SPAWAR
Systems Center
PACIFIC

Technical Report 2033
September 2013

Range Image Flow using High-Order Polynomial Expansion

A NISE funded Basic Research Project

Brian Okorn
Josh Harguess

SSC Pacific
San Diego, CA 92152-5001

SSC Pacific
San Diego, California 92152-5001

J. J. Beel, CAPT, USN
Commanding Officer

C. A. Keeney
Executive Director

ADMINISTRATIVE INFORMATION

This report was prepared by the Research and Applied Sciences Department (Code 71), SPAWAR Systems Center Pacific (SSC Pacific), San Diego, CA. This project is funded by the Naval Innovative Science and Engineering (NISE) Program at SSC Pacific as a Basic Research project.

Released by
Tracy North Pastore, Head
Unmanned Systems Branch

Under authority of
M. J. Machniak, Head
Advanced Systems & Applied Sciences Division

ACKNOWLEDGMENTS

We would like to give a special thanks to Dr. Steve Hobbs for his help with the high-order tensor calculations.

SB

EXECUTIVE SUMMARY

OBJECTIVE

This paper presents a novel algorithm for estimating the motion in a series of range images. First, each range image is approximated by applying a high-order polynomial expansion to local neighborhoods within the range image. Then, these approximations are used to derive the translation or displacement estimation from frame to frame within the series of range images (also known as range image flow). An iterative method for computing the translation is presented.

RESULTS

We evaluate the algorithm on several synthetic and real-world range image sequences with promising results.

RECOMMENDATIONS

Results in this paper are generated from a single iteration of the algorithm in space and time. Therefore, our next step is to improve the implementation of the algorithm so that using multiple spatial scales and past information improve the final flow estimation, as we would expect. Also, we will port the MATLAB® implementation to C++ to improve speed and meet the final integration goal of the algorithm.

CONTENTS

Figures

Tables

1. INTRODUCTION

Estimating motion in a video or series of images is an extremely important and difficult task in computer vision and has numerous applications, such as autonomous vehicle navigation [11]. Not much attention has been given to the estimation of motion between range images, however, estimating motion in video or image sequences, most commonly referred to as optical flow, has a long history of research. The two most common and prominent approaches to optical flow are known as local (or sparse) and global (or dense) methods. Local methods, such as the Lucas–Kanade method [14], estimate the motion of regions of interest between images using image registration and warping techniques. In contrast, global methods, such as Horn and Schunck's method [13], compute a dense motion field by estimating the motion of each pixel between images. In this work, we are concerned mostly with the latter method, particularly the work by Farnebäck [8], which approximates the image using a polynomial expansion of local patches and then uses the polynomial expansion to estimate the global displacements between images for each pixel.

Estimating optical flow for range images, also known as range flow, is the main topic of this paper. Very little current research exists on this topic, one of the few examples being that of Spies, Jähne and Barron [18], however, it is an extremely important problem in a growing field. The problem of range flow is different from optical flow for electro-optical images in the sense that every pixel value is a measure of distance instead of color or brightness. This difference makes it very difficult to apply existing and traditional two-dimensional (2D) optical flow methods to range flow. For instance, the brightness constancy constraint used by many optical flow methods is not valid for range images. Therefore, our approach in this work is to extend a well-known global optical flow method of motion estimation based on polynomial expansion [8] to range images. We extend the method by using a high-order polynomial expansion to include terms in the z direction (range distance to the sensor). We then formulate an iterative method to solve for displacement in the x, y, and z directions between range imagery. In addition, we perform the calculation at multiple scales for robustness and include displacement estimates from previous frames to improve the overall motion estimation. Promising results are presented on several sets of synthetic and real-world range images.

2. RELATED WORK

As previously mentioned, there is a vast amount of research in optical flow between color images, usually categorized into local and global methods, such as Lucas and Kanade [14] and Horn and Shunck [13], respectively. Many assumptions and constraints have been introduced in both approaches to deal with noise and smoothness of the solutions, such as the brightness constancy assumption, gradient constancy assumption and spatio-temporal smoothness constraints. This has led to a breeding ground of methods, such as the work of Bruhn, Weickert, and Schörr [3], which attempt to combine the local and global methods to address the drawbacks and assumptions of each individual method. The most popular and successful methods are covered in more detail in two benchmarking papers on the subject by McCane, Novins, Crannitch, and Galvin [15] and Baker et al. [1]. The two papers describe common databases, procedures, and results on comparing more than 20 optical flow methods, with the Baker et al. paper being the most recent and complete. Of particular interest to our research is the method introduced by Farnebäck in several papers that introduces the estimation of motion using the polynomial expansion [5–10, 16, 17, 19]. In Farnebäck's work, the local neighborhoods of each image are approximated by a polynomial expansion and an analytical solution for the displacement between images is derived. From this derivation, a robust algorithm is designed to compute the displacement, and thus, motion field, between two or more images in a sequence. The method has proven to be very accurate and robust for 2D images and has been included as a default algorithm in the OpenCV library [2].

The research of estimating the motion between range images, or range flow, is much more sparse. The term "range image flow" first appears in the work of Gonzalez [12], where he formulates a physics-based approach to estimate the motion of the range sensor relative to its environment. Our method uses the same basic physical model of the range sensor as that used by Gonzalez. One of the most popular and earliest papers on this topic, by Spies, Jähne, and Barron [18], notes the unique challenges of this problem and proposes a basic motion constraint equation on deformable surfaces. The constraint solutions are obtained in a total least squares framework and compute a dense range flow field from sparse solutions. While the results are promising for the range images presented in their paper, the method is not directly transferable to other domains, such as dense range flow in a moving scene due to the large displacements present. Therefore, the focus of our work is to extend the polynomial expansion method to range imagery to compute dense range flow on sequences of real-world range images.

3. RANGE IMAGE POLYNOMIAL EXPANSION

In our formulation, range image flow uses a polynomial expansion based approximation of the range image. This approximation is done using a set of quadratic basis functions, applied to the range data. The basis equation set is $\{1, x, y, x^2, y^2, xy\}$, which describes the variation of z, range from the sensor, as you vary x and y, azimuth and elevation with respect to the sensor. In addition to the basis functions, we incorporate a notion of the accuracy or importance of this data though a certainty matrix, as well as a proximity-based weight over the neighborhood in the form of an applicability matrix, as is done in Farnebäck's Ph.D. dissertation [8]. For the certainty matrix, a value of 1 was given to all pixels populated with valid data and 0 for any pixel with error or no data. A Gaussian kernel was used as the applicability measure. The weights of the basis were calculated using Equation (1) as described by Farnebäck,

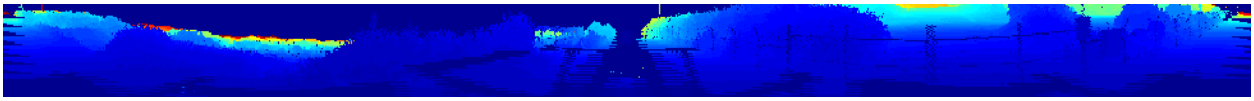
$$\mathbf{r} = (\mathbf{B}^T\mathbf{W_a}\mathbf{W_c}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W_a}\mathbf{W_c}\mathbf{f} \tag{1}$$

where $\mathbf{B}$ represents the basis functions, $\mathbf{W_a}$ and $\mathbf{W_c}$ are applicability and certainty, taken column-wise and diagonalized, and $\mathbf{f}$ is the range image data, taken column-wise. The values of these weights for a Velodyne® and Odetic lidar range image [4] can be seen in Figures 1 and 2, respectively. The approximated signal can then be reconstructed as $\mathbf{Br}$, reshaped column-wise into its proper neighborhood. This defines the value of each pixel as a function of its local coordinates, $\mathbf{x}$, described in Equation (2) as a quadratic matrix multiplication and, in Equation (3) in index notation,

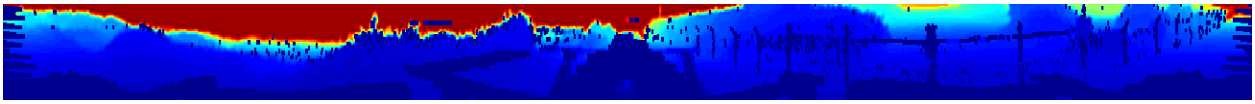
$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \tag{2}$$

$$f(\mathbf{x}) = a_{ij}x_ix_j + b_ix_i + c, \tag{3}$$

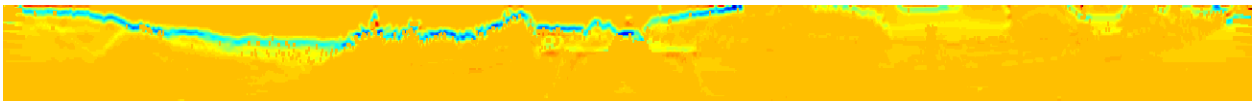
where $\mathbf{A} = \begin{bmatrix} r_{x^2} & \frac{r_{xy}}{2} \\ \frac{r_{xy}}{2} & r_{y^2} \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} r_x \\ r_y \end{bmatrix}$ and $c = r_1$. While the matrix form is more compact and easier to manipulate, the index notation becomes necessary as the model moves into higher-order approximations using larger degree tensors.



