Memoirs of the Scientific Sections of the Romanian Academy
Tome XXXV, 2012

COMPUTER SCIENCE

INFORMATION FUSION TECHNIQUES FOR SEGMENTATION OF MULTISPECTRAL IMAGES

SILVIU IOAN BEJINARIU, FLORIN ROTARU and CRISTINA DIANA NIȚĂ

Institute of Computer Science, Romanian Academy, Iasi Branch, Romania
Corresponding author: silviu.bejinariu@iit.academiaromana-is.ro

Image Fusion is the combining process of relevant information from one, two or more images to create a single image which is more complete than any of the input ones. In this paper it is proposed a new segmentation approach for edge detection in multispectral images using image fusion techniques. It was applied for circular objects detection in multispectral images. First, the multispectral images are pre-processed to reduce noise using a speckle filtering method based on homogeneity histogram analysis and then an edge detection method is applied. The results are combined to obtain a more accurate description of the objects in the input images.

Key words: Multispectral, fusion, segmentation, edge.

1. INTRODUCTION

The combining process of features extracted from two or more input images or from a single image but using different methods of extraction is known as Image Fusion [2]. In this paper it is proposed a new segmentation approach for edge detection in multispectral images using image fusion techniques. The procedure was applied for multispectral images to detect the circular objects in a scene. A multispectral image is represented by a set of images obtained for certain frequencies of the electromagnetic spectrum. The proposed algorithm allows multispectral segmentation by identifying the edges in a number of images of the same scene taken in different spectral bands, followed by combining the results. The images are assumed to be aligned.

Data Fusion is defined as the process of combining primary data from different sources and different means of information extraction using redundancy to obtain a more complete, efficient and correct information. Data Fusion methods are used in military and civil domains, for defense, security systems, robotics, medicine. The most usual objectives of data fusion applications are: target detection, recognition, identification, objects tracking, changes detection, decision process. *Information Fusion* is known as the process of combining data already preprocessed [2, 11].

Depending on the processing level where the fusion is applied, data fusion methods are classified as:

– Low level fusion (data integration) – combines primary data from different sources or sensors to obtain a more complete and synthetic single data set.
– Intermediate level (information fusion) – combines features extracted from input data sets to obtain a limited set of more relevant features.
– High level (decision fusion) – combines decisions from different human or artificial experts to obtain the optimal decision.

The second section contains a short description of data and image fusion concepts. In the third section there are described the image preprocessing method (speckle noise filtering) and the feature extraction method (edge detection).

The proposed procedure is presented in fourth section. The method was tested on images available in the „Multispectral Image Database" [34]. The final section gives some conclusions of this paper.

2. DATA FUSION AND IMAGE FUSION

Usually, the image fusion is performed in 2 steps: image registration and proper image fusion which can be done in the spatial domain or in a transform domain. The most used spatial domain fusion method is the Principal Component Analysis. Image fusion in a transform domain can be done with or without multi-scale decomposition of the transform domain [2].

In many cases, processing requires images with both spatial and spectral high-resolution, conditions which are not simultaneously satisfied by the acquisition equipments. In this case, fusion techniques are applied to allow integration of data from various sources (multi-sensor image fusion). Multi-sensor image fusion is applied for:

– recognition and classification of differently exposed images (lighting or focus),
– aerial and satellite image processing - high-resolution multispectral images are obtained from high resolution panchromatic images and low resolution multispectral images,
– medical image analysis - images from various sources are combined: magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET),

In general, image fusion techniques involve the following steps:

a. Image registration:

– Features selection – these features are used to establish correspondences between the original images. The features should be stable enough to be determined regardless of the resolution used or the orientation of the scene.
– Features correspondence – it is necessary to determine the coordinates of common features in all images by applying a "template matching" procedure.

– Determination of the geometric transformations among images. This is one of the most difficult problem because parameters like camera position and geometry are usually not known. The most known transformation functions are Thin-plate splines, Multiquadrics, Piecewise Linear and Weighted Linear.

