OPTIMIZING ESTIMATION ACCURACY: LEVERAGING THE WINNER-TAKES-ALL APPROACH

FuChe Wu[1] and Andrew Dellinger [2]

[1] Providence University, Taiwan
[2] Elon University, USA

ABSTRACT

This paper proposes an algorithm for improving estimation accuracy in industrial applications. Traditionally, a weighted sum method is used to map two views, but this approach often leads to blurry results. Instead, the winner-take-all approach is suggested as a means of achieving better accuracy. Three criteria are introduced for evaluating image quality, including sharpness (which measures the effect of motion artifacts from the RGB camera), flatness (which estimates the number of parts in the depth image that belong to plane parts), and fitness (which checks the match between the current view and existing map). While depth images provide 3D structure of the environment, they typically lack sufficient resolution to deliver accurate results. However, by estimating a plane, accuracy can be improved and a more precise boundary can be obtained from the higher resolution of the RGB image.

KEYWORDS

Accuracy,winner-take-all, depth image

1. INTRODUCTION

The depth camera is a useful tool for capturing 3D scenes, but its ability to estimate depth is limited by the need to use different patterns to find a correspondence pair of its stereo position. As a result, the resolution of the camera's raw data is often low, and the accuracy of its depth estimates is constrained by the Nyquist-Shannon sampling theorem. To enhance the quality of the final output, two approaches are worth considering. The first involves improving the quality of the raw data, for example, by increasing the resolution of the image. The second approach is to fit a model to the discrete data, thereby improving accuracy.

Super-resolution is a well-established problem in image processing, involving the fusion of a sequence of low-resolution (LR) noisy and blurred images to create a higher resolution image or sequence. The multiframe super-resolution problem was first addressed in [1], but we will not be focusing on it in this paper.

The limitations of sampling theory often require an increase in sampling frequency to achieve more accurate measurements, which in turn necessitates the use of a higher resolution camera to accurately measure distances in an image. However, as camera resolution is typically fixed, numerical continuity can be leveraged to estimate good values from raw data. Numerical continuity has two forms: spatial continuity and temporal continuity. Spatial continuity refers to the idea that surrounding points in an image may share the same plane, while temporal continuity suggests that pixels in successive images may occupy the same or a similar position. When

David C. Wyld et al. (Eds): SOEN, SIPP, PDTCA, ITE, CCSIT, NLPCL, DaKM, BIGML, AISC -2023
pp. 85-94, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131207

estimating image structure based on this continuity, different image regions and data quality can influence the estimation. Unlike the traditional statistical methodology, which often relies on averaging or median values, in pursuit of accuracy, we prioritize estimation of high-quality images, static images over dynamic ones, and flat areas over arbitrary ones. Additionally, consistency with past data is an important consideration in the evaluation of estimation results.

In this paper, our focus is on identifying the most trustworthy source of data. Single-point measurements can be sensitive and unstable, so we seek to identify lines or planes in images that are more resistant to noise. Plane estimation is particularly promising for improving odometry estimation, given its stability and accuracy. Different imaging devices have different characteristics, with depth cameras providing depth values but at a low resolution, and RGB cameras offering higher resolution but less stable feature tracking. In this work, we use an RGB camera to refine the boundaries of plane areas, leveraging its higher resolution to improve accuracy.

Rather than focusing solely on the precision of a single image, our primary objective is to generate a stable and accurate map for navigation purposes. To evaluate the quality of the captured data, we propose three metrics: sharpness, flatness, and fitness, which will be discussed in detail in Section Three.

2. RELATED WORK

Previous research has explored different approaches to enhance scene understanding by incorporating various constraints. For instance, some studies have focused on fitting feature points with a model to reduce the degree of freedom and obtain better results. This constraint acts as a transformation that converts lower-level features into higher-level features. Gao and Zhang [2] propose a method that extracts and selects planar point features with reliable depth values to reduce alignment errors in the Iterative Closest Point (ICP) algorithm.

Raposo and Barreto[3, 4]argue that Affine Correspondences (ACs) carry more information than Point Correspondences (PCs) which are commonly used as input in Structure-from-Motion algorithms.