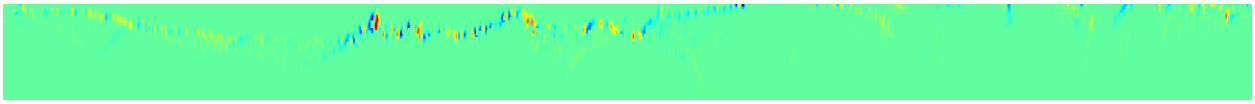
(a) Original Range Image



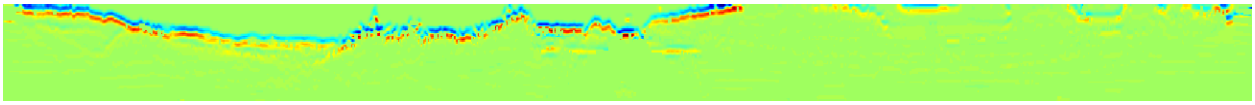
(b) 1 Coefficient Image



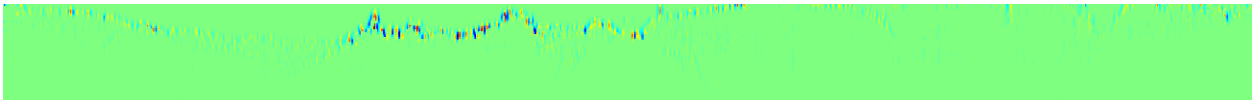
(c) y Coefficient Image



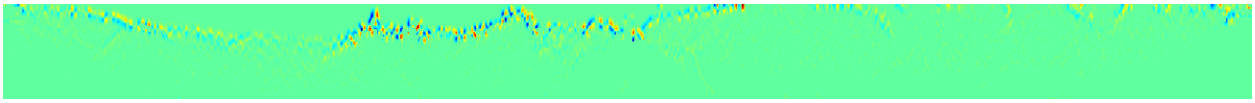
(d) x Coefficient Image



(e) y^2 Coefficient Image

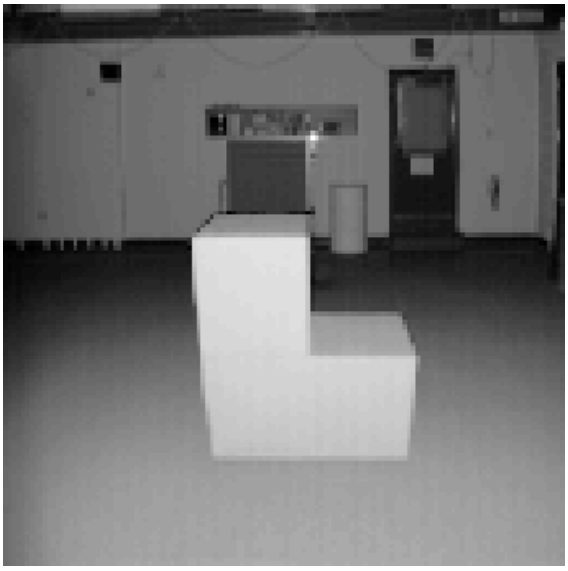


(f) x^2 Coefficient Image

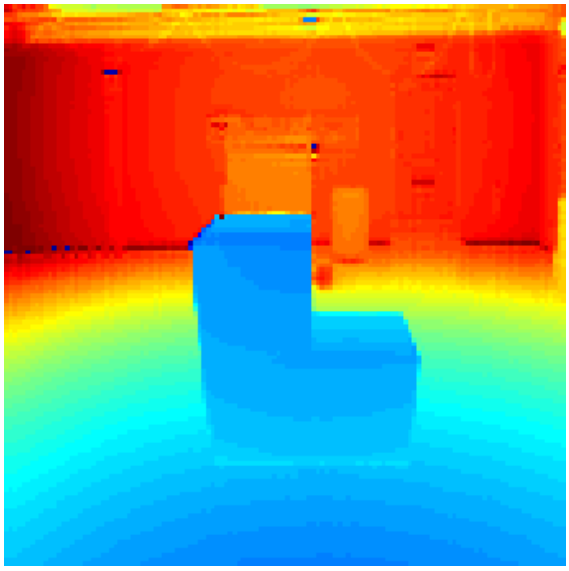


(g) xy Coefficient Image

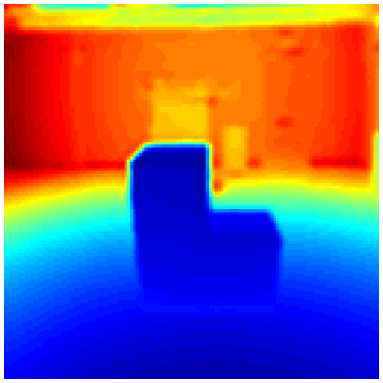
Figure 1. Velodyne® polynomial expansion.



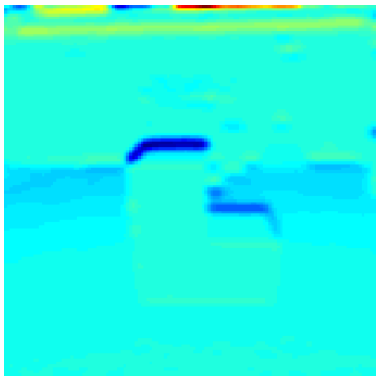
(a) Reflective Image



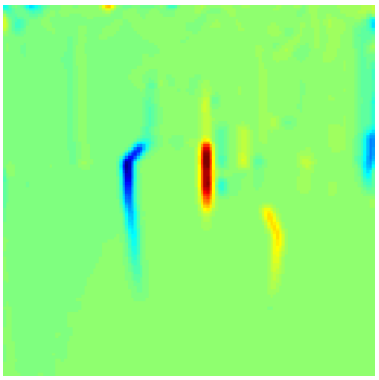
(b) Original Range Image



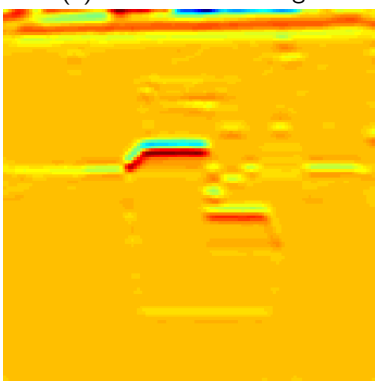
(c) 1 Coefficient Image



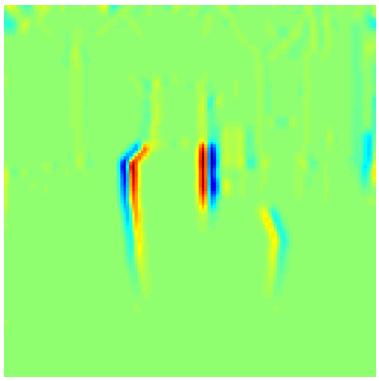
(d) y Coefficient Image



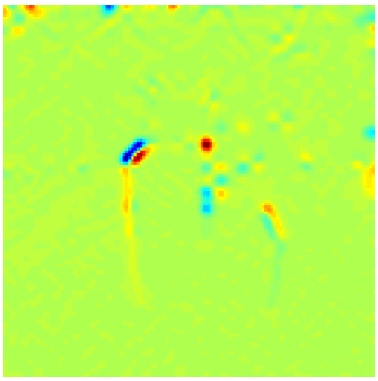
(e) x Coefficient Image



(f) y^2 Coefficient Image



(g) x^2 Coefficient Image



(h) xy Coefficient Image

Figure 2. Odetic polynomial expansion.

4. DISPLACEMENT CALCULATION

The polynomial expansion in 2D images allows displacement to be calculated analytically by looking at the effects of a displacement on the polynomial expansion coefficients. The effects of this displacement on the quadratic polynomial are derived in Equation (4).

$$\begin{aligned} f_1(\mathbf{x}) &= \mathbf{x}^T \mathbf{A_1} \mathbf{x} + \mathbf{b_1}^T \mathbf{x} + c_1 \\ f_2(\mathbf{x}) &= f_1(\mathbf{x} - \mathbf{d}) \\ &= (\mathbf{x} - \mathbf{d})^T \mathbf{A_1} (\mathbf{x} - \mathbf{d}) + \mathbf{b_1}^T (\mathbf{x} - \mathbf{d}) + c_1 \\ &= \mathbf{x}^T \mathbf{A_1} \mathbf{x} + (\mathbf{b_1} - 2\mathbf{A_1}\mathbf{d})^T \mathbf{x} + \mathbf{d}^T \mathbf{A_1} \mathbf{d} + \mathbf{b_1}^T \mathbf{d} + c_1 \\ &= \mathbf{x}^T \mathbf{A_2} \mathbf{x} + \mathbf{b_2}^T \mathbf{x} + c_2, \end{aligned} \tag{4}$$

leaving you with a new quadratic polynomial with different coefficients,

$$\mathbf{A_2} = \mathbf{A_1} \tag{5}$$

$$\mathbf{b_2} = \mathbf{b_1} - 2\mathbf{A_1}\mathbf{d} \tag{6}$$

$$c_2 = \mathbf{d}^T \mathbf{A_1} \mathbf{d} + \mathbf{b_1}^T \mathbf{d} + c_1. \tag{7}$$

With these new coefficients, $\mathbf{d}$ can be computed from Equation (6):

$$\mathbf{d} = -\frac{1}{2} \mathbf{A_1}^{-1} (\mathbf{b_2} - \mathbf{b_1}). \tag{8}$$

This method of displacement calculation was developed and tested by Farnebäck and is used as the starting point of the three-dimensional (3D) displacement calculation.

