International Journal of
Micro Air
Vehicles

Original Research

International Journal of Micro Air Vehicles
Volume 11: 1–20
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1756829318822327
journals.sagepub.com/home/mav
SAGE

Nonparametric background modelling and segmentation to detect micro air vehicles using RGB-D sensor

Navid Dorudian, Stanislao Lauria and Stephen Swift

Abstract

A novel approach to detect micro air vehicles in GPS-denied environments using an external RGB-D sensor is presented. The nonparametric background subtraction technique incorporating several innovative mechanisms allows the detection of high-speed moving micro air vehicles by combining colour and depth information. The proposed method stores several colour and depth images as models and then compares each pixel from a frame with the stored models to classify the pixel as background or foreground. To adapt to scene changes, once a pixel is classified as background, the system updates the model by finding and substituting the closest pixel to the camera with the current pixel. The background model update presented uses different criteria from existing methods. Additionally, a blind update model is added to adapt to background sudden changes. The proposed architecture is compared with existing techniques using two different micro air vehicles and publicly available datasets. Results showing some improvements over existing methods are discussed.

Keywords

Keywords

GPS-denied environments, dynamic environments, micro air detection, nonparametric background subtraction, background-model update, segmentation

Received 19 May 2017; Revised received 10 September 2018; accepted 27 October 2018

Introduction

In the last decades, the autonomous unmanned aerial vehicles (UAVs) have seen rapid progress. These vehicles are usually controlled and evaluated by external motion tracking system such as the VICON motion tracking system[1–3] or onboards visual sensors.[4–7]

Recently, using external sensors in GPS-denied environments has attracted many researchers.[3,8] In these methods localization of detected UAVs is crucial for collision-free path planning. While different techniques have been proposed for static objects, localization and detecting dynamic objects such as UAVs are still challenging and hard to implement due to the limitations of sensors.

A new evaluation system has been introduced in Baek et al.[8] where they have used RGB-D Kinect sensor for 3D measurements instead of VICON motion capture system which commonly used for verifying algorithm of UAV to control and self-localization in indoor area. For evaluation purpose, they have also applied a marker in order to recognise it. Their recognition algorithm includes two sections. In the first one, Gaussian mixture model (GMM) has been applied as a background subtraction method to find the area of interest.

A filter which adapts the labelling technique is also applied to identify a marker in the region of interest. The feasibility of this approach has been validated in a real experiment of using two kinds of UAVs.

Department of Computer Science, Brunel University, London, UK

Corresponding author:
Navid Dorudian, Department of Computer Science, Brunel University, London, UK.
Email: navid.dorudian@brunel.ac.uk

The authors have demonstrated that position tracking for the horizontal and vertical movement of a quadcopter is possible. However, the authors complain about some issues and limitation of the proposed method such as fluorescent lights, the accuracy of the position tracking and limit of the recognition range of quadcopter's location at distances of 1 to 3 m.

The main purpose of our paper is to address some of these limitations. In particular, the aim of this paper is to investigate the accuracy of the micro air vehicles (MAVs) detection and position tracking in challenging scenarios such as illumination changes in high-speed moving MAVs.

In order to achieve these goals, we have introduced a new object detection method based on motion detection algorithm using colour and depth data to produce the segmentation result. Our proposed method stores several colour and depth images as a model. It then compares each pixel from the new frames with the models in the same pixel location to identify the pixel as part of background or foreground. When the models have been created, they need a regular update to adapt to the changes in the scene.

To perform these updates, once the pixel is found to be part of the background, the system updates the model by finding the closest pixel to the camera and substitutes it with the current pixel if the new pixel is in the same or further location. To the best of the author's knowledge this segmentation method has never been tested before in this way. The approach to update the background model discussed in this paper is different from other classical methods which are updating the sample model with the new frames based on oldest values should replace first, mean or random substitutions.

Additionally, blind update is added to the model, in order for the system to adapt to the sudden changes in the background by updating the background as well as foreground pixels. After a sufficient number of sequences, for each pixel the background model swaps the current frame with one of the samples randomly in the model in the same location regardless of being foreground or background. Then, the proposed method is compared to the other state of the art methods. Results show that it is more accurate in object boundaries and it can tolerate more illumination changes.

Motion detection

The capability of motion detection is one of the most fundamental tasks in many computer vision applications, especially for dealing with automated visual surveillance and object tracking in real-time applications. By defining the recognised detection area as the region of interest (ROI), it will lead to additional tasks such as people counting,[9] wild-life and traffic monitoring,[10] robots localization and tracking[8] or safe UAVs navigation.[3]

The main goal of such an approach is to recognise foreground (moving object) that do not belong to the scene. One of the most popular method is comparing the current frame with previous frames. These previous frames are known "reference" in the literature. This reference typically is made from a single image or more complex model which is called scene model.[11] A scene model needs a regular update to adapt to the change of real-world practical conditions.

Generally motion detection methods can be divided into different categories such as optical flow,[12,13] cluster analysis,[14] median filtering,[15] running average,[16] frame differencing[17] and background subtraction.[18] Among them the last two are currently the most common methods.[19] On the other hand, statistical background models which have been widely used in object detection can be divided into different categories. These models are typically based on multimodal such as GMM,[20] mean-shift clustering,[21] hidden Markov models,[22] non-parametric kernel density estimation[23] or uni-model such as Gaussian[24] and Chi-square distribution.[25]

Background subtraction methods are typically based on a static background hypothesis. Often it has been assumed that in indoor environment, the scene does not have a periodic dynamic background. However, in practical scenarios, many situations could lead to background changes such as reflections, animated images on screen, moving curtains or chandelier by winds.

The existing states of the art background subtraction techniques have achieved significant success in many applications. However, these techniques only perform well under steady conditions and can lead to the failure in case of sudden illumination changes (etc. change of light), fast moving objects in the background (e.g. moving curtains) and changes in background objects (e.g. moving a table from one place to another).

Many object detection algorithm have been proposed to solve the problems by illumination changes.[19,26–28] These methods typically have a training stage after the changes and they are usually expensive in terms of computation.

A possible solution to reduce the impact of the previously mentioned phenomena could consist of using physical information of the scene. For instance, geometrical descriptions of buildings have been added to the model in order to assist to predict shadows.[29] We can obtain these 3D information of the scene from stereo devices, camera networks[30] and RGB-D cameras.

Currently, the production of low cost RGB-D cameras such as the Asus's Xtion Pro or the Microsoft's Kinect is totally changing the computer vision world. Many researchers are using these devices which can capture depth and colour images in the same time at frame rates of up to 30 fps which is widely available on the market. Depth data are very attractive and appropriate for applications based on moving object detection.

In the last few years many researchers have been investigating toward the use of depth data and colour information in video surveillance to segment background of the scene.[11,31–39]

The shapes of objects which are captured by depth sensor in the scene are not affected by shadows, illumination changes and interreflections. Therefore, depth information could help to provide much more robustness to such a phenomenon. However, background subtraction methods based on only depth data frequently produce invalid outcomes.[40,41] Depth data are usually noisy and have some restrictions for certain surfaces in measurement which typically is referred to as "holes"[31] or "Absent Depth Observations (ADO)" in the literature.[11] These failures come from several physical phenomena such as the production of depth camouflage, depth shadows, absorption by black objects, limitation on distances, lower sensitivity at longer distances and absent observations, etc. Figure 1 illustrates the amount of possible noise in each depth frame, for example the black speaker absorbs the signal and consequently the area is defined as absent of observation (shown by black points) or in some part of the cavity, depth is not available due to the characteristics of the scene. Since the depth frame is smaller than the colour frame, the black pixels on the edge of the picture are part of the outer boundary of the depth frame. Moreover, some points on the side of the frame reached to the maximum length of the sensor therefore the sensor is not able to return any value for those pixels. Therefore, we have introduced both colour and depth measurement in our approach to cover each other weakness in some challenging situations.

Despite all previous researches, the problem of moving object detection remains challenging and there is no universal technique to cover all practical scenarios which could detect the foreground of the scene without any noise. This motivated us to find a practical approach to combine the colour and depth data to obtain more precise and reliable background subtraction.

In this paper, a new object detection method based on background subtraction algorithm using colour and depth data is proposed. This method creates an individual model for colour images and another one for depth. Then by storing the previously observed pixels in these models, the system identifies each new pixel as foreground or background by comparing them. The models will be updated regularly after identifying the pixels as a by finding the smallest sample in pixel location within the depth model and swap it with value of the new pixel. In this way, the system will be able to adapt to great changes in the background scene by blind update which randomly swaps the pixels from the new frame with the model regardless of being foreground or background.

This method, when compared to the other state of the art algorithm, can tolerate more illumination changes, it is more accurate in general and around object boundaries. We will explain in more detail about our motion detection algorithm in the section Background segmentation and comparison results with other state of the art techniques in the section Results.

Related work

Camplani and Salgado[39] proposed a per-pixel background modelling method which combine different statistical classifiers based on colour and depth information which improves the background subtraction. For each pixel, the output from the mixture of the

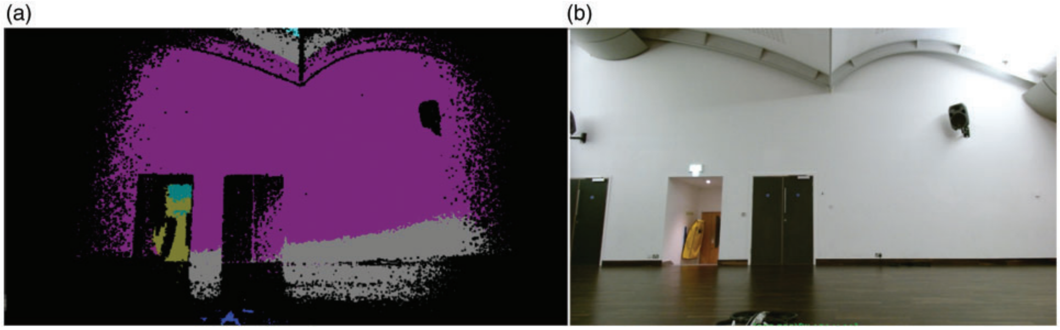

Figure 1. (a) Black points in image shows holes (ADO) and other colour coded shows the distance in depth frame, (b) Colour image.

two classifiers is gained through a weighted average to consider the characteristics of colour and depth information.

