Proceedings of Machine Learning Research 129:797–810, 2020 ACML 2020

A foreground detection algorithm for Time-of-Flight cameras adapted dynamic integration time adjustment and multipath distortions

Detong Chen CHENDETONG@SMARTTOF.COM
Shanghai Data Miracle Intelligent Technology Co., Ltd.
Ln.215, Gaoguang Rd, Shanghai City, China

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

There are two scenarios often appear in the use of a Time-of-Flight (ToF) camera. One is requiring dynamic adjustment of its integration time to avoid overexposure, the other is multipath distortions happen. In these two scenarios, the pixel values of depth map and intensity map will suddenly and greatly change, and it will effect ToF based applications that require foreground detection. Traditional foreground detection algorithms can not adapt to these scenarios well, since they are sensitive to the sudden large change of pixel values and the threshold of pixel values difference people pick. Therefore, this paper proposes a pixel-insensitive and threshold-free algorithm to deal with the above scenarios. It is an end-to-end model based on deep learning. It takes two intensity maps captured by a ToF camera as input, where one intensity map works as a background, and the other works as a contrast. Taking their actual differences, also called foreground, as a label. Then, using deep learning to learn how to detect foreground based on these inputs and labels. To learn the pattern, datasets are collected under various scenes by multiple ToF cameras, and the training datasets are enlarged through applying a series of random transformations on the foreground and introducing two-dimensional Gaussian noise. Experiments show the new algorithm can stably detect foreground under different circumstances including the two mentioned scenarios.

Keywords: Time-of-Flight, Foreground Detection, Deep Learning

1. Introduction

ToF imaging is a depth-sensing technology that has been attracting a lot of attention in recent years. It has a wide range of applications in many fields, such as passenger flow statistics Wang et al. (2019), volume measurement Verdú et al. (2013), gesture recognition Hwang and Kim (2019) and robot obstacle avoidance Shahnewaz and Pandey (2020). Some problems are still not well solved that effects ToF based applications, especially, multipath distortions Son et al. (2016) and requiring dynamic adjustment of integration time to avoid overexposure. In these senses, the pixel values of the entire depth map and the intensity map suddenly and greatly change (Figure 1).

The foreground detection belongs to traditional computer vision algorithms. The rationale in the approach is to compare the background learned by the model with the contrast image, and then employ a suitable threshold of pixel value difference to subtract the background from it. The remaining objects after the subtraction can be approximated as the

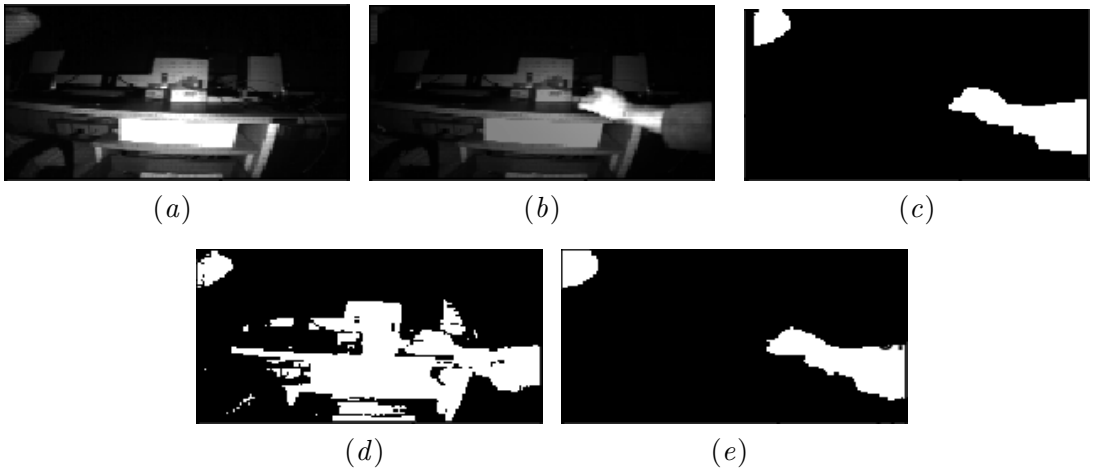


Figure 1: Parts (a) and (b) above show what is sudden large change in pixel values. Part (a) was taken under a strong light environment, while part (b) was taken under a weak light environment. Although they are in the same scene, there is a huge difference in pixel values. Here, the performances of using traditional foreground detection and the new algorithm are compared. (a) Intensity map acts as a background. (b) Intensity map acts as a contrast. (c) Foreground (actual difference): the region in white indicates foreground while the region in black represents background (d) The output of the traditional foreground detection algorithms. (e) The output of the new algorithm.

foreground objects. Although they work well in many scenes, they depend on an assumption, the behavior of each pixel is independent, that is often difficult to satisfy. To solve this problem, many advanced foreground detection algorithms are proposed, such as MOG, but they are based on the idea of learning the change of a single pixel, if the change in the pixel is too fluctuate or has not occurred in the learning period, it will lead to the failure of the algorithm Langmann et al. (2010). Besides, the performances of these algorithms all heavily rely on the threshold of pixel values difference people pick.

