Image Segmentation by Robust Binarization and Fast Morphological Edge Detection

Aishy Amer and Eric Dubois

INRS-Télécommunications

Montréal, Québec, Canada

amer@inrs-telecom.uquebec.ca

Abstract

In real-time vision (e.g., visual surveillance), fast image segmentation is required. In this paper, a fast unsupervised artifact-robust segmentation method is presented. Within this method the whole segmentation process is divided into simple tasks so that complex operations are avoided.

Herewith, the basic task is the binarization which aims at finding out significant large object regions. In the paper, two methods for binarization are proposed: 1) an interference-invariant thresholding function based on a combination of local (histogram-based) and global (block-based) decision criteria. Such a binarization is suitable to segment regions of low intensity variations; 2) a noise-robust binarization based on homogeneity tests which implicitly take image noise into account. This method is designed to detect objects with high variable intensity (texture).

The paper also contributes a novel formulation of binary morphological erosion and dilation and a novel morphological binary edge detector.

Simulations show the robustness of the proposed methods even in strongly corrupted images. The algorithms are demonstrated on low and high variable intensity regions. The segmentation uses few parameters which are automatically adjusted to the amount of noise in the image, to the local average contrast, and to the local standard deviation.

1 Introduction

Because of its multiple visual applications (e.g., coding, retrieval, surveillance) image segmentation is gaining more and more importance. However, the demands (e.g, robustness, complexity) are application dependent. While in some applications exact segments (e.g., in upconversion) are needed, approximate segments are satisfactory for the needs of other applications (e.g, retrieval). Another issue is that within a digital coding and storage imaging system, the performance of the system's segmentation technique in the presence of coding artifacts is an important question[2]. Furthermore, the complexity of segmentation methods is a key issue since the wide use of a segmentation tool strongly depends on its computational efficiency[7].

In this paper, *image segmentation* denotes the technique of extraction of image entities or structures (regions or objects) so that the outlines of these structures will coincide as accurately as possible with the physical 2-D object outlines. A *2-D object* is a projection of a 3D real-world object (e.g., a tree) onto the image plane. An object *region* (e.g., the leaves of a tree) is a set of image pixels which are similar with respect to a homogeneity criterion such as texture.

In general, image segmentation methods can be distinguished into two groups, region and contour-oriented methods[13, 5]. The advantage of region-oriented methods, e.g., region growing techniques, is robustness in noisy images. However, they require high implementation costs which make them less suitable for real-time vision applications. In general, contour-oriented methods have low calculation costs. Their main disadvantage is however the sensitivity towards degradation of image quality.

In this paper, a fast unsupervised hybrid (contour and region-based) image segmentation method of reduced complexity aimed at separating *significant large* objects is proposed. The proposed image segmentation consists of four tasks (Fig. 1, Fig. 4(b)–4(d)): separation of object regions (binarization) (Sec. 2,3), morphological edge detection (Sec. 3.5), contour analysis ([11, 3]), and object filling and reconstruction ([11, 3], Fig. 4(d)). In this paper, contributions are introduced to binarization and edge detection.

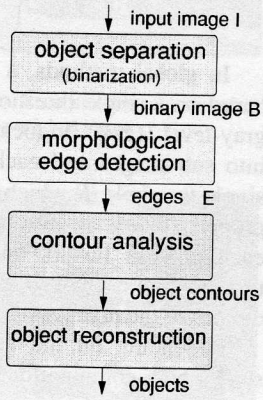


Figure 1: Multilevel Segm.

The critical task of the segmentation is clearly the binarization which needs to provide a binary image B in which smooth white regions representing image objects are separated by black pixels representing the object boundaries. Here, two binarization methods are proposed. The first (Sec. 2) aims at detecting regions of low variational intensity. The second method (Sec. 3) is designed to detect object regions with high variational intensity (i.e., texture).

Furthermore, novel definitions of morphological erosion and dilation and a novel morphological edge detector with significantly reduced computations and equal performance

compared to standard morphological detectors are proposed.

The proposed segmentation method has shown robustness with respect to white and impulsive noise and coding (e.g., MPEG-2) artifacts, has low computational cost, and has regular structure of main components (i.e., binarization and morphological operators). Furthermore, the whole segmentation uses very few parameters which are then automatically adjusted to the amount of noise and to the local image content. As a result of the segmentation process, a *list of object regions* with their features such as area (size in pixels), minimum bounding box (MBB), shape, intensity, and position is provided for further object-based processing.