The final classification is measured by the edges of depth and colour images and the previous foreground segmentation. They have used canny algorithm for edge detection. The colour-based classifier has a larger impact on segmentation of those pixels at object boundaries. This will reduce the noise of depth measurements neighbouring object borders. On the other hand, the depth-based classifier has more impact on the final segmentation in low gradient pixels. They believed that depth map ensures solid detected foreground regions and decrease the amount of errors due to illumination changes and shadows. The background model uses a mixture of Gaussian distributions. Camplani et al.[42] introduced another method based on the combination of several region-based classifiers. The authors believe that this approach cannot perform well in case of very fast moving object as this method is using the previous detections to recognise the areas where the foreground can occur. However, this hypothesis is not always correct.

Pixel-Based Adaptive Segmenter (PBAS) was proposed in Hofmann et al.[43] This nonparametric approach models the background by using the history of recently observed pixel values. The main mechanism of PBAS is the decision block. This decision is found on the per-pixel threshold for or against foreground based on background model and the current colour frame.

Moreover, the background model is updated time to time to be able to deal with the steady background changes. This update process relies on per-pixel learning parameter. The main novelty used in PBAS algorithm is that per-pixel thresholds metric modifies the estimate of the background dynamics.

Generic scene modelling (GSM) is a nonparametric method that uses both depth and colour information which is proposed in Moyà-Alcover et al.[11] Background model constructed using a kernel density estimation (KDE) process with a Gaussian kernel for each pixel of the scene. Unlike GMM model, in KDE no mixture parameters should be estimated. This helped them to estimate the density function without any assumption about density model. Consequently, it depends only on recent information of the scene.

A 3D kernel is constructed with one dimension for a depth data model and two for normalized chromaticity coordinates. Update model phase is performed using a first-in first-out in the queue. This means a new sample is added to the model and the oldest sample is discarded.

Recently, a new promising sample-based segmentation method is proposed for background subtraction called ViBe.[18] This method builds the model by collecting previously observed values for each pixel location. By having update phase in the processing stage, it can respond to the change in the background very fast by adding newly observed pixels directly in the models. The original ViBe demonstrated successful accuracy in many real-world scenarios such as dynamic backgrounds as well as being fast and simple to implement.

ViBe algorithm is fast and efficient which is widely used in background subtraction for moving object detection. On the other hand the original ViBe algorithm could easily produce ghost in the process of moving object detection.[44] Ghost described in Cucchiara et al.[45] as "a set of connected points detected as in motion by means of background subtraction, but not corresponding to any real moving object."

However, ViBe algorithm still suffers from some limitations in several challenging scenarios which can totally affect the outcome of ViBe algorithm such as sudden illumination changes, darker backgrounds, ghost and shadow production in frequent background changes.[46] This can lead to wrong classification of pixels and therefore to object detection failure.

In order to remove the ghost area in the process of foreground detection, different modified versions of ViBe algorithm have been introduced by other authors. For instance, Bo et al.[44] improved ViBe algorithm based on the theory that the histogram distribution characteristics of moving objects are different when a real object is moving. However, the histogram of ghost areas has a correspondence distribution characteristic.

According to Nyan and Grünwedel,[47] ViBe algorithm fails to detect object of interest when the lighting of the room is reduced by about half. In the same condition GMM and edge-based method could still detect it with more false positive. Once the light is off, the detection of both ViBe and GMM become very unreliable. Although their proposed method is able to detect in this condition, the performance still is poor.

Leens et al.[33] proposed a new ViBe approach which is using colour image and ToF (Time-of-Flight) sensors which is called indoor PMD (Photonic Mixer Device camera). Each model is created independently and then with logical operations combined the foreground masks. Segmentation results proved that the colour and depth are able to cover their limitations. For instance, depth contribution is important in the areas that colour segmentation typically fails. This is in case of illumination changes or when the colour of the object is identical to the backgrounds. On the other hand, when the object is very close to the background or the depth frame is too noisy to produce, a colour segmentation can produce a valid background mask.

However, mixture of colour and depth segmentation consists of sensor and RGB camera has couple of drawbacks: false detection in the persistence of fast movement by object and appearance of infrared shadows made by sensor. Solving these problems was the main goal of Pierard and Van Droogenbroeck[48] who tried to successfully improve the ViBe algorithm using colour and depth. Despite the decent outcomes, the authors complained about the problematic alignment in this system between the PMD and RGB cameras.

Shadow can reduce object detection rate and lead to rise the likelihood of tracking failure, which are very important measures of benchmarks in object detection-based system. In the last few years many shadow removal approaches have been introduced to improve detection ratio of state of the arts algorithms by using gradient amendment, edge-based, histogram, etc.[49–51] These methods usually face with some complex situation such as lack of information in darkness or low light condition and completely change of chromatic properties of the context.

Regardless of all previous researches, results show that shadow detection algorithms improved the average of the shadow detection rate. However, the rate of the detection still cannot completely meet the practical requirements. Major problem of existing shadow detection algorithms are the fracture of detected objects, particularly for pixel-based methods such as texture-based and chromaticity-based methods.[51]

In benchmark evaluation with other state of the art algorithms, ViBe proved that is robust to background motion and artefacts stemming from irregular motion (camera jitter).[52]

All previously mentioned researches have shown that nonparametric methods such as ViBe and PBAS have a successful accuracy in most cases except a few scenarios such as sudden illumination changes, poor lighting, and production of ghost. By having the physical information of the area, we are able to cover the weakness of these methods and significantly improve the overall accuracy of the moving object detection of the nonparametric methods. For this reason, we have added depth frame to RGB image in order to improve the outcomes.

Two main approaches exist to segment RGB-D data. In the first approach two independents segmentations are carried out. One on the colour image and the other one on the depth data, the two results are merged. The second approach fuses the RGB-D data before undertaking a joint segmentation.[36] We have used the second approach in our algorithm. This means we are considering jointly depth and colour to produce the segmentation.

For more clarification, we have included a complete version of our object detection algorithm in a C-like code in Appendix 1.

Background segmentation

Our proposed method follows a nonparametric background modelling pattern, similar to the previous works such as ViBe[18] and PBAS.[43] Consequently, the background model obtains by history of previously observed pixel values and the foreground segmentation depends on a threshold amount.

Using nonparametric methods such as ViBe algorithm with both colour and depth data is not totally new since this approach has been already applied for moving object detection.[33,36,48] However, in Leens et al.[33] and Pierard and Van Droogenbroeck[48] their vision system is made up with RGB camera and separate ToF camera. These systems need experimental calibration and align the both frames which are heavy and difficult. Instead some authors like Ottonelli et al.[36] used a simpler way of having only a standard stereo camera.

Recently with the rise of low-cost RGB-D camera researchers have started to use these sensors as they are able to produce better calibrated RGB and depth frames. These devices can capture up to 30 frames per second which would be beneficial for motion detection algorithms. These great benefits encourage us to start using RGB-D camera for the proposed background subtraction technique. Our method consists of different steps to be able to successfully use in live application and cope with the changes in the background. Figure 2 illustrates the flow chart of the proposed algorithm.

The system stores the first N number of frames in "system initialization" step to create colour and depth background models. "ADO removal filter" will apply to individual depth frames before going to the model to eliminate all the unknown values in the depth frame. Once the initialization has been completed, the background model will be ready and the system moves to the main loop. Each pixel of the new frame will be compared with the models to identify as foreground or background (Bg/Fg segmentation).

Those pixels identified as a background will be guided to update the background models. Additionally, after M number of frames, the system will use blind update to randomly swap foreground as well as the background pixels with the models. In the remaining of this section the key steps of Figure 2 are discussed in more detail.

System initialization

Background subtraction methods usually need a scene model to enable the system to compare and segment the regions of the new frames as a background or foreground. Meanwhile every model requires an

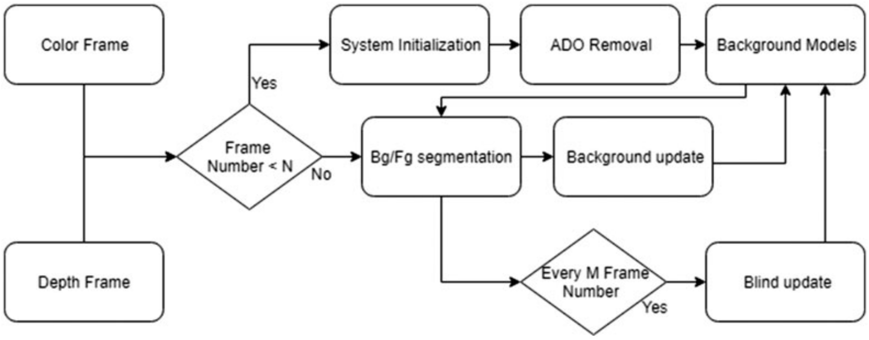

Figure 2. Flow chart of the proposed object detection method.

initialization process which has enhanced the importance of numerous popular methods described in publications, such as Elgammal et al.[23] which need various frames to initialize their model. These approaches are acceptable in terms of statistical point of view. Therefore, this can gather various amount of data which enables us to estimate the temporal distribution of the background pixels. However, these methods are not able to segment the foreground of video that is shorter than the normal initialization sequence needed by some background subtraction methods. On the other hand, other methods such as Barnich and Droogenbroeck[18] need plenty of time to complete the stored model.

The ability to provide an uninterrupted foreground detection is one of the most important factors in our application. This includes the sudden changes in light or shadow of the moving object on the wall, which cannot appropriately be handled by the regular initialization and update approach.

The possible answer to these issues could be introduced as an outstanding update model process which adapts the pixel models to the different lighting conditions. However, sudden illumination could completely change the chromatic properties of the context and even using such a dedicated update process could fail.

