Comparison of complex-background subtraction algorithms using a fixed camera

Geoffrey Samuel
Intelligent System and Robotics
University of Portsmouth
Geoffrey.Samuel@Port.ac.uk

Dr. Honghai Liu
Intelligent System and Robotics
University of Portsmouth
Honghai.Liu@Port.ac.uk

ABSTRACT

In this paper we look at the problem which is the background subtraction algorithms when dealing with complex, or real-world, background, as background subtraction is a vital step for Computer Vision. We will be measuring the effectiveness of each background subtraction algorithm of handling a real-world scene and subtracting the unwanted background elements as well as the time taken to complete each data set.
The data taken from this experiment can show that certain algorithms are suited to specific tasks and by joining them and we could successfully remove a complex background.

Keywords: *Computer Vision, Image Subtraction, Complexe scene*

1. INTRODUCTION

Background subtraction is a vital step for Computer Vision and it is the process of removing the background of an image in order to establish the motion and objects in the foreground for further analysis [13] and has many uses in topics such as surveillance based motion-tracking [5] and object classification [11] as well as monocular [1, 15], multi-camera motion capture methods [10, 6, 3, 14] and object reconsturction methods [9, 2]

Current areas of research includes background subtraction using unstable camera where Jodoin et al. [5] present a method using the average background motion to dynamically filter out the motion from the current frame, whereas Sheikh et al. [13] proposal an approach to extracting the background elements using the trajectories of salient features, leaving only the foreground elements.
Other indirect solutions to subtracting can be found with research into cameras capable of accurately capturing the depth as well as a full colour image by emitting light to the scene and measuring the time it takes to return to calculate the distance of an object [4],these approaches solve the background subtraction problem by allowing users to set depth thresholds, and ignoring any depth greater than the threshold. Research has also looked into creating the same effect buy with standard cameras without the use of any special kit or photography conditions, where the depth can be calculated using the amount of blur made by a coded aperture [7] as well as calculating the depth from a single image using a multi-scale local and global features Markov Random Field [12].

A complex background was described by Li et al. [8] and Sheikh et al. [13] as a background with moving background elements with are not wanted in the foreground, so a complex background can be thought of as a real-world background.
Our goal in this paper is to evaluate different background subtraction algorithms against using criteria such as the quality of the background subtraction and speed of algorithm to assess the algorithms value as a background subtraction algorithm and its projected performance as a real-time complex-background subtraction algorithm, and to suggest an algorithm or the combination of algorithms that can balances speed with quality.

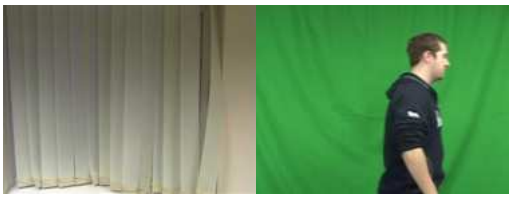
The remained of this artohicle is organized as follows. Section 2 gives an account of how we tested the algorithms. Out results are described in section 3. Section 4 discusses the results and Section 5 concludes the paper.

2. EXPERIMENT

2.1 The Experiment Plan

The purpose of the experiment is to measure the effectiveness of each background subtraction algorithm of handling a real-world scene and subtracting the unwanted background elements and leaving only the wanted foreground elements. To test each algorithm, a comparison would have to be made to determinant what pixels are foreground and which are background like a "Ground Truth". In order to test the algorithms on everyday type data, 7 different everyday motions were chosen to form our dataset. These motions were: Drinking, Jogging, Bending over, scratching head, Sitting down, Standing up and walking.