– Resampling – is accomplished using approximation methods like nearest-neighbor, bilinear and cubic convolution.

b. Image fusion (the combining process) can be approached from two different viewpoints:

– Spatial domain fusion – the most known combining methods are mediation, Principal Component Analysis and IHS (Intensity-Hue-Saturation) based methods.

– Transform domain fusion – the most used methods are discrete wavelet transform, Laplacian pyramid, curvelet transform.

In recent years, many methods for image fusion have been developed, which can be divided in two major groups:

– Methods based on multi-scalar decomposition;

– Methods not based on multi-scalar decomposition.

2.1. FUSION LINEAR MODELS FOR MULTI-SENSOR IMAGES

The multi-sensor image fusion is the combining process of data (images) from multiple sources. For satellite images, high-resolution sensors with few frequency bands and low spatial resolution sensors with high number of spectral bands are used. Most applications require a combination of these features to obtain a high spatial resolution images with rich spectral information.

Linear mixing model

For medium or low spatial resolution it is possible that a pixel value may be determined by several areas. This means that it is composed of signals coming from different individual components (mixed pixel).

In this case, the spectrum of a pixel is a linear combination of components spectra, weighted by the coverage.

$$p_i = \sum_{c=1}^{n_c} \left(r_{ci} * f_c \right) + e_i, \qquad i = 1, n_b \tag{2.1}$$

where:

p_i = pixel reflectance in *i* band,

r_{ci} = reflectance of *c* component in *i* band,

f_c = coverage of *c* component,

e_i = residual error in *i* band,

n_c = number of components,

n_b = number of spectral bands,

These equations can be re-written as:

$$P_{nb*1} = R_{nb*nc} \times F_{nc*1} + E_{nb*1} \tag{2.2}$$

Linear mixing method is simple and widely used to analyze mixed pixels.

Spatial de-mixing fusion

If there is information about components present in a certain scene, a de-mixing algorithm to determine the proportions of each component in each pixel can be applied. The proper identification of image components and their pure generated signal is required in this case. Because most pixels are mixed, these conditions are difficult to meet for images with heterogeneous components.

The method known as the "spatial de-mixing method" or the "method based on de-mixing fusion" consists of: (1) high-resolution image analysis to determine the classes of components, (2) compute the components coverage for each low resolution pixel, (3) de-mixing for each pixel, band and neighborhood of the pixel to be de- mixed, using its coverage and the spectral information and (4) image reconstruction by de-mixed pixels joining. De-mixing operation is performed on a neighborhood to have enough equations to solve the system of de-mixing problem. De-mixing can be spectral when the components present in the image are known and the de-mixing procedure is applied for each pixel and on all bands.

2.2. IMAGE FUSION USING MULTI-SCALE DECOMPOSITION

The multi-scale transformations are extremely useful for analyzing information content of the images to pursuit fusion. In this case, the main idea is to apply a multi-scale transformation (MST) to the images to be fused, combining images from decomposition followed by inverse transformation to obtain the fused image.

The most used decomposition methods [4, 28, 31] are: Pyramid Transform, Discrete Wavelet Transform and Discrete Wavelet Frame.

Some of the most recent methods on using the wavelet algorithm for image fusion are: the discrete wavelet transform, dual-tree complex wavelet transform and dyadic discrete wavelet transform [2, 21]. These transformations can be used to determine a multi-dimensional representation of the image edges and can be applied to fusion incomplete focused images, remote sensing and medical images (CT and MR).

