

Monte Carlo Based Algorithm for Fast Preliminary Video Analysis

Krzysztof Okarma and Piotr Lech

Szczecin University of Technology,
Faculty of Electrical Engineering,
Chair of Signal Processing and Multimedia Engineering,
26. Kwietnia 10, 71-126 Szczecin, Poland
{krzysztof.okarma,piotr.lech}@ps.pl

Abstract. In the paper a fast statistical image processing algorithm for video analysis is presented. Our method can be used on colour as well as grayscale or even binary images. The main component of the proposed approach is based on statistical analysis using the Monte Carlo method. A video's statistical information is acquired by specifying a logical condition for the Monte Carlo technique. The results of the algorithm depend on the correct choice of threshold values; thus the application area is limited by the adaptability of the thresholds to videos with large heterogeneity: e.g. videos with objects moving into and out of the scene, rapidly varying illumination, etc.

Keywords: statistical image analysis, Monte Carlo method.

1 Description of the Method

For a static image of analysed scene a constant value can be defined related to the number of pixels fulfilling specified logical condition. Such condition can be defined e.g. as the belonging of the image sample to the specified luminance range. In such case the algorithm works as the area estimator for the objects fulfilling the specified luminance criterion.

In the effect of analysis of the whole image the corresponding binary image is created which stores the values equal to 1 for the samples which fulfil the condition and 0 for the others. It gives the quantitative information related to the object's features described by the logical condition which can be obtained by the summation of all "ones".

The estimator \hat{L} given as:

$$\hat{L} = m, \quad (1)$$

where m stands for the total number of "ones" in the binary image, can be related to the area of the single object located in the empty scene, assuming the object's pixels fulfil the specified logical condition. After some additional morphological operations it is possible to easily estimate some other parameters such as object's perimeter, diameter, moments etc.

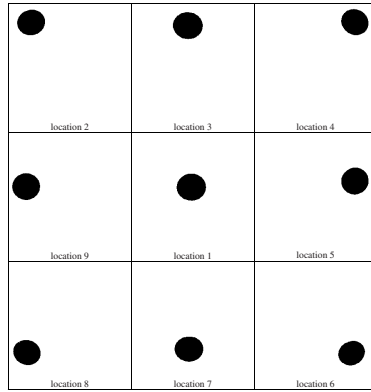


Fig. 1. The example of the distortions caused by camera lens

Counting of all "ones" for high resolution images may be time consuming because analysis of all image samples is required. In order to increase the speed of the algorithm the reduction of the number of analysed samples is sufficient. In that case application of a statistical experiment using the Monte Carlo method is useful. The number of analysed points is equal to the number of draws using pseudo-random generator with uniform distribution. Binary image samples can be stored in one-dimensional vector, numbered from 1 to N , where N is the total number of samples in the scene. Then n independent draws (with returns) are performed from the vector and the number of "ones" drawn (k) is stored. Estimated number of "ones" is equal to

$$\hat{L}_{MMC} = \frac{k}{n} \cdot N, \tag{2}$$

where: k - the number of "ones" drawn, n - the number of draws, N - the total number of samples.

The estimation error can be expressed as:

$$\varepsilon_\alpha = \frac{u_\alpha}{\sqrt{n}} \cdot \sqrt{\frac{K}{N} \cdot \left(1 - \frac{K}{N}\right)}, \tag{3}$$

where: u_α - the value denoting two-sided critical range, K - total number of "ones" in the entire image.

The considerations presented above are correct for the generator with uniform distribution. The prevention of the error's increase requires good statistical properties of the generator.

