

IDENTIFICATION OF OBJECT IN 3D FLASH LIDAR IMAGES

MADHUSUDHANARAO M.¹, KARTHEEK K.², RAJANIKANTH M.^{3*}

¹Sri Sunflower College of Engineering & Technology, Lankapalli, Challapalli, INDIA

²Sri Sunflower College of Engineering & Technology, Lankapalli, Challapalli, INDIA

³Head of Department of M.C.A Sri Sunflower College of Engineering & Technology, Lankapalli,Challapalli,INDIA

*Corresponding Author: Email- rajanikanthmeka7@yahoo.co.in

Received: September 29, 2011; Accepted: November 03, 2011

Abstract- In this we propose an algorithm to automatically find objects in 3D Flash Lidar images. Three-dimensional Flash Lidar camera is a new technology which allows a single camera to take multiple images in a fast succession. It produces a series of range delimited imagery using a single laser flash. The imagery can distinguish between objects at different distances, such as the sea-surface and underwater objects. We apply a transformation to reduce the computational complexity and apply stable unsupervised multiphase segmentation for object identification. Numerical results are presented to validate the approach both using scalar and Vector models.

Key words - Flash Lidar, segmentation, *Classification*, frequency coefficients

1.Introduction

Three-dimensional Flash Lidar imaging represents one of the latest advances in imaging technology [18] which uses a single laser pulse to generate a three-dimensional data. 3D Flash Lidar camera uses smart pixels for signal integrators: each pixel can accurately and independently count time to the target. The light reflected from a certain distance is captured in a plane of the image volume in a fast succession.

This imagery is different from the typical LIDAR (Light Detecting and Ranging) images, where several laser pulses are used to record the distance to the surface or objects. LIDAR has many applications in environmental and civil engineering and it is widely used for forestry, building constructions, and ground surface modeling as in [4, 7, 11, 14]. There are many references for LIDAR images, while 3D Flash Lidar image is a new approach. 3D Flash Lidar images capture the luminance of each distance, then produce a succession of 2D images at particular distances from the camera, that 3D Flash Lidar images have very different characteristics compared to LIDAR images.

One of the challenges of 3D Flash Lidar images comes from the images taken from a moving vehicle. For example, when the images are taken from an airborne vehicle, one object can be captured in few different 2D images each from a different distance from the camera. Figure 1 shows an example of 3D Flash Lidar image, a partial sequence of images taken from an airborne vehicle. The plane (a) depicts objects closer to the camera, and image (b) through (h) successively farther. Notice, the glint from the water surface begins in plane (b). The submerged object comes into view beginning with plane (f), then it is clearly visible in (g). It is evident

that the airplane is not parallel to the water, since the glint first appears on the right side of the planes but disappears on the left.

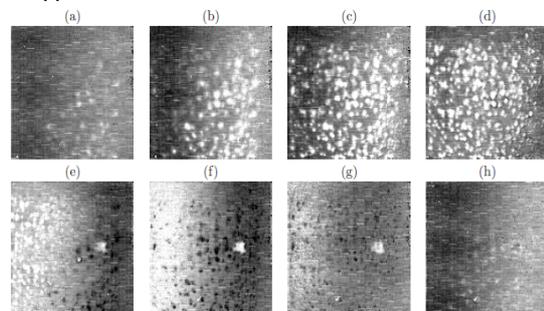


Fig. 1: An example of 3D Flash Lidar image taken from an airborne vehicle. Image (a)-(h) is a sequence of 2D image planes taken from each different distances from the camera to the floor. Image (a) and (b) are mid-air, image (b)-(e) show parts of the water surface, and image (e)-(g) captured underwater and floor.

The objective of this paper is to propose a method to easily identify objects submerged underwater. This is not a trivial issue, due to the glint, the tilt from the moving aircraft, and the volume of the data size of 3D Flash Lidar image. One can consider adjusting the tilt of the 2D planes by determining the oblique image plane that represents water surface. Instead, we propose directly finding an object without adjusting for alignment.

Here we propose a stable method by properly reducing the size and using unsupervised segmentation. In Section 2, we illustrate the procedure with the set up of the model, then a fast algorithm is explained in Section 3. Various numerical experiments are presented in Section 4, followed by concluding remarks in Section 5.