The difficulty of foreground detection based on ToF imaging can be addressed, if there is a method that can compare the regional features between the background image and the contrast image, such as contour information, rather than the values of individual pixels. Many studies have shown that deep learning is good at understanding regional features and semantic segmentation, such as the U-NET for medical image segmentation Ronneberger et al. (2015), which uses a U-shaped neural network for semantic segmentation. Inspired by these papers, this paper proposes a pixel-insensitive and threshold-free foreground detection algorithm based on deep learning that can deal with the above cases. The algorithm takes two intensity maps captured by a ToF camera as input, where one intensity map works as a background, and the other works as a contrast. Taking their actual differences, also called foreground, as a label. Then, using deep learning to learn how to detect foreground based on these inputs and labels. To learn the pattern, datasets are collected under various scenes by multiple ToF cameras, and the training datasets are enlarged through applying a series of random transformations on the foreground and introducing two-dimensional Gaussian

noise. Experiments show the new algorithm can stably detect a foreground under different circumstances including dynamic integration time adjustment and multipath distortions.

The contribution of the new algorithm can be summarized as the following two points:

• First time illustrates a pixel-insensitive and threshold-free foreground detection algorithm based on deep learning that can work stably when the pixel values of the image captured by ToF fluctuates.

• Augment training datasets through applying a series of random transformations on the foreground and introducing two-dimensional Gaussian noise to build a more robust algorithm.

2. Related Work

Foreground detection: The foreground detection belongs to traditional computer vision algorithms used for detecting changes in continuous pictures. The traditional algorithms include inter-frame differencing, average background method, MOG and so on. Inter-frame differencing is a method of comparing the former frame and latter frame, and then employing a suitable threshold of pixel value difference to detect the foreground Shahbaz et al. (2015). The average background method is a method of learning the average and variance of background pixels through multiply frames, and then using an appropriate threshold of pixel value difference to detect the foreground Shahbaz et al. (2015). MOG is a method of modeling each background pixel by a mixture of several gaussian distributions through learning from multiple frames, and then employing a suitable threshold of pixel value difference to detect the foreground Langmann et al. (2010). It can be seen that these algorithms are based on the logic of learning a single pixel, and rely on a suitable threshold of pixel value difference to detect the foreground.

Semantic segmentation: Semantic segmentation is an important task in computer vision. Its goal is to label each pixel of an image with a corresponding class. In the past few years, deep learning had a tremendous impact on semantic segmentation. Olaf et. al Ronneberger et al. (2015). proposed a neural network called U-Net to do biomedical image segmentation. The network consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. Chen et al. proposed a neural network called DeepLabv3+ to do semantic segmentation Chen et al. (2018). The network uses encoder-decoder to integrate multi-scale information and atrous separable convolution to reduce the computation complexity. These neural networks are all based on encoder-decoder architecture.

ToF data: There are many public ToF datasets on the Internet, for example, datasets used for depth calibration containing depth captured and ground truth Garro et al. (2013), datasets used for gesture recognition containing different gestures IEE (2014) and datasets used for Semantic segmentation containing different objects Dal Mutto et al. (2012). These datasets cannot be used in this paper, because of lacking multipath distortions, dynamic integration time adjustment and actual difference label between two images. Therefore, new datasets are collected and labeled.

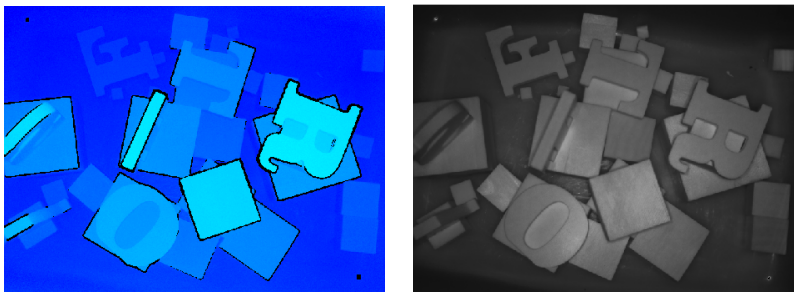
3. Methods and analysis

3.1. ToF imaging (depth map and intensity map)

The Time-of-Flight method of 3D imaging aims to continuously transmit light pulses towards the target, and then receive the light returned from the object, and finally obtain the target distance by detecting the flight (round trip) time of the light pulse Lange and Seitz (2001).

