the two images due to the different viewpoints.

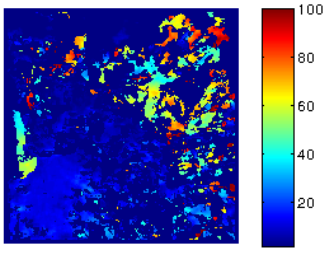


Fig. 5: Disparity map for the image pair in Fig. 4b.

3.4. 3D Point Cloud

The last step consists of mapping the disparity map to real world 3D coordinates. For every defined value of the disparity map, we compute the position of that point on both rectified images in the image plane, as shown by the red and blue dots in Fig. 6. We then consider the two rays going through those points and the device position. Those two rays intersect at the real world location of the captured point, shown in green in Fig. 6.

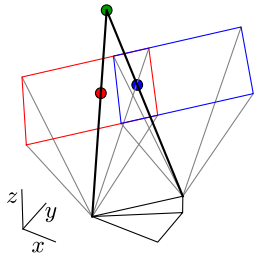


Fig. 6: Illustration of the computation of a point in 3D.

Fig. 7 shows the resulting point cloud for an image pair from the sequence. We distinguish two different clusters: The first is at an altitude of about 700 meters and corresponds to the cumulus cloud at the bottom right of the input image, while the second represents the higher cloud layer appearing at around 3000 meters.

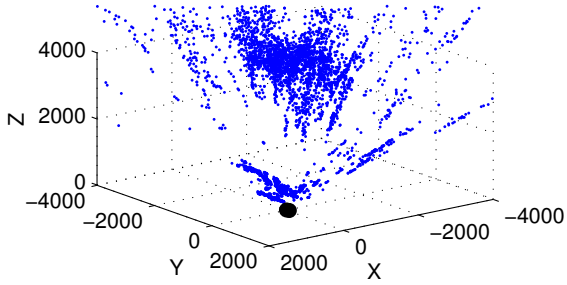


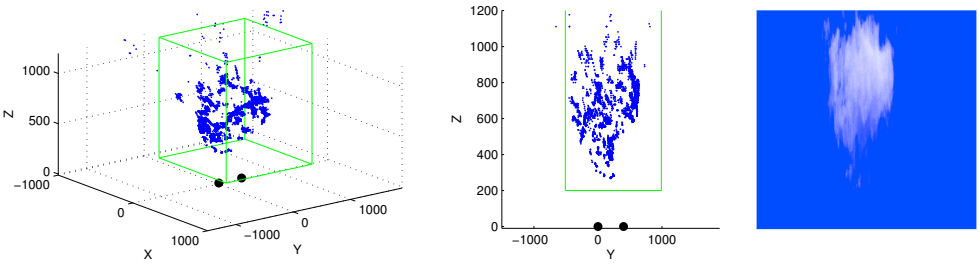
Fig. 7: 3D point cloud computed for a pair of images from the sequence.

4. VALIDATION

Since it is very difficult to measure the actual 3D location and shape of clouds, we use computer-generated images with a user-defined cloud bottom altitude for validating our approach. We use *Blender*,[1] an open-source 3D computer graphics program, for creating images using a procedural cloud shader and a model of the fish-eye lens of WAHRSIS. While being simpler than the reality, these images do have the advantage that the positions of the imagers and clouds can be user-defined.

For this experiment, we created a sequence of 21 frames with a cloud moving over the two cameras. The average cloud bottom altitude is 500 meters, and the distance between the cameras is 400 meters. Fig. 8 shows an example of such an image pair.

[1] http://www.blender.org/



(a) 3D view (b) 2D projection (c) Ground truth for (b)

Fig. 9: 3D point cloud for an image pair computer from the computer-generated sequence. The bounding boxes in which Blender generates the cloud are shown in green. The two black dots indicate the position of the imagers.

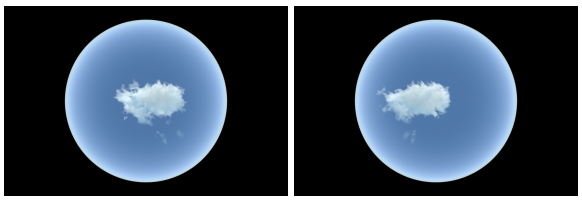


Fig. 8: Image pair from the computer-generated sequence.

Fig. 9 shows the resulting 3D point cloud for an image pair from the computer-generated sequence. It matches with the ground truth created by Blender. Averaging over the entire sequence, 82.4% of the reconstructed 3D points lie inside the bounding box in which the cloud is generated.

5. CONCLUSIONS

In this paper we have applied 3D scene flow to cloud base height estimation. We have introduced a workflow to a common scene flow algorithm in this context, which we validated with computer generated images.

Using sky imagers requires several calibration and alignment steps. In practice, the imagers are not perfectly horizontal and oriented towards a known azimuth angle. The GPS position measurements are also not accurate enough, especially for the altitude. We used manual calibration procedures to produce the results presented here, which we did not describe in this paper. Our current work includes the automation and the improvement of those procedures. This is needed to complement the primary results described in this paper with an extensive validation of the method on real images.

We will also conduct simulations with other distances between the imagers and cloud bottom altitudes in order to determine an optimal set-up of the imagers.

6. REFERENCES

[1] J. X. Yeo, Y. H. Lee, and J. T. Ong, "Performance of site diversity investigated through RADAR derived results," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 10, pp. 3890–3898, October 2011.

[2] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proc. International Conference on Computer Vision (ICCV)*, 1999.

[3] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "WAHRSIS: A low-cost, high-resolution whole sky imager with near-infrared capabilities," in *Proc. SPIE Infrared Imaging Systems*, 2014.

[4] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "Design of low-cost, compact and weather-proof whole sky imagers for high-dynamic-range captures," in *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015.

[5] M. C. Allmen and W. P. Kegelmeyer Jr., "The computation of cloud-base height from paired whole-sky imaging cameras," *Journal of Atmospheric and Oceanic Technology*, vol. 13, no. 1, pp. 97–113, February 1996.

[6] G. Seiz, E. P. Baltsavias, and A. Gruen, "Cloud mapping from the ground: Use of photogrammetric methods," *Photogrammetric Engineering and Remote Sensing*, vol. 68, no. 9, pp. 941–951, September 2002.

[7] E. Kassianov, C. N. Long, and J. Christy, "Cloud-base-height estimation from paired ground-based hemispherical observations," *Journal of Applied Meteorology*, vol. 44, no. 8, pp. 1221–1233, August 2005.

[8] Z. Peng, S. Yoo, D. Yu, D. Huang, P. Kalb, and J. Heiser, "3D cloud detection and tracking for solar forecast using multiple sky imagers," in *Proc. ACM Symposium on Applied Computing*, 2014.

[9] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Proc. International Conference on Computer Vision (ICCV)*, 2007.

[10] R. Li and S. Sclaroff, "Multi-scale 3D scene flow from binocular stereo sequences," *Computer Vision and Image Understanding*, vol. 110, no. 1, pp. 75–90, April 2008.

[11] J. Cech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.