Open Journal of Applied Sciences, **2013, 3, 58-64**
Published Online March 2013 (http://www.scirp.org/journal/ojapps)

Performance Evaluation of Various Functions for Kernel Density Estimation

Youngsung Soh, Yongsuk Hae, Aamer Mehmood, Raja Hadi Ashraf, and Intaek Kim
Department of Information and Communication Engineering, Myongji University, Yongin, 449-728, Korea
Email: soh@mju.ac.kr

Received 2012

ABSTRACT

There have been vast amount of studies on background modeling to detect moving objects. Two recent reviews[1,2] showed that kernel density estimation(KDE) method and Gaussian mixture model(GMM) perform about equally best among possible background models. For KDE, the selection of kernel functions and their bandwidths greatly influence the performance. There were few attempts to compare the adequacy of functions for KDE. In this paper, we evaluate the performance of various functions for KDE. Functions tested include almost everyone cited in the literature and a new function, Laplacian of Gaussian(LoG) is also introduced for comparison. All tests were done on real videos with varying background dynamics and results were analyzed both qualitatively and quantitatively. Effect of different bandwidths was also investigated.

Keywords: Background Model; Kernel Density Estimation; Kernel Functions

1. Introduction

The detection of moving objects is one of the challenging problems in video surveillance system due to changes of natural phenomena occurred in a scene. Background subtraction is commonly used for detecting moving objects especially when background has not much change. The most important issue in background subtraction is maintaining background. Many background modeling techniques were proposed by researchers. Among them are running Gaussian average[3], GMM[4], KDE[5], and eigenbackground[6]. Excellent reviews of these techniques are presented in [1,2]. In [1,2], GMM and KDE were shown similar performance and outstrip others. For KDE, the selection of kernel functions and their bandwidths is important in that they determine the underlying probability distribution and thus the quality of background modeling. While surveying the literature, we found one relevant work on kernel function comparisons. Zucchini[7] compared five kernel functions for KDE. He argued that Epanechnikov function performed best. The performance measure used was mean integrated squared error(MISE). He derived the results only in theoretical manner and never tested on real video.

In this paper, we tested nine kernel functions where eight of them are frequently cited in the literature. One new function, LoG, is introduced for comparison. All tests were done on real videos with varying background dynamics and results were analyzed both qualitatively and quantitatively. Effect of different bandwidths was also investigated. For quantitative comparison, we used recall and precision as performance measures and ROC curves were drawn to show the results.

The paper is structured as follows. In section 2, we describe the related work. Proposed method is introduced in section 3. Experimental results and analysis are explained in section 4. Finally section 5 gives conclusion and future work.

2. Related Works

Zucchini [7] compared the performance of five kernel functions for KDE. They are Epanechnikov, Gaussian, uniform, triangular, and bi weight functions. He used MISE as a performance measure. We follow the notations used in [7] to explain his approach below. Mean squared error(MSE) of estimated function is given as Equations (1), (2), and (3).

$$MSE(\hat{f}(x)) = E(\hat{f}(x) - f(x))^2 \quad (1)$$

$$MSE(\hat{f}(x)) = (E(\hat{f}(x) - f(x))^2 + E(\hat{f}(x) - Ef(x))^2 \quad (2)$$

$$MSE(\hat{f}(x)) = Bias^2(\hat{f}(x)) + Var(\hat{f}(x)) \quad (3)$$

Here $f(x)$ and $\hat{f}(x)$ represent original probability density function and estimated probability density function respectively. Bias and variance are two components of MSE. The theoretical derivation of bias and variance can be found in [7] and given as,

$$Bias(\hat{f}(x)) \approx \frac{h^2}{2} k_2 f^{//}(x) \quad (4)$$

$$Var(\hat{f}(x)) \approx \frac{1}{nh} f(x) \int K^2(z)dz \quad (5)$$

, where n and h represent the number of previous samples and bandwidth respectively. Substituting Equations (4) and (5) for Equation (3), we get

$$MSE(\hat{f}(x)) = \frac{1}{4} h^4 k_2^2 f^{//}(x)^2 + \frac{1}{nh} j_2 \quad (6)$$

, where $k_2 \approx \int z^2 K(z)dz$ and $j_2 \approx \int k(z)^2 dz$. Global accuracy of $\hat{f}(x)$ is MISE defined as in Equations (7) and (8).