A third dimension cannot simply be added to $\mathbf{d}$ because with the current model, it has no meaning as an input term. The function space is not defined for any value of z under this model so the model must be modified to explain behavior while displacing this third dimension. One possible solution could be that z displacement subtracts from the function value,

$$f(\mathbf{x} - \begin{bmatrix} 0 \\ 0 \\ d_z \end{bmatrix}) = f(\mathbf{x}) - d_z. \tag{9}$$

This model requires $\mathbf{A}$ and $\mathbf{b}$ be changed to

$$\mathbf{A} = \begin{bmatrix} r_{x^2} & \frac{r_{xy}}{2} & \gamma_1 \\ \frac{r_{xy}}{2} & r_{y^2} & \gamma_2 \\ -\gamma_1 & -\gamma_2 & 0 \end{bmatrix} \tag{10}$$

$$\mathbf{b} = \begin{bmatrix} r_x \\ r_y \\ 1 \end{bmatrix}. \tag{11}$$

This model depends on the values of γ_1 and γ_2, which if viewed as the coefficients of the xz and yz terms, should be 0, making $\mathbf{A}$ always singular. Otherwise $\mathbf{A}^{-1}$ is heavily affected by a term with no meaning to the polynomial equation.

To properly capture this higher-dimensional behavior, a higher-order equation is used to approximate this space. A linear spreading of the x and y data, Equation (12), as well as a constant increase, Equation (13), was used to approximate the behavior of the data as you move along the z dimension. This leads to a quartic polynomial, Equation (14), with the higher-order terms being very sparse tensors:

$$\begin{aligned} x' &= (\alpha_x z + 1)x \\ y' &= (\alpha_y z + 1)y \\ \mathbf{x_2'} &= \begin{bmatrix} x' \\ y' \end{bmatrix} \end{aligned} \tag{12}$$

$$f(\mathbf{x}) = f_{quad}(\mathbf{x_2'}) + z; \tag{13}$$

$$\begin{aligned} f(\mathbf{x}) = {} & \alpha_x^2 r_{x^2} x^2 z^2 + \alpha_x \alpha_y r_{xy} xyz^2 + \alpha_y^2 r_{y^2} y^2 z^2 \\ & + 2\alpha_x r_{x^2} x^2 z + (\alpha_x + \alpha_y) r_{xy} xyz + 2\alpha_y r_{y^2} y^2 z \\ & + r_{x^2} x^2 + r_{xy} xy + \alpha_x r_x xz + r_{y^2} y^2 + \alpha_y r_y yz \\ & + r_x x + r_y y + z + c. \end{aligned} \tag{14}$$

This can be combined into a tensor form similar to Equation (3):

$$f(\mathbf{x}) = g_{ijkl} x_i x_j x_k x_l + h_{ijk} x_i x_j x_k + a_{ij} x_i x_j + b_i x_i + c, \tag{15}$$

where the high-order tensors, $\mathbf{G}$ and $\mathbf{H}$, are sparse and the quadratic tensors are dense. For $\mathbf{G}$ and $\mathbf{H}$, the nonzero terms are

$$\begin{aligned} g_{0022} = g_{0202} = g_{2002} = ... &= \frac{\alpha_x^2 r_{x^2}}{6} \\ g_{0122} = g_{0212} = g_{2012} = ... &= \frac{\alpha_x \alpha_y r_{xy}}{12} \\ g_{1122} = g_{1212} = g_{2112} = ... &= \frac{\alpha_y^2 r_{y^2}}{6} \end{aligned} \tag{16}$$

$$\begin{aligned} h_{002} = h_{020} = ... &= \frac{2\alpha_x r_{x^2}}{3} \\ h_{012} = h_{120} = ... &= \frac{(\alpha_x + \alpha_y) r_{xy}}{6} \\ h_{112} = h_{121} = ... &= \frac{2\alpha_y r_{y^2}}{3}, \end{aligned} \tag{17}$$

while $\mathbf{A}$ and $\mathbf{b}$ are

$$\mathbf{A} = \begin{bmatrix} r_{x^2} & \frac{r_{xy}}{2} & \frac{\alpha_x r_x}{2} \\ \frac{r_{xy}}{2} & r_{y^2} & \frac{\alpha_y r_y}{2} \\ \frac{\alpha_x r_x}{2} & \frac{\alpha_y r_y}{2} & 0 \end{bmatrix} \tag{18}$$

$$\mathbf{b} = \begin{bmatrix} r_x \\ r_y \\ 1 \end{bmatrix}. \tag{19}$$

We now explore the effects of displacement, as Farnebäck did, in Equation (4):

$$\begin{aligned} f(\mathbf{x}) &= g_{ijkl}x_i x_j x_k x_l + h_{ijk}x_i x_j x_k + a_{ij}x_i x_j + b_i x_i + c \\ \tilde{f}(\mathbf{x}) &= f(\mathbf{x} - \mathbf{d}) \\ &= g_{ijkl}(x_i - d_i)(x_j - d_j)(x_k - d_k)(x_l - d_l) \\ &\quad + h_{ijk}(x_i - d_i)(x_j - d_j)(x_k - d_k) \\ &\quad + a_{ij}(x_i - d_i)(x_j - d_j) \\ &\quad + b_i(x_i - d_i) + c \\ &= g_{ijkl}x_i x_j x_k x_l \\ &\quad - (4g_{ijkl}d_l - h_{ijk})x_i x_j x_k \\ &\quad + (6g_{ijkl} - 3h_{ijk}d_k + a_{ij})x_i x_j \\ &\quad - (4g_{ijkl}d_j d_k d_l - 3h_{ijk}d_j d_k + 2a_{ij}d_j - b_i)x_i \\ &\quad + (g_{ijkl}d_i d_j d_k d_l - h_{ijk}d_i d_j d_k + a_{ij}d_i d_j - b_i d_i + c \\ &= \tilde{g}_{ijkl}x_i x_j x_k x_l + \tilde{h}_{ijk}x_i x_j x_k + \tilde{a}_{ij}x_i x_j + \tilde{b}_i x_i + \tilde{c}, \end{aligned} \tag{20}$$

leaving us with a new quartic polynomial with the following coefficients:

$$\tilde{g}_{ijkl} = g_{ijkl} \tag{21}$$

$$\tilde{h}_{ijk} = h_{ijk} - 4g_{ijkl}d_l \tag{22}$$

$$\tilde{a}_{ij} = a_i j - 3h_{ijk}d_k + 6g_{ijkl}d_k d_l \tag{23}$$

$$\tilde{b}_i = b_i - 2a_i j d_j + 3h_{ijk}d_j d_k - 4g_{ijkl}d_j d_k d_l \tag{24}$$

$$\tilde{c} = c - b_i d_i + a_i j d_i d_j - h_{ijk}d_i d_j d_k + g_{ijkl}d_i d_j d_k d_l. \tag{25}$$