In ToF-based applications, the depth map and intensity map are frequently used. In the depth map, each pixel value is determined by the distance between the camera and the object measured by the time of flight. Generally, the unit of the pixel value in the depth map is millimeter and the value range is from 0 millimeters to several thousand millimeters. In the intensity map, each pixel value is calculated by the amount of infrared light reflected back to the camera. Usually, the value range is from 0 to 4096. The value range of the depth map and intensity map may differ according to the type of ToF Imager.

In this paper, visualizing the depth map and intensity map is based on the following method. Selecting an expected value range, such as from 0 to 1,000, map it linearly to 0 to 255 (uint8 data type), and then convert the value of each pixel depending on the previous step, and finally use a pseudo color or grayscale color to display. Besides, pseudo color visualization will always be used on the depth map, while grayscale visualization will always be used on the intensity map (Figure 2).



(a) (b)

Figure 2: The figures above show the visualization of the depth map and intensity map. (a) Using pseudo color to visualize depth map. (b) Using grayscale color to visualize the intensity map.

3.2. Overexposure and dynamic integration time adjustment

Integration time: The duration that ToF sensor collects the photo-charge of infrared light emitted by the camera itself, the unit is microseconds.

Generally, the longer integration time, the better imaging quality is achieved, but long integration time may lead to regional overexposure, which is presented as error values (Figure 3). ToF cameras and RGB cameras share the same concept of overexposure that, the amount of light collected exceeds the range that the camera can withstand. In order to achieve the best imaging quality without overexposure. It is necessary to dynamically

adjust the integration time. For instance, increase the integration time when the image does not have overexposure, In contrast, decrease the integration time when the image has overexposure.

In addition, different integration time brings different depth measurement errors which makes accuracy foreground detection based on depth map difficult.

(a) (b)

Figure 3: The intensity maps with different integration times are compared here. Firstly, it is obvious that figure (b) is brighter than figure(a). Secondly, figure(a) does not have overexposure, while figure (b) has overexposure which is circled in red. (a) No-overexposure intensity map with integration time of 400 microseconds. (b) Overexposure intensity graph with integration time of 800 microseconds, the overexposed part is the red circled part.

3.3. Multipath distortions

There are two types of multipath distortions, multipath inference(MPI) and light scattering Shahnewaz and Pandey (2020). Although MPI will cause distance inaccuracy, it will not interfere with neighboring pixels. Only scattering will interfere with the pixel values of neighboring pixels.

MPI: All existing ToF cameras work under the assumption that each given pixel follows an optical path. In other words, the light in the scene is reflected only once. However, this assumption is impossible in reality. Light will be reflected many times in the scene. An object will not only reflect the modulated light emitted by the camera, but also reflect the light from other indirect paths. The interference between the reflected light from multiple sources will cause the error of depth and intensity measurement.

Light Scattering: Light scattering is similar to MPI (Light is reflected many times), but caused by multiple reflections of light in the lens.

In the case of multipath distortions, the change of the intensity map is complicated (the values in some places becomes higher, the values in some places becomes lower, but the contour of the object is still clearly visible), while the values of depth map become smaller as far objects get closer (Figure 4).

3.4. The problem of using traditional foreground detection algorithm

In the use of ToF cameras, when the integration time is fixed and no multipath distortions existed, the fluctuations of each pixel in the intensity map are relatively small, the

