Median mixture model for background – foreground segmentation in video sequences

Piotr Graszka

Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warszawa, Poland

P.Graszka@ii.pw.edu.pl

ABSTRACT

The purpose of this paper is to present a novel approach to the Gaussian mixture background modeling model (GMM) that we call the median mixture model (MMM). The proposed method is based on the same principles as the GMM, but all of the background model parameters are estimated in a much more efficient way resulting in accelerating the algorithm by about 25% without deteriorating the modeling results. The second part of this paper describes a method of uniting three MMMs where three different sets of input data undergo modeling in order to achieve even better results. This approach called the united median mixtures is more robust to random noise as well as unwanted shadows and reflections. Both algorithms are thoroughly tested and compared against the Gaussian mixture model, taking into consideration robustness to noise, shadows and reflections.

Keywords

Background – foreground segmentation, background modeling, Gaussian mixture model, image processing.

1. INTRODUCTION

The background – foreground segmentation or background subtraction is the first step in the majority of the automated video surveillance systems. The “background” is interpreted as a set of pixels that have constant over time properties like, for example, the color or the frequency of intensity changes [Gra13a]. However, to remove it, it is necessary to know how it looks like. There are a lot of specialized background modeling algorithms that consider different aspects of the problem. A good method should be able to adapt to three fundamental scenery changes [Gra13a]: a change of the brightness, such as the sun coming out from behind the clouds, a continuously repeating changes, such as a flag flapping in the wind and a change in the geometry, such as a car driving away from a parking space. The Gaussian mixture model (GMM) earned a label of a general purpose background modeling method as it considers all of the listed scenery changes and is known to behave well in the majority of indoor and outdoor scenes. However, it has also a couple of drawbacks that could be improved in future algorithms. The main is the lack of consideration of the correlation of the pixels in the image. There is also space for improvements when it comes to the accuracy of the results. Lastly, the speed of most of the implementations allows the algorithm to work in real time, however in all real life applications the background – foreground segmentation is just a first step of processing, so speeding it up would be a great advantage. This paper presents two novel approaches to the background modeling. The first one, called the median mixture model, distinguishes it self by a faster execution without deterioration of the results. The second one, called the united median mixture, has a much better accuracy and robustness to noise and distortions. Both of them are based on the same principles as the Gaussian mixture model, so to get acquainted with the idea the next part of this paper describes the most common approach to the GMM algorithm that was first proposed by C. Stauffer and W. Grimson and is currently very widely used [Sta99a].

2. GAUSSIAN MIXTURE MODEL

The GMM is a statistical algorithm that classifies a pixel as belonging to the background only if it is described by the historical statistics of the pixels previously observed in the analyzed image point.

The histogram of the previously observed pixel values is modeled for every pixel independently by 3 to 5 Gaussian distributions, hence the name of the algorithm – the Gaussian mixture model. Each of the distributions is described by three parameters: the mean value (μ), the variance (σ^2) and the weight (ω). The value for each of the parameters is estimated recursively basing on the previously approximated value. Only the Gaussian distribution that describes the currently analyzed pixel needs to be updated. A distribution describes a pixel only if the difference between the value of the pixel and the mean value of the distribution is less than $2.5\sigma^2$. The parameters of such a distribution are updated with the following formulas [Sta99a]:

$$\mu(x,y)_{n+1} = \alpha \cdot F(x,y)_n + (1-\alpha) \cdot \mu(x,y)_n \quad (1)$$

$$\sigma(x,y)^2_{n+1} = \alpha \cdot (F(x,y)_n - \mu(x,y)_n)^2 + (1-\alpha) \cdot \sigma(x,y)^2_n \quad (2)$$

$$\omega(x,y)_{n+1} = (1-\alpha) \cdot \omega(x,y)_n + \alpha \quad (3)$$

where:

$\mu(x,y)_{n+1}$ – currently approximated mean,

$\mu(x,y)_n$ – previously approximated mean,

$\sigma(x,y)^2_{n+1}$ – currently approximated variance,

$\sigma(x,y)^2_n$ – previously approximated variance,

$\omega(x,y)_{n+1}$ – currently approximated weight,

$\omega(x,y)_n$ – previously approximated weight,

$F(x,y)_n$ – pixel from the n-th frame,

α – estimation factor, usually $\alpha \in (0.01, 0.1)$.