Barnich and Droogenbroeck[18] introduced an appropriate technique for this issue which initializes the background model from single frame and gradually building more samples model. Even this technique is not able to cope with sudden illumination changes such as shadow of moving object. A more convenient solution to these issues is to use depth images which help us to understand a change in the physical position of each pixel in the real world. Therefore, in order for the system to be able to handle the sudden illumination changes, depth data added to the RGB in the proposed method.

Depth information is supposed to represent steady long-term description of the scene. Therefore, theoretically storing one model of the scene should be enough for the background model. However, we experienced that cheap sensors like Microsoft Kinect have a considerable amount of noise. In order to find the most accurate depth measure, we store the same number of depth as colour frame.

Unlike other approaches which need plenty of time and frame for initialization, our method required to finish initialization very quick and start tracking the moving object as soon as possible; therefore, the system blindly stores the first N number of colour and depth images (we recommend $N = 20$ samples) as a model and then gradually modifies the model during the update stage. This will allow us to start tracking our object rapidly.

Post initialization filtering (ADO removal)

As mentioned before, depth data could be very noisy and contain many ADO pixels. In order to reduce these noises, we need a hole filling strategy. The main goal of this strategy is to filter the unknown depth value and refining object boundaries. Each filtered frame is then used as a sample to build the depth model which helps to prevent with falling of the temporal variations of the depth distances. This will help to have more accurate depth pixel values in the model to identify the foreground and background.

Recently many researchers have been investigating *inpainting* depth frames to remove holes from the depth frames.[53,54] However, these methods are very expensive in terms of computation. One of the most common methods to remove the ADO pixels is to fill them by the neighbouring depth data.[54]

We have used this idea and made an assumption that neighbouring depth pixels are most likely to have similar value. We have used this assumption to remove the ADO pixels by replacing with the randomly nearest pixel values.

This simple and fast method will help to significantly reduce the number of errors. However, this method could also lead to more error in rare scenarios such

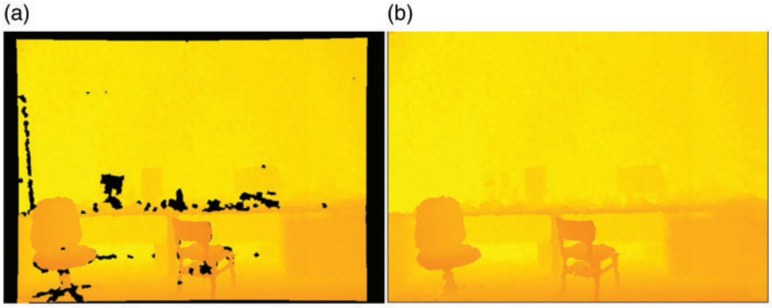

Figure 3. An example of depth image. Black pixels show holes in depth frame, (a) depth frame before ADO removal, (b) post initialization after ADO removal.

as an open area such as a corridor with a length of more than the sensor maximum range. However, these wrong values will be gradually corrected with more accurate values during the update process. Figure 3 illustrates an example of depth sample model before and after ADO removal.

Bg/Fg segmentation

Traditionally background subtraction techniques mainly rely on probability density function (PDF) or statistical parameters such as variance or the mean.

An alternative way is to consider statistical significance to build a model with previously observed real depth and colour data. This assumption is based on common sense that if the same pixel value has been observed many times in the same location, this pixel has a high probability of being background, compared to the values that never come across.

As part of our background subtraction, we want to classify each pixel as foreground or background. In order to do this, we are fusing the results from colour and depth models to produce the final decision.

Like Barnich and Droogenbroeck,[18] we create each background pixel with a set of samples instead of one background model. Accordingly, we have not used estimation of the PDF for the background classification. Instead in each location the current value of the colour pixel is compared to the collection of samples (colour model) to find out if the pixel value is close to some of the sample values instead of most of all samples in the same location.

In a similar way, depth pixels will be compared to depth model to check if the pixel has been in the same range or is closer to the camera.

In most cases depth and RGB have the same individual segmentation outcome. In other words, both separately agree whether the pixel is part of the background or not. However, in some challenging scenarios, they are strongly against each other. An example of these situations could be colour camouflage such as foreground having the same colour of the background or depth camouflage such as moving the hand on the wall.

In order to make the final decision we need to rely on colour or depth model, one more than the other. Recently with the production of new sensors such as ToF which has been used in Kinect V2 sensor, depth accuracy has been improved significantly.[55,56] On the other hand, illumination does not affect depth data. Therefore, we have relied more on the depth outcome to produce the result. This means if we could not find enough close samples in the depth model, then pixel will be classified as foreground regardless of colour outcome. In other words, if depth pixel is not available in any pixels, then we will only rely on the decision of the colour model on that pixel location.

All non-ADO pixels will be accepted as a background if they have some similarity with depth model. In the same way, each pixel can classify as foreground if they do not have some similarity with the depth model (by considering the tolerance amount). All other pixels will be decided by colour model.

In other words, if a pixel has close or greater distance to some of the depth sample values, it will be classified as a background. The main reason we added this condition is to detect shadows and colour camouflage as part of the background. An example of this is illustrated in Figure 4.

Those pixels which have not been assigned as a background will be then compared with the values of the depth again. However, this time the threshold will be increased. If the pixel cannot meet this condition, it will be considered as a foreground. All other pixels will be decided in the same way with colour model. Consequently, if they have some similarity with colour model, then we will classify them as a background, otherwise those pixels will be classified as a foreground. Figure 5 illustrates the proposed classification in flow chart diagram.

Formally, let us denote a 3D point as $X=(x, y, z) \in \mathrm{R}^3$, RGB-D camera produces a colour and depth

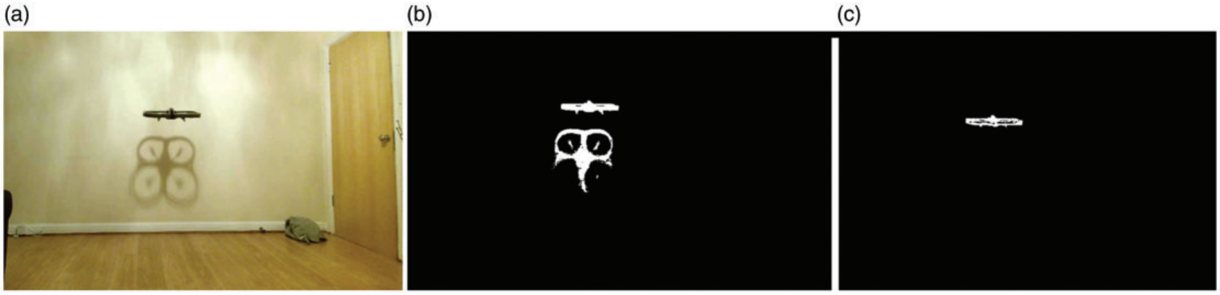

Figure 4. An example of shadow. (a) Colour image, (b) original vibe, (c) proposed method.

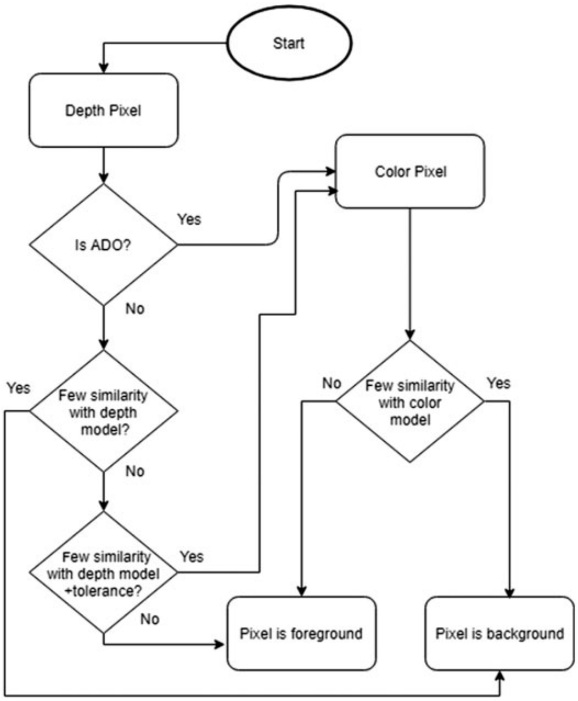
Start
Depth Pixel
Is ADO?
Yes
No
Color Pixel
Few similarity with depth model?
Yes
No
Few similarity with color model
No
Yes
Few similarity with depth model +tolerance?
Yes
No
Pixel is foreground
Pixel is background

Figure 5. Flow chart of the proposed classification method.

image. We denote $v(X)$ the value in a given colour and $d(X)$ the value in the depth taken by the pixel located at X in the new image frame and with an index of i in a background sample value of v_i and d_i. Each background pixel located at X is modelled by a collection of N background colour and depth sample values taken before as

$$M(X)_{\text{RGB}} = v_1, v_2, v_3, \ldots, v_n\} \quad (1)$$

$$M(X)_D = \{d_1, d_2, d_3, \ldots, d_n\} \quad (2)$$

In this paper, we refer to $M(X)_{\text{RGB}}$ as a background colour model and $M(X)_D$ as a background depth model. In order to classify each pixel of new frame as a background, we compare each depth pixel with the depth model $M(X)_D$ at location X. If the difference is equal or larger than h_D (acceptable depth threshold which is close to 0), we will count as the pixel is similar to that sample. Each pixel which could find more than cardinality denoted by $\#_{\text{Min}}$ (we recommend the value as $N/4$) similar pixels will be assigned as part of the background. On the other hand, if we could not find at least $\#_{\text{Min}}$ similar sample out of N number of samples at location X, the system will increase the h_D and do the last process again. This time if the system could not find $\#_{\text{Min}}$ similar sample, it will be count as foreground. All other pixels will be decided by comparing the colour values and $M(X)_{\text{RGB}}$ in the same way.

Background model update

In this section, we will explain how to continuously update the background model with the new frames over the time. The reason we have added this stage is that the system adapts to the changes in the background over the time. These changes could be illumination changes, appearing a new object in the scene or moving an object in the background completely to the different position.