Creating the ground truth image for each frame of each clip would very time consuming and creates a result which is bias against in the interoperation of the foreground and background elements. To automate this process, a method used mainly for visual effects, was employed to ensure the final result was not bias against interoperation or favour a certain algorithm. This method is called "Chroma Keying" where an actor is filmed against a green screen, and the green screen is removed, and allows the foreground actor element to be extracted. Although this is synthetic data, it comprises of all the elements which would be found if it was filmed all together, with only a few artefacts from the compositing stage and spill suppression stages. These artefacts can be seen as



(a) Background plate (b) Green Screen

(c) Composite Image (d) "Ground-Truth"

Figure 1: Creation of the synthetic dataset
(a) shows the clean background plate consisting of the complex background, (b) shows the actor on the green screen, (c) shows the final result when the green screen has been Chroma keyed out and superimposed over the background plate and (d) shows the auto generated "Ground-Truth" created by clamping the Chroma keying colour values.

the 1 pixel highlighted area around the top of the subjects head.

Fig. 1 shows the different stages of creating synthetic data and how the different elements came together to create the final data set. Fig.1.a shows a still frame from the complex background footage we filmed with the naturally flowing motion of the binds. Fig.1.b is a still frame from the green screen footage recorded in a controlled environment to minimize the time to extract the foreground elements. Fig.1.c shows a still frame from the final composite film where the foreground elements from the green screen have been placed in front of the complex background footage. Fig.1.d is the "Ground Truth" that was generated as a result of the Chroma keying process alpha channel has been converted to a Boolean value of true or false and then converted into a black and white image based of the value. This allows us to know if the subject was present in that pixel or weather the subject was not present.

As a number of different background algorithms were chosen to be compared; the algorithms would have to be in the same code format to ensure the speed was not being influenced by the code base of the platform it was running on. Because if its use in academic research and as researchers are more inclined to give code for testing than commercial companies, MATLAB was chosen. Three algorithms were used as they were all written by the same author and used MATLAB. These were the; "Frame Difference", "Approximate Median" and "Mixture of Gaussians" methods
A fourth algorithm was written by us using the most basic background subtraction method, which was written in the same coding structure as the frame difference method so to not bias the speed results by optimizing code. This is called the "Back Plate Difference" method.

2.2 The Algorithms

The Back Plate Difference algorithm is the most basic of the algorithms in where the pixel values are compared to an original clean back plate image and determines if the pixel is foreground of background.

$$|f_i - bgpt| > T_S$$

F is the frame, i is the frame number, bgpt is the clean back plate image, T is the threshold and s is the threshold value. This algorithm takes no motions nor any machine learning method so the back ground model will not change or update based off the complex background motion.

The second algorithm is the Frame Difference algorithm. The Frame Difference algorithm is similar to the Back Plate Difference method, but compares the frame with the frame before, therefore allowing for scene changes and updates.

$$|f_i - f_{i-1}| > T_S$$

F is the frame, i is the frame number, T is the threshold and s is the threshold value. This allows for slow movement update as the scene changes.
The Approximate Median algorithm takes the median value of a set number of previous frames to construct a back plate model, and compare the pixels in the same way as the Frame Difference algorithm. Unlike the Frame Difference, the background model is "burnt" by the foreground objects over time, allowing for foreground objects to move into the background if present for long enough.

$$(\widetilde{x} = (f_i - f_{i-1} - f_{i-2} \cdots f_{i-n}) > T_s) \rightarrow (\sigma_i + = 1) \rightarrow (\sigma_i - = 1)$$

$\widetilde{x}$ is the median image of the frames, f is the frame, i is the frame number, T is the threshold and s is the threshold value, and σ is the back plate model.
The last algorithm is the Mixture of Gaussians method. This algorithm converts each pixel into a Gaussian model and calculates the probably of the image based off the sum of the models.

$$f\left(i_t = \mu\right) = \sum_{i=1}^{k} \omega_{i,t} \cdot \eta\left(\mu, o\right)$$

f is the frame,$\eta\left(\mu, o\right)$ is the is the Gaussian component, K is the number of Gaussians per pixel.