Now we have a linear equation with $\mathbf{H}$ that could be used as an analytical solution to $\mathbf{d}$ but with our higher-order tensors being mostly sparse and mainly composed of the model coefficients as opposed to the expansion coefficients, we look to our lower order dense tensors to solve for $\mathbf{d}$

through numerical optimization. Using the symmetries and sparsities of the matrices, we are left with the following coefficients in our dense tensors. In this case, $\tilde{b}_2 = 1$ because it is created purely by the model of z motion and has information about the next image's polynomial expansion:

$$\begin{aligned}
\tilde{b}_0 &= b_0 - 2(a_{00}d_x + a_{01}d_y + a_{02}d_z) \\
&\quad + 6(h_{002}d_x d_z + h_{012}d_y d_z) \\
&\quad - 12(g_{0022}d_x d_z^2 + g_{0122}d_y d_z^2) \\
\tilde{b}_1 &= b_1 - 2(a_{01}d_x + a_{11}d_y + a_{12}d_z) \\
&\quad + 6(h_{012}d_x d_z + h_{112}d_y d_z)) \\
&\quad - 12(g_{0122}d_x d_z^2 + g_{1122}d_y d_z^2) \\
\tilde{b}_2 &= b_2 - 2(a_{02}d_x + a_{12}d_y) \\
&\quad + 3(h_{002}d_x^2 + h_{112}d_y^2 + 2h_{012}d_x d_y) \\
&\quad - 12(g_{0022}d_x^2 d_z + g_{1122}d_y^2 d_z + 2g_{0122}d_x d_y d_z) \\
&= 1
\end{aligned} \tag{26}$$

$$\begin{aligned}
\tilde{c} &= c - b_0 d_x + b_1 d_y + b_2 d_z \\
&\quad + (a_{00}d_x^2 + a_{11}d_y^2 + 2a_{01}d_x d_y + 2a_{02}d_x d_z + 2a_{12}d_y d_z) \\
&\quad - 3(h_{002}d_x^2 d_z + h_{112}d_y^2 d_z + 2h_{012}d_x d_y d_z) \\
&\quad + 6(g_{0022}d_x^2 d_z^2 + g_{1122}d_y^2 d_z^2 + 2g_{0122}d_x d_y d_z^2).
\end{aligned} \tag{27}$$

To calculate $\mathbf{d}$, we optimize the difference between the observed polynomial coefficients of the next image and the coefficients derived through displacing the coefficients of the first image using Equations (26) and (27). This optimization is done using the non-linear least squares version of Newton's method, the Gauss—Newton algorithm, on Equations (28) and (29). This optimization technique requires the Jacobian matrices of each coefficient with respect to $\mathbf{d}$, calculated using the partial derivative in Equations (30) and (31). Again, $\tilde{b}_2$ contains no polynomial expansion information, so its components of the Jacobian are 0.

$$\min_{\mathbf{d}} \sum (\tilde{b}_i^{(1)}(\mathbf{d}) - b_i^{(2)})^2 \tag{28}$$

$$\min_{\mathbf{d}} (\tilde{c}^{(1)}(\mathbf{d}) - c^{(2)})^2 \tag{29}$$

$$
\begin{aligned}
\frac{\partial \tilde{b}_0}{\partial d_x} &= -2a_{00} + 6h_{002}d_z - 3g_{0022}d_z^2 \\
\frac{\partial \tilde{b}_1}{\partial d_x} &= -2a_{01} + 6h_{012}d_z - 3g_{0122}d_z^2 \\
\frac{\partial \tilde{b}_2}{\partial d_x} &= -2a_{02} + 6(h_{002}d_x + h_{012}d_y) - 6(g_{0022}d_xd_z + g_{0122}d_yd_z) = 0 \\
\frac{\partial \tilde{b}_0}{\partial d_y} &= -2a_{01} + 6h_{012}d_z - 3g_{0122}d_z^2 \\
\frac{\partial \tilde{b}_1}{\partial d_y} &= -2a_{11} + 6h_{112}d_z - 3g_{1122}d_z^2 \\
\frac{\partial \tilde{b}_2}{\partial d_y} &= -2a_{12} + 6(h_{012}d_x + h_{112}d_y) - 6(g_{0122}d_xd_z + g_{1122}d_yd_z) = 0 \\
\frac{\partial \tilde{b}_0}{\partial d_z} &= -2a_{02} + 6(h_{002}d_x + h_{012}d_y) - 6(g_{0022}d_xd_z + g_{0122}d_yd_z) \\
\frac{\partial \tilde{b}_1}{\partial d_z} &= -2a_{12} + 6(h_{012}d_x + h_{112}d_y) - 6(g_{0122}d_xd_z + g_{1122}d_yd_z) \\
\frac{\partial \tilde{b}_2}{\partial d_z} &= -3(2g_{0122}d_xd_y + g_{0022}d_x^2 + g_{1122}d_y^2) = 0
\end{aligned}
\tag{30}
$$

$$
\begin{aligned}
\frac{\partial \tilde{c}}{\partial d_x} &= -b_0 + 2(a_{00}d_x + a_{01}d_y + a_{02}d_z) \\
&\quad - 6(h_{002}d_xd_z + h_{012}d_yd_z) + 12(g_{0022}d_xd_z^2 + g_{0122}d_yd_z^2) \\
\frac{\partial \tilde{c}}{\partial d_y} &= -b_1 + 2(a_{01}d_x + a_{11}d_y + a + 12d_z) \\
&\quad - 6(h_{012}d_xd_z + h_{112}d_yd_z) + 12(g_{0122}d_xd_z^2 + g_{1122}d_yd_z^2) \\
\frac{\partial \tilde{c}}{\partial d_z} &= -1 + 2(a_{02}d_x + a_{12}d_y) - 3(h_{002}d_x^2 + h_{112}d_y^2 + 2h_{012}d_xd_y) \\
&\quad + 12(g_{0022}d_x^2d_z + g_{1122}d_y^2d_z + 2g_{0122}d_xd_yd_z)
\end{aligned}
\tag{31}
$$

This leaves two remaining routes to solving for $\mathbf{d}$, through the changes in $\mathbf{b}$ or the changes in c, shown in Equations (32) and (33), respectively, each with its own advantage. The solution based on $\mathbf{b}$ tends to produce more accurate results for d_x and d_y because it captures the motion of the quadratic components seen mostly in edges. This solution does not tend to be as accurate for the d_z component because d_z is detected through the spreading of the quadratic x and y components. This effect can be small at long distances, where as d_z has a large effect on c. However, the effects on c do not track the quadratic components of d_x and d_y as directly as $\mathbf{b}$. To maximize the accuracy of $\mathbf{d}$, a weighted average of $\mathbf{d_b}$ and $\mathbf{d_c}$, shown in Equation (34), with β_{xy} relatively high and β_z low, where as γ_{xy} is relatively low and γ_z high.

$$
\Delta\mathbf{d_b} = (\mathbf{J_b}^T\mathbf{J_b})^{-1}\mathbf{J_b}^T(\tilde{\mathbf{b}}^{(1)}(\mathbf{d}) - \mathbf{b}^{(2)}) \tag{32}
$$

$$
\Delta\mathbf{d_c} = (\mathbf{J_c}^T\mathbf{J_c})^{-1}\mathbf{J_c}^T(\tilde{c}^{(1)}(\mathbf{d}) - c^{(2)}) \tag{33}
$$

$$\Delta\mathbf{d} = \begin{bmatrix}\beta_{xy}\\ \beta_{xy}\\ \beta_z\end{bmatrix}\Delta\mathbf{d_b} + \begin{bmatrix}\gamma_{xy}\\ \gamma_{xy}\\ \gamma_z\end{bmatrix}\Delta\mathbf{d_c} \tag{34}$$

Similar to Farnebäck, this algorithm estimates the displacement over a neighborhood I around $\mathbf{x}$ as opposed to a single point, minimizing the following equations:

$$\sum_{\Delta x\in I} w(\Delta\mathbf{x})\|(\tilde{b}_i^{(1)}(\mathbf{x}+\Delta\mathbf{x}-\mathbf{d}) - b_i^{(2)}(\mathbf{x}+\Delta\mathbf{x}))\|^2 \tag{35}$$

$$\sum_{\Delta x\in I} w(\Delta\mathbf{x})\|(\tilde{c}^{(1)}(\mathbf{x}+\Delta\mathbf{x}-\mathbf{d}) - c^{(2)}(\mathbf{x}+\Delta\mathbf{x}))\|^2, \tag{36}$$

where $w(\Delta\mathbf{x})$ is the neighborhood weighting function and the minimum steps are

$$\Delta\mathbf{d_b} = \left(\sum_{\Delta\mathbf{x}\in I} w(\Delta\mathbf{x})\mathbf{J_b}^T\mathbf{J_b}\right)^{-1}\sum_{\Delta x\in I} w(\Delta\mathbf{x})\mathbf{J_b}^T(\tilde{\mathbf{b}}^{(1)}(\mathbf{d}) - \mathbf{b}^{(2)}) \tag{37}$$

$$\Delta\mathbf{d_c} = \left(\sum_{\Delta\mathbf{x}\in I} w(\Delta\mathbf{x})\mathbf{J_c}^T\mathbf{J_c}\right)^{-1}\sum_{\Delta x\in I} w(\Delta\mathbf{x})\mathbf{J_c}^T(\tilde{c}^{(1)}(\mathbf{d}) - c^{(2)}). \tag{38}$$

5. RANGE IMAGE FLOW EXPERIMENTS

This algorithm has been tested using data from a Velodyne® HDL-64E. This sensor is a 360° field of view 3D lidar with 64 vertically mounted lasers on a spinning head. The lasers have a maximum range of 50 m and an accuracy of 2 cm. It is capable of spinning at 5 to 15 Hz, generating over 1.333 million points per second. For our tests, the sensor was set to 10 Hz, generating a horizontal resolution of 1800 returns per rotation. The lidar returns are assembled into a 64 x 1800 range image. The tests were done using a Ford F-150 with the Velodyne® mounted to the roof. The vehicle and mounting hardware are visible in the lidar scans, so all pixels within a threshold have been marked with a certainty value of 0, causing these values to have no effect on the polynomial expansion or the flow calculations.