When a pixel classified as a background, the system will randomly swap the value of new colour pixel with one of the samples in colour model in the same location. However, in depth model we will check the distance of the new pixel (in depth image) with background model distances. In order to compare the distance, the system finds the smallest sample in the depth model and compares it with the value of new depth frame in the same location. If it has the same distance or bigger, we see it as a good sample and will swap it with the smallest previous sample, otherwise it is a bad sample and will not change the samples model.

Figure 6 demonstrates an example of good and bad sample pixel. We have defined the good and bad samples based on the fact that, if we assume points A and B are different sample of background depth model in the

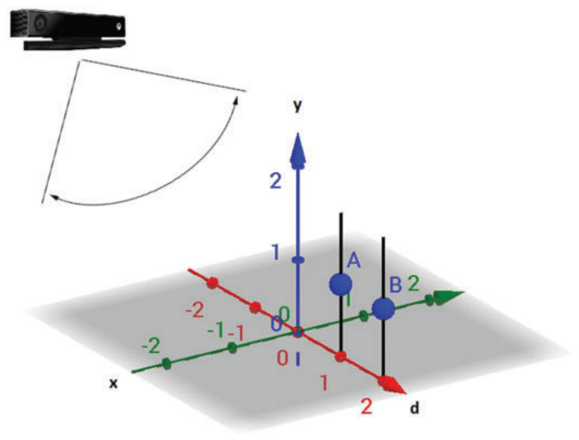

Figure 6. Point A and B are two points in the background depth model which they have identical X and Y with the different d. Point A is closer to the camera therefore it cannot be the background because another point B observed behind this point in this location.

same location, point A will be a bad sample as previously point B has been observed behind this point. This means in comparison of these two points with the same location of x and y with different dimensions (d) in a 3D space, the point with the absolute smaller number is closer to the camera; therefore, it cannot be part of the background. An exception of this rule could be the physical change in the background of the scene. An example of this could be moving forward a table in the background. This means some of the pixel values which belong to the model (it is part of the previous frames) do not exist anymore. However, in this case the system will be able to cope with the changes from time to time in the blind update stage.

Formally, when pixel $v(X)$ is identified as a background, the system will swap the $v(X)$ with randomly one of the colour model $M(X)_{\mathrm{RGB}}$.On the other side, the system finds the smallest distance value (d_{smallest}) in $M(X)_D$ and compare it with $d(X)$. If $d(X)$ is bigger than $d_{\mathrm{smallest}}(X)$, then we can accept that as a good background pixel and replace it with $d_{\mathrm{smallest}}(X)$, otherwise it is not a good sample and we will not change our model. This is one of the biggest differences with the current available methods which those are usually modify the background sample according to old replace with the new one, mean or random number. This enables us to improve our model and update with the new changes during the time as well as keeping the valid samples in the models. The only disadvantage of this selection is that if we move the background forward such as moving a table in the middle of the room, the system will always identify this as a foreground and will not change the model. For this reason, we have added a blind updated step into our algorithm which will allow the system to adapt with such changes in the background during the time.

Blind background model update

In this section, we will explain how to continuously update the background model with the new frames over the time. The reason we have added this step is that the system adapts to the changes in the background over the time. These changes could be appearing a new object in the scene or moving an object in the background completely to the different position.

In the classification step of our method, we update our background model by comparing the pixel of the new frame with the background model and replacing this with the new pixels if they are more valid. However, this will only allow us to replace those pixels which have already been identified as a background. Consequently, if we introduce a new object in the scene (as part of the background), because it has a smaller distance from the camera compared to all previous pixels in depth model and different colour to the colour model, it will never be recognised as a background. Therefore, it will never be part of the background samples.

The term which referred in the literature as background history or background memory has always raised a question in subtraction techniques that which sample we can keep in the model and for how long we can use that. For instance, one of the classical approaches for updating the background model is discarding the old pixel model and replacing with the new pixel after period of time or number of given frame (usually after couple of frame or seconds). These classical methods will update all the old pixels in the model where it is not always necessary to update the valid samples.

On the other hand, updating the model only by those pixels which identified as a background or including foreground pixels is always raised in background subtraction algorithms. In the literature, it has been described as a blind and conservative update procedure. A conservative approach only updates the model by pixels which are identified as a background and it never uses the pixel which belongs to the foreground. Conservative update could cause the background pixels being updated only and have a permanent misclassification. Most of the practical scenarios could reach to this situation.

Conservative approach can successfully detect the moving objects which do not have any similarity with the background. This is used in our background update stage (as illustrated in Figure 2). However, this can contribute to the creation of ghosts and failure in dynamic background scenarios.

Despite all the effort made by existing approaches, developing a fast approach to eliminate the ghost in dynamic background situations is still challenging for background detection techniques. For these reasons, as illustrated in the diagram in Figure 2, we have added the simple random background update phase for the colour and depth models which is called blind update.

Blind update will allow us to use any kind of pixel whether it is classified as a background or foreground and classify it as a background or foreground. The main downside of this method is the poor detection of slow moving object which are becoming part of the background model during the time. Several solutions have been introduced to solve this issue such as using background model of large size or first-in first-out which has been used. However, these solutions have negative sides such as higher computational and memory usage or time limiting.

Those pixels classified as part of the background in the scene, automatically will be used to update the background model. The method will swap the pixel from the new frame with the shortest in depth model if these pixels have better values (longer distance compared to the model). However, if our background will be dynamic, the system will permanently identify the background as part of the foreground. For each pixel, the system will swap the value of current depth (only if its non-ADO) and colour frame randomly from the model after M number of frames (we recommend this value as 30). This method has the advantage of a memoryless update strategy, producing a fast and efficient update. Moreover, a random sampling increases the time gaps and allows the adaptation of the background models that are classified as foreground.

Results

In this section, the results achieved by the proposed method are compared with alternative background/ foreground subtraction algorithms based on colour and depth data. We have tested the presented system in two different ways. First, we have evaluated the proposed moving object detection method with two datasets and then the entire system is tested via a live demonstration in indoor environment.

We have used two different indoor benchmark datasets. The first dataset contains sequences from two different types of MAVs. In the first sequence we have evaluated the detection accuracy using an AR. Drone[57] and in the second sequence we used a Crazyflie,[58] a smaller size quadcopter. To collect these two sequences, a Microsoft RGB-D Kinect V2 sensor has been used. The goal of this test is to measure the ability of the proposed method to detect a small and fast moving object such as the micro drones under different indoor challenging scenarios as detailed in Table 1. We have also generated hand-labelled ground truth for these sequences to measure the accuracy of each method used in the comparisons. In particular, we have compared the proposed method in this paper with $\mathrm{CL_W}$,[39] $\mathrm{MOG_{RGB-D}}$,[32] $\mathrm{GSM_{UF}}$ and $\mathrm{GSM_{UB}}$,[11] PBAS[43] and $\mathrm{ViBe_{bin}}$.[33]

It is worth mentioning that the original PBAS are using only colour frames. In this paper, these have been extended to use colour and depth (RGB-D) images in order to enable us to have same input for all methods. This has been done similar to Leens et al.[33] by fusing the result of colour and depth binary mask using a logical "OR" (non-exclusive). We refer to these methods as PBAS_{bin}.

We should state that all results for the proposed algorithm have been evaluated without using any post-filtering to compare the accuracy of the method. Clearly, the amount of noise will be reduced and the results will improve with post-filtering methods. For qualitative evaluation, a video is available on Dorudian[59] to show the accuracy of the proposed method in some challenging scenarios such as change in the background, removed object from the background (intermittent motion), change in the light and sunlight (illumination changes), micro UAV and appearance of shadow in wall and floor.

Additionally, we have tested the proposed method with the benchmark RGB-D dataset introduced in Camplani and Salgado[39] to compare and rank the algorithms to ensure that our proposed algorithm performs well among other currently available methods in different challenging scenarios. We have used the ground truth provided with these datasets in order to measure the performances. This dataset has four different sequences and each sequence has been made to test

Table 1. Details of our dataset which used for measuring the accuracy of the UAVs detection at 30 fps.

Sequence name	Number of frames	Frequency of frames used for ground truth	Number of ground truth	Number of frame where moving object is present	Objective
AR.Drone	350	Every 30 frames	12	220	Accuracy of UAV detection
Crazyflie	275	Every 30 frames	10	230	Accuracy of small UAV detection

the accuracy of the method in specific challenging scenario. DCamSeq and ColCamSeq ground truth has been produced to test the accuracy of individual method only in those sections in the images where each single problem is existing. This process guarantees that other challenging scenarios do not interrupt the algorithms segmentation. Table 2 shows the details of these sequences.

We have used the following metrics to measure the performance of the proposed algorithm in order to be able to compare and rank the results.

False Positive (FP): Part of the Bg pixels which are classified as Fg.

False Negative (FN): Part of Fg pixels which are classified as Bg.

Total Error (TE): The full number of misclassified Bg/Fg pixels which normalized according to the image size.

Similarity measure (S): Is non-linear metric that combine FN and FP which publicly known as Jaccard's index[60] and has been used in Li et al.[61] as

$$S(A, B) = \frac{|\mathrm{A}\ \cap \mathrm{B}|}{|A\ \cup \mathrm{B}|} \tag{3}$$

where A denoted as detected region and B is ground truth. Result closer to 1 shows Fg correctly identified similar to the ground truth, otherwise will be closer to 0.**Similarity measure (S_B):** To investigate the misclassified pixels near to the boundaries of moving objects. It is measured similar to S, but only considering the regions of 10 pixels surrounding the ground truth boundaries.

Additionally, we used the proposed evaluation method in Goyette et al.[52] to calculate the average ranking of method (RM) which combine the performance of each method across different metrics in each sequence and use overall ranking across category (RC) which shows in general how well an algorithm performs with respect to the other techniques by calculating an average (RM).