The remaining distributions for this pixel have their weights reduced according to the formula:

$$\omega(x,y)_{n+1} = (1-\alpha) \cdot \omega(x,y)_n \quad (4)$$

Such an approach to the update is called “a selective update”, because it updates only those parts of the background model that require it, preventing the pollution of the model with data that belongs to the foreground. However, if neither of the distributions describes the analyzed pixel, the distribution with the lowest weight is replaced with a new one with the mean value equal to the pixel value, large variance and low weight. This way the model can adapt to the changes in the background, for example lighting intensity changes or a new stationary object appearance.

Only the distributions with the highest weights – exceeding the threshold value usually equal to 0.5 – are used for the background – foreground pixel classification. If there is at least one distribution describing the analyzed pixel with the weight exceeding the threshold, then the pixel is classified as belonging to the background, otherwise the pixel is classified as a part of the foreground.

In the GMM method the classification and the model update is done in one step, which means that the algorithm needs to loop only once through the whole model in order to classify all the pixels and update all the distributions. The outcome of the algorithm is a binary image with white foreground areas and a black background corresponding to the pixels of the input frame classified as belonging to the foreground and background respectively.

The main advantage of this approach is that it is capable of modeling a rapidly changing background (for example a flag fluttering in the wind or a flickering light source) through the use of several Gaussian distributions describing different states of the changing background.

The main drawback, as in many other background modeling algorithms, is that it overlooks the information about the location of the pixel in the image together with its correlation with its neighborhood, which can be observed for example in the color gradient.

The results of the GMM method can be obtained in the real time and are very good in the majority of situations. The noise and distortions appear only if the light conditions change very rapidly. The foreground objects are reproduced quite precisely. This is the reason for labeling the method as a general purpose background modeling algorithm.

The next part of this paper describes the median mixture model which, as it was mentioned in the introduction, is based on the same principles as the Gaussian mixture model, but all of the background model parameters are estimated in a much more efficient manner. This way we obtain all the advantages of the GMM method along with a faster execution. There are also a couple of changes that make the median mixture model able to update the model better and to consider spatial correlations of the pixels in the image.

3. MEDIAN MIXTURE MODEL

Similarly to the GMM, the MMM is a statistical method that classifies a pixel as belonging to the background if it is described by the historical statistics of the pixels previously observed in the analyzed image point or in the nearest spatial neighborhood of that point.

The historical statistics in the case of the MMM approach is a bit different from the one in the GMM method. Each input frame pixel value is modeled by 5 distributions. These distributions have the same purpose as in the GMM approach, but their interpretation, as well as the parameters estimation formulas are different. Instead of the mean value (μ), the MMM uses the median value (m) and instead of the variance (σ^2), it uses the standard deviation (σ). The weight (ω) parameter has the same meaning as in the GMM method but a different estimation formula. The values of those three parameters are recursively approximated basing on the previously estimated values. Only the distribution with the highest weight among all the distributions that describe the currently analyzed pixel needs to be updated. The parameters of such a distribution are updated with the following formulas:

$$m(x,y)_{n+1}=\begin{cases} m(x,y)_n+1 & \text{for } F(x,y)_n>m(x,y)_n \\ m(x,y)_n-1 & \text{for } F(x,y)_n<m(x,y)_n \\ m(x,y)_n & \text{for } F(x,y)_n=m(x,y)_n \end{cases} \quad (5)$$

$$\sigma(x,y)_{n+1}=\begin{cases} \sigma(x,y)_n+1 & \text{for } d(x,y)_n>\sigma(x,y)_n \\ \sigma(x,y)_n-1 & \text{for } d(x,y)_n<\sigma(x,y)_n \\ \sigma(x,y)_n & \text{for } d(x,y)_n=\sigma(x,y)_n \end{cases} \quad (6)$$

$$\omega(x,y)_{n+1}=\begin{cases} \omega(x,y)_n+1 & \text{for } \omega(x,y)_n<\omega_{max} \\ \omega_{max} & \text{for } \omega(x,y)_n\geq\omega_{max} \end{cases} \quad (7)$$

where:

