Background subtraction in people detection framework for RGB-D cameras

Anh-Tuan Nghiem, François Bremond

Background subtraction in people detection framework for RGB-D cameras

Anh-Tuan Nghiem, Francois Bremond
INRIA-Sophia Antipolis
2004 Route des Lucioles, 06902 Valbonne, France
nghiemtuan@gmail.com, Francois.Bremond@inria.fr

Abstract

In this paper, we propose a background subtraction algorithm specific for depth videos from RGB-D cameras. Embedded in a people detection framework, it does not classify foreground / background at pixel level but provides useful information for the framework to remove noise. Noise is only removed when the framework has all the information from background subtraction, classification and object tracking. In our experiment, our background subtraction algorithm outperforms GMM, a popular background subtraction algorithm, in detecting people and removing noise.

1. Introduction

In 2010, Microsoft introduced Kinect, the first low cost RGB-D camera. Together with the camera, Microsoft also provided a library for people detection and skeleton detection. As stated in [9], this library employs a background subtraction algorithm to detect foreground regions. However, perhaps due to high variance of depth measurement when objects are too far from the camera, this library only detects people when they are in the range of 0.5 to around 4.5 m from the cameras. This is also mentioned in [10]. Therefore, this library is not suitable for a general video monitoring application when monitored people may be outside the working range of the library.

To overcome the shortcoming of the provided library, we construct a new people detection framework consisting of a background subtraction, a people classifier, a tracker and a noise removal component. In this paper, we present a new background subtraction algorithm within the people detection framework specific to handle different type of noise in video from RGB-D cameras.

We evaluate the proposed people detection framework on the videos of – project. In our experiments, the proposed people detection framework based on the new background subtraction algorithm achieves comparable detection results with the library from PrimeSense, the company behind Kinect sensors when people are in the working range of the library (0.5 - 4.5m). Beyond this range, the library stops working whereas our framework can still be able to detect people with few errors.

The paper is organised as follows. First we present related background subtraction algorithms, especially for RGB cameras in section 2. We then present the overall structure of the people detection framework in section 3. The details of our background subtraction algorithm are presented in section 4. Section 5 presents how our people detection framework removes noise. In section 6 we compare our algorithm with the library from PrimeSense and with GMM, a popular background subtraction algorithm for RGB cameras. Finally, the conclusion is presented in section 7.

2. Related works

For RGB cameras, background subtraction algorithms [11, 4, 2, 5] have been used extensively to automatically detect foreground regions (moving regions) in a video coming from fixed cameras. [7, 8] are extensive surveys on background subtraction algorithms.

For RGB-D cameras, one can consider depth videos as a special case of gray scale videos and apply the background subtraction algorithms of gray scale images to depth videos. However, due to high variance of depth measurement as illustrated in figure 1, current state-of-art background subtraction algorithms might not have good results in foreground / background classification.

Let's examine how some of these background subtraction algorithms can handle the depth video from RGB-D cameras.

In [11], the values at each pixel are modeled by a mixture of weighted Gaussian (GMM). A pixel value is classified as background if it belongs to a Gaussian with a heavy weight. For noisy data from depth videos, it is difficult to select the suitable threshold on the Gaussian weights to correctly classify foreground / background. Moreover, if people stand close to the background, only a small part of their body which is relatively distant from the background are de-

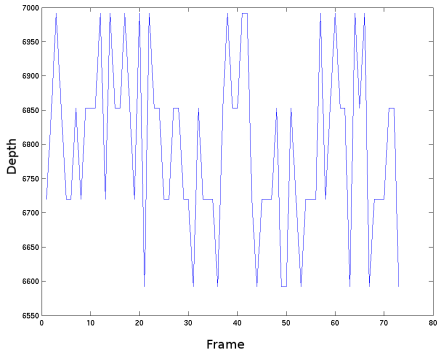

Figure 1. Depth value of a background pixel far from the camera. The difference between maximum and minimum value is as large as 40 cm.

tected as foreground as illustrated in figure 5, (Person close Background).