Let us denote the ranking of the ith technique for the metric m in the sequence sq as $rank_i(m, sq)$. Then the average ranking of technique i for the N_m number of metrics in the sequence sq is given as

$$RM_i = \frac{1}{N_m}\sum_{i=1}^{m} rank_i(m, sq) \tag{4}$$

Accordingly, the overall ranking among all the categories (RC_i) for N_i number of techniques is calculating by taking the mean of average ranking across all the sequence as

$$RC_i = \frac{1}{N_{sq}}\ \sum_{i=1}^{sq} RM_i \tag{5}$$

N_{sq} defined as the number of sequences which is 4 in the dataset demonstrated in Table 2. In general RM, RC, TE, FP and FN the lower amount demonstrate better performance, and higher S and S_B demonstrate more similarity with ground truth and therefore better performance.

Table 3 shows the result of the Crazyflies sequence. In this scenario, the moving object (Crazyflie) is fast and small. This will cause the sensor to frequently capture unmatched colour and depth images. This will make it more difficult for the tested algorithms to find the correct moving object. Additionally, some part of the UAV has an unknown pixel values in depth frames. For these reasons, all tested methods have a weak performance in this sequence. Figure 7 shows an example of this sequence and the binary mask of each method.

Table 3 shows that the proposed algorithm could obtain the lowest TE and FP. This shows that the system has less fault detection among other algorithms. However, the FN for the proposed method is very high which means the system could not successfully detect part of the foreground. Other methods also have the same problem except Pbas$_{\text{bin}}$ where instead, it has weak results in FP and S. On the other hand, the proposed method could achieve the highest similarity measure (S) which shows the closest result to the ground truth. Average ranking of the proposed method (RM) is the

Table 2. Details of dataset in Camplani and Salgado,[39] which is used for evaluation in this study.

Sequence name	Number of frames	Frequency of frames used for ground truth	Number of ground truth	Number of frame where moving object is present	Objective	Papers which also used these datasets
GenSeq	300	Every 8 frames	39	115	Overall performance	[11,38]
DCamSeq	670	Every 7 frames	102	400	Depth camouflage	
ColCamSeq	360	Every 8 frames	45	240	Colour camouflage	
ShSeq	250	Every 10 frames	25	120	Shadows impact	

Table 3. Crazyflies sequence results.

	TE		FN		FP		S		SB		
Method	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	RM
MOG$_{RGB-D}$	0.63	0.17	51.25	13.56	0.02	0.02	0.34	0.17	0.37	0.14	3.2
GSM$_{UB}$	0.08	0.01	55.96	23.49	0.03	0.03	0.27	0.13	0.29	0.09	3.8
GSM$_{UF}$	0.12	0.27	37.59	18.25	0.09	0.03	0.27	0.11	0.34	0.08	3.4
PBAS$_{Bin}$	0.63	0.06	0.20	0.57	0.63	0.06	0.11	0.04	0.44	0.07	3.2
VIBE$_{Bin}$	1.45	0.19	19.12	10.71	1.43	0.19	0.04	0.02	0.42	0.06	3.8
Proposed method	0.05	0.01	42.63	17.85	0.01	0.02	0.42	0.18	0.42	0.11	1.8

Lower TE, FN and FP show better result and higher S and S$_B$ demonstrate higher similarity to the ground truth.
FP: false positives; FN: false negatives; TE: total error; S: similarity measure; S$_B$: similarity measure in object boundaries.

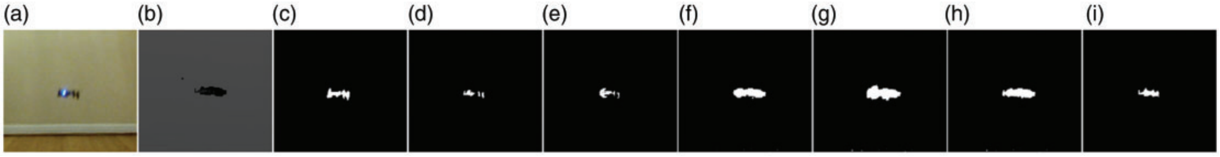

Figure 7. The result of micro UAV sequence. (a) Colour frame, (b) Depth frame, (c) Ground truth, (d) MOG$_{RGB-D}$ output, (e) GSM$_{UB}$ output, (f) GSM$_{UF}$ output, (g) PBAS$_{bin}$ output, (h) ViBe$_{bin}$ output, (i) Proposed method output.

Table 4. AR.Drone sequence results.

	TE		FN		FP		*S*		*SB*		
Method	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	RM
MOG$_{RGB-D}$	0.15	0.16	14.79	12.02	0.04	0.06	0.79	0.13	0.80	0.11	2.6
GSM$_{UB}$	0.49	0.29	74.97	11.66	0.01	0.01	0.25	0.12	0.26	0.12	4.2
GSM$_{UF}$	0.31	0.26	9.59	6.52	0.25	0.21	0.69	0.11	0.74	0.08	3.2
PBAS$_{Bin}$	1.25	0.33	0.25	0.80	1.25	0.33	0.33	0.09	0.66	0.10	4.2
VIBE$_{Bin}$	1.02	0.26	2.73	3.41	1.18	0.25	0.33	0.09	0.74	0.09	3.8
Proposed method	0.13	0.11	11.84	6.02	0.05	0.07	0.82	0.11	0.83	0.08	2.0

Lower TE, FN and FP show better result and higher S and S$_B$ demonstrate higher similarity to the ground truth.
FP: false positives; FN: false negatives; TE: total error; S: similarity measure; S$_B$: similarity measure in object boundaries.

lowest in this sequence which means it could achieve the best performance in overall.

Table 4 demonstrates the result for the AR.Drone sequence. This table shows that the accuracy of detection in AR.Drone is higher than smaller drones such as the Crazyflie. This can be explained with the AR.Drone having a bigger surface and receiving more accurate depth data from the sensor. Table 4 shows that the proposed method could achieve the lowest TE and FP which shows that this method has the lowest error compared to the other algorithms. Moreover, the similarity measures in the images (S) and around object boundaries (S$_B$) are the most similar to the ground truth. Similarly, average ranking (RM) shows the best performance for the proposed algorithm in overall by having the lowest amount among all other algorithms. Figure 8 illustrated an example of this sequences and the output of all compared methods.

In the remaining sections the benchmark RGB-D dataset introduced in Camplani and Salgado[39] are briefly discussed and then results are shown.

GenSeq sequences

This sequence has been designed to test the overall performance of the method in case of several possible error that may occur in one scene. This sequence contains a scene with individual person moving.

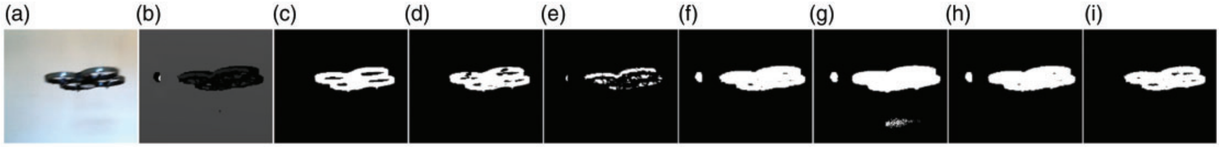

Figure 8. The result of AR.Drone sequence. (a) Colour frame, (b) Depth frame, (c) Ground truth, (d) MOG$_{RGB-D}$ output, (e) GSM$_{UB}$ output, (f) GSM$_{UF}$ output, (g) PBAS$_{bin}$ output, (h) ViBe$_{bin}$ output, (i) Proposed method output.

Table 5. GenSeq sequence results.

	TE		FN		FP		*S*		*SB*		
Method	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	RM
MOG$_{RGB-D}$	1.93	0.66	0.63	0.01	2.09	0.02	0.79	0.20	0.45	0.13	4.0
CL$_{W}$	1.30	0.42	1.49	0.02	1.27	0.01	0.83	0.21	0.53	0.14	3.0
GSM$_{UB}$	1.38	0.56	1.04	0.78	1.44	0.66	0.83	0.20	0.78	0.11	2.8
GSM$_{UF}$	1.30	0.52	4.08	15.38	1.30	0.60	0.83	0.20	0.78	0.14	3.2
PBAS$_{Bin}$	8.24	13.78	0.33	0.53	9.36	15.97	0.66	0.21	0.71	0.10	4.6
VIBE$_{Bin}$	2.32	0.58	1.59	1.52	2.43	0.56	0.77	0.16	0.75	0.09	4.6
Proposed method	1.09	0.46	2.85	7.43	1.02	0.56	0.88	0.14	0.79	0.12	2.0

Lower TE, FN and FP show better result and higher S and S$_{B}$ demonstrate higher similarity to the ground truth.
FP: false positives; FN: false negatives; TE: total error; S: similarity measure; S$_{B}$: similarity measure in object boundaries.

Table 6. DCamSeq sequence results.

	TE		FN		FP		*S*		*SB*		
Method	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	RM
MOG$_{RGB-D}$	2.11	1.29	15.25	0.09	1.31	0.02	0.61	0.14	0.61	0.11	2.6
CL$_{W}$	2.46	1.82	32.21	0.26	0.66	0.01	0.55	0.14	0.51	0.12	3.8
GSM$_{UB}$	1.74	1.70	20.45	10.73	0.46	1.57	0.64	0.17	0.54	0.14	2.0
GSM$_{UF}$	1.65	1.49	22.06	11.60	0.61	1.73	0.65	0.18	0.55	0.14	1.8
PBAS$_{Bin}$	6.66	14.29	46.98	31.45	4.69	15.17	0.32	0.22	0.38	0.23	7.0
VIBE$_{Bin}$	2.84	2.20	41.34	22.15	1.42	2.38	0.43	0.20	0.47	0.20	5.2
Proposed method	3.09	3.01	45.31	30.28	0.91	2.10	0.42	0.26	0.41	0.24	5.6

Lower TE, FN and FP show better result and higher S and S$_{B}$ demonstrate higher similarity to the ground truth.
FP: false positives; FN: false negatives; TE: total error; S: similarity measure; S$_{B}$: similarity measure in object boundaries.

Additionally, Table 5 shows the full results for all frames of this sequence.

Proposed method has the lowest amount of total error (TE) and highest similarity with the ground truth (S and S$_{B}$). Consequently, it has the lowest average ranking of method (RM) which shows that it has the best performance in this sequence among all methods.