$m(x,y)_{n+1}$ – currently estimated median,

$m(x,y)_n$ – previously estimated median,

$\sigma(x,y)_{n+1}$ – currently estimated standard deviation,

$\sigma(x,y)_n$ – previously estimated standard deviation,

$\omega(x,y)_{n+1}$ – currently estimated weight,

$\omega(x,y)_n$ – previously estimated weight,

$F(x,y)_n$ – pixel from the n-th frame,

$d(x,y)_n=|F(x,y)_n-m(x,y)_n|$ – current deviation,

ω_{max} – maximum weight value, usually $\omega_{max}\approx 200$.

The formula for the median estimation is taken from the approximated median filtering method proposed by N. McFarlane and C. Schofield [McF95a].

The distributions that do not describe the currently analyzed pixel have their weights reduced according to the formula:

$$\omega(x,y)_{n+1}=\begin{cases} \omega(x,y)_n-1 & \text{for } \omega(x,y)_n>1 \\ 0 & \text{for } \omega(x,y)_n\leq 1 \end{cases} \quad (8)$$

All the other distributions remain intact – this way we get the selective update feature.

If the weight of any of the distributions decreases to zero then such a distribution is not considered in the calculations at all. This means that we can model the background with a dynamically varying number of distributions – from a single distribution for the static background areas up to five distributions for the dynamic parts of the background.

If neither of the distributions describes the analyzed pixel, the neighboring distribution mixtures are checked. If one of the nearest neighbors contains a distribution that describes the analyzed pixel and its weight is greater than $0.5\omega_{max}$, a distribution with the lowest weight is replaced by a new one, with median value equal to the pixel value, large standard deviation, and a weight slightly above the $0.5\omega_{max}$ threshold. If the check against the neighborhood comes out negative, the weight of the replacing distribution is set to a low value.

The consideration of the neighboring distribution mixtures makes the MMM aware of the spatial correlations in the image. This feature provides a grater robustness to noise and makes it easier to discard the so called “ghost objects” – groups of pixels that were misclassified as belonging to the foreground. A ghost object can appear for example when a car drives away from a parking place (the parking place may become a ghost object as it is not described by the background model). The spatial correlation awareness makes it possible for the model to adapt faster to such situations because the pixels of the empty parking space are usually spatially correlated to the pixels of the rest of the car park which is already described by the background model.

The classification rule is the same as in the GMM method, but the classification step can be carried out independently from the update step. This is achieved by dividing the update procedure into two steps. The first one checks only the update rules and for each pixel it calculates the indexes of the distributions with the highest weight exceeding the threshold and the lowest weight – this data will be needed for the second step. The classification is conducted along with the first update step, which means that the classification results are available before the actual update is done. The second update step is conducted based on the data calculated in the first step and the binary classification results. Those are needed for the update to complete because if neither of the distributions describes the analyzed pixel, the algorithm would not have to check the neighborhood again as it was already done during the classification, so it can just use these results.

Such an approach to the background model update provides a smart feature we called “a fed update”. The classification results can be slightly changed before giving them to the second update step, which will have an impact on the final state of the background model after the update procedure is completed. This can be used to make the selective update even better, for example a denoising of the classification results would teach the model that the removed noise should not be reported as a part of the foreground. The same can be done with the ghost objects or any other phenomenon that should be classified as belonging to the background, but currently is not.