Lenac et al. [5] proposed a solution for the SLAM problem that is similar to the one proposed in this paper. They aimed to improve the stability of camera pose estimation by constructing planar surface segments on the local map and using them for registration to build the global map. Using features with structure is more powerful than using single points, which is why Köser and Koch [6] utilized the concept of differential correspondences for robot localization and camera pose estimation from a region of local image features.

Plane detection is a crucial step in our algorithm as it serves as the foundation for constructing a local scene. One approach to finding all possible plane areas in an image is proposed by Feng and Kamat[7], which divides the whole image into small pieces and determines whether each piece is a plane or not. If it is, neighboring pieces that belong to the same plane are merged to agglomerate a larger area. This region-growing-based method enables a quick coarse segmentation to extract the plane area in the image. However, the boundary obtained by this approach is not very precise. To locate a more accurate boundary, Wang et al. [8]use a Markov random field based optimization.

Another approach to plane detection is through the construction of a primitive element such as a mesh from the point cloud. Huang et al. [9] use a volumetric fusion approach to construct a surface first. Then, planes are detected by merging the overlapping area from a sequence of frames.

3. WINNER TAKE ALL

Although video is a convenient method for capturing the environment, it can result in a blurry representation of the scene as the quantity of data increases. Simply adding more data does not necessarily improve accuracy; in fact, it may even decrease it due to errors in computation when dealing with outlier data. Instead, accuracy is determined by the quality of the data. To assess the quality of the data, we propose three criteria: sharpness, flatness, and fitness. In the following sections, we will discuss these criteria in detail.

Motion blur artifacts can create holes or fake boundaries in corresponding depth frames.Gao et al.[2] proposed identifying error regions to eliminate these effects, but not all frames need to participate in map construction. Instead, we can choose a frame with less motion blur to contribute.

In general, a motion blur effect is defined as a point spread function $h(x, y)$ which is characterized by two parameters, namely blur direction and blur length.

The observed image I has the equation

$$I(x, y) = h(x, y) \otimes s(x, y) + n(x, y)$$

where $s(x, y)$ is the original image, $n(x, y)$ is the Gaussian noise, and. $\otimes$indicates the two-dimensional convolution.

A analysis of the blur effect usually occurs in the frequency domain [10]. Let F be a normalized Fourier Transform representation of image I. F is the absolute value of the centered Fourier transform of image I and its value is normalized between 0 to 1.

$$F = normalize(\text{absolute } (center(fourier(I))))$$

A sharp image has more pixels that have a higher frequency, and thus a higher $K_{sharpness}$value.

$$K_{sharpness} = \frac{Count(p_h)}{size(I)}$$

where $p_h$is the higher frequency pixel whose pixel value is higher than a predefined threshold.

Motion blur will result in obvious artifacts (holes or fake boundaries) in the corresponding depth frames. Gao et al. [11] try to identify the error regions to eliminate the effect. However, we don't need each frame to participate in the map calculation. We can choose a frame which less motion blur effect to contribute to the map construction.

From figure 1, we can see the difference between their Fourier transforms of a blurred image and a clear image.A clear image average has a higher value in the frequency domain on average.

To measure the quality of a plane area, flatness is defined as the area to error ratio.

$$K_{flatness} = \frac{area_{plane}}{Error_{depth}}$$

When we estimate a plane area, the $Error_{depth}$is defined as the summarization error, which measures the distance from each pixel to the plane area. Each area will combine with a $K_{flatness}$to identify its quality. If there are multiple planes overlapping on the global map, only leave the plane with the highest $K_{flatness}$.