2 Binarization by thresholding

2.1 Introduction

Thresholding methods can be divided into global, local, and dynamic methods[6]. Hereby, depending on a threshold T, each pixel I(x,y) in the input (gray-level) image I (with X columns and Y rows) is classified as belonging to an object and labeled white in a binary image B or to the background and labeled black (Eq. 1).

$$B(x,y) = \begin{cases} 1 & : \quad I(x,y) > T \\ 0 & : \quad I(x,y) \leq T \end{cases} \tag{1}$$

In global methods, a gray-level image is thresholded based on a single threshold T which depends only on the gray-level $I(x,y)$. In local methods, the image is portioned into sub-images and each sub-image is thresholded by a single threshold T which depends on $I(x,y)$ and the local property of its spatial neighborhood. In dynamic method, each pixel is thresholded based on a possible different threshold for each pixel, thus T depends on the spatial coordinates of the pixel.

Depending on the application, dynamic thresholding shows high computational costs[6, 4]. Besides, because it works very locally it is noise and illumination-change sensitive. For effective fast threshold determination, the combination of global and local criteria is needed. In this section, a thresholding function is presented which is based on both global (block-based) and local (histogram-based) decision criteria. Doing this, the threshold is adapted to the image contents and changes (e.g. noisy images Fig. 4(e) and MPEG-noisy image Fig. 4(f)).

2.2 Artifact-robust adaptive thresholding

Fig. 2 gives an overview of the binarization by thresholding. As can be seen, the image is first divided into N equal blocks. In order to take small regions, noise, and illumination changes into account, a histogram for each of the N blocks is computed. This histogram is then divided into M equal partitions. For each histogram partition, the most frequent gray level $G_{P_i}, i \in \{1 \cdots M\}$ is then determined. To take more global image content into the thresholding function, an average gray level $\mu_{B_k}, i \in \{1 \cdots M\}$ of each block is calculated. Finally, the threshold T (Eq.2) is calculated by averaging G_{P_i} and μ_{B_k} for all the N blocks. This thresholding function(Eq. 2) outstands for its interference invariance.

$$T = \frac{\sum_{k=1}^{N} (\sum_{i=1}^{M} (G_{P_i}) + \mu_{B_k})}{N \cdot M + N} \tag{2}$$

with

$$\mu_{B_k} = \frac{\sum_{x=1}^{W} [\sum_{y=1}^{H} (I(x,y))]}{W \cdot H} \tag{3}$$

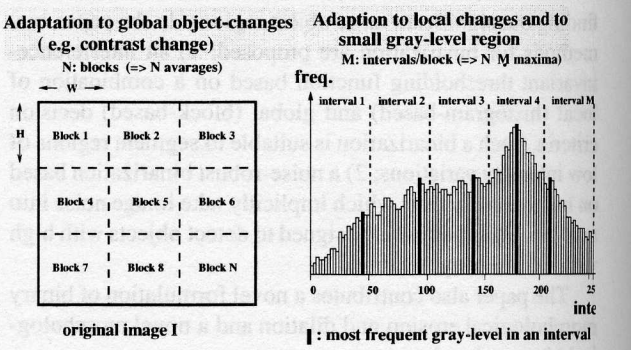
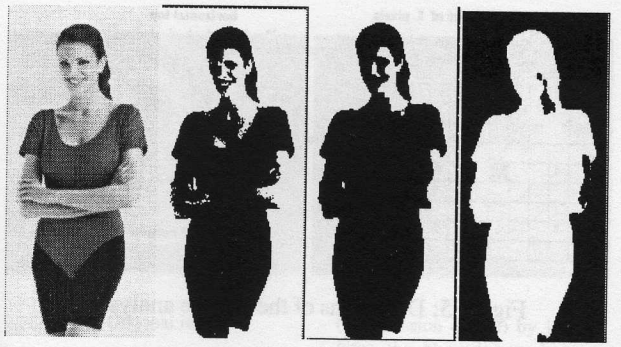


Figure 2: Extraction of image-global threshold

Background adaptation: In well-known thresholding methods[10, 1, 4, 6], significant objects are assumed to have gray-levels *greater* than the determined threshold. However, using only one threshold, no identification if the background is brighter (darker) than the objects can be achieved (Fig. 3(b), 3(c)). Therefore, a simple detection procedure is proposed which aims at detecting if significant objects are darker or brighter than the background. This detection procedure determines first the number of bright pixels (greater than the threshold T) card[Br] and compared it then to the number of dark pixels card[Dr] based on the determine global threshold T. Now, if the number of dark (bright) pixels is greater than the number of (dark) bright pixels in the input image, then a dark (bright) background is assumed and all pixels greater (smaller) than the determined threshold T are set to white in B (Eq. 4). This assumption is based on the observation that in many images especially in industrial applications such as factory automation and document production, either dark objects on bright background or bright objects on dark background are usually used. Furthermore, this procedure assumes that the area of objects is smaller



(a) Original image (observe the noise&shading) (b) Results using [10] (c) Results using [1] (d) Results using the proposed method

Figure 3: Adaptive background thresholding

than the area of the background.