Another approach is to model dynamic background using non-parametric methods. In [4], the values at each pixel is modeled by a history of N most recent values. The model construction and updating become simpler but the parameter selection to distinguish foreground / background is difficult. In [2, 5], background models are also a set of N values taken from input videos but not the most recent values. These N values are taken randomly from incoming pixel values. The more a pixel values appears, the higher chance that it is included in the background model. For noisy data from depth videos, similar to [11], these algorithms also suffer the same problems such as selecting appropriate parameters and distinguishing people close to background from noisy depth values of background.

In summary, for depth videos from RGB-D camera, most of current background subtraction algorithms try to solve the problem of foreground / background classification at pixel level. At this level, the information is quite limited to have good classification when the variance of depth measurement is too high.

In this paper, we propose a background subtraction algorithm specific for depth videos from RGB-D cameras. Embedded in a people detection framework, it does not classify foreground / background at pixel level but provides useful information for the framework to remove noise. Noise is only removed when the framework has all the information from background subtraction, classification and object tracking.

3. Background subtraction in a people detection framework for RGB-D camera

The proposed people detection framework for RGB-D camera consists of a background subtraction algorithm, a HOG-based people classifier (see [6]), a tracker, and a noise removal component. The noise removal component receives information from all the other components to effectively remove noise.

The background subtraction in this people detection framework, unlike other background subtraction algorithms for RGB cameras, assigns not only foreground or background label to each pixel but also labels corresponding to different types of noise specific to depth video.

The background model inside the background subtraction algorithm is updated selectively with the help of the feedback from the framework. Specifically, when the framework detects a person, the region corresponding to that person is not updated so that the background subtraction algorithm can detect that person even when that person stays in the same place for a long time.

4. Background subtraction algorithm

4.1. Modeling high variance of depth measurement

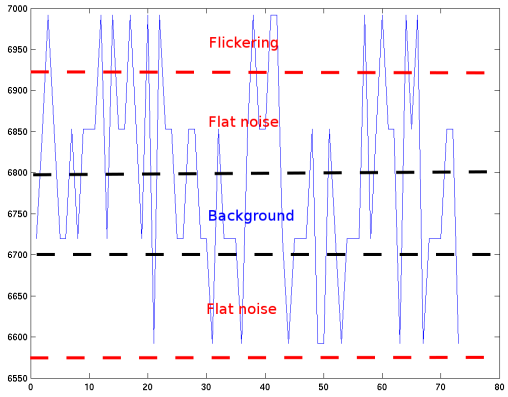

Figure 2. Range of depth pixel values is divided into three regions: Background, Flat noise, and Flickering noise. See text for more details.

To model the high variance of depth measurement, we divide the variance range into three separate regions: background, flat noise, and flickering noise as illustrated in figure 2.

In this figure, background region corresponds to the densest region of background depth values. Formally, a depth value d belongs to the background region if $|d-d^0| \leq \tau_B$ where d^0 is the mean of the depth values in the background region and τ_B is a small fixed threshold.

The Flat noise contains background depth values d within the range $\tau_B < |d-d^0| \leq \tau_F$ where τ_F is around 20 cm in our experiment. To distinguish the background depth values in this range from the depth values due to a person standing close to the background, we employ the classification and tracking information as explained in section 5.1.

The Flickering noise contains the remaining background depth values which are highly deviated from the mean value of the background region. Flickering noise can be detected based on the fact that the depth value in this region appears only in a very short period, normally only one or two consecutive frames. On the other hand, when a person stands close to the background, although the depth corresponding to the person is close to this range, the values are more stable and not flickered as in case of the flickering noise.

4.2. Background model

The purpose of our background model is to gather information to correctly model three types of regions described above: background, flat noise, and flickering noise.

Employing the idea from background subtraction algorithms for RGB cameras, we model depth values at each pixel by a set of codewords. Each code word is described by a tuple (m, n, t_s, t_l, n_c). In this tuple, m is the mean value of the depth values falling into this codeword. A depth value d is called "falling" into a codeword w if $|d - m^w| < \tau_B$ where τ_B is the predefined threshold mentioned in section 4.1. It is fixed and the same for every codeword. n is the number of depth values falling into this codeword. t_s is the frame number when this codeword is created. t_l is the frame number of the last time this codeword has a depth value falling into it. n_c: the number of times that this codeword has depth values falling into it consecutively. For example, if the depth values fall into the current codeword at frame 1, 2, 4, 5, 8, then $n_c = 3$ which corresponds to 3 group of consecutive falling. This is illustrated in figure 3.