$$MISE(\hat{f}(x)) = \int_{-\infty}^{\infty} Bias^2(\hat{f}(x))dx + \int_{-\infty}^{\infty} Var(\hat{f}(x))dx \quad (7)$$

$$MISE(\hat{f}(x)) = \frac{1}{4} h^4 k_2^2 \int f^{//}(x)^2 dx + \frac{1}{nh} j_2 \quad (8)$$

MISE measure is used to quantify the performance of the estimator. Optimal bandwidth can be calculated by minimizing the Equation (8) with respect to h and is given as

$$MISE_{OPT}(\hat{f}) = \frac{5}{4} \left(\frac{\beta(f) j_4^2 k_2^2}{n^4}\right)^{1/5} \quad (9)$$

, where $\beta(f) = \int f^{//}(x)^2 dx$. Wand and Jones[8] used MISE given in Equation (9) to measure the performance of various kernel functions and found that Epanechnikov kernel is the best.

Assuming the efficiency of Epanechnikov function is 100% , the efficiency of other kernels were calculated and given as in **Table 1**. As can be seen, not much difference is observed among various kernel functions though Epanechnikov function achieves the best.

Table 1. Kernel functions and their efficiencies

Kernel	Efficiency
Epanechnikov	100%
Bi weight	99.39%
Triangular	98.59%
Gaussian	95.12%
Rectangular	92.95%

3. The Proposed Method

We follow the notations used in [5] to explain the proposed method. Let $x_1, x_2, \ldots x_N$ be previous N samples of intensity values for some pixel. Given these samples, KDE is used to estimate probability density at any intensity value of the pixel. Let x_t be an intensity value of the pixel at time t. Then we can estimate probability density for pixel value x_t as in Equation (10).

$$P_r(x_t) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(x_t - x_i) \quad (10)$$

where K is a kernel function and σ is bandwidth. For more than one dimension, Equation (11) is used.

$$P_r(x_t) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} K_\sigma(x_{t_j} - x_{i_j}) \quad (11)$$

, where K_σ is a kernel function for d dimensional space. In our work, we assume d = 1. The pixel is considered to be foreground if the above probability estimate is less than some threshold value.

3.1. Kernel Functions

Kernel function *K (t)* described in Equation (11) should satisfy three conditions . They are :

1) *K (t)* >= 0,
2) *K (t)* should be symmetric, and
3) $\int K(t)dt = 1$.

We collected almost all the kernel functions cited in the literature that were used for KDE. There were eight candidate functions: uniform, triangular, quartic, tri weight, tri cube, cosine, Epanechnikov, and Gaussian functions. We add one more function, LoG, for comparison. Their names, formula with value range, and graphs are given in **Table 2**. For LoG, since negative value violates condition 1) above, we use the range where the function value is nonnegative.

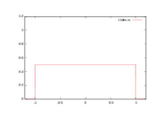
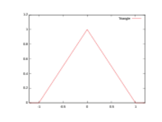
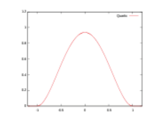
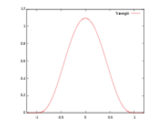
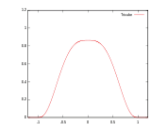
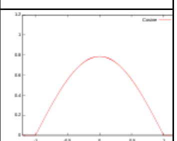
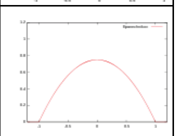
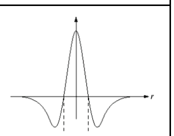
3.2. Selection of Threshold

Elgammal, Duraiswami, Harwood, and Davis[5] seemed to select threshold value empirically for Gaussian kernel to differentiate between background and foreground. Threshold selection guideline for all other kernels we considered in this paper is rarely found in the literature. Intensive empirical study led us to the conclusion that around 85% of the maximum probability density value that each kernel function can provide gave the best results. To reduce the computation time that is the major drawback of KDE, we built lookup table having pre-calculated function values for all possible domain values for each kernel.

3.3. Selection of Bandwidth

Elgammal, Duraiswami, Harwood, and Davis [5] showed how to select optimal bandwidth for Gaussian kernel. However, bandwidth selection guideline for all other kernels we considered is rarely found in the literature. Thus we empirically chose several bandwidth values and analyze the behavior of each kernel function.