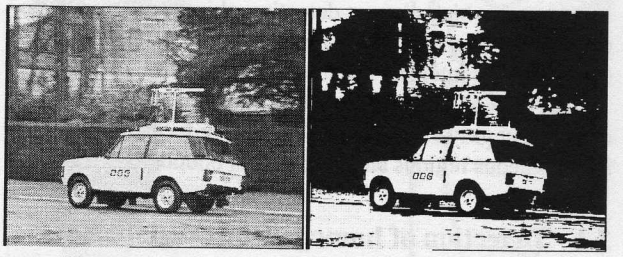
$$B(x,y) = \begin{cases} \text{if } \text{card[Br]} \gg \text{card[Dr]} \\ \quad \begin{cases} 1 & : \ I(x,y) > T \\ 0 & : \ I(x,y) \leq T \end{cases} \\ \text{if } \text{card[Dr]} \gg \text{card[Br]} \\ \quad \begin{cases} 0 & : \ I(x,y) > T \\ 1 & : \ I(x,y) \leq T \end{cases} \end{cases} \tag{4}$$

where card[·] gives the cardinal number of a set, Br (Dr) is the set of bright (dark) pixels, and $x \gg y$ means x is significantly greater than y.

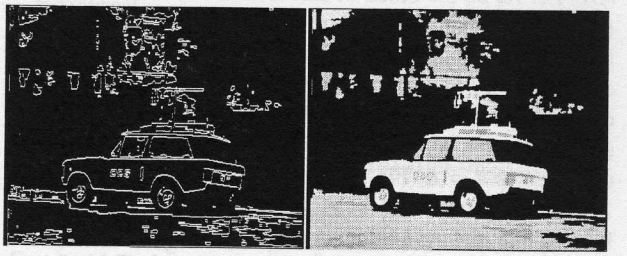
2.3 Experimental results

The presented thresholding method has been tested in various images and results have been compared to frequently used methods[10, 1]. The proposed thresholding function outstands for its interference invariance. On average, the algorithm needs 0.05 seconds[1] for thresholding. Otsu[10] method needs in average 0.05 and Abutaleb[1] algorithm needs 0.67. Fig. 3 gives comparative results of these methods. As can be seen, the proposed method separates the bright background and dark objects. On the other side, in Fig. 4(b), the algorithm detects that the objects have brighter gray levels than the background. Furthermore, the thresholding-based object segmentation was verified in many simulations showing that it stays robust, even on very low-bit-rate MPEG-2 decoded images (Fig. 4(f), 2Mbit/s bitrate) and on heavy noisy images (Fig. 4(e), white and impulsive noise added).

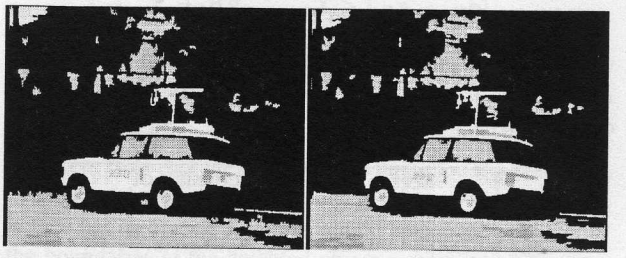
[1]Time given in the paper is measured in seconds by the 'clock' command of 'C' on a SUN4 machine with a SPARC 167 Mhz processor for one single standard ITU-R-601 field (720 × 288)



(a) Original image (b) Binary image



(c) Gap-free edges (d) Reconstructed objects of the original image



(e) Reconstructed objects of the noisy original image (f) Reconstructed objects of the 2Mbit-MPEG-decoded original image

Figure 4: Artifact-robust segmentation based on thresholding

3 Variation-adaptive binarization

The principal disadvantage of thresholding methods is that they are designed for specific type of images and that they fail to detect object region in case of complex intensity variations resulting in incomplete segmentation. In this section, a binarization method based on intensity homogeneity tests is proposed which aims at detecting object region with homogeneous intensity. Here, also regions with high variable (non uniform) intensity (henceforth texture) are detected.

The goal of the binarization is to differently label pixels of the input gray-level image I which are located in texture-homogeneous regions and pixels at its boundaries.

3.1 Detection of homogeneous regions

Let I represent the input (gray-level) image with X columns and Y rows and let B denote the binary image of the same size. Let R_c be the block of size $L \times L$ centered at position (x, y). Let the eight neighboring blocks of R_c corresponding to the eight directions ($2\times$ horizontal, $2\times$ vertical, $4\times$ diagonal, Fig. 5) of the same size be $R_0, \cdots, R_7$. Let the distance (in pixels) between the central pixel of block R_c and the centers of the 8 blocks be d. Note that in this processing step $d = L$ (Fig. 5).