2 Influence of Geometrical Distortions

In order to simplify the considerations it is assumed that the goal of the algorithm is the estimation of the object's area. The sources of the geometrical distortions

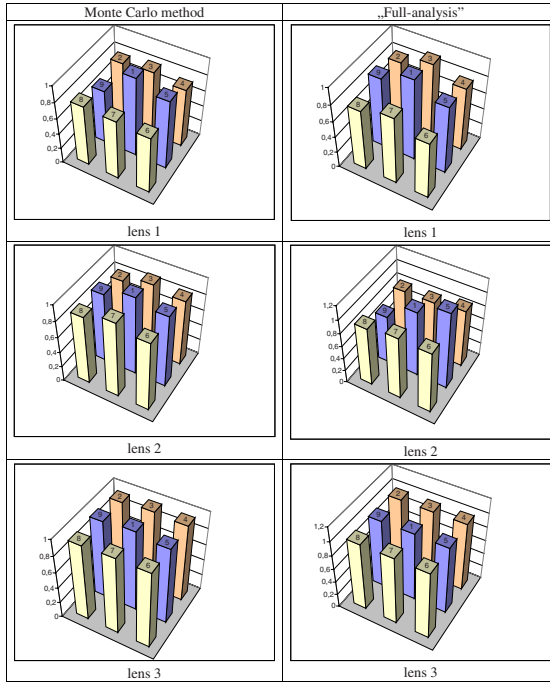


Fig. 2. The comparison of the normalised results for various locations and camera lens

are mainly related to the camera optics and may be observed mostly near the image corners as shown in Fig. 1.

The experiment has been performed using the digital camera with adjustable optical parameters (various lens), the first test with the most visible distortions and the third one with almost invisible ones. The camera has been installed directly over the scene containing single object of known dimensions. In this experiment the scene has been constantly lightened by uniformly distributed light in order to eliminate the influence of its changes. The location of the object has been changed (see Fig. 1).

The comparison of the results obtained using three different lens for the classical "full-analysis" procedure (counting of all the pixels belonging to the object) and the Monte Carlo method is illustrated by Fig. 2. All presented results are normalised assuming the exact object's area is equal to 1. They show the relevance of the influence of lens quality for both methods and using the Monte Carlo method with good quality lens does not introduce significant errors. In systems with poor quality cameras some additional digital image correction algorithms may be needed to ensure high accuracy.

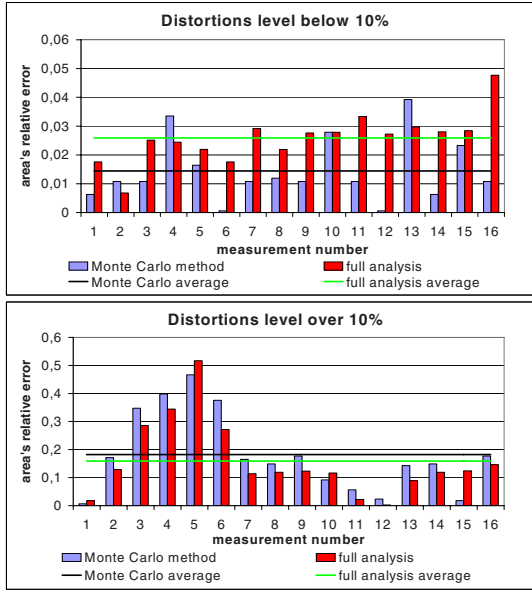


Fig. 3. Relative errors of area estimation for distortions level below 10% and over 10%

3 Influence of Lighting Conditions

Regardless of the static errors dependent on the lens quality, in most practical applications some dynamic ones, caused by changes of local or global lighting conditions, may also appear. In order to analyse the influence of lighting it is assumed that the object is located in the centre of the scene and the best lens is used.

The experiment has been performed for a small amount of distortions (below 10% of distorted pixels from the whole scene comparing to the uniformly lightened scene) and the significantly higher number of distorted pixels. All the distortions have been caused by the light changes with the direct influence on the number of pixels fulfilling the same logical condition in each case. The threshold value of 10% distorted pixels has been chosen experimentally for specified lens and the analysed static scene for the better illustration of observed effects.