A global map consists of multiple local maps. When a local map finds the best fit location among the global map, there may exist some overlapping areas. Fitness is defined as the average error between the overlapping areas.

$$K_{fitness} = \frac{area_{overlap}}{Error_{plane}}$$

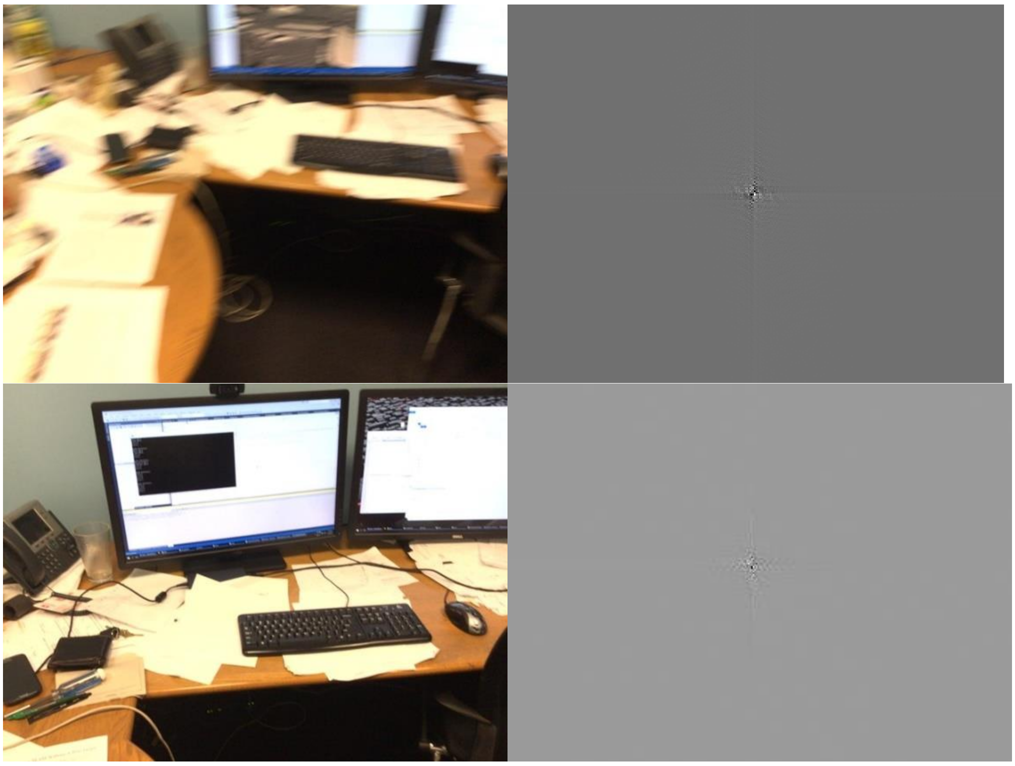

Figure 1.examples of a blurred image and a clear image and their Fourier transform

Captured data can provide local map information. To fuse the local maps to construct the global map, a traditional approach is based on the weighted sum technique. But this usually means that a bad local map will affect the precision of the final result. Thus, we try to define the measure quality method. When there exist multiple data for an area, a quality coefficient can be used to discriminate which data is better to keep.

4. Plane Boundary Refinement

We adopt Feng's work to segment the plane area. First, a set of planes are detected from a depth image. For each plane area, the RANdomSAmple Consensus (RANSAC) algorithm is used to refine plane detection. However, since the resolution of the depth camera is limited, we will try the assistant from the RGB image to improve the quality of plane segmentation.

To translate the boundary from the depth image into the RGB image, a closed path is identified for each plane area on the depth image first. Then, because of the Epipolar geometry, each depth pixel can project back to its corresponding position in the RGB image. Within these candidate positions, we will locate all the possible positions to get a better plane region.

Each plane can be represented as

$$n_1.(x - p_1) = 0$$

where $n_1$is the normal, and $p_1$is a point on the plane. Thus, the plane also has the form as

$$ax + by + cz = d$$

Each frame has a local coordinate. A point p in the ith depth frame can be represented as p_i^D , similar p_i^{RGB} represented as the location in ith RGB frame. Assume the rotation and translation are R and T, respectively.

$$p_i^{RGB} = R * p_i^D + T$$

More specifically, the boundary in the RGB frame should be in the range

$$R * \left(-\frac{1}{2}, -\frac{1}{2}\right) + T + p_i^{RGB} \leq \hat{p}_i^{RGB} \leq R * \left(\frac{1}{2}, \frac{1}{2}\right) + T + p_i^{RGB}$$

Thus, each pixel in the boundary from the depth frame can find its corresponding boundary at the RGB frame. Assume that the general case on the boundary from the RGB frame also has texture variation. Thus, from this potential region, we can find a new boundary that has a similar shape to the original boundary that extracts from the depth frame.