A pixel $\mathbf{p} = (x, y)$ of I is labeled white in B if R_c is located in a *texture-homogeneous sector* $S_* \in S$, with $S = \{S_{to}, S_{bo}, S_{ri}, S_{le}, S_{ab}, S_{be}, S_{re}\}$ as defined in Table 1 and shown in Fig. 5. S_* is texture-homogeneous if

$$\begin{aligned} |\mu_{R_c} - \mu_{R_i}| < T_\mu \quad \wedge \\ |\sigma_{R_c} - \sigma_{R_i}| < T_\sigma, \qquad \forall R_i \in S_* \end{aligned} \tag{5}$$

where μ_i and σ_i denote the mean and standard deviation of R_i, $T_\mu = \psi(\sigma_n)$, and $T_\sigma = \phi(\sigma_{S_{re}})$ (Eq. 6). Both $\psi(x)$ and $\phi(x)$ are strictly monotonic functions.

	Sector	consists of
S_{to}	"horiz. top"	R_c,R_0,R_1,R_2,R_3,R_4
S_{bo}	"horiz. bottom"	R_c,R_0,R_4,R_5,R_6,R_7
S_{ri}	"vert. right"	R_c,R_0,R_1,R_2,R_6,R_7
S_{le}	"vert. left"	R_c,R_2,R_3,R_4,R_5,R_6
S_{ab}	"diag. above"	R_c,R_1,R_2,R_3,R_4,R_5
S_{be}	"diag. below"	R_c,R_0,R_1,R_5,R_6,R_7
S_{re}	"rect."	R_c,$R_0 - R_7$

Table 1: Sectors of the texture analysis

This detection is done hierarchically, i.e., the labeling starts with a block size $L \times L$ (Fig. 6(c)) ($L = 2n - 1, n \in \mathcal{N}, n > 2$) and then the block size is set to $(L-2) \times (L-2)$. This is repeated until the block size is 3 (Fig. 6(d),7(b)). In each step, non-labeled pixels are examined and eventually set to white. As can been seen in Fig. 6(d), in this step, the interior of each region is detected and set to white. Observe that the smallest block is 3, i.e., the smallest surrounding area S_{re} is 9 and that a large $d = L$ is taken in this step. This conservative setting, which is necessary to prevent incorrect region merging, does not allow pixels which are located at a region's boundaries to be labeled.

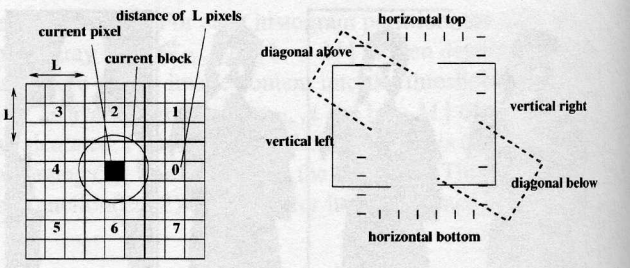


Figure 5: Directions of the texture analysis

3.2 Completing object boundaries

Because of the before mentioned large masks, pixels which are located at a region's boundaries are not labeled. In this step, such pixels are completed using the same strategy as described above but with variable $d = 3, 2, 1$. However, before setting pixels, a test is made if this setting would merge two regions. Only pixels which lay at object boundaries or lay inside a binary region are set. Thus a region's separation is preserved but boundaries and small holes are completed.

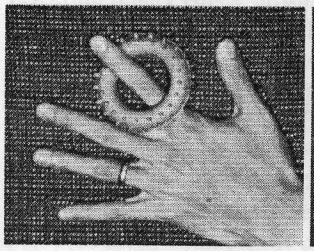
3.3 Adaptation to image changes

Because of noise and other artifacts, homogeneities could be lost, and therefore the thresholds T_μ and T_σ should be adapted to the amount of noise present in the image, to the local average contrast, and to the local standard deviation.

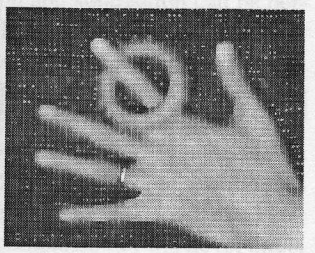
First white Gaussian noise is estimated. Then, T_μ and T_σ are adapted as follows: let a "clean" region R_c be a set of pixels x_i with the mean μ_{R_c} and variance $\sigma^2_{R_c}$. If noise (with estimated variance σ^2_n) is added to that "clean" region and if we assume that there is no correlation between the noise and the image signal, then the new mean and variance of the region are: $\mu_{R_n} = \mu_{R_c} + \mu_n$ and $\sigma^2_{R_n} = \sigma^2_{R_c} + \sigma^2_n$. In this paper, the adaptation functions are:

$$T_\mu = \alpha + \beta\sigma^2_n, \qquad T_\sigma = \delta + \gamma\sigma^2_{S_{re}} \tag{6}$$