2.3 Testing of the Algorithms

To test the effectiveness of the algorithms, each pixel in each of the images was tested against the pixels from the ground truth. As each image is either black or white, it could be through of a Boolean value, that if it is black it is false and therefore background. The time results of each data set were normalized to allow for direct comparison between methods.
Testing the speed of each algorithm required the algorithm to be run 100 times and retreating the average time. This was to compensate for any background tasks taking place in either the programme or the operating system. This was not preformed on the mixture of Gaussian algorithm as each frame took an average of 11 seconds, and the data set was too large to run through 100 times.

3. RESULTS

As Table one shows the quality of the each dataset using the different Algorithms, where all methods preformed between 80-90
Table two shows the average speed of each of the algorithms, for each of the datasets, when run 100 times. Method 4, Mixture of Gaussian, was the only method not 100 times as each frame took an average of 11 seconds to compute. These figures would suggest that these all but the Mixture of Gaussian method, would be suitable for real-time applications where the algorithm would be required to run between 30-60 times a second.

Table 1: Percentages match between methods and "Ground Truth"

Motions	A	B	C	D
Drinking	90.78%	82.12%	89.52%	83.78%
Jogging	88.24%	88.88%	92.14%	88.20%
Bend over	91.26%	88.22%	83.40%	90.19%
Scratch head	88.18%	84.78%	90.56%	86.15%
Sitting down	88.51%	80.07%	82.28%	81.68%
Standing up	89.40%	83.82%	80.99%	83.78%
Walking	88.47%	89.81%	94.22%	90.01%

Table 2: Average Speed of Algorithm

Motions	A	B	C	D
Drinking	0.0507	0.0004	0.3301	10.6954
Jogging	0.0507	0.0025	0.0691	10.8219
Bend over	0.0492	0.0819	0.0730	12.2895
Scratch head	0.0450	0.0850	0.0718	10.6132
Sitting down	0.0420	0.0692	0.0662	10.850
Standing up	0.0416	0.0747	0.0529	12.7196
Walking	0.0319	0.0129	0.0541	10.5202

4. DISCUSSION

As table 1 indicate, the percentages of correctly identified pixels are between 80-94%. Back plate difference has the highest number of correctly identified pixels in four out of the seven data sets, with Approximate Median claiming the highest number of correctly identified pixels in the other three data sets.
As table 2 shows, the time taken to complete an average frame of the data set. The time taken varies from 0.0004 seconds to 12.7196 seconds per frame. Each of the algorithms were run 100 to calculate the average time for each frame to ensure that the operating system did not interfere or influence the speed results, apart from the Mixture of Gaussian
The Back plate difference algorithm provides constantly good results, but as the algorithm is based around removing a static background, the complex back ground could still be present. The algorithm did the worse of correctly identifying pixels in the walking data clip, but overall maintains an identification rate of between 88-91%. The speed of the Back plate difference algorithm was the fastest in four out of the seven datasets.
With the Frame Difference algorithm, the correct identification rate is lower than of the Back Plate Difference, ranging between 66-88%. As the method takes into the account the scene transforming and moving over time, this method has a better idea of what is going on in the scene, therefore being able to suppress the background motion. This is the only algorithm to correctly identify the motion of the complex background, but has problems with identifying the foreground objects correctly. The Frame difference algorithm was statically the second faster's algorithm, being the fastest time for three of the seven data sets.
The Approximate Median method provided three of the top seven results in terms of highest percent of correct pixel identification, but also had the most varied results with a range between 80-94%. The main problem caused by this problem is the "burring" effect that allows for static foreground objects to become part of the background. The resulting image is visual similar to the result of that of the Back plate Difference. The Approximate Median algorithm was a close third place on the speed tests, being only slightly slower than that of the Frame Difference or the Back Plate Difference speed test times.
The Mixture of Gaussian algorithm produced medico results, which varied between 81-90%. This method even got the worse percentage of correctly identified pixels for one clip. The rustling image also looked rather noisy in places. With the speed tests, the algorithm was not able to be run 100 to get an average running time, and taking an average of 11 seconds per frame.

With the Mixture of Gaussian method providing medico results and a slow speed, in comparison to the rest of the speed results, the Mixture of Gaussian should not be in any application designed for real-time usage, it does not seem to update or handle the complex background at any stage.