Let $B_d B_{rgb}$ be the boundary extracted from the depth frame and RGB frame, respectively. We will try to merge two neighboring line segments to form a long line from the boundary. Thus, a boundary consists of many line segments.

$$B_d = \{L_1, L_2, \ldots L_n\}$$

we will find a best-matched line segment in the RGB frame for each line segment in the depth frame.

Thus,

$$B_{rgb} = \{M(L_1), M(L_2), \ldots M(L_n)\}$$

where $M(L_1)$ represents the best-matched line segment.

After finding the boundary segments in the RGB frame, we also can find the mapping plane.If we find a plane in the depth frame as $n_1 * (x - p_1) = 0$, then a plane in the RGB frame should be

$$R * n_1 * (x - (R * p_1 + T)) = 0$$

Thus, we can find the RGB plane's 3D location by finding the intersection between the mapping plane and the ray from the focus to the line segment.

5. ODOMETRY ESTIMATION

Usually, to estimating the camera position is based on tracking the corresponding feature points. The new camera pose can be calculated by the corresponding pair. For getting a more stable result, we try to solve this problem by finding the corresponding plane. In frame n, area $A_i$located in frame n should have some overlap area with area A_j or in its neighboring area.

A maximum area A_i is selected as the first target. Then, each of its neighboring areas will be tested in sequence to find an optimized solution.

Assume that an area A_j is a corresponding area to A_i and their normal directions are n_i, n_j respectively.
Then, a normal direction $n_q = n_i \times n_j$ is the normal direction of the rotation plane from n_i to n_j.

And we can find the best matching area for each area from its neighboring area by comparing the normal directions of their rotation planes.

If A_k is a possible corresponding area of A_i, then its rotation plane should consistent with n_q, thus $\|n_q - n_i \times n_j\|$ should be the minimum value among all its neighboring areas. A quaternion vector q represents a rotation about a unit vector (μ_x, μ_y, μ_z) through an angle θ. A unit quaternion itself has unit magnitude, and can be written in the following vector format.

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) \\ sin\left(\frac{\theta}{2}\right)\mu_x \\ sin\left(\frac{\theta}{2}\right)\mu_y \\ sin\left(\frac{\theta}{2}\right)\mu_z \end{bmatrix}$$

As a rotation matrix R can be formed from a quaternion q as follows.

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Thus, the error of a matched area should have the minimized cost if their surface normal fit with the rotation matrix.

$$cost = \sum \|R \cdot n_i - n_j\|$$

After getting the rotation matrix, we need at least one feature point between different poses to estimate the translation. We want the feature point as stable as possible. Thus, we use two methods to locate this feature point. One is a geometry-based approach, which will be more robust. If there exist three linear independent planes that matched on the different poses, we can find one exact intersection point among these three planes as the feature point.

Another method to locate this feature point is a texture-based approach. It tries to guarantee a solution.First, the feature extraction algorithm, such as the SIFT method, is applied to extract some good features on the RGB image. We need to find the corresponding relationship between two frames. We divide feature points into the following categories and select them in this order.

Among multiple areas, the priority selection is the center area. We will check whether any area occupies the center area. If not, the next candidate will be chosen by its size. The rotation matrix will apply to each selected area and let them have a similar pose.

Then we will check whether or not the extracted feature points belong to this area. We will check the whether or not the number of feature points in the selected area and corresponding area is the same. If the number is the same, we can find a center among these feature points as the local original in this area.

Each feature point calculates its polar coordinate. Then, it can map to another frame by its angle order to find its corresponding relationship.

Then, the translation is

$$T = \frac{\sum_{i=1}^{m} p_i^n - p_i^{n+1}}{m}$$

Assume that there are m feature points. A feature point in frame n is represented as p_i^n.

The average error is

$$\epsilon = \frac{\sum_{i=1}^{m} \| p_i^n - p_i^{n+1} - T \|}{m}$$

If the error is less than a predefined threshold, then it is a solution. Otherwise, we should discard the mapping pair.