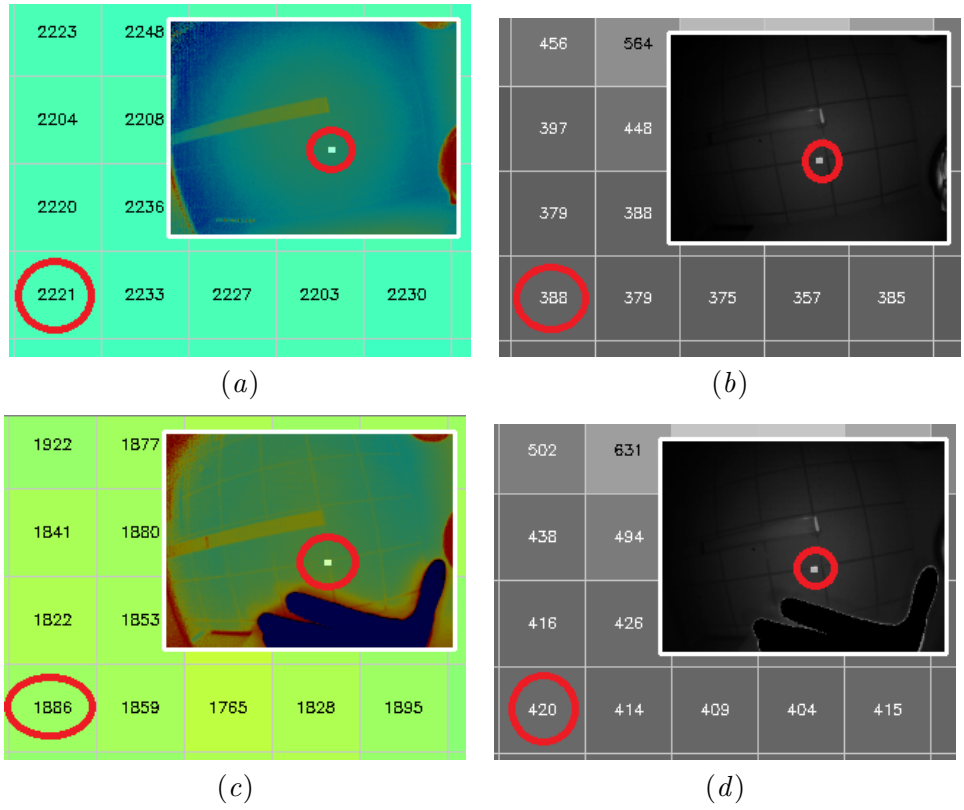


Figure 4: An intuitive display of how multipath distortions effect the depth map and intensity map is shown. The ToF camera is placed on the table toward to the ceiling. When the hand is closing enough to the camera which causes multipath distortions, the depth value and the intensity value of the ceiling obviously changed. (a) Without multipath distorts, the value of a pixel in the depth map is 2221mm, and the fluctuation range is 30mm. (b) Without multipath distorts, the value of a pixel in the intensity map is 388, and the fluctuation range is 5. (c) With multipath distorts, the value of the corresponding pixel in Figure (a) becomes 1886mm, which changes 355mm. (d) With multipath distorts, the value of the corresponding pixel in Figure (b) becomes 420, which changes 32.

traditional foreground detection algorithms based on single pixel value judgment can work nicely when a suitable threshold of pixel values difference is picked. When the integration time is dynamically adjusted to avoid overexposure and multipath distortions encountered, it will cause a sudden change in the value of each pixel in the image. If just look at a single pixel, this change (dynamic integration time and multipath distortions) is completely different from the previous change (fixed integration time and no multipath distortions). Without the help of surrounding pixels, there is almost no clue to find the rule, resulting in misjudgment of foreground. For instance, MOG algorithm is based on the idea that using multiple gaussian model to learn each pixel changes independently. If the change in pixel is too fluctuate or has not occurred in the learning period, it will lead to the failure of the algorithm. However, when the dynamically adjusted integration time is applied, the change

of the pixel values of the entire intensity map is approximately proportional to integration time, and the change of depth map might vary due to different calibration work. Meanwhile, when multipath distortions is encountered, it will cause complex changes in both the depth map and the intensity map, which will eventually be demonstrated in the large changes of pixel values. In the case of multipath distortions, the change of the intensity map is complicated (the values in some places becomes higher, the values in some places becomes lower, but the contour of the object is still clearly visible), while the values of depth map become smaller as far objects get closer.

Traditional algorithms use the assumption (the behavior of each pixel is independent) to independently learn the changes of each pixel to detect foreground. However, in the two cases mentioned earlier, when only looking at each pixel independently, it is an irregular change and difficult to learn and extract the foreground through a suitable threshold.

3.5. New foreground detection algorithm based on deep learning

Human can clearly distinguish the actual difference between two intensity maps including the scenes that traditional algorithm cannot adapt well, because human can compare the regional features between the background image and the contrast image, such as contour information, rather than the values of individual pixels. Using the intensity map instead of the depth map is based on two considerations: (1) In the case of continuously capturing data from a completely static field, the fluctuation of pixel values in the intensity map is relatively small compares to depth map. (2) Different ToF manufacturers perform different filters and calibration on the depth map, which makes it difficult to have a uniform standard, while the intensity map does not include these.