2. Multiphase segmentation for 3D Flash Lidar image:

Three-dimensional Flash Lidar images capture multiple 2D images of the same scene, where each image is from different distances to the camera. Let an image domain be $\Omega \subset \mathbb{R}^2$, and the captured image I be $I : \Omega \times D \rightarrow \mathbb{R}^m$ where $D = \{1, 2, \dots, m\}$ is the (finite) discrete values representing different distance from the camera. $I(\Omega, 1) : \Omega \rightarrow \mathbb{R}$ is the image closest to the camera, $I(\Omega, 2)$ the next closest to the camera, and $I(\Omega, m)$ represents the farthest from the camera. We use the notation (x, y, z) to represent each pixel in the image, $(x, y) \in \Omega$ and $z \in D$. Note that the z -coordinate is different from xy -coordinates of Ω . 3D Flash Lidar images capture what can be seen from the camera, that the depth information is not complete. It cannot find an object hidden behind a solid object, and the planar dimension should not be treated equally to the depth z direction. This is also noticed in [3].

To properly utilize the distance information, we consider the profile of the image in the z -direction,

$$P_{xy}^{(z)} = I(x, y, z) \quad (1)$$

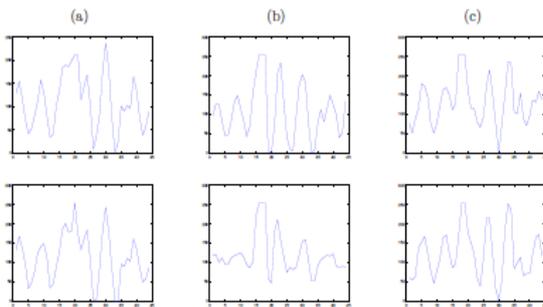


Fig. 2: Example of the profile function $p_{xy}(z)$ in (1) for the airborne example in Figure 1. The graphs in the same column are taken from locations next to each other, which shows the similarities. When the profiles are compared against different columns, the differences are more noticeable.

Figure:2 illustrates some of profiles $P_{(x,y)}(z)$ for different locations of $(x, y) \in \Omega$ for the airborne example in Figure 1. The graphs in the same column are taken from locations next to each other, which shows the similarity between them. When the profile graphs are compared against different columns, the differences are more noticeable. We capture the differences in these profiles, then cluster the profiles according to the difference by a segmentation method. Since the main idea is to capture the difference in the depth profile, we can consider reducing the size of the data for easy computation.

Step 1: Reducing the size of information.

Notice from the profile graphs in Figure 2, that the graphs are all oscillatory. Therefore, to capture the difference in these profiles, Fourier transformation with trigonometric function is a good choice for the base

function. We apply a Fast Fourier Transformation (FFT) to the z direction,

$$a_j(x, y) = \sum_{z=1}^m p_{xy}(z) e^{-i2\pi z/m} \quad (2)$$

Here $z \in D$ and m are as defined above. We pick the first few frequencies, which correspond to larger differences, and define a vectorial image on Ω as

$$u = (a_1, a_2, a_3, \dots, a_n) : \Omega \rightarrow \mathbb{R}^n \quad (3)$$

Where n is the no. of frequencies used and in this paper, we used $n = 3$. The process savings are realized when n is much smaller than m , and the elimination of the higher frequencies is a common noise suppression technique which validates our choice of using only few beginning coefficients. In our experiments, it was often sufficient to use an Image such as $\bar{u} = \sqrt{|a_1|^2 + |a_2|^2 + |a_3|^2}$, and apply scalar multiphase image segmentation. This \bar{u} is different from simply adding some planes of the image I directly. Since this \bar{u} emphasizes the differences captured by the transformation, there is no need to hand pick important planes directly.

Step 2: Segmentation and Classification.

Image segmentation is a widely studied image processing task, where it simplifies (partitions) the image, making it easier to identify certain objects or features in the image. There are various different approaches: Geman and Geman's mixture random field models [5], Mumford and Shah's piecewise smooth variational image models [12], the graph-cutting and spectral method of Shi and Malik [16], and the data-driven Monte-Carlo Markov chain model (DDMCMC) of Tu and Zhu [20] are some of

the classical well-known examples. In variational settings, Mumford-Shah model [12] and Chan-Vese (CV) model [1] are well-known and CV model has been extended to multiphase segmentation as in [2, 8, 9, 13].