During the experiment 16 measurements have been performed in various lighting conditions (changes of the number of light sources, their locations and parameters). The results obtained for two cases described above are presented in Fig. 3.

Comparing the relative errors obtained for two cases analysed in the paper it is worth noticing that for higher amount of distortions the Monte Carlo approach leads to similar results comparing to "full analysis". However, for lower distortion level the advantage of the Monte Carlo method is much better visible.

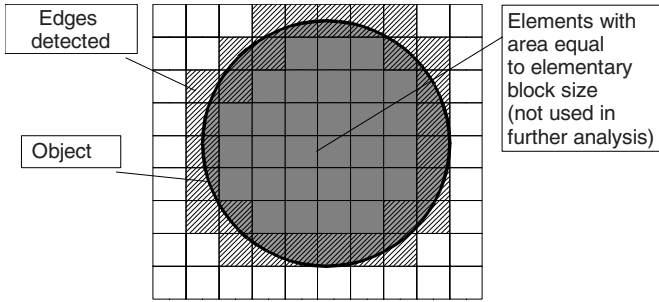


Fig. 4. Idea of the edge detection using the Monte Carlo approach

4 Applications of the Method

Universality of the Monte Carlo method makes possible to use it in many areas of digital image and video analysis applications. There are many articles and books related to more or less advanced statistical techniques applied for video analysis published in the recent years. A popular approach seems to be the usage of Sequential Monte Carlo [5] or Markov Chain Monte Carlo methods [9] e.g. for video text segmentation [1] as well as some tracking purposes [6]. Nevertheless, such algorithms usually require relatively high computational power so their applications in some real-time systems is limited. Apart from that, a good example of reasonable requirements (Pentium PC) can be the real-time low bit-rate video segmentation approach presented in the paper [4] also designed for specific applications.

The most crucial features of proposed approach are its easy implementation and low computational complexity. All the applications where analysis of the whole image is not necessary and the high accuracy of results is not required are the potential field of its usage.

4.1 Estimation of Geometrical Features

A typical application of the Monte Carlo method is the area estimation. However, presented idea can be applied also for the estimation of some other geometrical features. The simplest one is the perimeter estimation by edge detection. Analysed binary image should be divided into $T \times S$ elements of $r \times r$ pixels each using the square grid with the usage of the Monte Carlo method for the area estimation. All the blocks with the area equal to zero or equal to the size of elementary block are not used in further analysis because they do not correspond to the figure's edge as shown in Fig. 4.

In the next step the area of each object's fragment in elementary square elements is calculated. The estimated values are stored in the array of $T \times S$ elements.

On the base of the binary image the array K of the same size is created. The elements of that array have the following values: zero if the corresponding element's value in the binary image is equal to the size of elementary block

(all pixels belong to the object) and none of its neighbouring blocks (using 8-directional neighbourhood) has zero value, zero if the corresponding element's value in the binary image is equal to zero (background), one for the others (representing the edge). The array K shows the projection of the edges detected from the source image. Counted number of non-zero elements of array K represents the estimated value of object's perimeter expressed in the squares of $r \times r$ pixels.

In order to obtain a better estimation in the final step the number of square elements should be increased (smaller values of parameter r) and then the first steps are repeated. Using additionally the array K achieved in the third step, the analysis of blocks in the binary image with zero values is not needed, so further analysis is performed only for strongly limited number of indexed elements corresponding to the edge obtained in the third step.

The limit accuracy of the algorithm is determined by the size of the elementary block equal to 1 pixel what is equivalent to using convolution edge detection filters.

Dividing the scene into smaller squares it is possible to easily estimate some motion parameters such as direction and velocity applying the Monte Carlo procedure for each block. If the whole binary image is divided into $T \times M$ square blocks containing $r \times r$ pixels each, there is also a possibility to estimate some additional geometrical parameters which may be treated as local (e.g. mean diameter or average area) or global ones (e.g. number of objects inside the given area).