Table 2. Kernel functions, formula, and their graphs

Kernel	Equation K(u)=	Diagram
Uniform	$\left\{\begin{matrix} 1/2 \quad 1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Triangu-lar	$\left\{\begin{matrix} 1-\lvert \mu \rvert \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Quartic	$\left\{\begin{matrix} \frac{15}{16}(1-u^2)^2 \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Tri weight	$\left\{\begin{matrix} \frac{35}{32}(1-u^2)^3 \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Tri cube	$\left\{\begin{matrix} \frac{70}{81}(1-\lvert u \rvert^3)^3 \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Cosine	$\left\{\begin{matrix} \frac{\pi}{4}\cos(\frac{\pi}{2}u) \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Epa-nechni-kov	$\left\{\begin{matrix} \frac{3}{4}(1-u^2) \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	
Lapla-cian of Gaussian	$\left\{\begin{matrix} \frac{1}{\pi}(1-\frac{u^2}{2})\ e^{-\frac{u^2}{2}} \ =1 \quad if\ \lvert u \rvert \le 1 \\ 0 \quad Otherwise \end{matrix}\right\}$	

4. Experimental Results

In this section, we compare the performance of various kernel functions for KDE both qualitatively and quantitatively. We use three test video sets that were used for the competition of background and foreground separation in VSSN2006 Conference[9]. Following are abbreviations for each data set.

STATIC : Background is almost static.(749 frames)

MILD : Background has mild dynamic behavior.(749 frames)

SEVERE : Background has severe dynamic behavior.(819 frames)

For all three, background is real and foreground is artificial, i.e., graphically generated objects are inserted and animated in real background video. The reason for doing this is in the easiness of getting ground truth. The size of the image is 320x240 for all test videos.

4.1. Qualitative Analysis

Figure. 1 shows the result for STATIC. **Figure. 1(a)** is

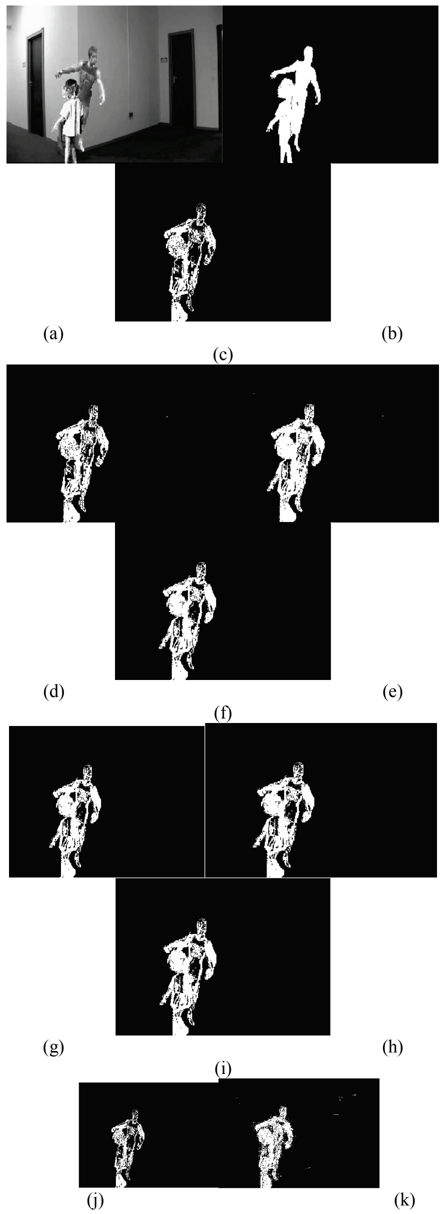


Figure 1. (*a*) Original frame, (*b*) Ground Truth, (*c*) Triangular, (*d*)Gaussian, (*e*) Epanechnikov, (*f*) Quartic, (*g*) Tri weight, (*h*) Tri cube, (i) Cosine, (j) Uniform, and (k) Laplacian of Gaussian