The data set shown in Figures 5, 6, 7, 8, and 9 was collected along Woodward Road at SSC Pacific on a foggy day. The images contain mostly data from the bushes and other vegetation surrounding the road, but do eventually show a parking lot, seen in Figure 9. No ground truth is currently available for the flow fields so only qualitative analysis could be done.

Most of the flow field appears accurate, though some regions still contain peculiar behavior. Certain regions of the flow image act as sources or sinks to the flow fields, such as in Figure 6, where in the middle right portion of the image the flow field moves away from a central point. The boundary of the scan also seems to be less accurate than the central areas. Areas such as that shown in Figure 8 are relatively uniform, making flow calculations more difficult. Despite this, the algorithm performed reasonably well, though it does have regions with sporadic flow behavior. Further work and testing will be done to this algorithm to improve its performance and a ground truth test will be run to quantitatively evaluate the performance of this algorithm.

6. POLYNOMIAL CLASSIFICATION AND PLANAR REGION DETECTION

Using Odetic Laser Range Finder range images [4] of a series of geometric objects, we were able to test the polynomial coefficients associated with 3D features. The coefficients associated with specific 3D features are shown in Table 1. The primary features of interest are planes, edges, and corners. Figure 3 shows the comparison between the real image patches (3a, 3c, and 3e) associated with these features with their respective polynomial expansions (3b, 3d, and 3f, respectively). Planar features, such as the one shown in Figure 3a, have quadratic components close to zero and linear components dependent on the normal of the plane. This planar feature has an expansion, $\{r_1, r_y, r_x, r_{y^2}, r_{x^2}, r_{xy}\}$, of $\{68.0, 0.19, -0.11, 0.00, 0.02, -0.01\}$. The polynomial expansion shows the very low quadratic components indicative of planar regions. Edge features, however, have quadratic features depending on the direction of the edge and linear features dependent on the location of the edge. In Figure 3c, there is a horizontal edge with an expansion of $\{85.7, -11.19, 0.26, 2.95, 0.01, -0.10\}$. In this case, the expansion shows the relatively high x and x^2 components found in horizontal edges. Finally, corner features have nonzero zero values for all of their linear and quadratic components. The expansion of Figure 3c shows an example of this, $\{137.1, -19.28, 24.1, 2.13, 2.64, 7.53\}$, showing all nonzero linear and quadratic terms.

Table 1. Polynomial Expansion of 3D Features.

Polynomial Coefficients					
r_y	r_x	r_{y^2}	r_{x^2}	r_{xy}	Feature Type
any	*any*	≈ 0	≈ 0	≈ 0	Plane
≈ 0	≈ 0	$\not\approx \mathbf{0}$	≈ 0	≈ 0	Vert. Linear
$\not\approx \mathbf{0}$	≈ 0	$\not\approx \mathbf{0}$	≈ 0	≈ 0	Vert. Edge
≈ 0	≈ 0	≈ 0	$\not\approx \mathbf{0}$	≈ 0	Horz. Linear
≈ 0	$\not\approx \mathbf{0}$	≈ 0	$\not\approx \mathbf{0}$	≈ 0	Horz. Edge
≈ 0	≈ 0	*any*	≈ 0	$\not\approx \mathbf{0}$	Diag. Linear
$\not\approx \mathbf{0}$	≈ 0	*any*	≈ 0	$\not\approx \mathbf{0}$	Diag. Edge
≈ 0	≈ 0	≈ 0	*any*	$\not\approx \mathbf{0}$	Diag. Linear
≈ 0	$\not\approx \mathbf{0}$	≈ 0	*any*	$\not\approx \mathbf{0}$	Diag. Edge
≈ 0	≈ 0	$\not\approx \mathbf{0}$	$\not\approx \mathbf{0}$	*any*	Spherical
$\not\approx \mathbf{0}$	$\not\approx \mathbf{0}$	$\not\approx \mathbf{0}$	$\not\approx \mathbf{0}$	*any*	Corner

This leads to the development of a planar segmentation algorithm. Using the polynomial expansion of the range image, the planar regions can be differentiated from the non-planar regions by the quadratic terms of the expansion. From there, a region-growing, flood-fill technique is used to determine which regions belong to the same plane. In this case, the pixels at the frontier of the region are checked for low quadratic components and if they are found to be planar, then their linear components are compared to the frontier's linear components. If the linear components are sufficiently different, the pixel becomes the seed to a new region; otherwise, it is added to the frontier and the region continues to grow. This method was tested on the Odetic range images and an example is shown in Figure 4. The floor is correctly segmented as a single plane and the wall is correctly segmented into two planes, with a break caused by a raised wire covering. The box structure is also correctly segmented.

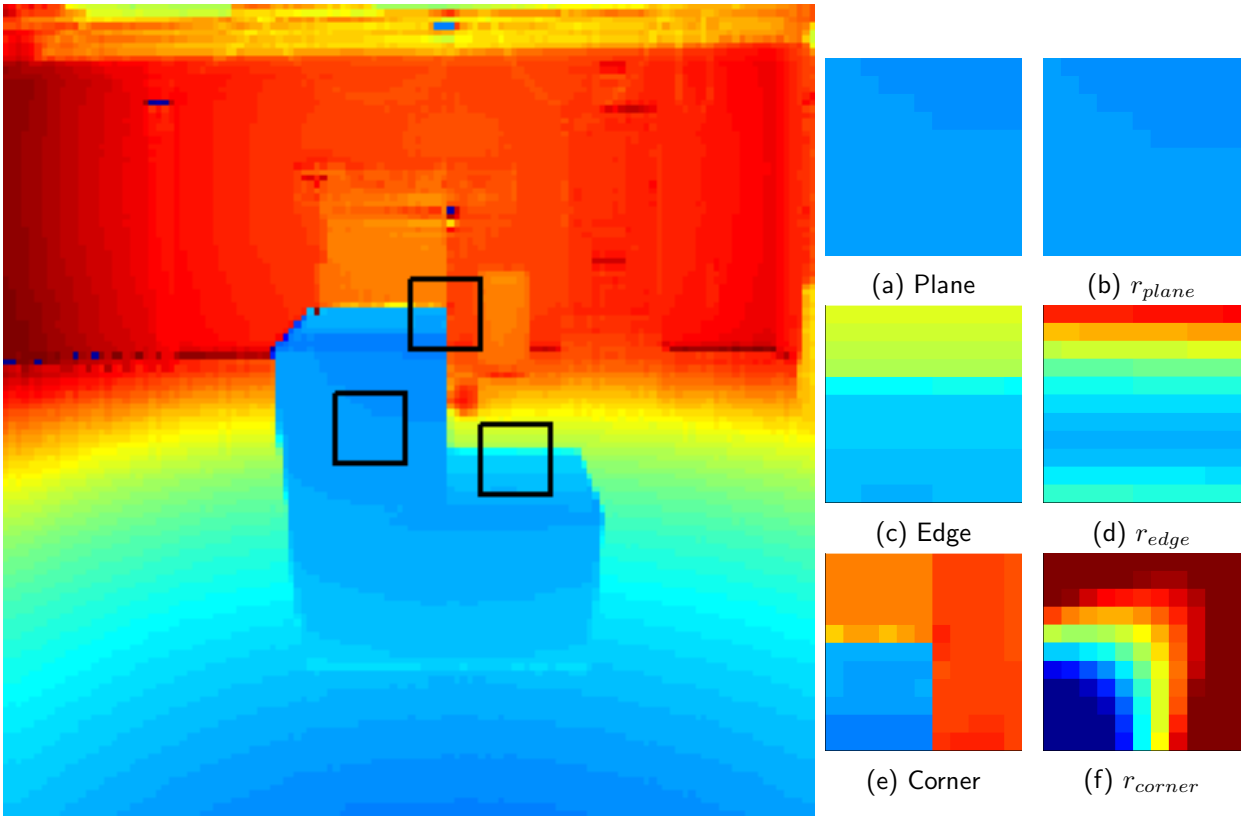


Figure 3. Outlined planar (a), edge (c) and corner (e) features with their respective polynomial expansions (b, d and f, respectively).