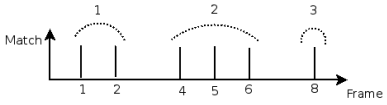

Figure 3. Illustration of computing n_c. (See text for details).

At a given pixel, for an incoming depth value d, the algorithm will find the corresponding codeword that d can fall into. If there is no such codeword, a new codeword with $m = d$ is created and inserted into the background model. Otherwise, the found codeword will be updated with d.

Updating codewords is straightforward for most of the features of codewords except the mean value m. In our algorithm m is updated as follows:

$$m_{new} = \frac{min(N, n)m_{old} + d}{min(N, n) + 1} \quad (1)$$

where N is a predefined value to help d to affect the value of m even when n is big. In our experiment, we set $N = 100$.

4.3. Depth value classification

As described in section 3 about overall structure of the framework, the classification of foreground / background pixels is realised at two levels: pixel level of background subtraction algorithm and object level when the framework has the feedback from the classification and tracking step.

At the pixel level, the background subtraction algorithm assigns one of the following label to each incoming depth value: $\{BG, FL, FLK, TB, FG\}$. BG/ FL/ FLK means depth values are in the background / flat / flickering region in figure 2. TB means that depth values belong to an uninterested object newly appearing in the camera view. For example, when a chair is moved or a bag is put on a table, the pixels belonging to those objects should have label TB. FG means that depth values do not belong to one of the above background labels. In other words, these depth values correspond to moving objects.

To explain the classification of depth values, let's assume that we have a set $W = \{w\}$ of codewords for a given pixel P and we want to classify a depth value d. In W, assuming that w^0 is the codeword with the highest number of depth values falling into it.

To classify d, the algorithm firstly finds the codeword w^* whose the mean value m^* is close to d ($|d - m^*| < \tau_B$). If there is no such codeword, the algorithm assigns label FG to d. Otherwise, the label of d is determined based on the characteristics of w^*.

The algorithm assigns the label BG to d if w^* is also w^0.

The algorithm assigns the label FL to d if w^* is not w^0 but $\tau_B < |m^* - m^0| \leq \tau_F$ τ_F is a small threshold mentioned in section 4.1.

The algorithm assigns the label FLK to d if w^* satisfies two conditions. Firstly, when several consecutive depth values fall into the same codeword w^*, the duration of this consecutiveness is no longer than β frames ($\beta n_c^* > n^*$, $\beta = 3$ in our experiment). Secondly, this phenomena should occurs once in at least every γ frames ($(t - t_s)/\gamma \leq n_c^*$ where t is the current timestamps). $\gamma = 100$ in our experiment.

In our algorithm, if a depth value satisfies the constraints of both $FLICKERING$ and $FLAT$, it will carry both labels for later processing.

The algorithm assigns the label TB to d if w^* satisfies two constraints of a temporal background like a moved chair. Firstly, the chair should stay in the new place long enough to be considered as background ($n^* > \alpha$, $\alpha = 200$ in our experiment). Secondly, after the chair has been moved to a new place, at each pixel in the region corresponding to the new place of the chair, the depth value measured by RGB-D camera is always the same except when there is another object moving by ($(t_l^* - t_s^*)/n^* < \psi$ where $\psi = 0.8$ in our experiment).

d is assigned the label FG if the corresponding codeword w^* does not belong to one of the above conditions.

5. Eliminating noise

As explained in section 3, at the pixel level, the background subtraction algorithm does not have enough information to remove noise. Therefore, noise is only removed when there is information from the classification and tracking.

Two types of noise regions are considered: foreground regions containing only noise pixels and foreground regions containing both noise pixels and real people. Let's call the first type as pure noise and the second type as mixed noise.