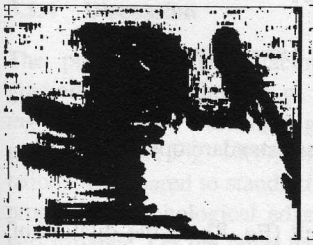
with $\alpha, \beta, \gamma, \delta \in \mathcal{R}$, $\beta = \psi(\sigma_{n_{max}})$, $\gamma = \phi(\sigma_{S_{re_{max}}})$, $\sigma_{n_{max}}$ ($\sigma_{S_{re_{max}}}$) is the maximum assumed noise (local standard deviation), and α and δ are the minimum thresholds. Current research is aimed at enhancing this artifact-adaptation function, e.g., through statistical tests.



(a) Original image

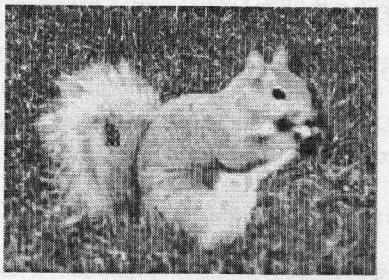


(b) Diffusion of 6(a) by [12]. Note the blurred boundaries and the non-uniform background diffusion (see text)

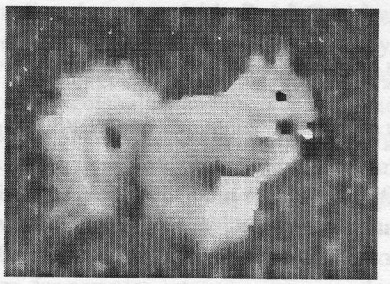


(c) Binarization, block size 11×11

(d) Binarization, block sizes $11 \times 11 \rightarrow 3 \times 3$



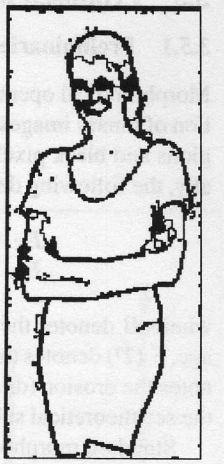
(e) Original image



(f) Diffusion of 6(e) by[12]. Note the inter-region (grass and object) blurring making region separation difficult

Figure 6: Hierarchical Binarization: results & comparison

(a) Original image (observe the noise and shading)



(b) Binarization, block sizes $11 \times 11 \rightarrow 3 \times 3$

Figure 7: Binarization results

3.4 Experimental results and discussion

The proposed binarization method was verified by simulation with images undergoing noise and artifacts. It was shown to be noise-robust. Fig. 6 shows a segmentation of a natural image. Although this image is overlaid with noise (about 30dB PSNR), the hand and both rings are separated from the background. Also in textured images, the segmentation yields comparable basic results to high-performance image segmentation algorithms[13]. On the other hand, the proposed method needs significantly less computations.

Note that all the results are obtained with the same parameter setting. No image-wise manual adjustment was undertaken. Note that the main goal of the proposed method is to detect and separate robustly object regions. The emphasis is here not on detected precise region boundaries. Although the boundaries of the detected object regions are more than one-pixel wide (as it is the case in[13]), such results are useful in applications where approximate segments are satisfactory such as in object-based retrieval applications.

Furthermore, this binarization method had been compared with a binarization based on anisotropic diffusion followed by thresholding[12], Fig. 6(b). A basic problem of using anisotropic diffusion is that while it sharpens edges it blurs small discontinuities (like the ones between the large ring and the hand in Fig. 6(a), Fig. 6(b)). It fails also to detect textured regions (note the background in Fig. 6(b)). These properties question the use of diffusion-based segmentation especially in the presence of non uniform gray-level or textured image regions. Furthermore, anisotropic diffusion is a computational expensive method.

3.5 Proposed morphological operations

3.5.1 Preliminaries

Morphological operations are very effective for edge detection of binary images, where white pixels denote uniform regions and black pixels denote region boundaries[8, 9]. Usually, the following detectors are used:

$$E_{\mathcal{E}} = B - \mathcal{E}[B, K_{(n\times n)}], \quad E_{\mathcal{D}} = B - \mathcal{D}[B, K_{(n\times n)}) \tag{7}$$

where B denotes the binary image, E denotes the edge image, $\mathcal{E}$ ($\mathcal{D}$) denotes the erosion (dilation) operator, $K_{n\times n}$ denotes the erosion (dilation) $n \times n$ kernel used, and $-$ denotes the set-theoretical subtraction.