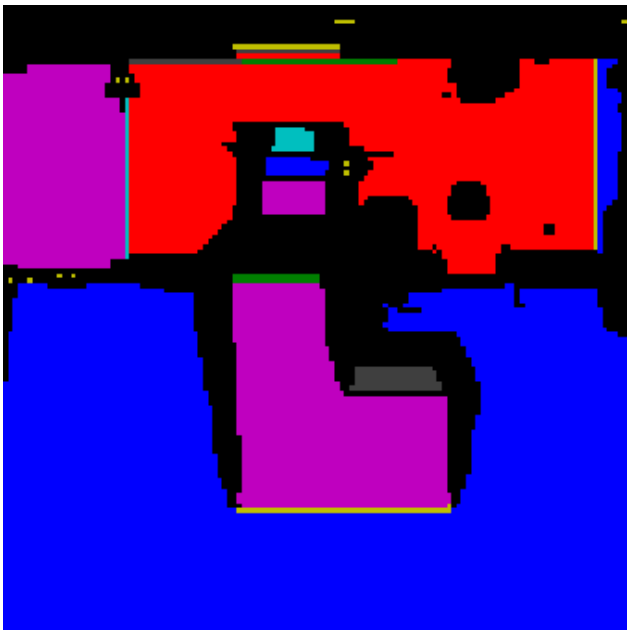


Figure 4. Planar region detection of Odetic range image. Non-connected colors are independent planar regions.

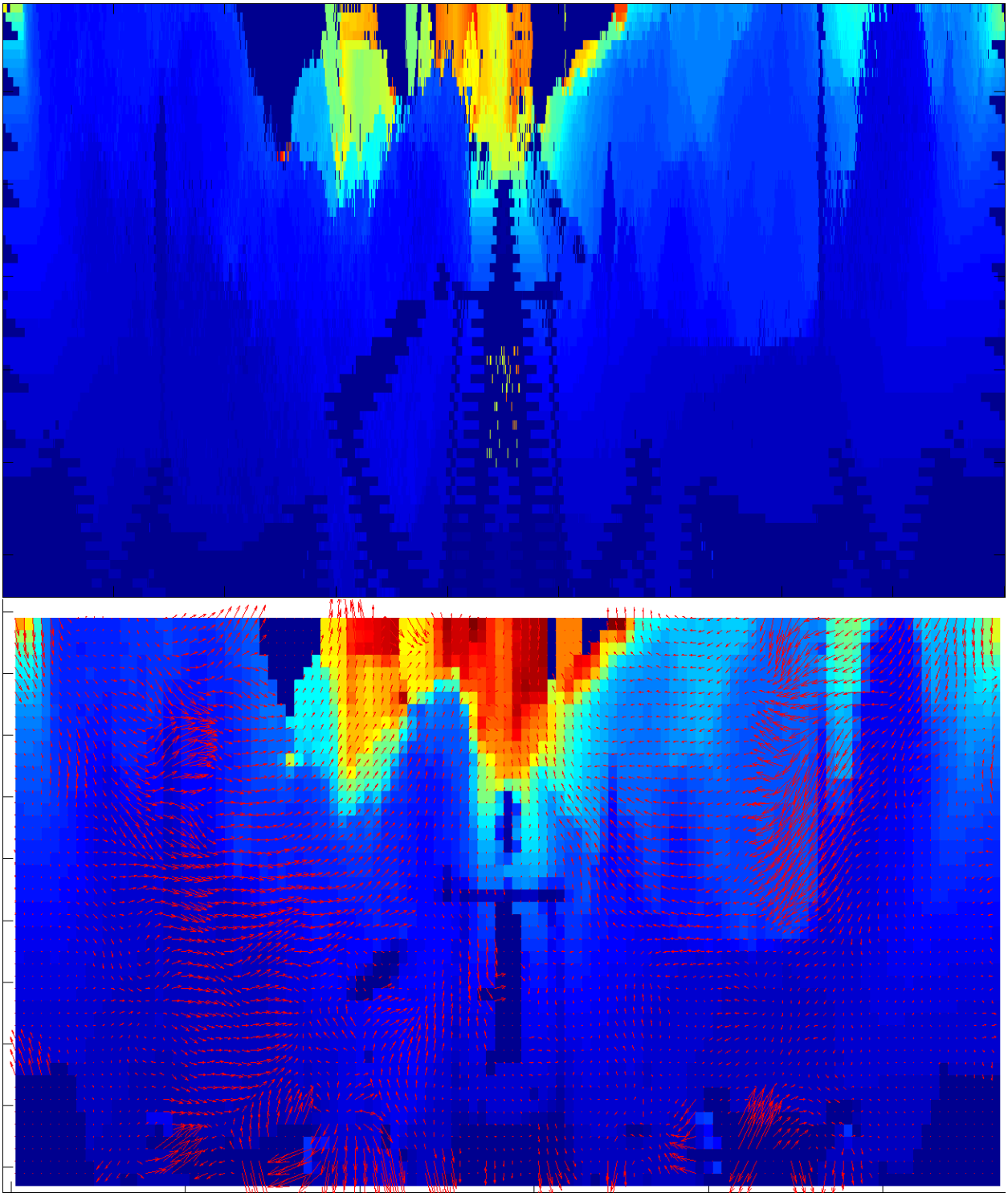


Figure 5. Range image flow on Velodyne® scan 100.

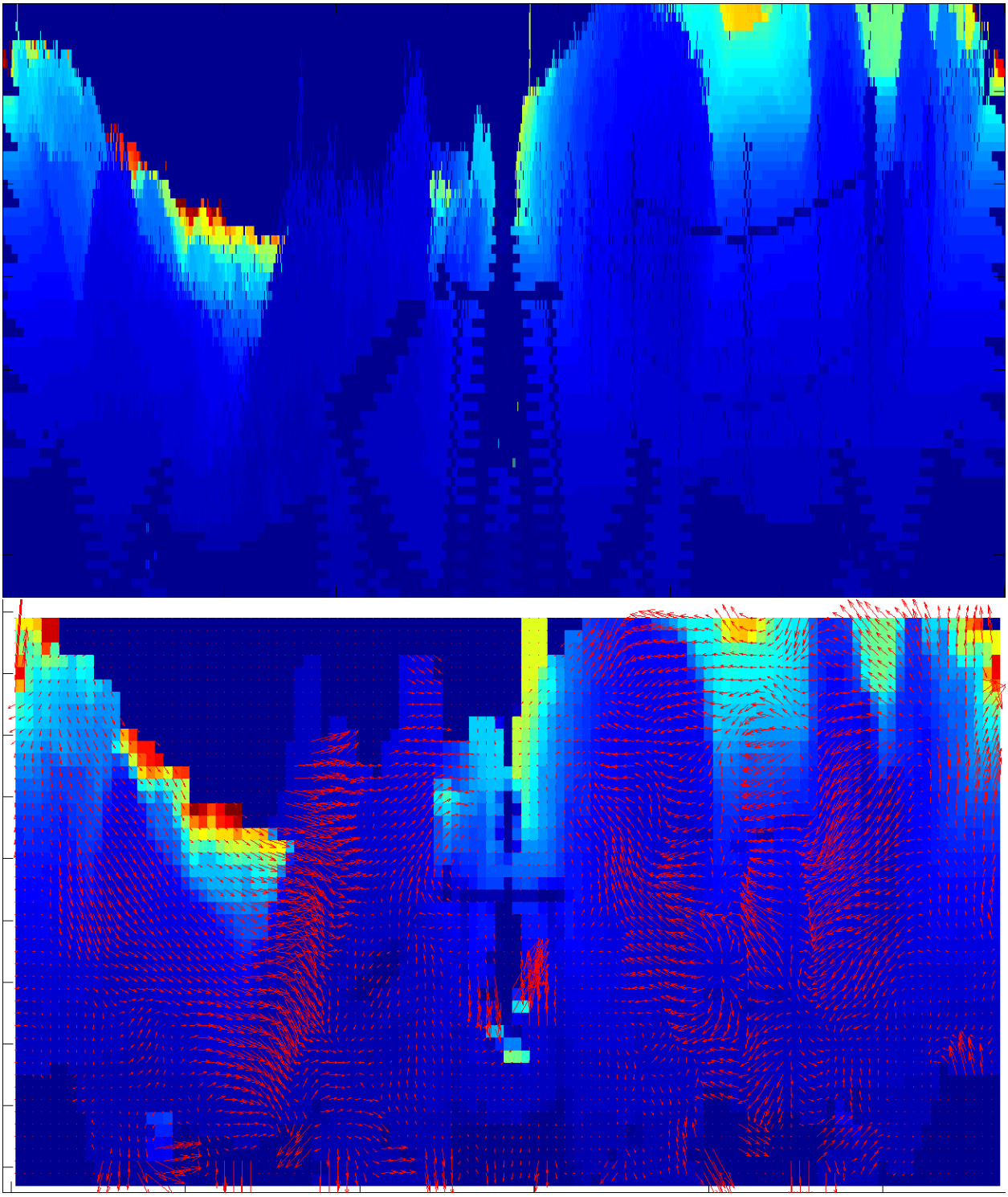


Figure 6. Range image flow on Velodyne® scan 200.