DCamSeq sequences

The goal is to investigate the tolerance of the algorithms in case the depth camouflage occurs. As Table 6 illustrates the result of this sequence, the total error (TE) and false negative (FN) of the proposed method is very high which shows poor detection in this sequence. Accordingly, after PBAS it has the highest RM compared to other methods which demonstrate a weakness of the proposed method. The reason is that the depth model is not able to detect the entire hand when it is on top of the cupboard. GSM$_{UB}$ and GSM$_{UF}$ have achieved the lowest RM and shown a great result.

ColCamSeq sequences

It has been made to investigate the possible error of the algorithms in the case of colour camouflage. As Table 7 illustrates the result of this sequence, the proposed method could achieve the highest similarity measure

Table 7. ColCamSeq sequence results.

Method	TE		FN		FP		*S*		*SB*		
	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	RM
MOG$_{RGB-D}$	3.49	3.40	3.38	0.02	6.13	0.14	0.91	0.09	0.81	0.08	4.4
CL$_{W}$	3.20	2.77	3.52	0.09	2.92	0.10	0.89	0.15	0.77	0.16	4.4
GSM$_{UB}$	2.30	2.26	7.10	14.5	3.21	6.30	0.90	0.15	0.52	0.11	4.4
GSM$_{UF}$	2.20	2.27	2.94	5.53	4.36	6.42	0.92	0.08	0.53	0.09	3.2
PBAS$_{Bin}$	10.04	13.61	0.48	1.41	20.66	23.62	0.79	0.22	0.80	0.11	5.0
VIBE$_{Bin}$	3.16	2.72	1.08	2.95	7.19	7.13	0.91	0.08	0.86	0.07	3.4
Proposed method	2.61	2.84	1.99	3.52	5.54	8.27	0.93	0.09	0.89	0.07	2.4

Lower TE, FN and FP show better result and higher S and S_B demonstrate higher similarity to the ground truth.
FP: false positives; FN: false negatives; TE: total error; S: similarity measure; S_B: similarity measure in object boundaries.

Table 8. ShSeq sequence results.

	TE		FN		FP		*S*		*SB*		
Method	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	Avg.	St. Dev	RM
MOG$_{RGB-D}$	3.94	1.54	0.59	0.02	4.50	0.07	0.77	0.09	0.66	0.05	5.6
CL$_{W}$	0.81	0.35	1.60	0.05	0.68	0.02	0.94	0.04	0.71	0.07	3.0
GSM$_{UB}$	0.87	0.33	0.98	0.88	0.88	0.42	0.93	0.03	0.76	0.06	3.4
GSM$_{UF}$	1.66	0.38	0.14	0.19	1.92	0.44	0.89	0.04	0.65	0.05	3.8
PBAS$_{Bin}$	3.92	2.73	0.35	0.31	4.48	0.10	0.78	0.11	0.60	0.03	5.4
VIBE$_{Bin}$	3.72	0.99	0.06	0.15	4.31	1.17	0.78	0.07	0.64	0.03	4.4
Proposed method	0.80	0.41	0.88	0.70	0.81	0.48	0.95	0.03	0.82	0.06	2.0

Lower TE, FN and FP show better result and higher S and S_B demonstrate higher similarity to the ground truth.
FP: false positives; FN: false negatives; TE: total error; S: similarity measure; S_B: similarity measure in object boundaries.

and average in FP, FN and TE which lead to get the lowest RM. This means that the proposed method performs well in this scenario which is able to almost completely detect the white board from the same colour background. The reason behind is that our depth model strongly believes the board is not part of the background and therefore can detect it as a foreground.

ShSeq sequences

This sequence considered to test the impact of shadows in the scene. As Table 8 illustrates the result of this sequence, the proposed method could successfully detect the foreground object and avoid the shadow of the box on the floor. The total error shows the lowest amount of error and highest similarity measures (S and S_B) compared to the other methods. Accordingly, this allowed the proposed method to achieve the lowest RM which demonstrates the best performance among all other methods. Figure 9 illustrates an example from ShSeq sequences which has also been demonstrated in Camplani and Salgado.[39]

Figure 10 summarizes results shown in Tables 5 to 8 by illustrating the RC and RM for each individual method in each sequence. The lower amount for RM and RC shows better result. As illustrated in Figure 10, our method could achieve the lowest RM in GenSeq, ColCamSeq and ShSeq compared to the other five algorithms. This shows the best overall performance in all these benchmark datasets. However, in the DCamSeq the proposed method has the highest RM value. This means the proposed method is not able to perform well in case of depth camouflage but in all the other scenarios; it is able to demonstrate the best result. Indeed, according to RC values which calculated the overall performance of the algorithms in these four sequences, the proposed method outperforms among these six methods as it could achieve one of the lowest amounts of RC. Despite the positive result PBAS previously achieved in colour only datasets, in these RGB-D sequences, it presented the weakest performance in all four scenarios by achieving the highest RM and RC. The main reason for this failure is that PBAS was originally introduced only for colour frames and it cannot tolerate the noise of depth frames. We

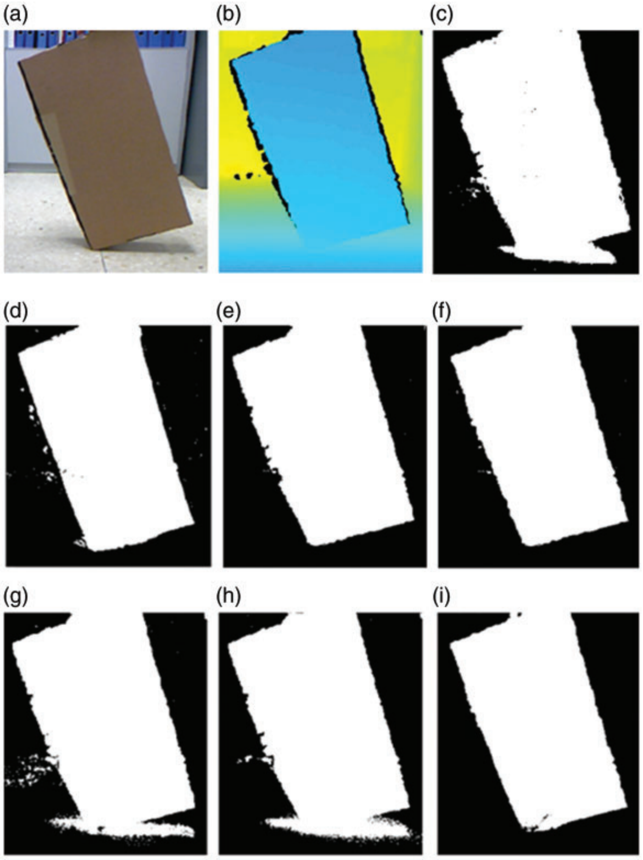

Figure 9. An example of ShSeq sequences (a) Colour data, (b) depth data codified in colour, (c) MOG_{RGB-D}, (d) CL_W output, (e) GSM_{UB} output, (f) GSM_{UF} output, (g) PBAS output, (h) $ViBe_{bin}$ output, (i) Proposed method output.

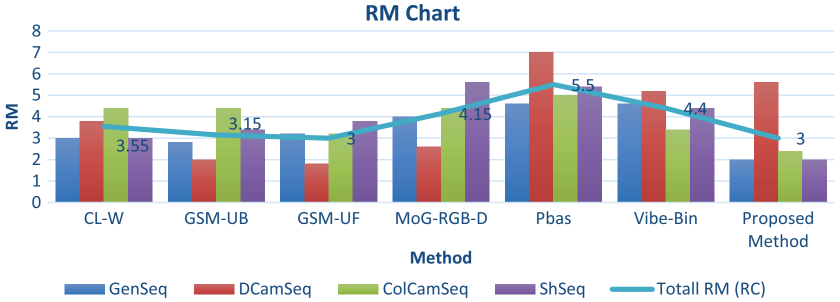

Figure 10. RM chart shows the overall performance of CL_W, GSM_{UF}, GSM_{UB}, Mog_{RGB-D}.PBAS and Vib_{bin} and proposed method in GenseqSeq, DCamSeq, ColCamSeq and Shseq sequences. The lower then RM and RC values are the better the performance is.

have used the original parameters in this comparison. However, it might be possible to achieve better results by changing the parameters.

Real-time experiment

The proposed system has been tested in a live application within an indoor environment. A basic control system based on the proposed approach has been implemented controlling an MAV. The computational cost of the algorithm is calculated as the mean rate of the processing time of the algorithm. The test was performed on a laptop with an Intel(R) Core(TM) i7-6700HQ CPU @2.6 GHz and 8 GB RAM along with Microsoft Kinect v2 sensor and a parrot AR.Drone.[57] As illustrated in Figure 11, coloured cover has been added to the front of the drone for more accurate depth data on the drone body and recognition of front of the drone for navigation by control system.

It is worth mentioning that the proposed algorithm used during these tests has been implemented in C++ and OpenCv library[62] without any specific code optimisation, as the aim of this experiment is to show that the proposed algorithm can be successfully run at real-time frame rates and therefore no effort has been made to optimise the code/set-up.

The quadcopter could successfully land on the floor and the total flying time was 210 seconds with mean processing time of 68.8 ms, and looking at the frame rate, it is about 15 fps. Demonstration showed that the proposed system could safely control the behaviour of the quadcopter in typical processing time required by other state of the art systems[63,64] for each frame at real-time. Figure 12 shows the total time obtained by the system for the first 200 frames of the test. The minimum time for a single frame on this test was 55 and the maximum was 82 ms.

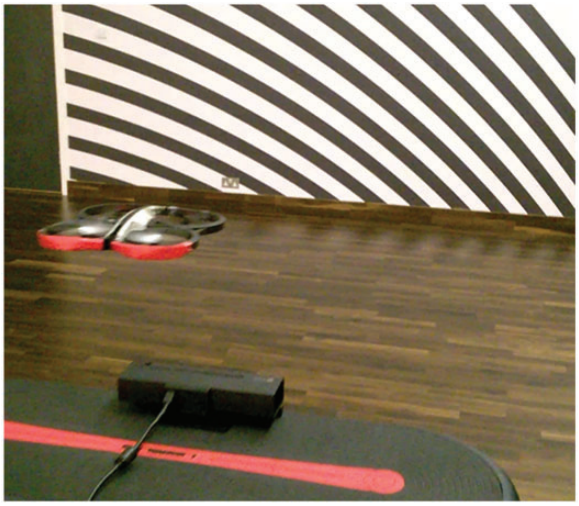

Figure 11. An AR.Drone was used to test the proposed method in real-time.