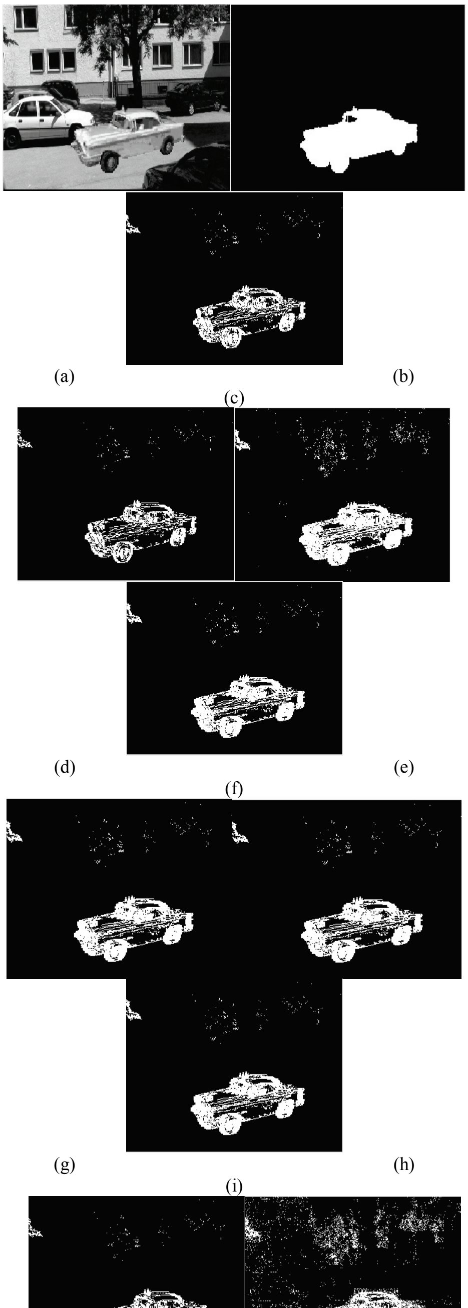


Figure 2. (*a*) Original frame, (*b*) Ground Truth, (*c*) Triangular, (*d*)Gaussian, (*e*) Epanechnikov, (*f*) Quartic, (*g*) Tri weight, (*h*) Tri cube, (i) Cosine, (j) Uniform, and (k) Laplacian of Gaussian

the original frame, 1(b) the ground truth, and 1(c) through 1(k) the foreground detection results for nine kernel functions respectively. Bandwidths used for all kernel functions were set to 20**.** Since the video contains almost no dynamic background activities, all functions performed about equally well, though uniform kernel found somewhat less true positives.

Figure. 2 and **Figure. 3** depict the results for MILD and SEVERE videos respectively. Conventions for each image in **Figure. 2** and **Figure. 3** are the same as the one for **Figure. 1**. LoG seems to be the worst in terms of de tecting false positives and uniform kernel performed

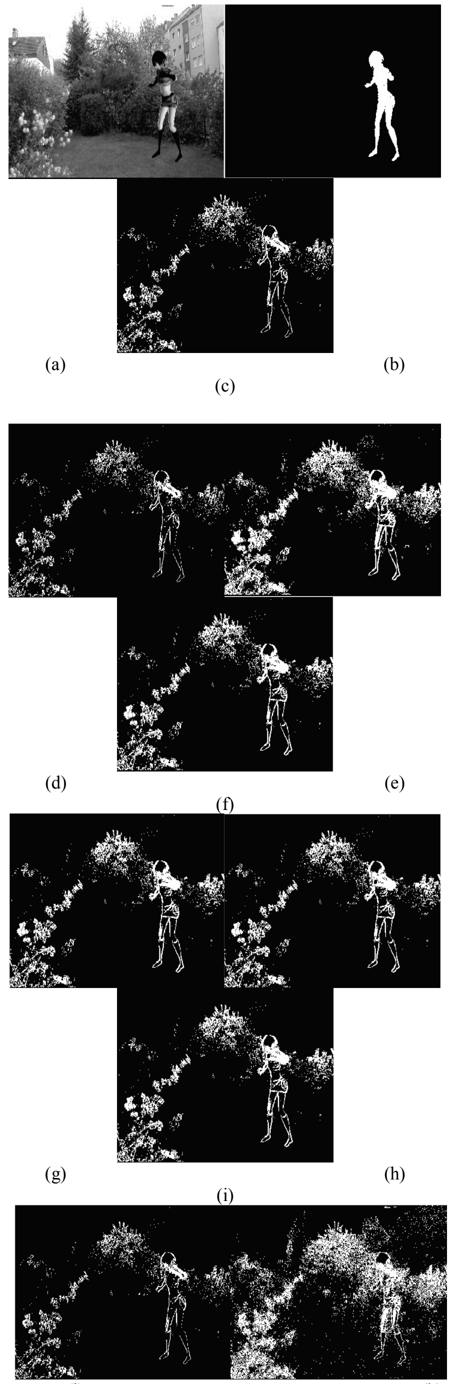


Figure 3. (*a*) Original frame, (*b*) Ground Truth, (*c*) Triangular, (*d*)Gaussian, (*e*) Epanechnikov, (*f*) Quartic, (*g*) Tri weight, (*h*) Tri cube, (i) Cosine, (j) Uniform, and (k) Laplacian of Gaussian