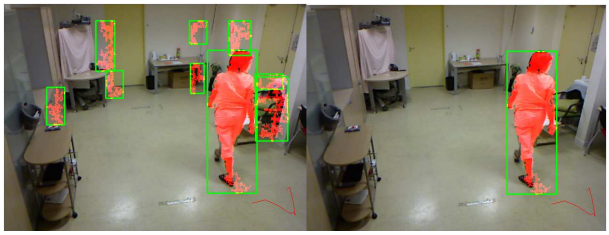

Figure 4. The performance of noise removal.

5.1. Eliminating pure noise

Assuming that at frame t, a foreground region R_t is classified as a human by the classification task with probability $P_H(R_t)$. Moreover, R_t is linked by the tracker to the foreground regions R_{t-1} at frame $t-1$, R_{t-2} at frame $t-2$ etc.

We define the probability of being the noise of type l for R_t in single frame t as $P^l(R_t) = N^l/N$. Here N^l is the total number of foreground pixels inside R_t with the noise label l assigned by the background subtraction algorithm and N is the total number of pixels in R_t.

Then R_t is classified as not noise of type l if:

$$P_H(R_{t-i})(1 - P^l(R_{t-i})) > \varphi \tag{2}$$

In our experiment, $\varphi = 0.8$. If in M consecutive frames, R_t is not classified as a noise region of type l, future foreground regions linked with R_t will also be considered as not noise of type l by the above criteria.

Let's take an example on how to distinguish a person and a foreground region corresponding to flat noise to understand how pure noise is removed.

When a person enters a room, he should be somewhere in the middle of the room before approaching the wall where flat noise may occur. Therefore, at least in some consecutive frames, the probability of being classified as potential flat noise $P^{FL}(R_t)$ is small. Moreover, if the people classifier can recognise the person with high probability ($P_H(R_t)$) during these frames, the value on the left hand side of the equation (2) should be high. Consequently, in the subsequent frames, the person will not be classified as flat noise even when he stays close to the wall for a long time.

On the other hand, when a foreground region corresponding to flat noise is detected, in all frames during its lifetime, the probability that this foreground region is classified as flat noise should always be high. Beside that, if the framework has a good people classifier, the probability that this foreground region is classified as person should be small. Combining these two conditions, the value on the left hand side of equation (2) should be small and this foreground region will be classified as noise by the framework.

The above method fails when the person stays only in regions very close to the wall (background) as soon as he enters the room. However, this case is not common in the reality.

We can apply the same reasoning for flickering and temporal background noise.

5.2. Eliminating mixed noise

Mixed noise corresponds to foreground regions containing both people and noise pixels. Because it contains people, mixed noise cannot be detected and removed by the method presented in section 5.1. This type of noise is detected based on the comparison between the size of the foreground region and the normal size of human body. Specifically, the probability $P_M(R_t)$ that a foreground region R_t is classified as mixed noise is computed as follows:

$$P_M(R_t) = max(P_1(R_t), P_2(R_t), P_3(R_t)) \tag{3}$$

where $P_1(R_t)$, $P_2(R_t)$, $P_3(R_t)$ corresponds to the probability that the height, width, area of R_t are bigger than those of a normal person. Because noise can only increase the size of foreground regions corresponding to people, we model those possibilities using half of a normal distribution as follows:

$$P_i(R_t) \propto \begin{cases} 0 & \text{if } x \leq T_x \\ 1 - N(x|T_x, \sigma_x) & \text{otherwise} \end{cases} \tag{4}$$

where x corresponds to height, width, area of the foreground region. T_x, σ_x are the mean and variance of the normal distribution of corresponding human size.

A foreground region is classified as mixed noise if its probability of mixed noise is larger than a certain threshold.

To remove noise from a foreground region corresponding to mixed noise, we simply remove foreground pixels with label noise assigned by the background subtraction algorithm and keep only the pixels with label FG.

With equation (4) a person pulling some objects or a group of people might be classified as mixed noise. However, because the noise removal process only removes pixels with noise labels, those people are often not affected.

	PrimeSense	GMM	Proposed
Precision	0.979	0.979	0.998
Sensitivity	0.987	0.987	0.995
F-Score	0.983	0.983	0.996

Table 1. Performance of people detection on close range videos.