With such a structure, the median mixture model inherits the capability to model a rapidly changing background from the Gaussian mixture model along with its other advantages. At the same time the MMM approach considers the spatial correlations in the image eliminating the main disadvantage of the GMM method.

The main drawback of the described background modeling algorithm is that by itself the MMM is more prone to noise than the GMM. However, this can be minimized with an appropriate use of the fed update feature as it is a way of teaching the model what is wrong with the current results.

The next part of this paper describes a background – foreground segmentation algorithm called the united median mixtures. The UMM approach makes use of the features of the MMM method by uniting the results of the three independently working median mixture models in order to get even better overall background modeling results.

4. UNITED MEDIAN MIXTURES

The concept behind the UMM method is much more general than for the previously described approaches, so this paper presents an example of an implementation that uses the median mixture model algorithm, but it might as well be implemented with a different background modeling method underneath.

The united median mixtures model structurally consists of three independently working MMMs called “cores”. Each of the cores conducts the background modeling using a different set of input data that come from the same input video sequence. A flowchart of the UMM is shown in the Figure 2.

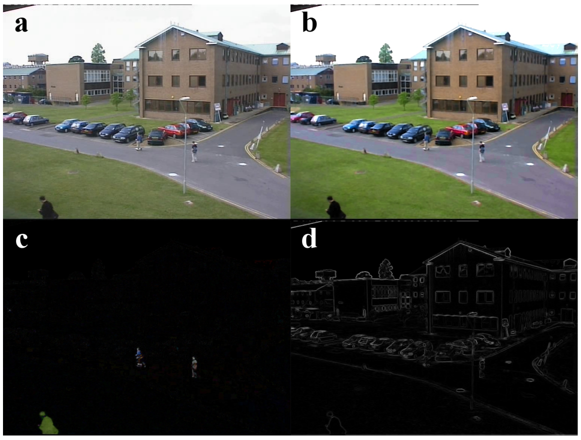

Figure 1. Inputs of the subsequent UMM's cores: a – input image [PETS01a], b – color core input, c – difference core input, d – gradient core input.

The first core, called the color core, models the background simply in the color space, so the input data goes only through the image quality enhancement. It consists of three parts: the median filter that minimizes the salt and pepper noises, the contrast enhancement and the saturation enhancement. The last two are conducted using factors that need to be empirically chosen beforehand. Comparing to the raw input, the input image of the first core is sharper and has better color dynamics. An example of such an input data is shown in the Figure 1b. The modeling results of this core are shown in the Figure 3b and are very similar to those