worst in terms of detecting true positives. All the other kernels performed equally well except Epanechnikov kernel where more false positives are seen. Since it is not easy to compare the performance exactly just by visual observation, we resort to quantitative comparison in next section.

4.2. Quantitative Analysis

Figure. 4 shows the ROC curves for nine kernel functions for STATIC. Horizontal axis and vertical axis correspond to recall and precision respectively. **Figure. 5** and **Figure. 6** depict the ROC curves for nine kernel functions for MILD and SEVERE respectively and axis convention is the same as the one in **Figure. 4**. Uniform kernel was the worst and cosine kernel seems to be the best for all the videos. Among the others, LoG and Gaussian kernels showed relatively poor performance. As we go from STATIC to MILD and from MILD to SEVERE, all kernels performance deteriorated due to increasing background dynamics.

4.3. Bandwidth Analysis

Figure. 7 shows the ROC curves of different bandwidths for each kernel function for STATIC. Bandwidths tested were 20, 40, and 60. For all kernels bandwidth of 20

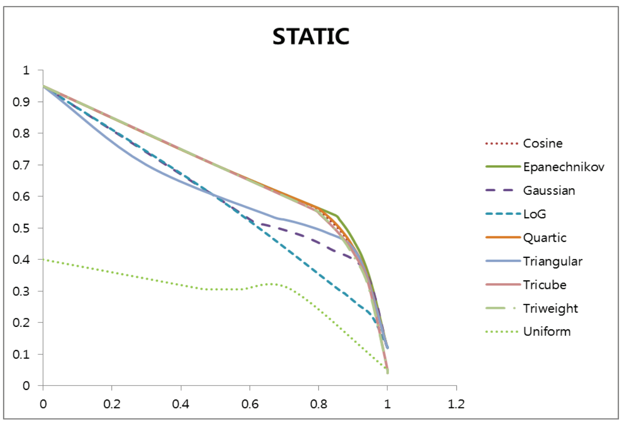


Figure 4. ROC Curves for STATIC

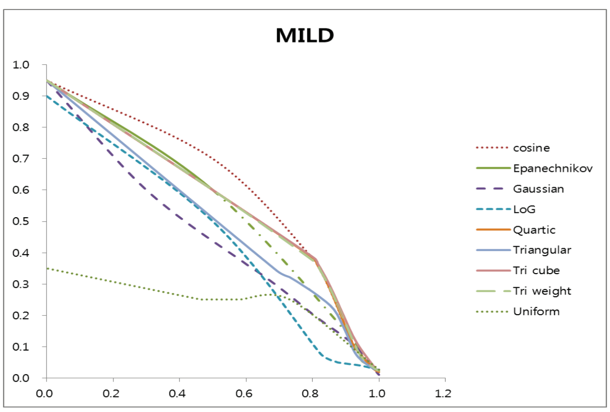


Figure 5. ROC Curves for MILD

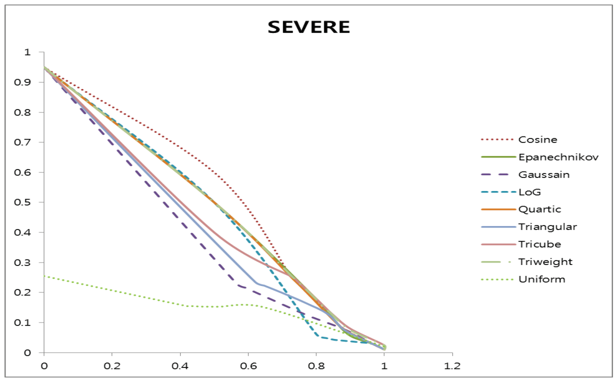


Figure 6. ROC Curves for SEVERE

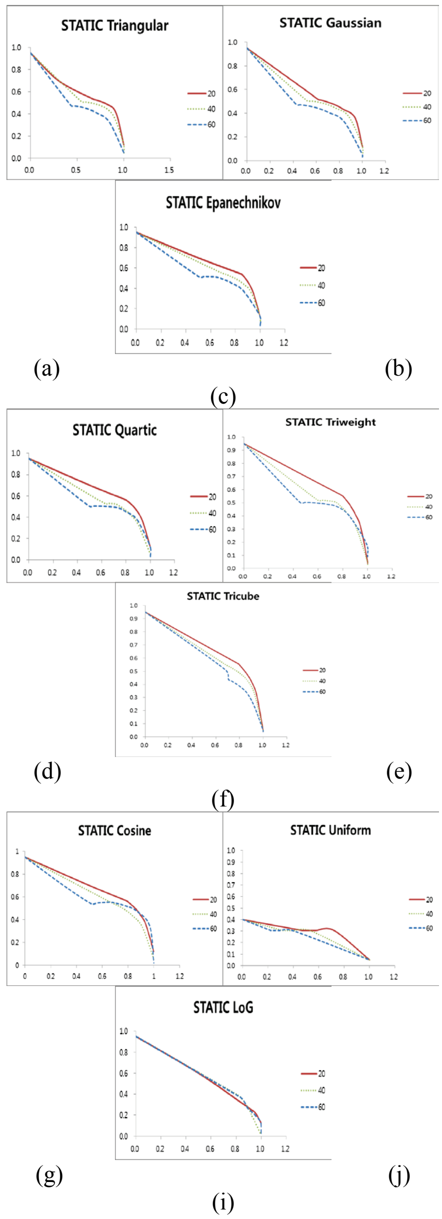
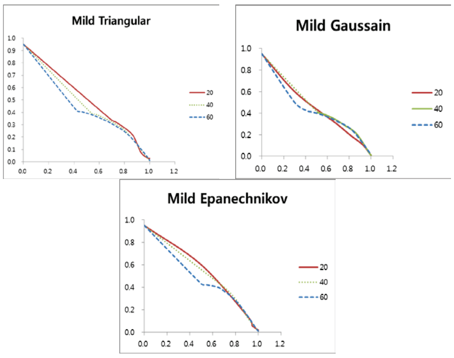
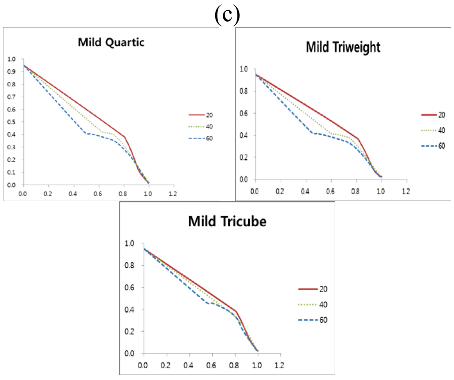


Figure 7. (*a*) Triangular, (*b*)Gaussian, (*c*) Epanechnikov, (*d*) Quartic, (*e*) Tri weight, (*f*) Tri cube, (*g*) Cosine, (*j*) Uniform, and (*i*) Laplacian of Gaussian



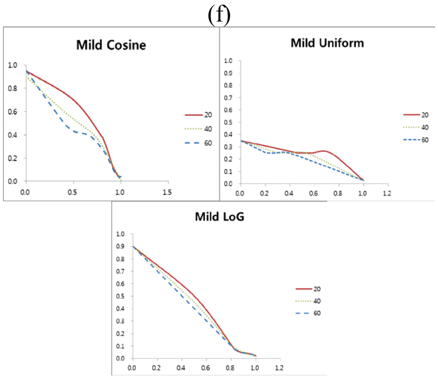
(a) (b)

(c)



(d) (e)

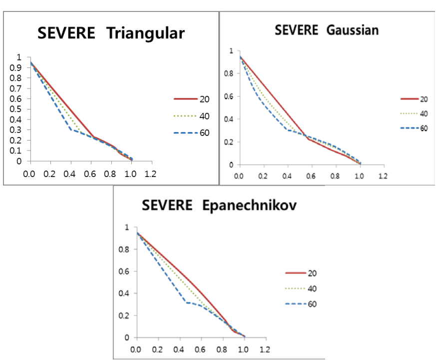
(f)



(g) (j)