6. EXPERIMENTS

Our primary objective is to develop an automated system to measure the size of objects on the production line accurately. We assume that the objects under consideration are stationary and placed on a stable platform. After each measurement, the object is reset to its initial position, indicating its readiness for the next measurement cycle. To commence the measurement process, the system waits for the object to remain stationary for a predefined duration, ensuring its stability before taking the measurements. By following this approach, we can ensure that the measurements are precise and accurate, as the object remains static throughout the measuring process.

With the plane constraint, we can significantly improve the estimation accuracy, making it suitable for a wide range of applications, including object size measurement. This is particularly important in the intelligent logistics industry[12], where precise measurements of packages and objects are essential for efficient and effective transport and delivery.

To begin the measurement process, the camera system must first be calibrated to determine the transformation relationship between the depth camera and the RGB camera. This involves capturing a set of calibration images using a calibration pattern, as shown in Figure 2. For our experiments, we used a Realsense depth camera D435i, which has been shown to achieve an RMS depth error of less than 10 mm with plane constraint when the depth is not much more than 1.5 meters [13].

Once the camera system is calibrated, the measurement process can begin. The object to be measured is placed on the platform, and the camera captures a sequence of images as the object moves along the production line. Each movement represents a reset, indicating that the object is ready for the next measurement.

The measurement process starts when the object is stationary for a period of time, which allows the system to detect the presence of the object and calculate its size. This is achieved by segmenting the object from the background using the plane constraint and the sharpness, flatness, and fitness criteria discussed earlier. The system then calculates the size of the object based on the segmented image data.

Overall, our approach provides a robust and accurate solution for object size measurement, with potential applications in a variety of industries and settings.

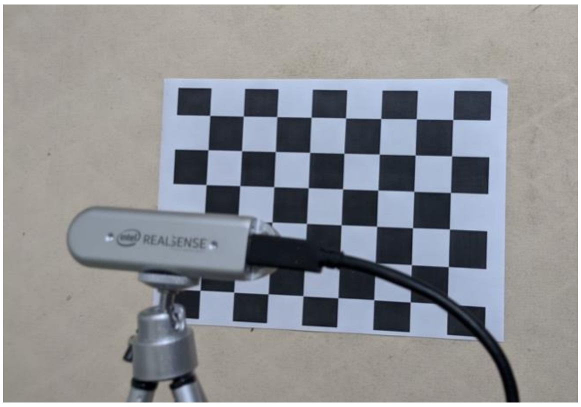

Figure 2. Camera calibration

To accurately measure the dimensions of a box, we can establish a fixed relationship between the camera and the target platform and assume that the platform is a static background. This can greatly simplify the process of dimension measurement. At the beginning of the measurement process, a good calibration process is essential. With a well-calibrated system, we can accurately determine the intrinsic and extrinsic parameters of both the RGB camera and the depth camera. This calibration process is crucial for achieving accurate and reliable measurements.

Once the calibration process is complete, we can easily measure the dimensions of any box placed on the platform by filtering out the background. To achieve this, we can use image segmentation to separate the box from the background. With the box isolated, we can then accurately calculate its dimensions.

It is worth noting that an RGB image typically has a better resolution than a depth image. This means that the dimensions measured from RGB segmentation are generally more accurate than those measured from a depth image alone. However, both the RGB and depth images provide important information for accurately measuring dimensions.

To provide a quantitative comparison of the accuracy of dimension measurement using RGB segmentation and depth images, we conducted an experiment where we placed the target object 300 mm in front of the camera. We compared the RMS error of dimension measurement using both RGB segmentation and depth images. The results showed that RGB segmentation provided more accurate measurements with a lower RMS error compared to depth images alone.

Device/resolution	RMS error
Depth/ 1280 x 720	2mm
RGB/ 1920 x 1080	1mm

As figure 3 shows, the boundary from a depth image translates into a path of an RGB image. Since the resolution increases, the precision also improves.