Inspired by how human distinguish actual difference, an algorithm based on an end-to-end deep learning model is proposed. The algorithm takes two intensity maps captured by a ToF camera as input, where one intensity map works as background and the other works as contrast. Taking their actual differences, also called foreground, as label. Then, using deep learning to learn how to detect foreground based on these inputs and labels. Assume the shape of the image is (height, width, 1), then the input will be a stacked image with shape (height, width, 2) which stacked the background map and the contrast map, and the label will be a binary image with shape(height, width, 1) where 0 represents the background and 1 represents the foreground. In order to reduce the computation complexity, the input and label images can be scaled down by the same factor (Figure 5).

In a fixed installation scene, the first frame taken by the ToF camera can be used as the background intensity map. In a dynamic scene, the previous frame can be used as the background intensity map.

The algorithm is a lightweight end-to-end neural network model. The model consists of a head, a contraction path and an expansion path (Figure 6). The head is used to increase receptive Field. The contraction path is a typical convolutional network, consisting of repeated application of convolutions, each with a rectified linear unit (ReLU) and Max Pooling operation. The expansion path combines features and spatial information through a series of upward convolutions and high-resolution features from the contraction path. Zeiler and Fergus (2014) shows the deeper the network is, the higher the level of semantics network can understand. Due to the foreground detection does not need to understand the

high level of semantics in the image, only the regional difference, the deep network is not necessary.

To train the model, Binary Cross Entropy based on each pixel is used by the model.

$$L(y, \widetilde{y}) = -\frac{1}{m}\sum_{i=1}^{m}[y_i log(\widetilde{y}) + (1 - y_i) log(1 - \widetilde{y})]$$

Training details: The model is implemented by using keras based on tensorflow. In order to improve the robustness of the model, during training, random cropping and flipping are applied. The model uses the Adam optimizer with an initial learning rate of 1e-3 to train, and then keeps reducing learning rate when the metric has stopped improving. After learning rate is reduced to zero, the model reaches its optimal state.

(a) (b) (c)

Figure 5: The input and label of the algorithm for training are shown here. (a) Intensity map acts as background for the input. (b) Intensity map acts as contrast for the input. (c) Foreground (Actual Difference) acts as label for output.

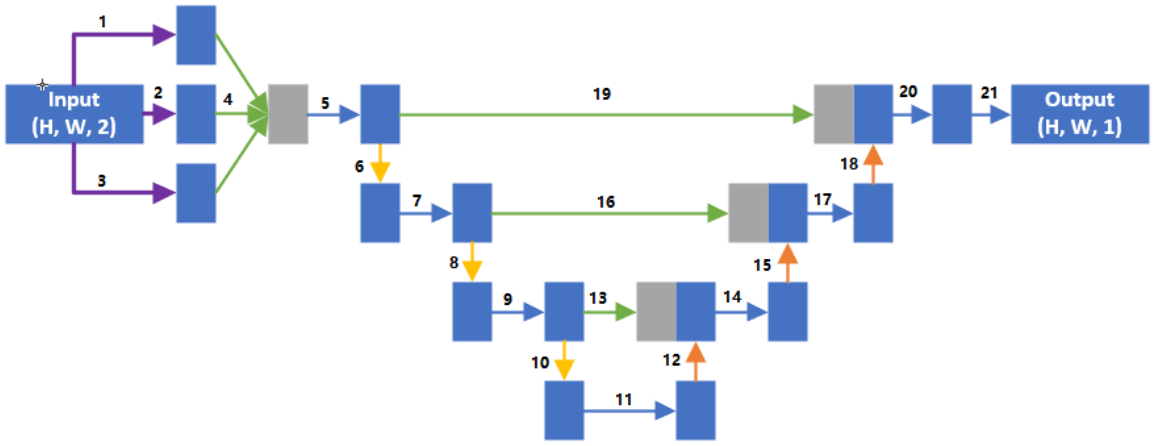


Figure 6: The architecture of the model is shown here. It contains a head, a contraction path and an expansion network. The information of each layers are displayed in Table 1.