6. Experiments

For RGB cameras, there are standard dataset such as the Change Detection Challenge [3] to compare the performance of different background subtraction algorithms. However, for RGB-D cameras, there has been no such standard dataset yet. Therefore, to test our background subtraction algorithm, we use some videos from the project Dem@care [1] and specific videos expressing several difficult problems of background subtraction algorithm for depth videos. The frame rate of these videos is around 4 frames /s.

The ground truth of these videos are object bounding boxes drawn on selected frames. The frames are selected if they contain moving people. A person is considered as detected (true positive) if the area of the intersection between the bounding box of the ground truth and the bounding box given by the algorithm is larger than 70% of the area of both bounding boxes.

In our experiment, we compare our background subtraction algorithm with GMM [11] and with the people detection library of PrimeSense, the constructor of the RGB-D camera Microsoft Kinect.

For GMM, we replace our background subtraction algorithm in the people detection framework with GMM and turn off noise removal algorithms. Therefore, GMM has the ability to update its background selectively according to the results of people classification and tracking. To handle high variance of depth measurement, we set up parameters of GMM so that it favors removing noise than detecting moving objects.

For the library of PrimeSene, it is a full people detection framework and we only take the output of the library to compare with the results of the other two algorithms.

In the first experiment, we compare the performance of PrimeSense and our people detection framework with GMM and with the proposed background subtraction algorithm on close range videos when the distance between the camera and the farthest point in the scene is less than 5m. Therefore, the variance of depth measurement on these videos is small. There are three close range videos consisting of 12446 frames. Among these frames there are 114 ground truth's bounding boxes. As shown in table 1, our algorithm with noise removal can achieve comparable results with those of PrimeSense library and GMM.

In the second experiment, we compare the performance of our background subtraction algorithm with the performance of GMM on long range depth videos in which the distance between the camera and the farthest point is around 7m (figure 5, Long range). From this experiment, we do not compare the performance of PrimeSense library because it can only detect people if they are within 5m away from the camera. From the results of this experiment (table 2, dataset Long), we see that the precision of GMM is not as good as the precision of our background subtraction algorithm with noise removal. This reflects the fact that when the distance increases, the proposed algorithm is better than GMM in handling high variance of depth measurement.

In the third experiment, we compare the performance of our background subtraction algorithm with the performance of GMM on a depth video of a very long corridor (figure 5, Corridor). In this video, the variance of depth measurement at pixels corresponding to the end of the corridor is quite high. From the results of this experiment (table 2, dataset Corridor) we see that, our background subtraction algorithm with noise removal has a higher precision index than the precision index of GMM. This means that it can better handle the high variance of depth measurement when objects are far from the camera. The sensitivity of object detection also shows that both GMM and the proposed people detection framework can detect people even when they are very far from the camera.

In the final experiment, we employ a video in which there is a dark bookcase on the right of the frame (figure 5, Dark bookcase). Because this region does not reflect much the infra-red emitted from the depth-camera, the depth-camera can measure valid depth values intermittently in only few frames. Most of the time, the depth-camera assigns zero depth value to pixels in this region. Consequently, GMM produces many noise in this region which decrease the precision dramatically (table 2, dataset Dark region). For our algorithm, it cannot remove this kind of noise using flat noise removal as the permanent background is zero which is very different from measured depth values when they are available. However, we see that the valid depth values in this region has the characteristic of flickering noise. Therefore, with the help of the temporal filter our algorithm can remove most of flickering noise. The precision of our algorithm in this experiment is not as high as the precision in other experiments because when the treadmill heavily occludes the person, the person region is split into some small regions which are considered as noise.