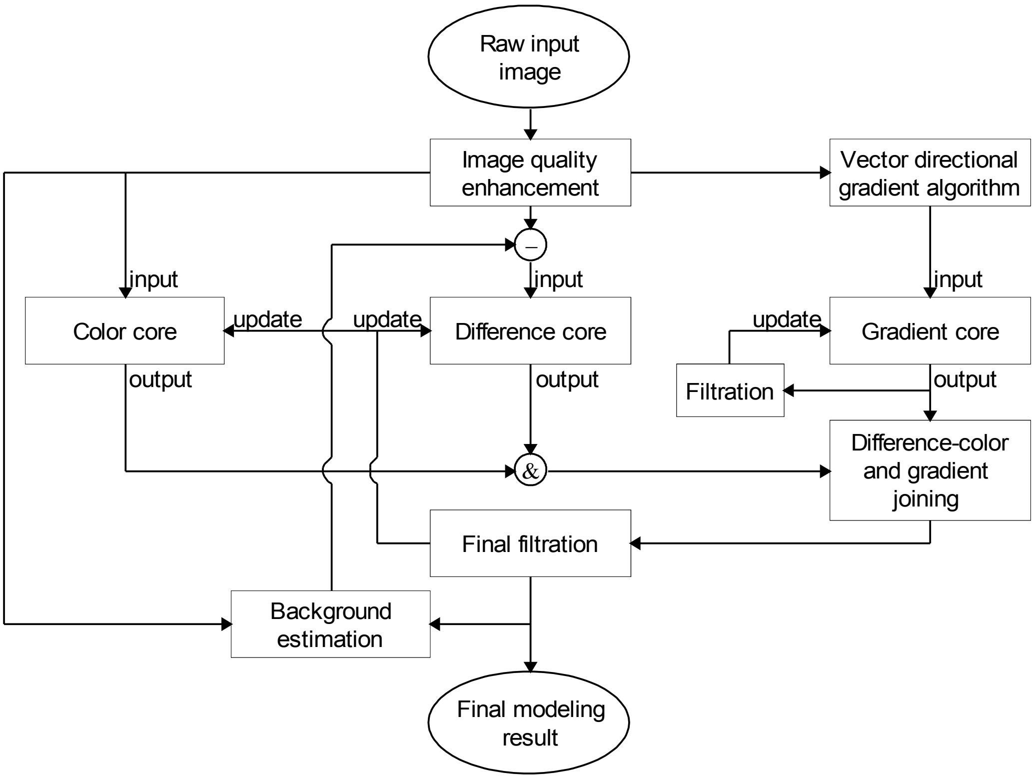

Figure 2. Flowchart of the united median mixtures algorithm

of the MMM or GMM methods as they also model background in the color space.

The input of the second or the difference core is a difference of the color core input frame and the estimated background frame. The background frame is estimated at the very end of the loop of the algorithm by coping to the background accumulator only those pixels of the currently analyzed frame that were classified as belonging to the background. All the other pixels that were held in the accumulator stay intact. This way the estimated background frame always contains the most recent background look. Usually, the input image of the difference core looks as if the static background was erased and everything that is left has distorted colors. An example of the input data for the second core is shown in the Figure 1c. The modeling results of this core are shown in the Figure 3c and are similar to those of the first one – the biggest differences occur in the distribution of the noise.

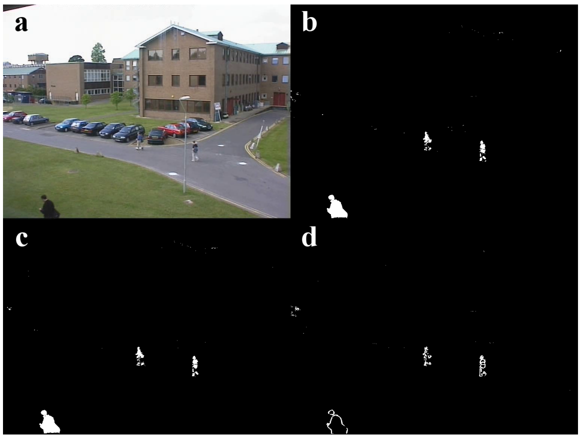

Figure 3. Outputs of the subsequent UMM's cores: a – input image [PETS01a], b – color core output, c – difference core output, d – gradient core output.

In order to join the results of the first two cores it is enough to calculate the logical conjunction of them as they are both in a binary form – white foreground and black background. The resulting difference-color image contain much less noise as the logical conjunction preserves only those values that occur in both operands.

The third core, called the gradient core, models the background in the image gradient space. Its input data is calculated from the color core input data by the vector directional gradient algorithm (VDG) [Luk06a]. An example of the input data for the gradient core is shown in the Figure 1d. The modeling results of this core are shown in the Figure 3d and they contain only the contours of the foreground objects. Those contours do not include shadows or reflections that occurred in the input image because those phenomena usually do not have sharp edges – their color gradient is relatively small.