We apply an unsupervised multiphase image segmentation method [15]. Different from the case of two-phase identification, multiphase segmentation has sensitivity issues: choosing an initial condition, and pre-assigning the number of phases. An unsupervised segmentation model [15] automatically gives a reasonable number of phases while it segments the image. Remote sensing and its application to 3D Flash Lidar image is an excellent application of unsupervised multiphase model, since most objects and backgrounds are not predetermined

and the number of phases are unknown. An unsupervised method can automatically give a reasonable segmentation result, and it is based on minimizing the following functional:

$$[k, \{X_i\}, \{c_i\}] \bar{u} = \mu \left(\sum_{i=1}^k \frac{P(X_i)}{|X_i|} \right) \sum_{i=1}^k F_{(X_i)} + \sum_{j=1}^n \sum_{i=1}^k \int_{X_i} |u^j - c_i^j|^2 dx$$

Here the super index j indicates the length of the vectorial image u . When a scalar image is used, set $n = 1$. The notations $P(A)$ and $|A|$ represent the one-dimensional Hausdorff measure and the Lebesgue measure of a phase A respectively. The term $\frac{P(X_i)}{|X_i|}$ is an inverse scale term that under minimization tends to favor larger objects. Note that each phase, $\{X_i\}$, intensity average $\{c_i\}$, and the number of phases k are all unknowns, only the observed image \bar{u} is given. The proposed method is these two steps: apply FFT in z -direction to reduce the size of data, then apply a fast and stable multiphase segmentation algorithm to few low frequency coefficients. By utilizing the profile, we can directly apply the algorithm to the 3D Flash Lidar image. This is more efficient compared to adjusting the 2D plan or dealing with complication of 3D reconstruction.

3. Details of the segmentation algorithm

When image segmentation is applied, we represent each segment as a characteristic function X_i for $i = 1, \dots, K$. The minimum of the energy is computed by considering the change directly as in [6, 15, 17]. When a pixel (x, y) in phase l moves to another phase j , the energy change is computed by

$$\Delta E_{lj} = \Delta F + \sum_{i=1}^k [(u^i - c^i)^2 \frac{n_i}{n_{j+1}} - (u^i - c^i)^2 \frac{n_i}{n_{l-1}}] \tag{5}$$

3D Flash Lidar Image segmentation Algorithm:

- Input a 3D Flash Lidar image l .
- Use a transformation such as FFT in z -direction, pick few frequency values a_1, a_2, \dots, a_n , and define $u = (a_1, a_2, \dots, a_n)$.
- Apply multiphase segmentation to the reduced image u by:

Set an initial phase: $|X_1| = |\Omega|$ with $k_0 = 1$.

Iterate

1. At each pixel $(x, y) \in \Omega$ (with $x_1(x, y) = 1$ and $x_i(x, y) = 0$ for $\forall i \neq 1$), Compute value $= \min_j \{\Delta E_{ij} | j \neq 1, j = 1, \dots, k + 1\}$, with ΔE_{ij} in (5) and let $h = \arg \min_j \{\Delta E_{ij} | j \neq 1, j = 1, \dots, k + 1\}$. Here $k + 1$ refers to a new phase.

Then,

$$\begin{cases} \text{if value} < 0, \text{ set } X_h^{(x,y)} = 1 \\ \quad \text{and } X_1^{(x,y)} = 0, \\ \text{if value} > 0, \text{ do nothing} \end{cases}$$

2. Update $k = h$, calculate $n_i = |X_i|$ and c_i for each phase $i = 1, \dots, k$.

Here the super index j indicates the length of the vectorial image u as before, and for scalar case, we set $n = 1$. c_i is the average of each phase i , and n_i is the number of pixels in phase i , i.e. area $|X_i| = n_i$.