#	**Layers**
1	Convolution: 16 filters, kernel size = 3, stride = 1
2	Convolution: 16 filters, kernal size = 3, stride = 1, dilation = 3
3	Convolution: 16 filters, kernal size = 3, stride = 1, dilation = 5
4	Concatenate
5	Convolution: 48 filters, kernal size = 3, stride = 1 Batch Normalization Relu
6, 8, 10	MaxPooling: kernal size = 2, stride = 2
7	Convolution: 32 filters, kernal size = 3, stride = 1 Batch Normalization Relu
9	Convolution: 16 filters, kernal size = 3, stride = 1 Batch Normalization Relu
11	Convolution: 16 filters, kernal size = 3, stride = 1 Batch Normalization Relu
12, 15, 18	UpSampling: kernal size = 2, stride = 1
13, 16, 19	Skip / Concatenate Connection
14	Convolution: 32 filters, kernal size = 3, stride = 1 Batch Normalization Relu
17	Convolution: 48 filters, kernal size = 3, stride = 1 Batch Normalization Relu
20	Convolution: 16 filters, kernal size = 1, stride = 1 Batch Normalization Relu
21	Convolution: 1 filter, kernal size = 3, stride = 1 Sigmoid

Table 1: The information of all layers in the model are shown here.

4. Data collection and augmentation

The datasets used in this paper was collected by two ToF cameras, one is based on Espros EPC660 imager, and the other is based on Sony IMX556 imager. The datasets are a collection of members that each one includes one intensity map acts as background, one intensity map acts as contrast and their actual difference. The resolutions of EPC660 and IMX556 are 320*240 pixels and 640*480 pixels separately. In order to unify the shape of input and improve the performance of model, the images they captured are all reduced to 160*120 pixels.

The datasets contain a total of 560 sets that contains different integration times and multipath distortions, such as a pair of the intensity map without multipath distortions

captured under 1000us integration time and the intensity map with multipath distortions captured under 600us integration time. During the experiment, 448 sets of datasets are used for training and 112 sets of datasets are used for testing.

The datasets contain diverse indoor scenes, such as office rooms, meeting rooms and laboratories. When capturing a set of datasets, the ToF camera was mounted on a fixed bracket for prevent vibrating. After camera was mounted, use an appropriate integration time that will not cause regional overexposure, but can capture clear intensity map to capture an intensity map as background. Then, put an object, named "A", into the scene and repeat the previous step (select another integration time) to capture an intensity map as contrast (multipath distortions can be reproduced if the object is close enough to the camera). Finally, label the actual difference (the object "A") between the previous two intensity maps on a binary image.

Data collection and labeling are laborious. The following method can be used to simulate the multipath distortions and dynamic integration time adjustment on the original datasets for augmenting the training datasets (Figure 7). Firstly, cutout the actual difference between two intensity maps. Secondly, apply a same series of transformations, such as translation, rotation and shear, on both actual difference image and actual different cutout. Thirdly, paste the transformed actual difference cutout on the intensity map acts as background to create a new intensity map to simulate objects at different location. Fourthly, blend the new intensity map and noise image (put several random two-dimensional gaussians with random height, width, μ and σ at different locations) to create another intensity map to simulate different integration times and multipath distortions.

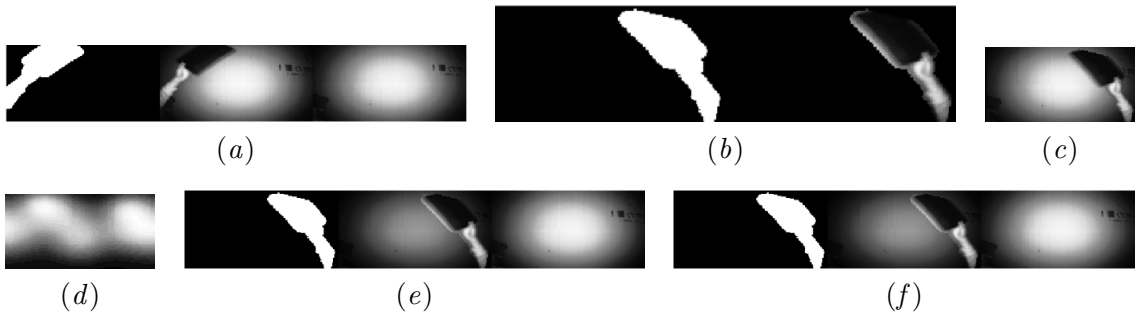


Figure 7: How to simulate the multipath distortions and dynamic integration time adjustment is shown here. (a) Select a set from datasets which contains actual difference, intensity map acts as contrast and intensity map acts as background. (b) Apply a same series of transformations on actual difference and actual different cutout. (c) Paste the transformed actual difference cutout on the intensity map acts as background. (d) Create noise image(put several random two-dimensional gaussians with random height, width, μ and σ at different locations). (e) Blend the intensity map from part (c) and noise image from part (d). (f) Create a new set of datasets by using the previous images.