Joining the third core results with the results of the other two cores leads to the elimination of the shadows and reflection areas misclassified by the first two cores as belonging to the foreground. The joining procedure consists of three steps:

- Gradient results filtration – filtering the gradient core results using a median filter and then using the filtered image as a marker image for the binary morphological reconstruction by dilation [Vin93a]. This two phase filtering approach removes all the noise the median filter removed and does not deteriorate the shape of the non noise objects by reconstructing them from the filtered image.
- Finding contours of the areas left after filtration – this step consists of a couple of sub-steps: dilating the filtered image, finding contours of the objects from the dilated image using the border following algorithm [Suz85a], finding convex hulls of the contours using the Sklansky algorithm [Skl82a], connecting the closely lying convex hulls with four iterations of the binary morphological close operation, and finally, finding convex hulls of the contours of the objects from the morphologically closed image. The result after performing all the sub-steps is a gradient contour image containing only the contours of the objects after the filtration.
- Separation and removal of the shadows and reflections areas from the difference-color image – closing the holes in a difference-color image with four iterations of the binary morphological close by reconstruction, separation of the shadows and reflections by subtracting the gradient contour image from the closed difference-color image and finally removing the separated shadows and reflections using the filtered gradient image as the marker image for the binary morphological reconstruction by dilation of the image with separated shadows and reflections.

The resulting difference-color joined image contains the objects from the difference-color image, less noise and no shadows or reflections areas.

After uniting the modeling results of all of the three cores, a final median filtering with a morphological reconstruction by dilation is conducted in order to eliminate any remaining noise, not deteriorating the shapes of the found objects.

Finally, the first two cores are updated with the filtered united results and the gradient core is updated with the filtered gradient results using the fed update feature. This way the cores are taught that all the foreground pixels that were removed from or added to their original results were misclassified and from now on they should be recognized correctly, which means that the final UMM result designate the correct classification result for the cores.

The united median mixtures method gives better results than the previously described algorithms. In particular, the UMM method is more accurate than the GMM method. The final results contain practically no noise at all and the shapes of the foreground objects are more complete. Thanks to the analysis of the gradient data the algorithm manages also to reduce the impact of the shadows and reflections on the final modeling results.

The main drawback of the current united median mixtures method implementation is its speed. The run time of the algorithm is about 5 to 7 times longer than for the median mixture model approach and about 3.5 to 6 times longer than for the GMM method. There are two reasons for that. The first one is the fact that the run time length depends on the number of the foreground objects in the currently analyzed scene. The more foreground objects there are, the more time consuming calculations the algorithm has to conduct. To be precise, the calculations in question are related to the removal of the shadows and reflections. Such a dependency does not occur neither in the MMM nor in the GMM.

The second reason for the long run time of the described UMM method is the fact that the implementation is still work in progress so there are a lot of additions to the code that make it possible to see and analyze the partial results of the algorithm. However, the main problem of the current implementation is that although each core works independently from the remaining ones, they are started sequentially, which means that their run times sum up, whereas they should be run in parallel, for example, on a different threads. This way the run time of all the cores would take up only as much as the run time of the slowest one.

The next part of this paper summarizes the results of the test and experiments that were conduced after implementing all of the described algorithms.

5. RESULTS

Comparative tests of the median mixture model and the Gaussian mixture model methods proved the initial assumptions to be correct. The results of both approaches are very similar in terms of the quality and accuracy and are obtainable in the real time. Figure 4 shows an outdoor scene with three moving objects – two pedestrians and one cyclist [PETS01a]. All of them should be recognized as belonging to the foreground and both algorithms managed to do that very well. Although the proposed approach is more prone to noise than the GMM method, in this case results of both algorithms are almost identical. However, the median mixture model is about 25% faster than the Gaussian mixture model, which makes it possible to model greater resolution video sequences or to add additional processing to the background modeling pipeline without compromising the possibility to work in the real time.

Figure 4. Comparison of the Gaussian mixture model and the median mixture model.

As it was mentioned before, the united median mixtures model gives even better results in terms of quality and accuracy than the Gaussian mixture model or the median mixture model. Figures 5 – 9 show and compare the behavior of all three approaches in different modeling conditions.

Figure 5. Comparison of all three methods in an extreme situation – lots of reflections, from top left: input image, the results of the GMM, the results of the MMM, the results of the UMM.