Standard morphological erosion and dilation are defined around an origin, where its position is crucial for the detection. For each step of an erosion (dilation), only one pixel is set (at a time) in B. To achieve precise edges with single-pixel width, 3×3 kernels (defined around the center) are used: when a 3×3 *cross* kernel is used, an incomplete corner detection is obtained (Fig. 8(a)); a 3×3 *square* kernel gives complete edge but requires more computation (which grows rapidly with increased input data, Fig. 9(a)); and the use of a 2×2 kernel will produce incomplete edges (Fig. 8(b)).

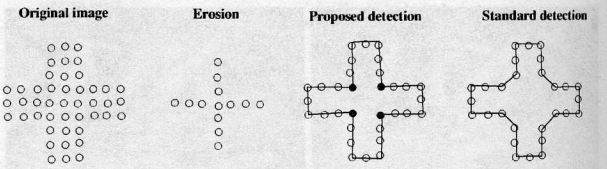
To avoid these drawbacks, new operational rules for edge detection by erosion (dilation) are proposed which use a fixed sized (square 2×2) kernel and set all four pixels of the kernel at a time in B. For edge detection based on the novel rules, accurate complete edges are achieved and the computational cost is significantly reduced. Since the wide use of image segmentation strongly depends on its computational efficiency, this is a key point[7].

3.5.2 Efficient erosion and dilation

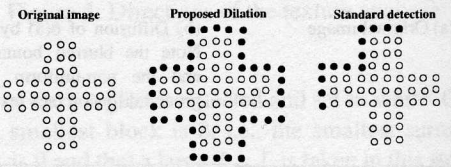
Proposed erosion: *if all four binary-image pixels inside the 2×2 kernel are white, then all four pixels in the output image are set (at a time) to white, if they were not eroded in a previous step. If at least one of the four binary-image pixels inside the kernel is black, then all the four pixels in the output image are set (at a time) to black.*

Proposed dilation: *If at least one of the four binary-image pixels inside the kernel is white, then all the four pixels in the output image are set (at a time) to white.*

Set-theoretical formulation: An advantage of the proposed operations is that they can be formally defined based on set-theoretical intersection, union, and translation in analogy to the formal definitions of the standard operation[8]. The standard erosion and dilation satisfy the following two properties[8]: 1) the erosion of an image by the union of kernels is equivalent to erosion by each kernel independently and then intersecting the result (Eq. 8); 2) the dilation of an image by the union of kernels correspond to dilation by each kernel and then performing union of the resulting images (Eq. 9). Given image A and kernels B and C in $\mathcal{R}^2$,



(a) Erosion by a 3×3 cross kernel



(b) Dilation using a 2×2 square kernel, origin at left upper pixel

Figure 8: Proposed versus standard operations

$$\mathcal{E}_s[A, B \cup C] = \mathcal{E}_s[A, B] \cap \mathcal{E}_s[A, C] \tag{8}$$

$$\mathcal{D}_s[A, B \cup C] = \mathcal{D}_s[A, B] \cup \mathcal{D}_s[A, C] \tag{9}$$

where $\mathcal{E}_s$ ($\mathcal{D}_s$) denotes the standard erosion (dilation). The proposed erosion and dilation are then defined by:

$$\mathcal{E}_p[A, K_{2\times2}] = \mathcal{E}_s[A, S_{3\times3}] =$$

$$\mathcal{E}_s[A, K^{ul}_{2\times2} \cup K^{ur}_{2\times2} \cup K^{ll}_{2\times2} \cup K^{lr}_{2\times2}] =$$

$$\mathcal{E}_s[A, K^{ul}_{2\times2}] \cap \mathcal{E}_s[A, K^{ur}_{2\times2}] \cap \mathcal{E}_s[A, K^{ll}_{2\times2}] \cap \mathcal{E}_s[A, K^{lr}_{2\times2}] \tag{10}$$

$$\mathcal{D}_p[A, K_{2\times2}] = \mathcal{D}_s[A, S_{3\times3}] =$$

$$\mathcal{D}_s[A, K^{ul}_{2\times2} \cup K^{ur}_{2\times2} \cup K^{ll}_{2\times2} \cup K^{lr}_{2\times2}] =$$

$$\mathcal{D}_s[A, K^{ul}_{2\times2}] \cup \mathcal{D}_s[A, K^{ur}_{2\times2}] \cup \mathcal{D}_s[A, K^{ll}_{2\times2}] \cup \mathcal{D}_s[A, K^{lr}_{2\times2}] \tag{11}$$

where $\mathcal{E}_p$ ($\mathcal{D}_p$) denotes the proposed erosion (dilation), $S_{3\times3}$ is a 3×3 square kernel, and $K^{ul}_{2\times2}$ is a 2×2 kernel with origin at the upper left (equivalently upper right, lower left, lower right) corner.