Fusion using wavelet transforms

In literature, several 2D image fusion techniques have been described in [6, 13, 16, 19, 22, 26, 27, 30, 33]. In all these works, wavelet transforms W are first calculated for two input images $I_1(x, y)$ and $I_2(x, y)$, then the results are combined using ϕ fusion rules. Finally, the inverse wavelet transform W^{-1} is computed and the image fusion $I(x, y)$ reconstructed.

$$I(x, y) = W^{-1}(\phi(W(I_1(x, y)), W(I_2(x, y)))) \tag{2.3}$$

Fusion schemes based on the wavelet transforms have the following advantages in relation with pyramid transform schemes:

- Wavelet transforms provide information about edge directions (Li *et al.* [16]);
- For the pyramid representation, the features from the areas with large differences between original images are not stored in the fused image (Li *et al.* [16]);
- Fusion images obtained using wavelet transforms have a signal to noise ratio (SNR) better than those obtained using the pyramidal representation, when the same fusion rules are used (Wilson *et al.*, 1995 [30]).

Since wavelet coefficients with high absolute values contain information about relevant features of the image, such as edges and lines, it is recommended as a fusion rule to select the maximum coefficient (in absolute value). A more advanced rule is the rule for selecting areas, proposed by Li [16]. In this case a binary decision tree is built and the maximum absolute value in a window is used as measure of relevance for the central pixel.

In the method proposed by Burt and Kolczynski [3], the resulting coefficients are obtained by a weighted average based on local activity levels in each sub-band image.

2.3. FUSION PROCESS QUALITY EVALUATION

Quality assessing of the fusion process is application dependent. Usually, the results are interpreted by a human observer. Human perception is therefore a criterion for evaluation. There are two situations when an objective performance evaluation may be applied [23, 32]:

- If a reference image is available, the following measures are used for evaluation: root mean square, correlation, peak signal to noise ratio, mutual information and quality index.
- If a reference image is not available, the standard deviation, entropy or the overall cross-entropy of source images and fused image are used for the fusion results evaluation.

3. FEATURES EXTRACTION

The earliest edge detection methods were based on an image enhancement operator (like Sobel [18], Roberts [25]) followed by an image thresholding operation. Marr and Hildreth [17] introduced the Gaussian smoothing as a pre-processing step and used the zero-crossings of the Laplacian of Gaussian to locate edges. Another Gaussian edge detector based on optimizing criteria was proposed by Canny [5]. Other methods were developed by: Deriche, a recursive implementation in [7] or Rothwell which designed a spatially adaptive operator in [29].

Newer approaches [1, 14] use the multiresolution representations obtained by convolving the original image with Gaussian filters of different sizes which are integrated afterward to produce the edge map. Other approaches (Nalwa and Binford [20]) use parametric fitting methods, in which the image is fitted with a parametric model whose parameters are determined by minimizing the fitting error.

The proposed method uses the detected edges as input for the fusion process. Edges are represented by local changes of intensity in the image that occur on the boundary between the regions of the image. The edges can be used to extract other important features: lines, corners. Edges detection is performed usually in the following steps [12]: smoothing, used to reduce the noise without affecting the edges, sharpening, used to enhance the quality of the edges, edge detection, determines the pixel candidates to be part of a final edge, edge thinning and linking to obtain the exact position of an edge.

3.1. IMAGE PRE-PROCESSING

Before applying the edge detection filters, a preprocessing step is needed to reduce the noise. In this section a speckle filtering approach is discussed.

Speckle filtering based on 2D homogeneity histogram analysis

The described method is a modified version of the algorithm proposed in [10].

The general model of an image affected by speckle noise is:

$$g(i,j) = f(i,j)u(i,j) + \zeta(i,j) \tag{3.1}$$

where:

– $g(i,j)$ is the initial noisy image;

– $f(i,j)$ is the initial clean image;

– $u(i,j)$ is the multiplicative component of the speckle noise;

– $\zeta(i,j)$ is the additive component of the speckle noise;

– $0 \le i \le H-1; \quad 0 \le j \le W-1,$ with (H, W) image dimension.