Figure 5 shows a comparison of all three methods in an extreme situation, when there are a lot of reflections in the scene caused by the headlights of the passing cars. In this particular situation, the GMM method misclassified the reflections that occurred on the road and on the pavement. The MMM method gave slightly better results – less reflections were misclassified due to the spatial correlation awareness of this approach. The UMM's results are undoubtedly the best. There is only a very small fragment of the headlights reflection misclassified just in front of one of the cars, where the intensity of the reflection was the highest. Moreover, the shapes of both cars recognized by the united median mixtures model are more complete and accurate than for the other two algorithms.

Figure 6. Comparison of all three methods in extreme situations – sudden light change, from top left: input image, the results of the GMM, the results of the MMM, the results of the UMM.

Figure 6 shows the input scene and the modeling results just after a sudden increase of the light intensity. The change caused all parts of the scene to become brighter, however it was the most noticeable in already bright areas like the wall of the building or the lines on the street. Those parts were misclassified by both the GMM and the MMM methods, whereas the UMM method did very well in this situation – it misclassified only one small spot corresponding to the brightest part of the wall. The UMM approach has also recognized the shape of a passing car as more complete than the two other algorithms.

Figure 7. Comparison of all three methods in extreme situations – reflection and 5 people, from top left: input image, the results of the GMM, the results of the MMM, the results of the UMM.

Figure 7 shows the input scene with a huge reflection just in the middle of the image caused by the headlights of a passing car. Besides the car there are also three people localized around the reflection and two people in the back. Both the GMM and the MMM methods recognized all those six moving objects correctly, but unfortunately neither of them managed to classify the reflection area correctly. On the other hand, the UMM method managed the reflection area very well, but it also filtered out two smallest objects, probably considering them as a noise.

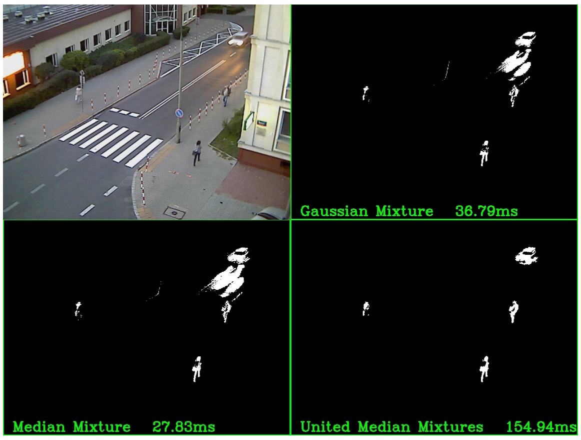

Figure 8. Comparison of all three methods in extreme situations – reflection and 3 people, from top left: input image, the results of the GMM, the results of the MMM, the results of the UMM.

Figure 8 shows the input scene similar to previously described scene from Figure 7. Three people are walking on the pavement and the headlights of a passing car cause a reflection on the street. Although this time all three algorithms recognized the moving objects very well, only the UMM method managed the reflection area correctly.

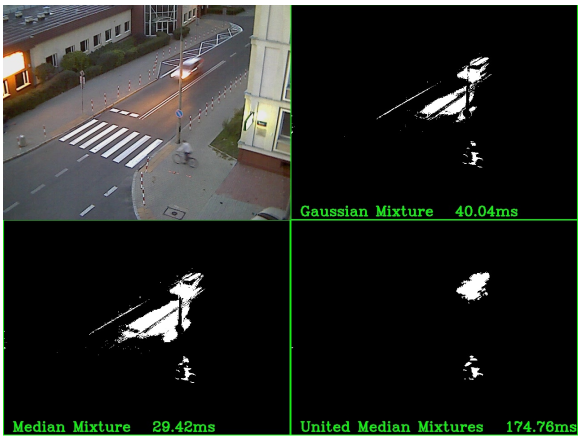

Figure 9. Comparison of all three methods in extreme situations – huge reflection and a fast moving cyclist, from top left: input image, the results of the GMM, the results of the MMM, the results of the UMM.