For a single object on the image plane the most interesting parameters are those ones which are insensitive to image deformations introduced during acquisition and typical geometrical transformations such as scaling, translation and rotation. In such sense the usefulness of the simplest parameters, such as area or perimeter, is strongly limited but many other factors, such as moments, can be often determined on the basis of the simplest ones. Some typical geometrical parameters used in the image analysis are horizontal and vertical projection's lengths (easily extended by the analysis of the presence of concavities in the object's shape) and Feret's diameters as the measure of horizontal and vertical object's maximum size [3].

An interesting group of parameters are the shape coefficients because of the possibility of fast estimation and wide opportunities of their usage for the classification and recognition purposes. Most of them can be easily calculated on the base of area, perimeter or Feret's diameters. Another group of parameters used during the binary image analysis is represented by the linear moments e.g. the first order ones related to the object's centre of gravity and the second order moments used as the object's inertia measures.

The binary array used in the Monte Carlo approach can be treated as the equivalent of reduced resolution one so geometrical parameters can be expressed in blocks of $r \times r$ pixels instead of pixels. The processing time is then significantly shortened due to the reduction of the number of analysed pixels and performed

Table 1. Area and perimeter values (in pixels) obtained in the experiments for various numbers of analysed points in the image and in the block respectively

Method Points (N)	Full image Monte Carlo					Monte Carlo 32x32 pixels				
	100	200	500	1000	5000	10	20	50	100	500
Area	31518	26127	29030	28698	29528	32957	31938	30945	30530	30276
Perimeter1	—	—	—	—	—	640	672	736	768	768
Perimeter2	—	—	—	—	—	437	417	440	436	434

experiments showed a good accuracy of their computation (sufficient for some typical image recognition applications).

Performed tests have shown that proposed approach can be valuable for the estimation of some shape coefficients which are dependent on the area and the perimeter. In that case the estimation errors depend on the accuracy of the area and perimeter estimation. It is worth noticing that the approximation of moments and geometrical parameters based on the perimeter requires the usage of the Monte Carlo method with division of the image into blocks while for the geometrical parameters based only on the area or projection lengths the standard Monte Carlo method is sufficient. In order to determine the projection lengths the additional storing of the minimum and maximum coordinates of the analysed pixels representing the object is necessary.

The estimation of the moments require slightly more sophisticated algorithm because the accuracy limited to the size of the block ($r \times r$ pixels) is not sufficient for most applications. The simplest solution is similar to the technique used for the perimeter estimation. Blocks corresponding to the object’s contour can be divided into smaller elements and then the adjustment vectors for the moments should be calculated.

Table 1 illustrates the results of the statistical experiment performed for the estimation of the area and the perimeter of a single object located in the scene. The perimeter has been estimated using two approaches: counting of all 32x32 pixels blocks representing the contour (Perimeter1) and using the square root of the estimated area for each block (Perimeter2). The first method leads to the overestimation and the second one produces too small values. Correct values of the area and the perimeter obtained by the analysis of all pixels are 30286 and 552 pixels respectively.

4.2 Fast Image Quality Estimation

Conventional objective full-reference image quality metrics, mainly based on the Mean Square Error [2], are poorly correlated with the Human Visual System so some other proposals have been presented in recent years. One of the most popular seems to be the Universal Image Quality Index proposed by Wang and Bovik [7]. Such measure models image distortions as the combination of three elements: loss of correlation, luminance distortion and loss of contrast. Assuming $x_{i,j}$ and $y_{i,j}$ are the values of the luminance for the pixel (i, j) of the original

and distorted image respectively, it is defined as the local index for the single image block (NxN pixels - usually N=8) as

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2) \cdot [(\bar{x})^2 + (\bar{y})^2]}, \tag{4}$$

where \bar{x} and \bar{y} are the mean values and σ stands for the standard deviation in the original and distorted image blocks respectively [7]. The overall quality index is defined as the mean value of metrics (4) obtained for all blocks using sliding window approach.