The approximate median has promising results, similar to that of Back Plate Difference in the speed result and background/foreground detection, but the "burning" can be a issues with fast moving motion especially when the subject has stayed still to allow the burning to accumulate before the subject moves on, creating a patch of false negative.

With the back plate difference, the results were the top four for the number of matching pixels and the top four fastest datasets. The results show that this is defiantly a quick and relatively easy way to remove a background. When looking at the image data produced by the back plate difference, the complex background is instantly noticeable. Although the results are the highest in all the tests, the algorithm does not remove or to some degree even suppress the unwanted motion of the complex background.

The only algorithm capable, within this experiment, of removing a complex background is the frame difference algorithm. This is because the model updates each frame, and checks for movement or motion through a frame, allowing for the algorithm to ignore background motion and identify foreground elements. The problem with this is that it does not handle slow moving foreground objects well.

5. CONCLUSIONS

In conclusion, we have looked at the pros and cons of

each of the algorithms and how only one algorithm is able to remove unwanted background motion commonly seen in a complex scene, the Frame Difference algorithm, but how this lacks the capabilities to identify slow moving sections of the foreground. Whereas the Back plate difference method allows for fast and relatively accurate foreground extraction, but cannot deal with the complex background as the background model used does not update or change. What could be done is use the both the Frame Difference combined with the Back Plate Difference using a special function to suppress the complex background motion.

ACKNOWLEDGMENTS

The project is supported by the UK Engineering and Physical Science Research Council under grant No. 06002299 The author would like to thank Seth Benton for providing his background subtraction algrothims.

6. REFERENCES

[1] *Y. Chen, J. Lee, R. Parent, and R. Machiraju. Markerless monocular motion capture using image features and physical constraints. In* Computer Graphics International, *volume 1. Citeseer, 2005.*

[2] *G. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In* IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, *volume 2. Citeseer, 2003.*

[3] *E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. Seidel, and S. Thrun. Performance capture from sparse multi-view video.* ACM Transactions on Graphics (TOG), *27(3):98, 2008.*

[4] *R. Gvili, A. Kaplan, E. Ofek, and G. Yahav. Depth keying.* SPIE Elec. Imaging, *5006:564–574, 2003.*

[5] *P. Jodoin, J. Konrad, V. Saligrama, and V. Veilleux-Gaboury. Motion detection with an unstable camera. In* Proc. IEEE Signal IntŠl Conf. Image Process, *pages 229–232, 2008.*

[6] *R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. In* IEEE computer society conference on computer vision and pattern recognition, *volume 2, page 129. IEEE Computer Society; 1999, 2005.*

[7] *A. Levin, R. Fergus, F. Durand, and W. Freeman. Image and depth from a conventional camera with a coded aperture. In* ACM Transactions on Graphics, *volume 26, page 70. ACM, 2007.*

[8] *L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Foreground object detection from videos containing complex background. In* MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, *pages 2–10, New York, NY, USA, 2003. ACM.*

[9] *V. Lippiello and F. Ruggiero. Surface model reconstruction of 3d objects from multiple views. In* IEEE International Conference on Robotics and Automation, *2009.*

[10] *B. Michoud, E. Guillou, H. Briceno, and S. Bouakaz. Real-time marker-free motion capture from multiple cameras. In* IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, *pages 1–7, 2007.*

[11] *E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, and S. Lepchev. A real-time system for classification of moving objects.* Pattern Recognition, International Conference on, *3:30688, 2002.*

[12] *A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images.* Advances in Neural Information Processing Systems, *18:1161, 2006.*

[13] *Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In* IEEE International Conference on Computer Vision, *2009.*

[14] *D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In* ACM SIGGRAPH 2008 papers, *page 97. ACM, 2008.*

[15] *E. Yu and J. Aggarwal. Detection of stable contacts for human motion analysis. In* Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, *page 94. ACM, 2006.*