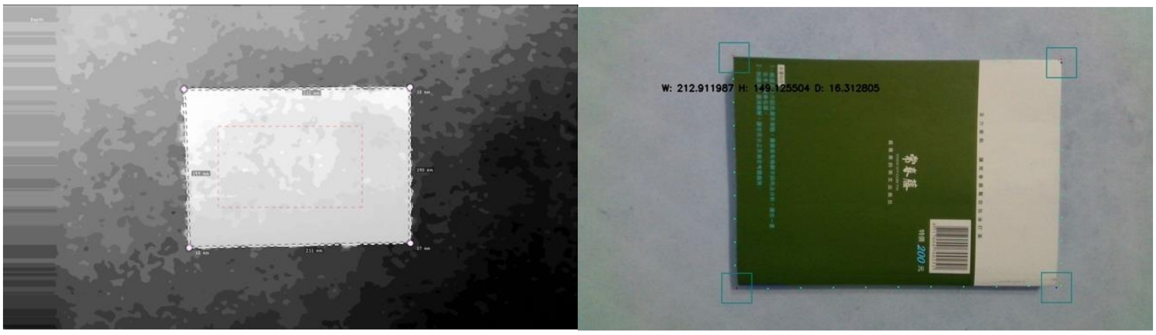

Figure 3. Dimension measurement from a depth image and an RGB image

7. CONCLUSIONS

If we are unsure which data is better, having more candidates may seem like a good solution. However, using statistical approaches such as the majority vote or mean value to decide which data is better can be affected by outlier data and may not be suitable for applications that require high accuracy. In such cases, data quality is more important than data quantity. Utilizing features with more constraints, such as planes, can lead to a more stable solution.

To improve accuracy, we propose an algorithm that incorporates a plane constraint to achieve more precise results. We utilize sharpness, flatness, and fitness criteria to filter out any bad input that may affect computation. Furthermore, we combine the different characteristics of the depth camera and RGB camera to provide higher resolution for the captured data.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under grant no. MOST 110-2221-E-126-007.

REFERENCES

1. Tsai, R., Multiframe image restoration and registration. Advance Computer Visual and Image Processing, 1984. 1: p. 317-339.
2. Gao, X. and T. Zhang, Robust RGB-D simultaneous localization and mapping using planar point features. Robotics and Autonomous Systems, 2015. 72: p. 1-14.
3. Raposo, C. and J.P. Barreto. Theory and practice of structure-from-motion using affine correspondences. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
4. Raposo, C. and J.P. Barreto. $$\pi $$ Match: Monocular vSLAM and Piecewise Planar Reconstruction Using Fast Plane Correspondences. in European Conference on Computer Vision. 2016. Springer.
5. Lenac, K., et al., Fast planar surface 3D SLAM using LIDAR. Robotics and Autonomous Systems, 2017. 92: p. 197-220.
6. Köser, K. and R. Koch. Differential spatial resection-pose estimation using a single local image feature. in European Conference on Computer Vision. 2008. Springer.

7. Feng, C., Y. Taguchi, and V.R. Kamat. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. in 2014 IEEE International Conference on Robotics and Automation (ICRA). 2014. IEEE.
8. Wang, C. and X. Guo. Plane-based optimization of geometry and texture for rgb-d reconstruction of indoor scenes. in 2018 International Conference on 3D Vision (3DV). 2018. IEEE.
9. Huang, J., et al., 3Dlite: towards commodity 3D scanning for content creation. ACM Trans. Graph., 2017. 36(6): p. 203:1-203:14.
10. De, K. and V. Masilamani, Image sharpness measure for blurred images in frequency domain. Procedia Engineering, 2013. 64: p. 149-158.
11. Gao, Y., et al., Depth Error Elimination for RGB-D Cameras. Vol. 6. 2015: Association for Computing Machinery. Article 13.
12. Park, H., A. Van Messemac, and W. De Neveac. Box-Scan: An efficient and effective algorithm for box dimension measurement in conveyor systems using a single RGB-D camera. in Proceedings of the 7th IIAE International Conference on Industrial Application Engineering, Kitakyushu, Japan. 2019.
13. Grunnet-Jepsen, A., J.N. Sweetser, and J. Woodfill, Best-Known-Methods for Tuning Intel® RealSense™ D400 Depth Cameras for Best Performance. New Technologies Group, Intel Corporation, Rev. 1.