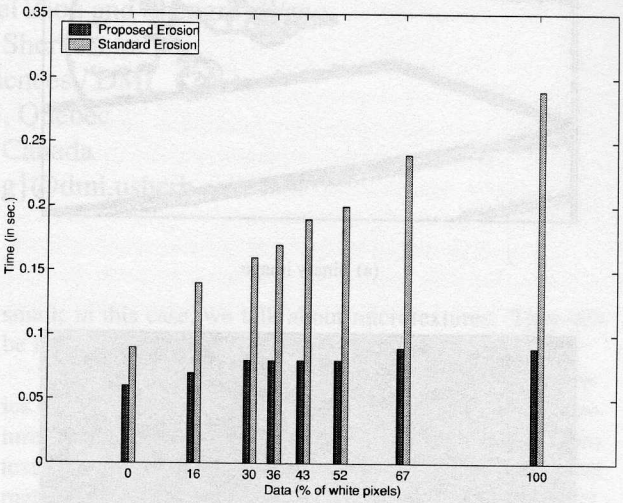
3.5.3 Novel morphological binary edge detection

In this section, the need of using two operations (Eq. 7) for a binary morphological edge detection is questioned. When

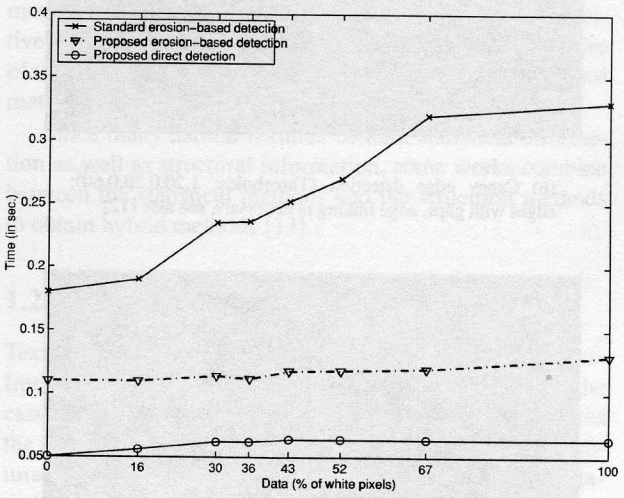
detecting binary edges, erosion and subtraction can be performed implicitly which would further reduce the complexity of morphological edge detection. Such an implicit detection is proposed as follows: *if at least one of the four binary-image pixels inside the* 2×2 *kernel is black, then set all four pixels of the kernel (at a time) to white if they are white in the binary image*. Thus if the 2×2 kernel fits in a white area it is implicitly eroded but edges (kernel does not fit) are kept. Fig. 9(b) gives a complexity comparison of the novel direct edge detection, edge detection using the proposed erosion and edge detection using standard erosion (a 3×3 square kernel). As can be seen, the cost of edge detection is significantly reduced.

3.5.4 Discussion

The proposed edge detectors have been compared to gradient-based methods (e.g., Canny, Fig. 10(b), *edge linking* is necessary) showing higher detection accuracy (e.g., gap free edges, Fig. 10(c)) and significantly lower computations. Compared to standard morphological detectors, the proposed morphological edge detectors achieve the same performance. On the other hand, Fig. 9(a) gives a computation comparison of the proposed and standard morphological erosion applied to various natural image data. As can be seen, the computational cost using the standard erosion with 3×3 square kernel grows rapidly with the amount of input data, while the cost of the proposed erosion stays almost constant. The computations can be further reduced by applying the novel morphological edge detection with implicit erosion (Fig. 9(b)).