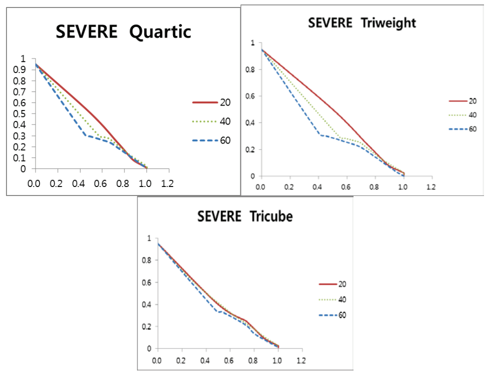
(i)

Figure 8, (*a*) Triangular, (*b*)Gaussian, (*c*) Epanechnikov, (*d*) Quartic, (*e*) Tri weight, (*f*) Tri cube, (*g*) Cosine, (*j*) Uniform, and (*i*) Laplacian of Gaussian



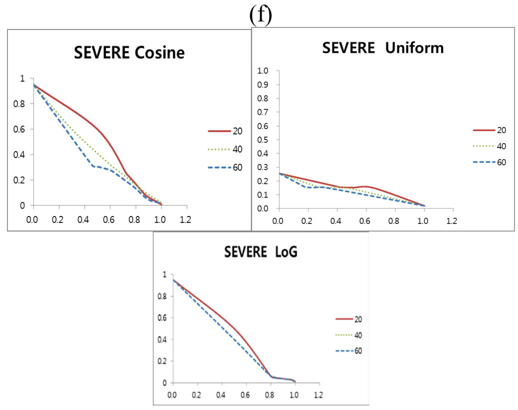
(a) (b)

(c)



(d) (e)

(f)



(g) (j)

(i)

Figure 9, (*a*) Triangular, (*b*)Gaussian, (*c*) Epanechnikov, (*d*) Quartic, (*e*) Tri weight, (*f*) Tri cube, (*g*) Cosine, (*j*) Uniform, and (*i*) Laplacian of Gaussian

showed best result and as bandwidth increases, the performance gets worse. **Figure. 8** and **Figure. 9** depict the ROC curves of different bandwidths for each kernel function for MILD and SEVERE. We can observe the same performance characteristic as in STATIC.

5. Conclusion

KDE, along with GMM, is known to be the best background modeling method. The performance of KDE greatly depends on kernel functions and their bandwidths. In this paper, we analyzed the performance of nine kernel functions on real videos having various levels of background dynamics. Eight out of nine kernel functions were collected through literature survey and one more kernel function, LoG, was added for comparison. Through quantitative analysis, we found that cosine kernel performed best and, LoG and uniform kernels were worst. All other kernels were in between. By bandwidth analysis, we found that bandwidth of 20 performed best and as bandwidth increases, the performance deteriorates.

In this work, all the thresholds were selected empirically. It would give better results if automatic selection of thresholds is possible. This is intended for future research.

6. Acknowledgment

This work(Grants No. C0005448) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded by Korea Small and M.

REFERENCES

[1] M. Piccardi, “Background subtraction techniques: a review,” 2004 IEEE International Conference on Systems, Man, and Cybernetics, 2004, pp.3099-3104

[2] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger,“Review and evaluation of commonly implemented background subtraction algorithms,” 19th International Conference on Pattern Recognition, 2008 ,pp.1- 4

[3] C. Wren, A. Azarbayejani, T. Darrel, and A.Pentland,“Pfinder: real-time tracking of the human body,” IEEE Transactions on Pattern analysis and Machine Intelligence, vol.19, no.7, 1997, pp.780-785.

[4] P. Wayne Power, Johann A. Schoonees, “Understanding Background Mixture Models for Foreground Segmentation,” Proc. Image and Vision Computing New Zealand, 2002.

[5] A. Elgammal, R. Duraiswami, , D. Harwood, and L. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance,” Proc. of IEEE, VOL. 90, NO. 7, 2002

[6] N. Oliver, B. Rosario, and A. Pentland, “A Bayesian computer vision system for modeling human interactions,” IEEE Transactions on Pattern analysis and Machine Intelligence, vol.22, no.8, 2000, pp.831-843.

[7] W. Zucchini, Applied smoothing techniques, Part 1 Kernel Density Estimation., 2003

[8] M. P. Wand ,M. C. Jones, Kernel Smoothing, Monographs on Statistics and Applied, Probability Chapman & Hall, 1995

[9] http://mmc36.informatik.uni-augsburg.de/VSSN06_OS