For some specific images, by neglecting the additive speckle noise component, the model (3.1) simplifies to:

$$g(i,j) = f(i,j)u(i,j) \tag{3.2}$$

Applying a logarithmic function, the multiplicative component $\mathrm{u(i,j)}$ becomes the additive component $\mathrm{u_l(i,j)}$:

$$g_l(i,j) = f_l(i,j) + u_l(i,j) \tag{3.3}$$

where $g_l(i,j), f_l(i,j), u_l(i,j)$ are the logarithms of $g(i,j), f(i,j), u(i,j)$ images.

The method proposed in [10] estimates the homogeneity of a bimodal histogram resulted from the texture analysis of $g_l(i,j)$ image. The homogeneity is used to evaluate the speckle characteristics. A high homogeneity value indicates a homogenous region, otherwise the region has strong texture characteristics due the speckle noise. The result of the homogeneity matrix processing is a threshold used to assign the image pixels to "Hs" class (homogenous) or to "NHs" class (non-homogenous). Finally, non-homogenous regions are filtered using directional average filters in order to reduce the speckle noise, preserving in the same time the edges.

To conclude, the above method is a four steps approach:

1. Using the texture indicators, a bi-dimensional homogeneity histogram is built.
2. Compute a threshold from the homogeneity histogram using an entropy measure and classify each image pixel in "Hs" or "NHs" class.
3. Filter the non-homogenous areas by directional average filters.
4. Repeat steps 1–3 until a homogenous ratio is reached.

To get the texture features, the following operators are used:

$$\begin{aligned} L5 &= \begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix} \\ E5 &= \begin{bmatrix}-1 & -2 & 0 & 2 & 1\end{bmatrix} \\ S5 &= \begin{bmatrix}-1 & 0 & 2 & 0 & -1\end{bmatrix} \\ W5 &= \begin{bmatrix}-1 & 2 & 0 & -2 & 1\end{bmatrix} \\ R5 &= \begin{bmatrix}1 & -4 & 6 & -4 & 1\end{bmatrix} \end{aligned} \tag{3.4}$$

where:

– $L5$ is the average level operator;
– $E5$ is the edge operator;
– $S5$ is the spot operator;
– $W5$ is the ripple operator;
– $R5$ is the wave operator.

Using the masks:

$$L5^{\mathrm{t}}\,xE5=\begin{bmatrix}-1 & -2 & 0 & 2 & 1\\ -4 & -8 & 0 & 8 & 4\\ -6 & -12 & 0 & 12 & 6\\ -4 & -8 & 0 & 8 & 4\\ -1 & -2 & 0 & 2 & 1\end{bmatrix}\quad L5^{\mathrm{t}}\,xS5=\begin{bmatrix}-1 & 0 & 2 & 0 & 1\\ -4 & 0 & 8 & 0 & 4\\ -6 & 0 & 12 & 0 & 6\\ -4 & 0 & 8 & 0 & 4\\ -1 & 0 & 2 & 0 & 1\end{bmatrix}$$

$$E5^{\mathrm{t}}\,xL5=\begin{bmatrix}-1 & -4 & -6 & -4 & -1\\ -2 & -8 & -12 & -8 & -2\\ 0 & 0 & 0 & 0 & 0\\ 2 & 8 & 12 & 8 & 2\\ 1 & 4 & 6 & 4 & 1\end{bmatrix} \qquad (3.5)$$

$$S5^{\mathrm{t}}\,xL5=\begin{bmatrix}-1 & -4 & -6 & -4 & -1\\ 0 & 0 & 0 & 0 & 0\\ 2 & 8 & 12 & 8 & 2\\ 0 & 0 & 0 & 0 & 0\\ -1 & -4 & -6 & -4 & -1\end{bmatrix}$$

for each pixel, a texture indicator $f(x,y)$ is calculated:

$$f(i,j)=\sqrt{(f_{L5^t\,xE5}(i,j))^2+(f_{L5^t\,xS5}(i,j))^2+(f_{E5^t\,xL5}(i,j))^2+(f_{S5^t}} \qquad (3.6)$$

The texture indicator (3.6) is then normalized:

$$F(i,j)=\frac{f(i,j)-f_{\min}}{f_{\max}-f_{miin}} \qquad (3.7)$$

where $f_{\max}=\max\{f(i,j)\},f_{\min}=\min\{f(i,j)\},0\le i\le H-1,0\le j\le W-1$.