Methods	**mIOU**
Inter-frame differencing (Threshold >=20)	23.4%
Inter-frame differencing (Threshold >=30)	31.8%
Mog (Threshold >=20)	21.3%
Mog (Threshold >=30)	23.8%
K-Nearest (Threshold >=20)	23.3%
K-Nearest (Threshold >=30)	24.8%
Ours	**82.3%**

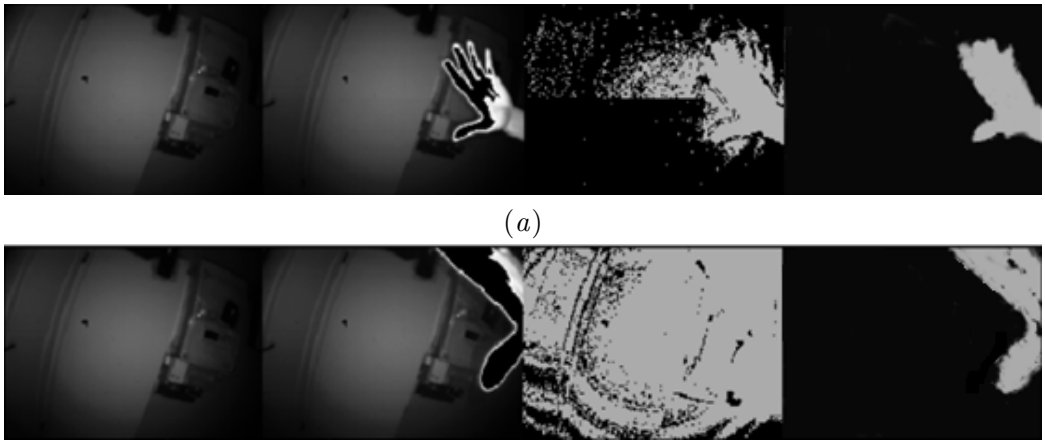
Table 2: The performance of different algorithms on the test datasets are compared by using the metric called mean Intersection Over Union. The inter-frame differencing algorithm requires only one background image to work, therefore the datasets in this paper can be nicely adopted to it. The Mog and K-Nearest algorithm requires multiply images to learn the background, therefore random scaling all the values of entire background image is used to simulate multiple background images.

5. Experiment and result

In the experiment, the error of foreground detection algorithms is quantified by the metric called mean Intersection Over Union (Table 2). The reason why pixel accuracy is not used here is when the area of actual difference between two intensity maps is relatively small, it cannot nicely represent the performance of foreground detection. In the case of using fixed integration time and no multipath distortions existed, the performances of the new algorithm and the traditional foreground detection algorithms are close. In the case of using dynamic integration time and no multipath distortions existed, the performance of the new algorithm dramatically surpasses the traditional one. Besides, another advantage of the new algorithm is no need to set any parameters, such as the threshold of pixel values difference. The traditional algorithms are sensitive to the threshold of pixel values difference which is difficult to pick, therefore, different thresholds are used in the test to detect the foreground.

Performance under multipath distortions: In the case of multipath distortions existed, the intensity map fluctuates greatly. The traditional algorithm treats a large amount of background as the foreground, but the new algorithm can detect the foreground well. The performances of both algorithms can be seen in Figure 8, and the multipath distortions were reproduced by controlling the distance between the hand and the camera.

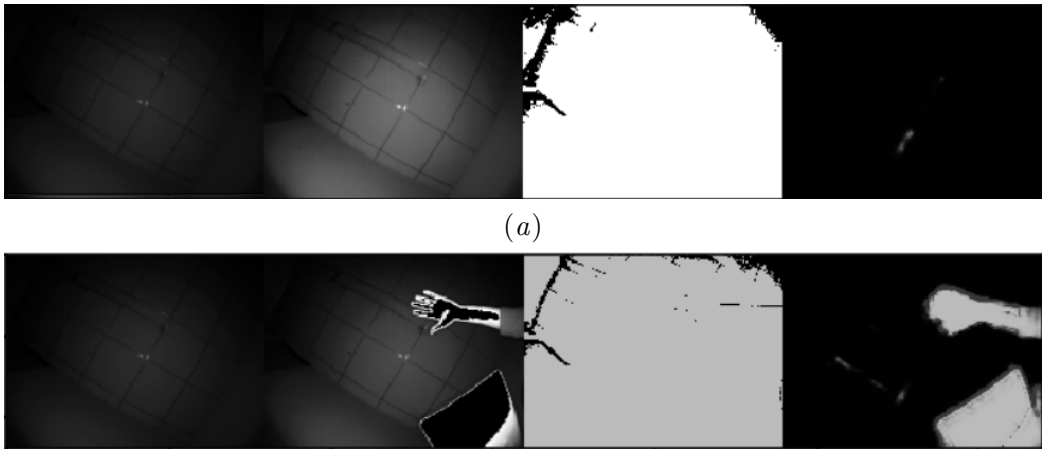
Performance under dynamic integration time adjustment: In the case of dynamically adjusting the integration time, the change of value in intensity map is approximately proportional to the integration time. When the integration time changes greatly, the traditional algorithm will consider most areas of the entire image as foreground, in contrast, the new algorithm can detect the foreground pretty well. Figure 9 demonstrates the performances of both algorithms, 500us integration time and 800us integration time are used separately.