This energy difference (5) can be easily evaluated by computing each terms. The term ΔF is the change in the

total length and the scale term, which can be simply expressed as $\Delta F = Tl(S_j - S_l) + S_j \Delta T$. Here $S_j = \sum \frac{P(X_j)}{n_j}$ when (x, y) in phase j . Tl is the total length when (x, y) in phase l , which is computed by adding all the difference of the characteristic function X_l such as $T_l = \sum_{i=1}^k P(X_i) = \sum_{i=1}^k \sum_{(x,y) \in \Omega} (|X_i(x+1,y) - X_i(x,y)| + |X_i(x,y+1) - X_i(x,y)|)$

The difference of total length, ΔT , can be also easily computed by considering the addition of change in the length in phase j and phase l , $\Delta T = \Delta P(X_j) + \Delta P(X_l)$. The change in length of each phase is a function of the dimension of the space. If a pixel joins or leaves a phase, the change in length can be computed using its neighbors, and can be simply computed by $\Delta T = -2(\sum_{(x,y) \in \Omega} X_j^{(x,y)} - \sum_{(x,y) \in \Omega} X_l^{(x,y)})$. (See [15])

for related details.) By combining all these terms, the change in energy (5) can be computed. Now, the minimum can be found by computing the difference in energy. If $\Delta E_{lj} > 0$, the pixel will not change to phase j since that will increase the energy. If this value ΔE_{lj} is negative, it is better to move (x, y) to phase j .

The scalar versions of the functional (4) and the algorithm (5) demonstrate stability for general images when no prior information is given, and related properties and stability of these approach can be found in [15] or [10].

4. Numerical Experiments

One of the best applications of this approach is in identifying objects underwater. Figure 1 shows an example of 3D Flash Lidar image of a underwater target. It is showing 8 planes

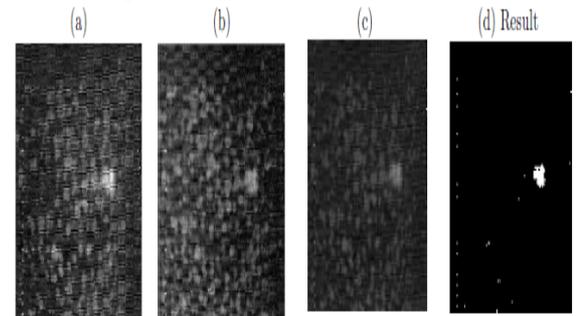


Fig. 3: Images (a) and (b) are a_2 and a_3 : the frequency response of FFT applied to z -direction of 3D Flash Lidar image in Figure 1. They capture both the object and glints of the water. (c) The scalar image \bar{u} . (d) The result using a scalar multiphase segmentation on \bar{u} , with $\lambda = 0.0001$. Image (d) clearly identifies the object separately from the glares of the water.

Here the object can be identified as one phase. We apply the proposed method. The top row of Figure 3 (a)-(b) shows the frequency responses: a_2 and a_3 . The primary characteristics of larger targets are evident in these lower frequency images, and the frequency images (a)-(b) include the object as well as all the glints from the water surface. Image (c) is the scalar image \bar{u} of these lower frequency images. To clearly identify the

object, we apply multiphase image segmentation to image (c) and Figure 3 (d) shows the result. This result (d) is one phase out of the 8 phases automatically segmented using $\lambda = 0.0001$ in (4). This method clearly separates the object from the glint of the water effectively without any prior knowledge of the sense.

As a comparison, Figure 4 shows results using a two phase segmentation [1] for the airborne example in Figure 1. This example shows that regardless of using scalar or vectorial model of CV model, two phase segmentation is not suitable for automatically finding multiple different objects and classifying the different objects. The glints as well as the object are in one phase of the segmentation due to the limitation of finding only two phases.

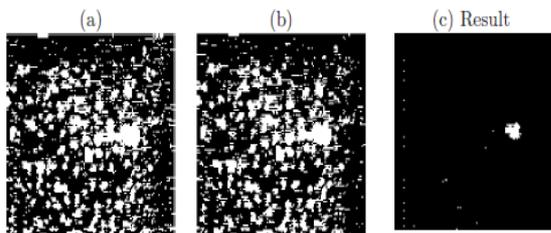


Fig. 4: (a) The result using two phase segmentation [1] on a scalar image \bar{u} . (b) The result using two phase vectorial segmentation on a vectorial image u . These experiments show that two phase segmentation is not suitable for automatically finding multiple objects. The glint and the object are in the same phase, due to the limitation of finding only two phases. (c) The result using the proposed approach.

The top row of Figure 5 is another example of a 3D Flash Lidar image, showing an object on the ground. Figure 5 second row shows segmentation results, using scalar and vectorial models. Both results show reasonable results; well-identifying the location of the object. Image (i) is one phase out of nine and image (j) is one out of seven automatically segmented phases. Using the vectorial model gives a cleaner result compared to the scalar case, since the vectorial model can be more stable against noise introduced from one of the images a_2, a_3 or a_4 .