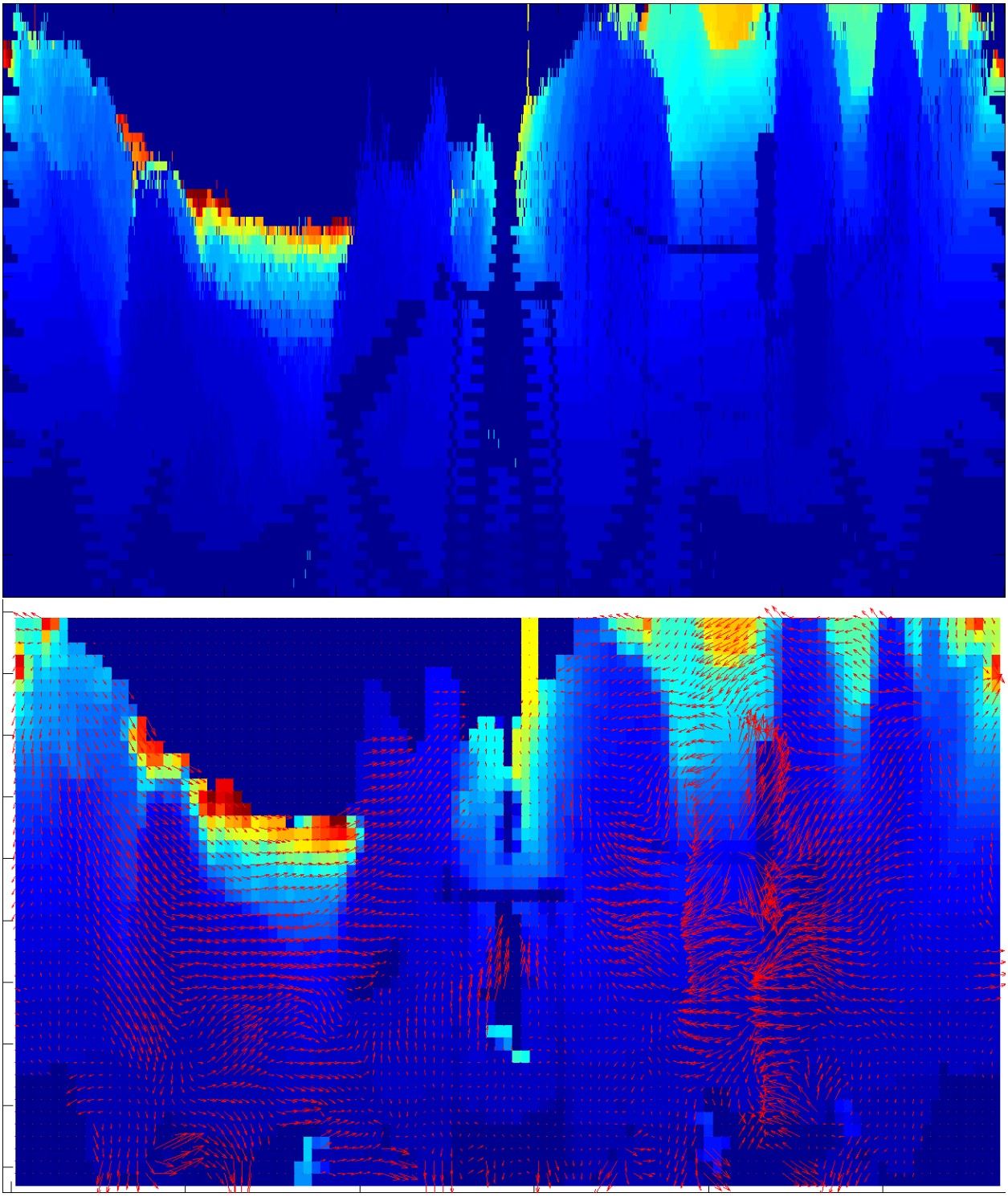


Figure 7. Range image flow on Velodyne® scan 204.

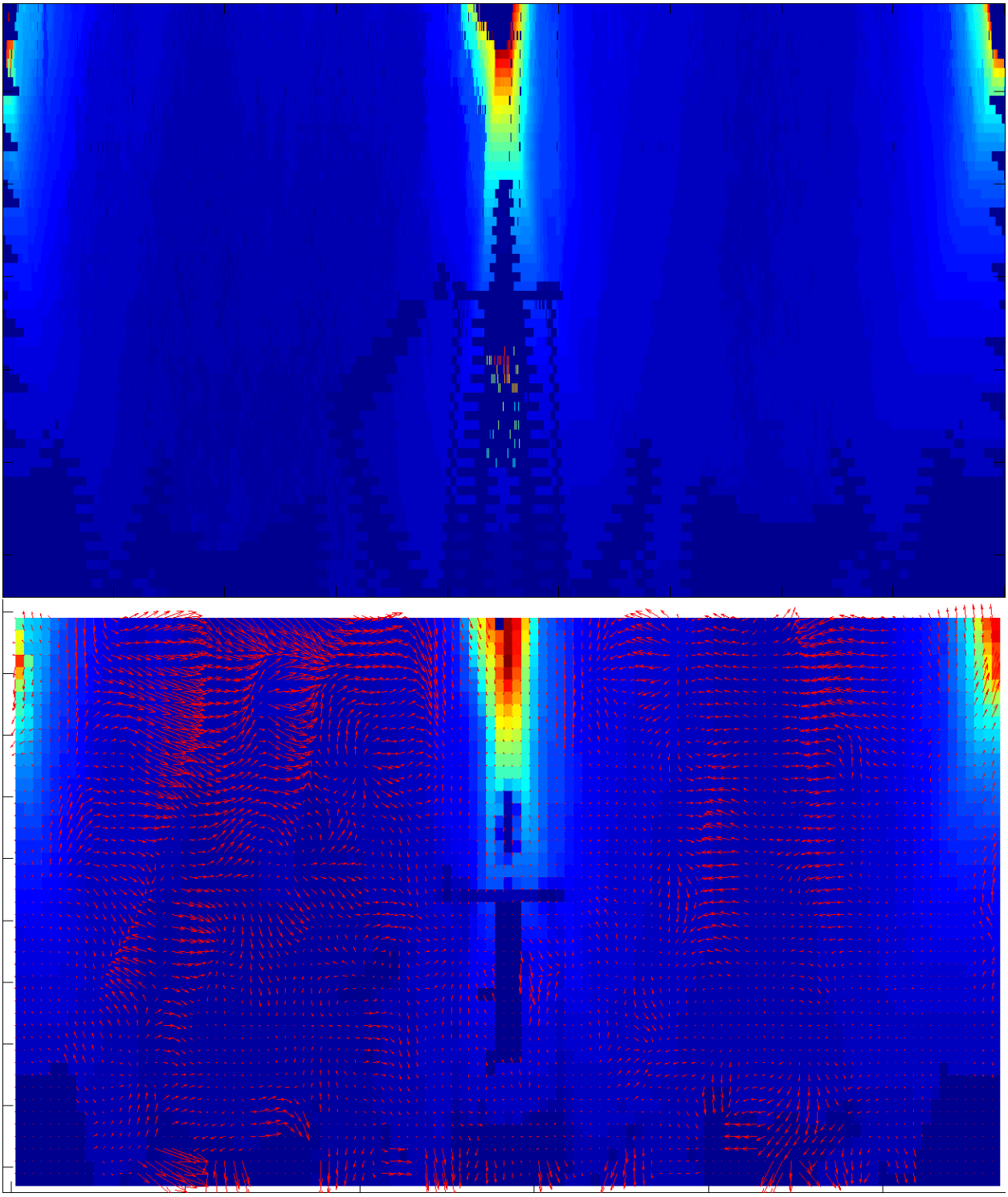


Figure 8. Range image flow on Velodyne® scan 400.