(a) Proposed vs. standard erosion



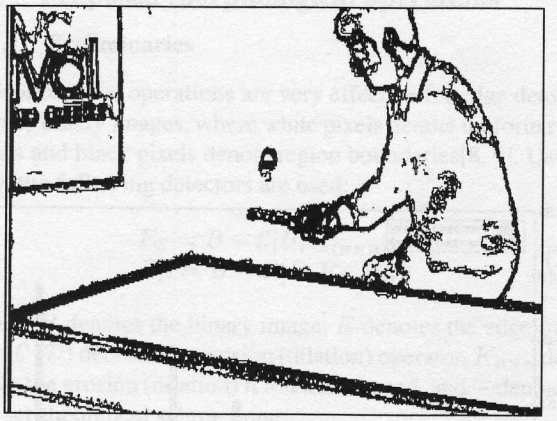
(b) Proposed vs. standard detection

Figure 9: Computational efficiency comparison

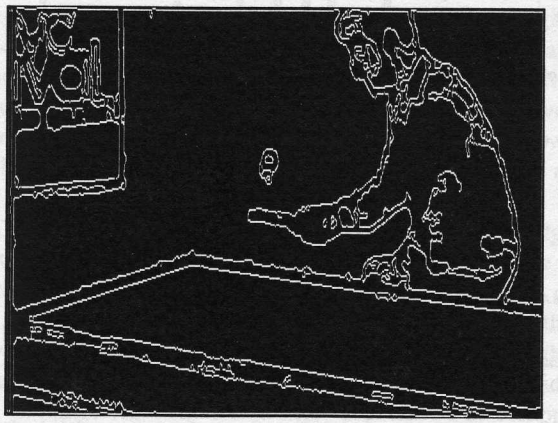
4 Conclusion

In this paper, a four-step (binarization, edge detection, contour analysis, and filling) image segmentation of reduced computations aiming at detecting significant large objects is presented. The proposed edge detectors have been compared to gradient-based methods showing higher detection accuracy (e.g., gap-free edges) and significantly less computations. Two binarization methods are proposed: the first is a thresholding which shows very noise-robustness by various simulations; the second binarization 1) separates texture homogeneous regions based on the mean and standard deviation, 2)is noise robust, and 3) has regular structure and low computations. The robustness is due to the adaptation to the amount of noise and to local and global image content.

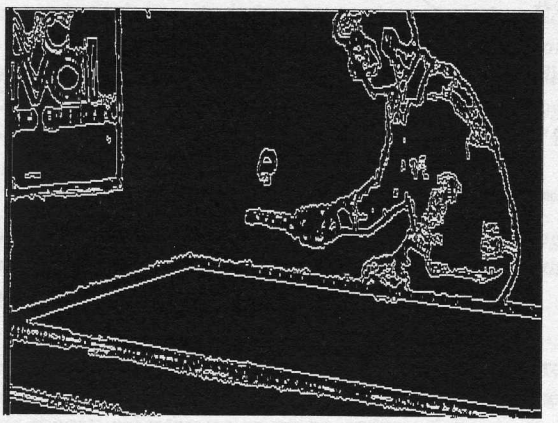
However, for complex intensity variation, the mean and standard variation alone are not sufficiently discriminatory. In future work, other discrimination factors will be introduced, e.g., the correlation function of the block of pixels. Further research is oriented at introducing local and global contextual criteria such as edge continuities. Further work



(a) Binary image



(b) Canny edge detection (Thresholds: 1.20,0.20,0.60): edges with gaps, edge linking is necessary, see also [12]



(c) Morph. edge detection: good region separation, one-pixel wide, gap-free edges

Figure 10: Edge detection comparison

is also aimed at enhancing the noise adaptation function through statistical tests.

References

[1] A. Abutaleb, "Automatic thresholding of gray-level pictures using two-dimensional entropy," *Computer Vision, Graphics, and Image Process.*, vol. 47, pp. 22–32, 1989.

[2] A. Amer and E. Dubois, "Segmentation-based motion estimation for video processing using object-based detection of motion types," in *Proc. SPIE Visual Communications and Image Process.*, (San Jose, CA), Jan. 1999.

[3] A. Amer and S. Reichert, "A multilevel method for robust object segmentation in video sequences," in *Proc. 9^{th} Aachen Symposium on 'Signaltheorie'*, (Aachen, Germany), pp. 43–46, Mar. 1997. ISBN 3-923219-10-5, in German.

[4] J. Bernsen, "Dynamic thresholding of grey-level images," in *Proc. 8^{th} Int. Conf. on Pattern Recognition (ICPR8)*, (Paris, France), pp. 1251–1255, Oct. 1986.

[5] A. Chehikian, "Image segmentation by contours and regions cooperation," *Signal Process.*, vol. 78, pp. 329–347, Nov. 1999.

[6] M. Dai, P. Baylou, L. Humbert, and M. Najim, "Image segmentation by a dynamic thresholding using edge detection based on cascaded uniform filters," *Signal Process.*, vol. 52, pp. 49–63, Apr. 1996.

[7] L. Garrido, P. Salembier, and D. Garcia, "Extensive operators in partition lattices for image sequence analysis," *Signal Process.*, vol. 66, pp. 157–180, 1998.

[8] C. Giardina and E. Dougherty, *Morphological Methods in Image and Signal Processing.* New Jersey: Prentice Hall, 1988.

[9] R. Haralick and L. Shapiro, *Computer and Robot Vision.* Reading: Addison-Wesley, 1992.

[10] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Mach. and Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[11] T. Pavlidis, *Algorithms for Graphics and Image Processing.* Maryland: Computer Science Press, 1982.

[12] P. Perona and J. Malik, "Scale-space and edge detection using ansotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, July 1990.

[13] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900, Sept. 1996.