For each pixel, the homogeneity parameter is normalized into [0, K] interval:

$$Ho(i,j)=[Kx(1-F(i,j)] \qquad (3.8)$$

In [10] the normalizing constant is $K=100$. A local average value of homogeneity is then computed for w = 5:

$$\overline{Ho}(i,j) = \frac{1}{w \times w} \sum_{m-i-(w-1)/2}^{i+(w-1)/2} \; \sum_{n-j-(w-1)/2}^{j+(w-1)/2} Ho(m,n) \tag{3.9}$$

In (3.9) $w = 5$ because the masks (3.5) are 5 × 5 matrices. Finally a bidimensional homogeneity histogram is built:

$$h_{Ho,\overline{Ho}}(m,n) = \sum_{0 \le i \le H-1, 0 \le j \le W-1} \delta(Ho(i,j) - m, \overline{Ho}(i,j) - n) \tag{3.10}$$

where:

$$Ho_{\min} \le m \le Ho_{\max}, \overline{Ho}_{\min} \le n \le \overline{Ho}_{\max}, \quad \delta(p.q) = \begin{cases} 1 & p = q \\ 0 & p \ne q \end{cases} \tag{3.11}$$

and $Ho_{\min}, Ho_{\max}$ are the minimum and maximum values of $Ho(i,j)$ elements, while $\overline{Ho}_{\min}, \overline{Ho}_{\max}$ are the minimum and maximum values of $\overline{Ho}(i,j)$ elements. In the second step of the algorithm, from $h_{Ho,\overline{Ho}}(m,n)$ histogram a threshold $T(Ho_{th}, \overline{Ho}_{th})$ is computed in order to assign each pixel to Hs or NHs class.

Let $Hop(i,j)$ be the probability distribution for the homogeneity i and the average homogeneity j, with $\mathrm{i,j} = 1,2,\ldots..,\mathrm{N}, \mathrm{N} = [\max(Ho(m,n), \overline{Ho}(m,n))]$. Assuming a threshold $T(Ho_{th}, \overline{Ho}_{th})$ was provided, the homogeneity space can be divided in two clusters: "foreground" HoF and "background" HoB. The entropies of these clusters are defined by the relations:

$$H_{\mathrm{HoB}}(s,t) = -\sum_{i=1}^{s} \sum_{j=1}^{t} \frac{Hop(i,j)}{HoP(s,t)} \ln \frac{Hop(i,j)}{HoP(s,t)} \tag{3.12}$$

$$H_{HoF}(s,t) = -\sum_{\mathrm{i=s+1}}^{\mathrm{N}} \sum_{\mathrm{j=t+1}}^{N} \frac{Hop(i,j)}{1 - HoP(s,t)} \ln \frac{Hop(i,j)}{1 - HoP(s,t)} \tag{3.13}$$

where:

$$Hop(i,j) = \frac{1}{HxW} h_{\mathrm{Ho},\overline{\mathrm{H}o}}(i,j) \tag{3.14}$$

$$HoP(s,t) = \sum_{i=1}^{s} \sum_{j=1}^{t} Hop(i,j) \tag{3.15}$$

The threshold is done by:

$$T(Ho_{\mathrm{th}}, \overline{Ho}_{\mathrm{th}}) = Arg \max_{1 \le \mathrm{s} \le \mathrm{N}, 1 \le \mathrm{t} \le \mathrm{N}} \{H_{\mathrm{HoF}}(s,t) + H_{\mathrm{HoB}}(s,t)\} \tag{3.16}$$