7. Conclusion

In this paper, we present a background subtraction algorithm in a people detection framework for RGB-D camera. The proposed background subtraction algorithm differs from other pixel-based background subtraction algorithm in two aspects. Firstly, our algorithm does not try to classify

Data			GMM			Proposed		
Name	#frames	#GT	Precision	Sensitivity	F-score	Precision	Sensitivity	F-score
Long	12355	258	0.89	0.97	0.93	0.95	0.97	0.96
Corridor	1069	210	0.5	0.9	0.64	0.98	0.97	0.97
Dark region	610	132	0.44	0.97	0.61	0.85	0.94	0.89

Table 2. Performance of GMM and the proposed background subtraction algorithm on three dataset. (#frames: number of frames, #GT: number of ground truth bounding boxes)

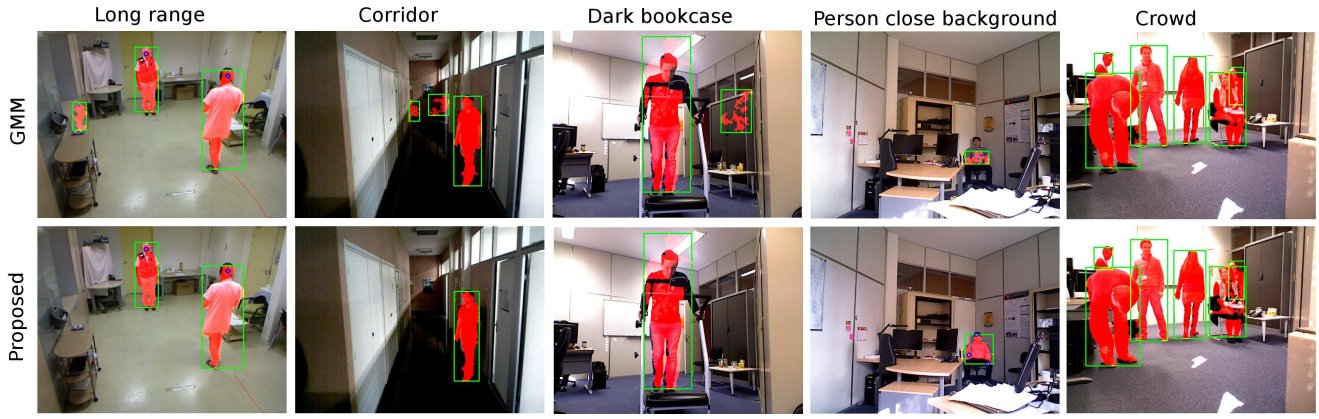

Figure 5. The detection results of GMM (upper) and the proposed algorithm (lower) on some videos from RGB-D cameras.

foreground / background at pixel level. Instead it only gathers information about noise at pixel level for noise removal component in the framework. Noise is only removed when the noise removal component has additional information from classification and tracking. Secondly, the proposed background subtraction algorithm is designed specific to depth video data. Therefore, it can handle various level of variance in depth measurement with default parameter setting. The experiment has shown that for close range videos our algorithm has similar performance as the performance of PrimeSense library. For long range videos, our algorithm outperformed GMM, a typical pixel based background subtraction algorithm.

The general idea of a background subtraction algorithm inside an object detection framework and the idea of flat noise and flickering noise can be generalised for videos of RGB camera. Therefore, in the future, we also would like to combine this idea to the case of people detection in videos from RGB camera.

References

[1] Dementia ambient care: Multi-sensing monitoring for intelligent remote management and decision support, 2014. http://www.changedetection.net.

[2] O. Barnich and M. V. Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences, 2011. IEEE Transactions on Image Processing.

[3] CVPR2014. A video database for testing change detection algorithms, 2014. http://www.changedetection.net.

[4] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction, 2000. IEEE Frame-Rate Workshop.

[5] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter, 2012. Computer Vision and Pattern Recognition Workshops.

[6] A. Nghiem, E. Auvinet, and J. Meunier. Head detection using kinect camera and its application to fall detection, 2012. 11th International Conference on Information Sciences, Signal Processing and their Applications.

[7] M. Piccardi. Background subtraction techniques: a review, 2004. IEEE International Conference on Systems, Man and Cybernetics.

[8] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey, 2005. IEEE Transactions on Image Processing.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images, 2011. IEEE Conference on Computer Vision and Pattern Recognition.

[10] L. Spinello and K. O. Arras. People detection in rgb-d data, 2011. International Conference on Intelligent Robots and Systems.

[11] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking, 1999. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.