(a)

(b)

Figure 8: The performances of traditional algorithms and new algorithm are compared based on two sets of test datasets. In each part, intensity map acts as background, intensity map acts as contrast, performance of traditional algorithm, performance of new algorithm are displayed sequentially. (a) Performances of traditional algorithms and new algorithm in the case of multipath distortions with light magnitude. (b) Performances of traditional algorithms and new algorithm in the case of multipath distortions with heavy magnitude.



(a)

(b)

Figure 9: The performances of traditional algorithms and new algorithm are compared based on two sets of test datasets. In each part, intensity map acts as background, intensity map acts as contrast, performance of traditional algorithm, performance of new algorithm are displayed sequentially. (a) Performances of traditional algorithms and new algorithm in the case of dynamic integration time adjustment. (b) Performances of traditional algorithms and new algorithm in the case of dynamic integration time adjustment.

6. Discussion and Conclusion

Using neural networks to detect foreground on ToF images is a novel algorithm. Despite the additional computation introduced by neural networks, it is acceptable on mainstream computing platforms, and the input image size can be scaled down to reduce the computation complexity. The new algorithm can be used in many ToF camera-based applications, such as passenger flow statistics, object detection on convey belt. It can also be a sub-module in the machine vision platforms.

In this paper, it shows the feasibility of using neural networks to learn and detect the foreground on the intensity map of ToF, especially, dynamically adjusting the integration time or encountering multipath distortions. In order to realize this learning process, two real ToF cameras are used to collect comprehensive datasets. In the experience, it shows the performance of the new algorithm significantly surpasses the traditional algorithms in the scenes mentioned above. This also gives people who develop ToF based applications a new way of thinking.

References

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. Fusion of geometry and color information for scene segmentation. volume 6, pages 505–521. IEEE, 2012.

Valeria Garro, Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. Edge-preserving interpolation of depth data exploiting color information. *annals of telecommunications-annales des télécommunications*, 68(11-12):597–613, 2013.

Tae-Hoon Hwang and Jin-Heon Kim. A real time low-cost hand gesture control system for interaction with mechanical device. *Journal of IKEEE*, 23(4):1423–1429, 2019.

Hand gesture recognition with leap motion and kinect devices, 2014. IEEE.

Robert Lange and Peter Seitz. Solid-state time-of-flight range camera. *IEEE Journal of quantum electronics*, 37(3):390–397, 2001.

Benjamin Langmann, Seyed E Ghobadi, Klaus Hartmann, and Otmar Loffeld. Multi-modal background subtraction using gaussian mixture models. In *ISPRS Symposium on Photogrammetry Computer Vision and Image Analysis*, pages 61–66, 2010.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Ajmal Shahbaz, Joko Hariyono, and Kang-Hyun Jo. Evaluation of background subtraction algorithms for video surveillance. In *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 1–4. IEEE, 2015.

Ali Shahnewaz and Ajay K Pandey. Color and depth sensing sensor technologies for robotics and machine vision. In *Machine Vision and Navigation*, pages 59–86. Springer, 2020.

Kilho Son, Ming-Yu Liu, and Yuichi Taguchi. Learning to remove multipath distortions in time-of-flight range images for a robotic arm setup. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3390–3397. IEEE, 2016.

Samuel Verdú, Eugenio Ivorra, Antonio J Sánchez, Joel Girón, Jose M Barat, and Raúl Grau. Comparison of tof and sl techniques for in-line measurement of food item volume using animal and vegetable tissues. *Food control*, 33(1):221–226, 2013.

Weihang Wang, Peilin Liu, Rendong Ying, Jun Wang, Jiuchao Qian, Jialu Jia, and Jiefeng Gao. A high-computational efficiency human detection and flow estimation method based on tof measurements. *Sensors*, 19(3):729, 2019.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.