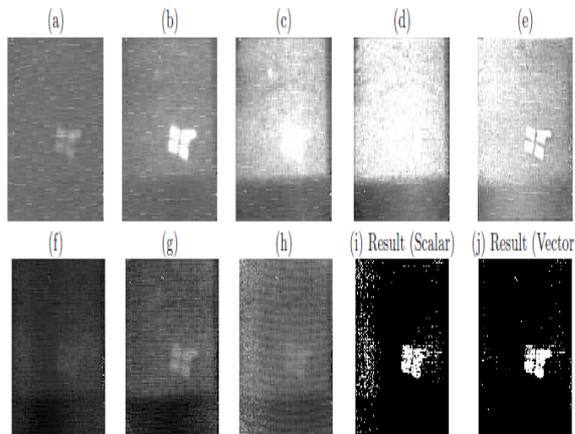


Fig. 5: The top row are few images of a 3D Flash Lidar image showing a single object on the ground. Image (f)-

(h) are a_2, a_3 and a_4 , respectively. Image (i) is a result using a scalar case of \bar{u} , using $\lambda = 0.0001$ and Image (j) is a result using a vectorial case of $u = (a_2, a_3, a_4)$, using vectorial model with $\lambda = 0.0005$. Using the vectorial model gives a cleaner result compared to the scalar case, since the vectorial model can be more stable against noise introduced from one of the images a_2, a_3 and a_4 .

5. Concluding Remark

Three-dimensional Flash Lidar cameras are one of the latest advances in imaging technology [18]. It produces a series of range delimited imagery using a single laser flash. We propose an algorithm to find objects in the 3D Flash Lidar image using a variational model of unsupervised multiphase image segmentation. The imagery can distinguish between objects at different distances, such as the sea-surface and underwater objects. The proposed method particularly works well when objects are underwater with presence of high frequency noise, as in Figure 3. A step of FFT helps to reduce the size of the image, and makes the segmentation process stable by reducing the noise. This is a first work on underwater object identification of 3D Flash Lidar images, and there are many interesting problem: identifying tall objects where parts of such image are spread across many different planes is a challenging problem, especially since only discrete height information is available. Three-dimensional Flash Lidar images have a rich information set and various interesting properties can be explored

References

- [1] Chan T. and Vese L. (2001) *IEEE transactions on image processing*, 10, 2.
- [2] Jason T. Chung and Luminita A. Vese (2003) *UCLA CAM Report* 03-53.
- [3] Crosby F. and Zhou H. (2010) *Journal of Photogrammetric Engineering and Remote Sensing*, 76, 12.
- [4] Elmqvist M., Jungert E., Lantz F., Persson A. and U. Söderman (2001) *International Archives of Photogrammetry and Remote Sensing*, vol. 34, 3/W4, pp. 219-226, 2001.
- [5] Geman, S. and Geman, D. (1984) *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-6:721-741..
- [6] Gibou F. and Fedkiw R. (2005) *Proceedings of the 4th Annual Hawaii International Conference on Statistics and Mathematics*, pp. 281-291.
- [7] Brenner C., Haala N. (1998) *International Archives of Photogrammetry and Remote Sensing*, vol. 32, part 4, pp. 77-84.
- [8] Jung Y. M., Kang S. H. and Shen J. (2007) *SIAM Journal on Applied Mathematics*, vol. 67, no. 5, pp. 1213-1232.
- [9] Lie J., Lysaker M. and Tai X.-C. (2006) *AMS Mathematics of Computation*, vol. 75, pp. 11551174.

- [10] Kang S. H., Sandberg B. and Yip A. (2010) *UCLA CAM Report* 10-53.
- [11] Maas H.-G. and Vosselman G. (1999) *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 54, pp. 153-163.
- [12] Mumford D. and Shah J. (1989) *Comm. Pure Appl. Math.*, vol. 42, pp. 577-685.
- [13] Pan Y., Birdwell D. and Djouadi S. (2006) , *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 1171-121.
- [14] Roggero M. (2001) *International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4 Annapolis, MD*, 22-24.
- [15] Sandberg B., Kang S. H. and Chan T. (2010) *IEEE Trans. in Image Processing, Volume: 19 Issue: 1*, 119 - 130.