In the paper [8] the definition (4) has been extended into the Structural Similarity (SSIM) introducing the possibility of choosing the importance exponent for each of three factors in Eq. 4 with additional stability enhancement for the regions where \bar{x} or σ_x^2 are close to zero. The modified expression based on the usage of two coefficients preventing such instability can be expressed as

$$SSIM = \frac{(2\bar{x}\bar{y} + C_1) \cdot (2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_1) \cdot [(\bar{x})^2 + (\bar{y})^2 + C_2]}, \tag{5}$$

where C_1 and C_2 are the small values chosen experimentally as suggested by authors of the paper [8].

However, analysis of all the image pixels is time consuming and in many applications the exact image quality assessment is not the most crucial element, because the image quality estimation should be performed fast and not necessarily very accurately. Besides all the image quality metrics should be actually treated as estimators, because there is no ideal objective image quality measure.

For the applications where the image quality estimation should be fast enough to avoid the introduction of additional delays the calculation of the SSIM index using the Monte Carlo approach is proposed. Estimation of the local SSIM index should be performed only for some randomly chosen pixels inside the current sliding window. Assuming a good quality of the pseudo-random generator the expected number of the drawn pixels inside the window in each position should be almost the same. The advantage of that approach is an equal chance to analyse each pixel of the image so there is no need to use any sophisticated method for the decrease of the resolution in order to preserve some patterns.

The results of the image quality estimation using proposed approach for the test images after low-pass and median filtration, JPEG compression and contamination by an achromatic impulse noise are shown in the Tables 2 - 4. The usage of the limited number of randomly distributed pixels during the calculations of the Structural Similarity index may lead to a good quality estimation, assuming the pseudo-random generator with the uniform distribution, even for a low number of samples for some typical distortions. The results obtained for the image 'Baboon' (Table 3) differ from the other images because of the specific character of the image with many details. It leads to the lower quality index for the lossy JPEG compression (many pixels differ from their originals) and a better one for the images contaminated by an impulse ('salt and pepper') noise.

Table 2. The SSIM index obtained for various distortions using the Monte Carlo approach with various number of points - the image 'Kodim'

Number of points	low-pass	low-pass	median	median	5%	20%	JPEG	JPEG
	3x3	5x5	3x3	5x5	noise	noise	60%	10%
50	0.9280	0.8581	0.9418	0.8607	0.0880	0.0613	0.9448	0.8495
100	0.9355	0.8780	0.9471	0.8997	0.1144	0.0799	0.9472	0.8448
200	0.9366	0.8811	0.9437	0.8982	0.1094	0.0682	0.9428	0.8448
500	0.9405	0.8803	0.9508	0.9013	0.1183	0.0723	0.9450	0.8475
1000	0.9384	0.8718	0.9489	0.8945	0.1217	0.0768	0.9472	0.8502
5000	0.9387	0.8737	0.9498	0.8956	0.1220	0.0759	0.9454	0.8447
10000	0.9392	0.8762	0.9492	0.8961	0.1847	0.0746	0.9457	0.8485
all	0.9393	0.8768	0.9493	0.8974	0.2888	0.0741	0.9455	0.8468

Table 3. The SSIM index obtained for various distortions using the Monte Carlo approach with various number of points - the image 'Baboon'

Number of points	low-pass	low-pass	median	median	5%	20%	JPEG	JPEG
	3x3	5x5	3x3	5x5	noise	noise	60%	10%
50	0.6919	0.4456	0.7333	0.4613	0.5661	0.2323	0.8987	0.7059
100	0.6917	0.4790	0.7123	0.5160	0.5043	0.1826	0.9003	0.6790
200	0.6794	0.4540	0.7039	0.4633	0.5281	0.2342	0.8986	0.6838
500	0.6906	0.4593	0.7178	0.4846	0.5674	0.2363	0.8992	0.6880
1000	0.6877	0.4629	0.7168	0.4841	0.5409	0.2254	0.8954	0.6802
5000	0.6860	0.4591	0.7162	0.4796	0.5548	0.2270	0.8992	0.6790
10000	0.6846	0.4591	0.7125	0.4798	0.5532	0.2318	0.8977	0.6778
all	0.6861	0.4608	0.7147	0.4815	0.5518	0.2295	0.8986	0.6800