Figure 9 shows yet another input scene with a huge reflection in the middle caused by the headlights of a passing car. This scene contains also a cyclist who is moving very fast towards the pedestrian crossing. Similarly to the previously described situations only the united median mixtures model managed to classify the reflection area correctly. Moreover in this particular situation the UMM method recognized the shapes of the passing car and the cyclist more precisely than the Gaussian mixture model or the median mixture model despite the blur of those objects caused by their speed.

Generally, the conducted tests and experiments confirmed all the assumptions about the high processing speed of the median mixtures model and the very good quality and accuracy of the united median mixtures model.

6. CONCLUSION

The novel approaches to the background – foreground segmentation described in this paper closely rely on each other. As a matter of fact, the concept of uniting the background modeling results of different data sources came up first and then a need for a fast and robust background modeling algorithm emerged as it was necessary for the cores of the united model. A deep survey of the present solutions for the problem [Gra13a] led to a choice of the Gaussian mixture model as the archetype of the core algorithm. It was chosen because of its robustness and a broad spectrum of applications, which means it would not need special conditions to work correctly. From there it was all about making the algorithm faster, not loosing those two key features along the way. The result of that effort is the median mixture model.

Having the MMM as a fine candidate for the cores, it was possible to test the concept of uniting their results together. The specific solutions for joining the results of the cores were developed by an extensive testing and experimenting with different approaches. The methods that performed the best are the ones that are described in this paper.

The final shape of both the MMM and the UMM met the expectations. The median mixture model is as robust to noise and distortions as the Gaussian mixture model with much shorter run time. There is about 25% time gain compared to the GMM, which means that if for a certain video sequence the GMM method has about 20FPS, the MMM method would have almost 30FPS for the same video sequence.

The united median mixtures is still, as it was said before, work in progress hence the run times of the implementation in the tests. However, the results are very good compared to both the GMM and the MMM methods. In most cases the UMM is more precise in determining the final shapes of the foreground objects. The recognized shapes are more complete than in the other two methods. Although the normal spot noise and distortions do not have any impact on the final modeling results of all three approaches, the UMM method behave a lot better in deteriorated conditions when there are sudden light changes and intensifying noises or distortions. Such a resistance to noise sometimes effects in not recognizing the foreground objects that are to small – they are simply considered to be noise.

The further work needs to focus on improving the united median mixtures model implementation. As it was already mentioned, the main drawback of the approach is its run time. The conducted experiments show that the biggest improvement can be made by paralleling the independent code parts. The best example for this are the cores that currently work sequentially. Excluding the unnecessary code and making the rest of the implementation more efficient would also noticeably influence the run time. Summing up, the final results are more than satisfactory, but there is still more work to do.

7. REFERENCES

[Gra13a] Graszka P., Background Modeling in Video Sequences, Conf.proc. ICSS 2013, Vol.240, pp.195-205, 2013.

[Luk06a] Lukac R., Plataniotis K., Color Image Processing: Methods and Applications, CRC Press, pp.89-99, 2006.

[McF95a] McFarlane N., Schofield C., Segmentation and tracking of piglets in images, MVA, Vol.8, Issue 3, pp.187-193, 1995.

[PETS01a] PETS 2001 dataset, The University of Reading, UK, 2001.

[Skl82a] Sklansky J., Finding the Convex Hull of a Simple Polygon, Conf.proc. PRL Vol.1, Issue 2, pp.79-83, 1982.

[Sta99a] Stauffer C., Grimson W., Adaptive background mixture models for real time tracking, in Conf.proc. CVPR, pp.246-252, 1999.

[Suz85a] Suzuki S., Abe K., Topological Structural Analysis of Digitized Binary Images by Border Following, Conf.proc. CVGIP, Vol.30, Issue 1, pp.32-46, 1985.

[Vin93a] Vincent L., Morphological Gray Scale Reconstruction in Image Analysis: Applications and Efficient Algorithms, IEEE Transactions on Image Processing, Vol.2, No.2, 1993.