Conclusion

In this paper, we presented a novel nonparametric approach for the detection of MAVs using background modelling and segmentation of moving object by the history of previously observed pixel values similar to the previous work of ViBe and PBAS algorithm.

Our system produces one model for colour and one for depth. By combining colour and depth model together to produce the final classification, we could improve the background segmentation accuracy of similar methods in some challenging scenarios. These models get updated for each pixel which identified as a background. Additionally, after M frames, the system blindly updates the models regardless of the pixel being background or foreground.

These updates allow us to create more accurate depth model regardless of noisy depth frames. For this reason, the depth model has a greater influence in the final segmentation. In particular, the system relies more on colour model when depth is not available or cannot surly decide the foreground/background (e.g. near object boundaries). This helped us to significantly reduce the amount of false detection in case of sudden illumination changes and shadow on the floor.

The proposed method has four steps: initialization, post initialization filtering, classification and update. The results and evaluation section demonstrated that the proposed algorithm in our two sequences could achieve the best performance by having the lowest RM. However, the FN is high in both sequences which indicate some part of the foreground has been identified as a background due to the size, speed and surface of the UAV.

In the other four public datasets, the proposed method has the most accurate and reliable outcomes in comparison with other state of the art methods. Furthermore, we have shown that the GSM_{UF} and the proposed method achieved the best overall results by having the lowest RC as illustrated in Figure 10. This system also improved the overall performance of the detection of high-speed moving MAVs by

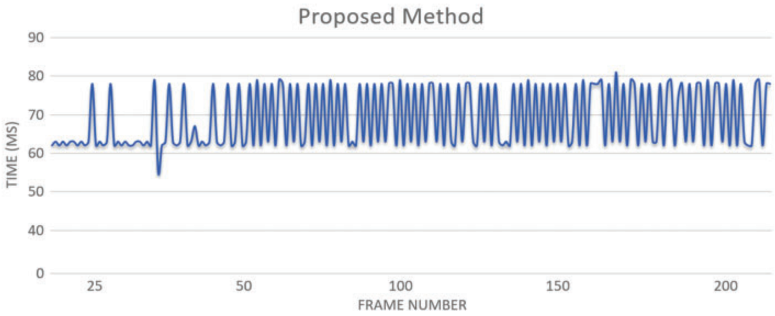

Figure 12. Computational time for the system to control the quadcopter. The time for the first 200 frames is shown.

combining the depth and colour model to produce the segmentation and update models to make it more accurate time after time.

These outcomes are also supported by Tables 5 to 8, where it has been highlighted the robustness of the proposed method by achieving the lowest value of RM in three sequences and only a poor performance in one sequence (DCamSeq) as the method has difficulty in detecting the moving object in occurrence of depth camouflage. Further improvements of the depth camouflage problem with the proposed method can be obtained by reducing the acceptable threshold amount and using a more accurate depth sensor.

The system also is able to cope with the dynamic background by using blind update which randomly exchanges pixels regardless of being background or foreground in every couple of frames. This will help the system to have a more valid and accurate model. However, it also leads to weak detection in case of a very slow moving object.

As future works, we are interested in the use of several quadcopters in the scene to perform autonomous flights using the detection system proposed here.

Acknowledgements

The data underpinning this publication can be accessed from Brunel University London's data repository, Brunel figshare here under a CCBY license: www.figshare.com/s/664c580196f139fb2d64

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has been partially supported by Brunel University.

References

1. Muller M, Lupashin S, D'Andrea R. Quadrocopter ball juggling. In: *2011 IEEE/RSJ international conference on intelligent robots and systems*. Piscataway: IEEE, 2011, pp. 5113–5120.
2. Durrant-Whyte HF, Roy N and Abbeel P. *Robotics : science and systems VII*. Cambridge: MIT Press, 2012, pp. 177–184.
3. Carrio A, Vemprala S, Ripoll A, et al. Drone detection using depth maps. CoRR. 2018; abs/1808.0.http://arxiv.org/abs/1808.00259
4. Henry P, Krainin M, Herbst E, et al. RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. *Int J Rob Res* 2012;31:647–663.
5. Li D, Li Q, Cheng N, et al. Combined RGBD-inertial based state estimation for MAV in GPS-denied indoor environments. In: *2013 9th Asian control conference (ASCC)*. Piscataway: IEEE, 2013, pp. 1–8.
6. Faessler M, Fontana F, Forster C, et al. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. *J F Robot* 2016;33:431–450.
7. Santana LV, Sarcinelli-Filho M and Carelli R. Estimation and control of the 3D position of a quadrotor in indoor environments. In: *2013 16th International conference on advanced robotics (ICAR)*. Piscataway: IEEE, 2013, pp. 1–6.
8. Baek J, Park S, Cho B, et al. Position tracking system using single RGB-D camera for evaluation of multi-rotor UAV control and self-localization. 2015;735:1283–1288.
9. Hou Y-L and Pang GKH. People counting and human detection in a challenging situation. *IEEE Trans Syst Man Cybern A Syst Humans* 2011;41:24–33.
10. Yong S-P, Deng JD and Purvis MK. Novelty detection in wildlife scenes through semantic context modelling. *Pattern Recognit* 2012;45:3439–3450.
11. Moyà-Alcover G, Elgammal A, Jaume-i-Capó A, et al. Modeling depth for nonparametric foreground segmentation using RGBD devices. *Pattern Recognit Lett* 9676–85 (2017).
12. Horn BKP and Schunck BG. Determining optical flow. *Artif Intell* 1981;17:185–203.
13. Brox T, Bruhn A, Papenberg N, et al. *High accuracy optical flow estimation based on a theory for warping*. Berlin: Springer, 2004, pp. 25–36.
14. Papageorgiou TD, Curtis WA, McHenry M, et al. Neurofeedback of two motor functions using supervised learning-based real-time functional magnetic resonance imaging. In: *2009 Annual international conference of the IEEE engineering in medicine and biology society*. Piscataway: IEEE, 2009, pp. 5377–5380.
15. Cheung SS and Kamath C. Robust techniques for background subtraction in urban traffic video. Proc. SPIE 5308, Visual Communications and Image Processing 2004, (18 January 2004). DOI: 10.1117/12.526886
16. Tang Z, Miao Z and Wan Y. Background subtraction using running Gaussian average and frame difference. In: *Entertainment Computing – ICEC 2007*. Berlin: Springer, 2007, pp. 411–414.
17. Zhan C, Duan X, Xu S, et al. An improved moving object detection algorithm based on frame difference and edge detection. In: *Fourth international conference on image and graphics (ICIG 2007)*. Piscataway: IEEE, 2007. pp. 519–523.
18. Barnich O and Droogenbroeck MV. *ViBe : a universal background subtraction algorithm for video sequences*.in *IEEE Transactions on Image Processing*, vol. 20, no. 6, 2011, pp. 1709–1724.
19. Chun-hyok PAK, Hai Z, Hongbo ZHU, et al. *A novel motion detection approach based on the improved ViBe algorithm*.2016 Chinese Control and Decision Conference (CCDC), Yinchuan, 2016, pp. 7081–7086.
20. Stauffer C and Grimson WEL. Adaptive background mixture models for real-time tracking. In: *Proceedings*

1999 IEEE computer society conference on computer vision and pattern recognition (Cat. no. PR00149), IEEE Comput. Soc,vol. 2, pp. 246–252.