Table 4. The SSIM index obtained for various distortions using the Monte Carlo approach with various number of points - the image 'Lena'

Number of points	low-pass	low-pass	median	median	5%	20%	JPEG	JPEG
	3x3	5x5	3x3	5x5	noise	noise	60%	10%
50	0.9212	0.8533	0.9324	0.8807	0.2892	0.1098	0.9441	0.8438
100	0.8967	0.8123	0.9072	0.8398	0.3052	0.0948	0.9175	0.7926
200	0.9043	0.8130	0.9175	0.8456	0.3393	0.1115	0.9258	0.8121
500	0.9087	0.8305	0.9202	0.8565	0.3411	0.0920	0.9234	0.8136
1000	0.9091	0.8309	0.9210	0.8585	0.3469	0.0953	0.9250	0.8173
5000	0.9105	0.8340	0.9215	0.8612	0.3432	0.0935	0.9256	0.8159
10000	0.9097	0.8355	0.9203	0.8613	0.3391	0.0930	0.9243	0.8161
all	0.9103	0.8347	0.9212	0.8612	0.3397	0.0945	0.9251	0.8164

Analysing presented results for many typical distortions, obtained relative errors of the SSIM values estimated using strongly limited number of pixels are about 1-2%, so proposed fast Monte Carlo SSIM estimation can be treated as an interesting alternative for the applications in lower performance systems with a smaller amount of memory.

5 Conclusions

Considering the fact that the Monte Carlo method is much faster than full image analysis it seems to be a good alternative for the classical methods in a wide area of applications. Presented examples are not comprehensive but it is worth noticing that presented approach can be very useful especially in embedded systems with a low computational power and a limited amount of memory.

References

1. Chen, D., Odobez, J.-M.: Sequential Monte Carlo Video Text Segmentation. In: International Conference on Image Processing ICIP 2003, vol. 3, pp. 21–24. IEEE Press, New York (2003)
2. Eskicioglu, A., Fisher, P., Chen, S.: Image Quality Measures and Their Performance. *IEEE Trans. Comm.* 43(12), 2959–2965 (1995)
3. Kindratenko, V.: Development and Application of Image Analysis Techniques for Identification and Classification of Microscopic Particles. PhD thesis, Antwerp University (1997)
4. Luo, H., Eleftheriadis, A., Kouloheris, J.: Statistical Model-Based Video Segmentation and its Application to Very Low Bit-Rate Video Coding. *Signal Processing: Image Communication* 16(3), 333–352 (2000)
5. Quan, G., Chelappa, R.: Structure from Motion Using Sequential Monte Carlo Methods. *Int. Journal of Computer Vision* 59(1), 5–31 (2004)
6. Vermaak, J., Ikoma, N., Godsill, S.J.: Sequential Monte Carlo Framework for Extended Object Tracking. *IEE Proc. Radar Sonar Navig.* 152(5), 353–363 (2005)
7. Wang, Z., Bovik, A.: A Universal Image Quality Index. *IEEE Signal Process. Letters* 9(3), 81–84 (2002)
8. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image Quality Assessment: From Error Measurement to Structural Similarity. *IEEE Trans. Image Process.* 13(4), 600–612 (2004)
9. Zhai, Y., Shah, M.: Video Scene Segmentation Using Markov Chain Monte Carlo. *IEEE Trans. on Multimedia* 8(4), 686–697 (2006)