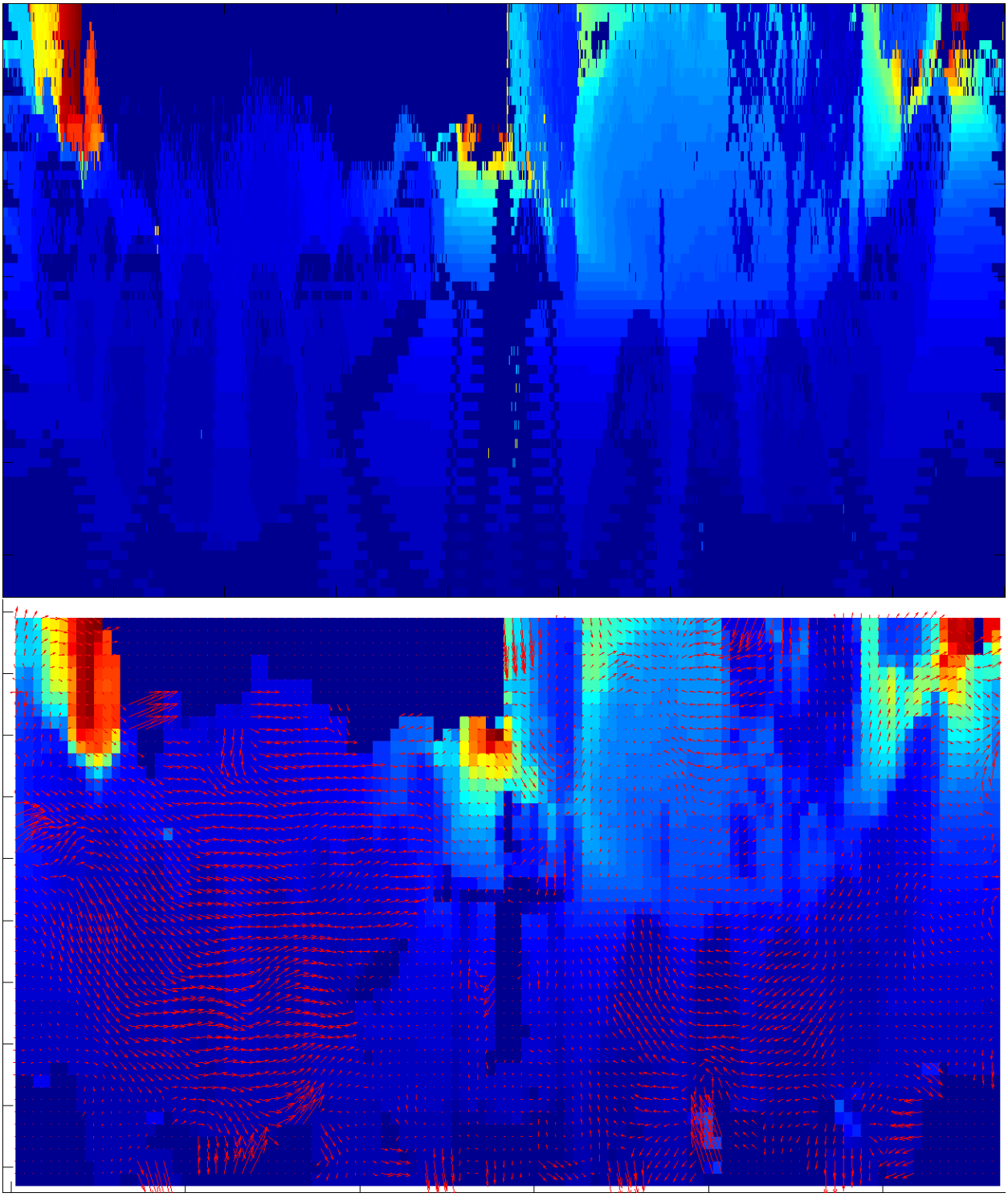


Figure 9. Range image flow on Velodyne® scan 1000.

7. CONCLUSION

The Polynomial Expansion-based method is effective in calculating the flow of range data, though further testing should be done. In addition, this method can detect simple 3D features and calculate planar segmentation on range image data. In future work, quantitative testing will be done to validate the qualitative results, using ground truth data to run large scale quantitative tests. The feature detection and planar segmentation can be incorporated into the flow calculation. Certain features have more stable information about their motion and as such should take on greater weights in the flow calculations. Planer regions, for example, contain very little information about the data's motion, but the edge and corner features contribute a large amount of flow information. Using the planar segmentation, the flows calculated at the corners and edges of planar regions could be interpolated across the region. In addition to the range image work, future work will focus on calculating the rotational components of the flow vectors with respect to a single image location, useful in image stabilization and unmanned aerial vehicle orientation determination.

REFERENCES

1. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. 2011. “A Database and Evaluation Methodology for Optical Flow.” *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31.

2. G. Bradski and A. Kaehler. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Sebastopol, CA.

3. A. Bruhn, J. Weickert, and C. Schörr. 2005. “Lucas/Kanade meets Horn/Schunck: Combining Local and Global Optic Flow Methods,” *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231. Available at http://dx.doi.org/10.1023/B%3AVISI.0000045324.43199.43

4. CESAR Lab at Oak Ridge National Laboratory. 1994. “Odetics LADAR Camera Images.” Available at marathon.csee.usf.edu/range/DataBase.html

5. G. Farnebäck. 2000. “Fast and Accurate Motion Estimation using Orientation Tensors and Parametric Motion Models." *Proceedings of the 15th International Conference on Pattern Recognition (ICPR-2000)*, *Volume 1* (pp. 135–139), 3–7 September, Barcelona, Spain. IEEE.

6. G. Farnebäck. 2000. “Orientation Estimation based on Weighted Projection onto Quadratic Polynomials.” *Vision, Modeling and Visualization 2000 Proceedings* (pp. 89–96), 22–24 November, Saarbrücken, Germany. IEEE.

7. G. Farnebäck. 2001. “Very High Accuracy Velocity Estimation using Orientation Tensors, Parametric Motion, and Simultaneous Segmentation of the Motion Field.” *Proceedings of the Eighth International Conference on Computer Vision (ICCV 2001,) Volume 1* (pp. 171–177), 7–14 July, Vancouver, Canada. IEEE.

8. G. Farnebäck. 2002. “Polynomial Expansion for Orientation and Motion Estimation.” Ph.D. thesis, Linkoping University, Sweden.

9. G. Farnebäck. 2003. “Two-frame Motion Estimation based on Polynomial Expansion.” *Proceedings of the IEEE International Conference on Image Processing, Volume 3* (pp. 363–370), 14–18 December, Barcelona, Spain. IEEE.

10. G. Farnebäck and C-F Westin. 2006. “Affine and Deformable Registration based on Polynomial Expansion.” *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006* (pp. 857–864, 1–6 October, Copenhagen, Denmark. Springer.

11. A. Giachetti, M. Campani, and V. Torre. 1998. “The Use of Optical Flow for Road Navigation.” *IEEE Transactions on Robotics and Automation,* vol. 14, issue 1 (pp. 34–48).

12. J. Gonzalez. 1996. “Recovering Motion Parameters from a 2D Range Image Sequence.” *Proceedings of the 13th International Conference on Pattern Recognition*, *Volume 1* (pp. 433–440), 25–29 August, Vienna, Austria. IEEE.

13. B. K. Horn and B. G. Schunck. 1981. “Determining Optical Flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203.

14. B. D. Lucas and T. Kanade. 1981. “An Iterative Image Registration Technique with an Application to Stereo Vision." *International Joint Conference on Artificial Intelligence, Volume 2* (pp. 674–679), 24–28 August, Vancouver, B.C., Canada. IJCAI.

15. B. McCane, K. Novins, D. Crannitch, and B. Galvin. 2001. “On Benchmarking Optical Flow.” *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143.21.

16. K. Nordberg and G. Farnebäck. 2003. “A Framework for Estimation of Orientation and Velocity.” *Proceedings of the International Conference on Image Processing (ICIP 2003), Volume 3* (pp. III–57, 14–18 September, Barcelona, Spain. IEEE.

17. K. Nordberg and G. Farnebäck. 2005. “Estimation of Orientation Tensors for Simple Signals by Means of Second-order Filters.” *Signal Processing: Image Communication*, vol. 20, no. 6, pp. 582–594.

18. H. Spies, B. Jähne, and J. L. Barron. 2002. “Range Flow Estimation.” *Computer Vision and Image Understanding*, vol. 85, no. 3, pp. 209–231.

19. Y.-j. Wang, G. Farnebäck, and C.-F. Westin. 2010. “Multi-affine Registration using Local Polynomial Expansion," *Journal of Zhejiang University SCIENCE C*, vol. 11, no. 7, pp. 495–503.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-01-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
September 2013	Basic Research	

4. TITLE AND SUBTITLE

Range Image Flow using High-Order Polynomial Expansion

A NISE funded Basic Research Project

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHORS

Brian Okorn
Josh Harguess

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SSC Pacific, 53560 Hull Street, San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR 2033

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Innovative Science and Engineering Program
SSC Pacific, 53560 Hull Street, San Diego, CA 92152–5001

10. SPONSOR/MONITOR'S ACRONYM(S)

NISE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper presents a novel algorithm for estimating the motion in a series of range images. First, each range image is approximated by applying a high-order polynomial expansion to local neighborhoods within the range image. Then, these approximations are used to derive the translation or displacement estimation from frame to frame within the series of range images (also known as range image flow). An iterative method for computing the translation is presented.

We evaluate the algorithm on several synthetic and real-world range image sequences with promising results. Results in this paper are generated from a single iteration of the algorithm in space and time. Therefore, our next step is to improve the implementation of the algorithm so that using multiple spatial scales and past information improve the signal flow estimation, as we would expect. Also, we will port the MATLAB® implementation to C++ to improve speed and meet the signal integration goal of the algorithm.

15. SUBJECT TERMS

polynomial expansion
range image flow
planar region detection
displacement calculation
multi-segmented magnetic robot
conformal wheel

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Brian Okorn
U	U	U	U	29	**19B. TELEPHONE NUMBER** *(Include area code)* (619) 553-6915

INITIAL DISTRIBUTION

84300	Library	(2)
85300	Archive/Stock	(1)
71710	Brian Okorn	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

Approved for public release.

SSC Pacific
San Diego, CA 92152-5001