21. Comaniciu D and Meer P. Robust analysis of feature spaces: color image segmentation. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. Piscataway: IEEE Computer Society, pp. 750–755.
22. Stenger B, Ramesh V, Paragios N, et al. Topology free hidden Markov models: application to background modeling. In: *Proceedings eighth IEEE international conference on computer vision ICCV* 2001. Piscataway: IEEE Computer Society, pp. 294–301.
23. Elgammal A, Harwood D and Davis L. Non-parametric model for background subtraction. *Comput Vision ECCV 2000* 2000;1843:751–767.
24. Wren CR, Azarbayejani A, Darrell T, et al. Pfinder: real-time tracking of the human body. *IEEE Trans Pattern Anal Mach Intell* 1997;19:780–785.
25. Cavallaro A, Steiger O and Ebrahimi T. Semantic video analysis for adaptive content delivery and automatic description. *IEEE Trans Circuits Syst Video Technol* 2005;15:1200–1209.
26. Dong Y and Desouza GN. Adaptive learning of multi-subspace for foreground detection under illumination changes. *Comput Vis Image Underst* 2011;115:31–49.
27. Shakeri M and Zhang H. Object detection using a moving camera under sudden illumination change. In: *Proceeding on 32nd Chinese Control Conference,IEEE,* Xi'an, 2013, pp. 4001–4006.
28. Wang H, Wang Q, Li Y, et al. An illumination-robust algorithm based on visual background extractor for moving object detection. In: *2015 10th Asian Control Conf Emerg Control Tech a Sustain World, ASCC* 2015. *IEEE,Kota Kinabalu,* 2015.doi: 10.1109/ASCC.2015.7244840, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp = &arnumber = 7244840&isnumber = 7244373
29. Rogez M, Tougne L and Robinault L. A prior-knowledge based casted shadows prediction model featuring OpenStreetMap data. *En VISAPP* 2013;vol. 1, 602–607.
30. Cristani M, Farenzena M, Bloisi D, et al. Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP J Adv Signal Process* 2010. 2010: 343057. https://doi.org/10.1155/2010/343057
31. Braham M, Lejeune A and Van Droogenbroeck M. A physically motivated pixel-based model for background subtraction in 3D images. In: *2014 International conference 3D imaging, IC3D*, Liege, 2014, pp. 1-8.doi: 10.1109/IC3D.2014.7032591,URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp = &arnumber = 7032591&isnumber = 7032567
32. Gordon G, Darrell T, Harville M, et al. Background estimation and removal based on range and color. In: *Proceedings 1999 IEEE computer society conference on computer vision and pattern recognition (Cat no. PR00149)*. Piscataway: IEEE Computer Society, 1999, pp. 459–464.
33. Leens J, Piérard S, Barnich O, et al. *Combining color, depth, and motion for video segmentation*. Berlin: Springer, 2009, pp. 104–113.
34. Dahan MJ, Chen N, Shamir A, et al. Combining color and depth for enhanced image segmentation and retargeting. *Vis Comput* 2011;28:1181–1193.
35. Mirante E, Georgiev M and Gotchev A. A fast image segmentation algorithm using color and depth map. In: *2011 3DTV conference: the true vision - capture, transmission and display of 3D video (3DTV-CON)*. Piscataway: IEEE, 2011, pp. 1–4.
36. Ottonelli S, Spagnolo P, Mazzeo PL, et al. Improved video segmentation with color and depth using a stereo camera. In: *2013 IEEE international conference on industrial technology (ICIT)*. Piscataway: IEEE, 2013, pp. 1134–9.
37. Bleiweiss A and Werman M. *Fusing time-of-flight depth and color for real-time segmentation and tracking*. Berlin: Springer, 2009, pp. 58–69.
38. Camplani M, Del Blanco CR, Salgado L, et al. Advanced background modeling with RGB-D sensors through classifiers combination and inter-frame foreground prediction. *Mach Vis Appl* 2014;25:1197–1210.
39. Camplani M and Salgado L. Background foreground segmentation with RGB-D Kinect data: an efficient combination of classifiers. *J Vis Commun Image Represent* 2014;25:122–136.
40. Francois E and Chupeau B. Depth-based segmentation. *IEEE Trans Circuits Syst Video Technol* 1997;7:237–240.
41. Doulamis ND, Doulamis AD, Avrithis YS, et al. Efficient summarization of stereoscopic video sequences. *IEEE Trans Circuits Syst Video Technol* 2000;10:501–517.
42. Camplani M, del Blanco CR, Salgado L, et al. Multi-sensor background subtraction by fusing multiple region-based probabilistic classifiers. *Pattern Recognit Lett* 2014;50:23–33.
43. Hofmann M, Tiefenbacher P and Rigoll G. Background segmentation with feedback: the pixel-based adaptive segmenter. In: *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. Piscataway: IEEE, 2012, pp. 38–43.
44. Bo G, Kefeng S, Daoyin Q, et al. Moving object detection based on improved ViBe algorithm. 2015;9:225–232.
45. Cucchiara R, Grana C, Piccardi M, et al. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans Pattern Anal Mach Intell* 2003;25:1337–1342.
46. Bouwmans T. Traditional and recent approaches in background modeling for foreground detection: an overview. *Comput Sci Rev* 2014;11:31–66.
47. Nyan B and Grünwedel S. PhD Forum: illumination-robust foreground detection for multi-camera occupancy mapping.*2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, Hong Kong, 2012, pp. 1-2.URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp = &arnumber = 6470166&is number = 6470120
48. Pierard S and Van Droogenbroeck M. Techniques to improve the foreground segmentation with a 3D

camera and a color camera. In: *20th Annual Workshop Circuits, Systems and Signal Processing*, 2009, pp. 247–250.

49. Rabha JR. Background modelling by codebook technique for automated video surveillance with shadow removal. In: *2015 IEEE international conference on signal image processing application (ICSIPA)*, Kuala Lumpur, IEEE, 2015, pp. 584–589.doi: 10.1109/ICSIPA.2015.7412258
50. Huerta I, Holte MB, Moeslund TB, et al. Chromatic shadow detection and tracking for moving foreground segmentation. *Image Vis Comput* 2015;41:42–53.
51. Chen Z, Zhao Y, Huang X, et al. An improved shadow removal algorithm based on gradient amendment. In: *International conference on signal processing (ICSP).2014 12th International Conference on Signal Processing (ICSP)*, Hangzhou, IEEE, 2014, pp. 1190-1194.doi: 10.1109/ICOSP.2014.7015188
52. Goyette N, Jodoin P-M, Porikli F, et al. Changedetection.net: a new change detection benchmark dataset. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Piscataway: IEEE, 2012, pp. 1–8.
53. Camplani M and Salgado L. Efficient spatio-temporal hole filling strategy for Kinect depth maps..Proc. SPIE 8290, Three-Dimensional Image Processing (3DIP) and Applications II, 82900E (30 January 2012); doi: 10.1117/12.911909; https://doi.org/10.1117/12.911909
54. Hsieh C-F, Yih C-H and Hsieh C-T. An improved depth image inpainting. In: *Proceedings of International Research Conference on Information Technology and Computer Sciences (IRCITCS)*, 28–29 September 2013, Kuala Lumpur. Taipei: Asia-Pacific Education & Research Association, 2014, pp. 15–23.
55. Yang L, Zhang L, Dong H, et al. Evaluating and improving the depth accuracy of kinect for Windows v2. *IEEE Sens J* 2015;15:4275–4285.
56. Pinto AM, Costa P, Moreira AP, et al. Evaluation of depth sensors for robotic applications. *Proc 2015 IEEE Int Conf Auton Robot Syst Compet ICARSC* 2015; 2015: 139–143.
57. Piskorski S, Brulez N, Eline P, et al. AR.Drone developer guide. Parrot, sdk. 2012.
58. Crazyflie 2.0 | Bitcraze, www.bitcraze.io/crazyflie-2/ (accessed 15 May 2017).
59. Dorudian N. Quadcopter detection, https://figshare.com/s/deed56cf5dd1c333fedf (2018, accessed 26 December 2018).
60. Real R, Vargas JM and Olmstead R. The probabilistic basis of Jaccard's index of similarity. *Syst Biol* 1996;45:380–385.
61. Li L, Huang W, Gu IY-H, et al. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans Image Process* 2004;13:1459–1472.
62. Intel Corporation, Willow Garage I. Open source computer vision library, http://opencv.org (2018, accessed 26 December 2018).
63. Santos MCP, Santana LV, Martins MM, et al. Estimating and controlling UAV position using RGB-D/IMU data fusion with decentralized information/Kalman filter. In: *2015 IEEE international conference on industrial technology (ICIT)*. Piscataway: IEEE, 2015, pp. 232–239.
64. Gongora A and Gonzalez-Jimenez J. Enhancement of a commercial multicopter for research in autonomous navigation. In: *2015 23rd Mediterranean Conference on Control and Automation (MED)*. Piscataway: IEEE, 2015, pp. 1204–1209.

Appendix I

C-like source code for proposed method

Pseudo-code for the main part of our algorithm for grayscale depth and colour images.

Default values for all the parameters of the algorithm is also given in the below code.

```
int width, height;
// Total number of samples
int N = 20;
// Random frame frequency (blind update frequency)
int M = 40;
// Minimum number of close samples
int $\#_{Min}$= N/4;
// Input Current Colour Image
byte ColourImage[width][height];
// Input Current Depth Image
byte DepthImage[width][height];
// Background Colour Model
byte ColourModel[N][width][height];
// Background Depth Model
byte DepthModel[N][width][height];
// Output Bg/Fg segmentation Mask
byte segMask[width][height];
byte background = 0;
byte foreground = 255;
int NoTolerance = 5;
int colourTolerance =DepthTolerance = 30;
int ADO = 650; // or 0
//For each pixel
for ( int i = 0;i < width; i++)
{
  int ioff= step*i;
  //compare with all pixels Models
  for (int j = 0; j< height; j++)
  {
    int countColor =0, index = 0, countDepth = 0,
    countDepthNoTolerance = 0;
// 1. Compare color and depth pixel to the back-
      ground models
while(index<N)
{
  //difference of two colour pixels
  int dist= ColourModel[index][i][j]- ColourImage
  [i][j];
```

```
    if (dist<= ColorTolerance && dist>= ColorTolerance)
    countColor ++;
  //difference of two depth pixels
  dist= DepthImage [i][j]- DepthModel [index][i][j];
  if(Depthsample != ADO)
  {
    If (dist+ DepthTolerance >= 0 )
      countDepth ++;
    if (dist2+ NoTolrance > 0)
    countDepthNoTolerance++;
  }
  index++;
}
// 2. Classification
bool isBackground=false;
  //If depth is ADO, Only rely on color frame
  if (DepthImage[i][j]== ADO) // 0 or 650
    {
      if(countColor>=$\#_{Min}$ )
        isBackground=true;
      else
        isBackground=false;
    }
  //If depth is strongly saying is background then the system will accept it
  else if (countDepthNoTolerance > $\#_{Min}$)
    isBackground=true;
  // If depth is strongly saying is not background then the system will accept it
  else if (countDepth < $\#_{Min}$)
    isBackground=false;
   //All remaining pxiels will be decided by color frame
  else if ( countColor >= $\#_{Min}$)
      isBackground=true;
  //3. Update the model by background pixels
  if (isBackground)
    {
      segMask[i][j] =0;
      int SmallestDepthAmount = 0;
      int SmallestDepthNumber = 0;
    //find the smallest depth amount and the position in the model
      findThesmallestDepth( SmallestDepthAmount, SmallestDepthNumber);
        //randon number (0-N)
        rand= GetRandomNumber(0,N);
        //randomly swap the pixel with the model
            ColourModel[rand][i][ j]= ColourImage [i][j];
      If ( (SmallestDepthAmount < DepthImage[i][j]) && (DepthImage[j][j] != ADO) )
         DepthModel [SmallestDepthNumber][i][ j]= DepthImage [i][j];
    }
    else
      segMask[i][j] =255;
      //4. Blind randomly update the models
      //Update after N number of frame
      if (FrameNumber%N == 0)
      {
      // replace randomly chosen sample
      rand= GetRandomNumber(0,N);
      ColourModel[rand][i][ j]= ColourImage[i][j];
      If (DepthImage != ADO)
      DepthModel [rand][i][ j]= DepthImage [i][j];
      }
  }
```