Given the threshold $T(Ho_{th}, \overline{Ho}_{th})$, the two clusters Hs, NHs are:

$$Hs = \{P(i,j) \mid Ho(i,j) \geq Ho_{th}, \overline{Ho}(i,j) \geq \overline{Ho}_{th}\} \tag{3.17}$$

$$NHs = \{P(i,j) \mid Ho(i,j) < Ho_{th}, \overline{Ho}(i,j) < \overline{Ho}_{th}\} \tag{3.18}$$

To reduce the speckle noise, the NHs cluster pixels are filtered by the directional average filters:

$$\overline{g}(i,j) = \begin{cases} g(i,j) & (i,j) \in Hs \\ DAFg(i,j) & (i,j) \in NHs \end{cases} \tag{3.19}$$

where DAF is the directional average filter.

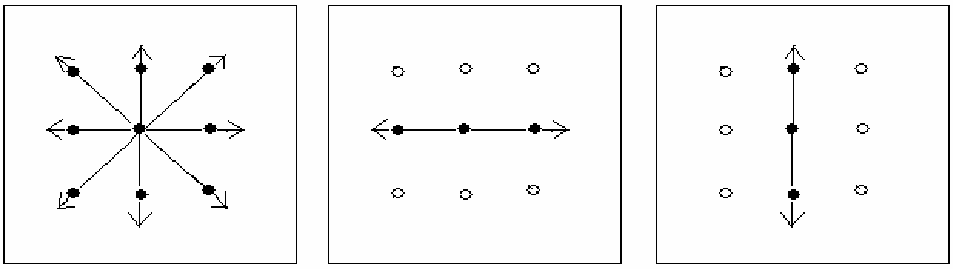


Fig. 1. Directional filter (from [10]).

As illustrated above, the filter direction is calculated from the NHs neighbors of the current NHs pixel. A Sobel operator is applied on a neighborhood of the current pixel $g(i,j)$ $(i,j) \in NHs$. The results $e_h(i,j)$ and $e_v(i,j)$ are normalized:

$$Eh(i,j) = \frac{e_h(i,j) - e_{\min}}{e_{\max} - e_{\min}} \tag{3.20}$$

$$Ev(i,j) = \frac{e_v(i,j) - e_{\min}}{e_{\max} - e_{\min}} \tag{3.21}$$

where

$e_{\max} = \max(e_h(i,j), e_v(i,j)), e_{\min} = \min(e_h(i,j), e_v(i,j))$ $(0 \leq i \leq H-1, 0 \leq j \leq W-1)$.

Applying successively the masks $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$:

$$\mathrm{M}_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathrm{M}_2 = \frac{1}{3}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathrm{M}_3 = \frac{1}{3}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.22}$$

on the original image $\mathbf{I}$ the images $\mathbf{R}_1 = Cov(\mathbf{M}_1, \mathbf{I})$, $\mathbf{R}_2 = Cov(\mathbf{M}_2, \mathbf{I})$ and $\mathbf{R}_3 = Cov(\mathbf{M}_3, \mathbf{I})$ are obtained.

The filter DAF from (3.19) is defined by:

$$DAF(g(i,j)) = \begin{cases} \mathrm{R}_1 & E_h(i,j) = E_v(i,j) \\ \mathrm{R}_2 & E_h(i,j) > E_v(i,j) \\ \mathrm{R}_3 & E_h(i,j) < E_v(i,j) \end{cases} \tag{3.23}$$

As already mentioned, the filtering process continues if a homogeneity condition is not meet. This condition is given by:

$$HR = \frac{Num(Hs)}{H \times W} \tag{3.24}$$

where $Num(Hs)$ is the number of „homogenous" pixels. If the new HR is smaller than a predefined threshold the filtering process is resumed with step 1. Due to the fact HR diminishes after steps 1–3, the process is convergent.

Figure 2 from [10] illustrates the results of various filtering methods applied on synthetic images, versus the presented method results.

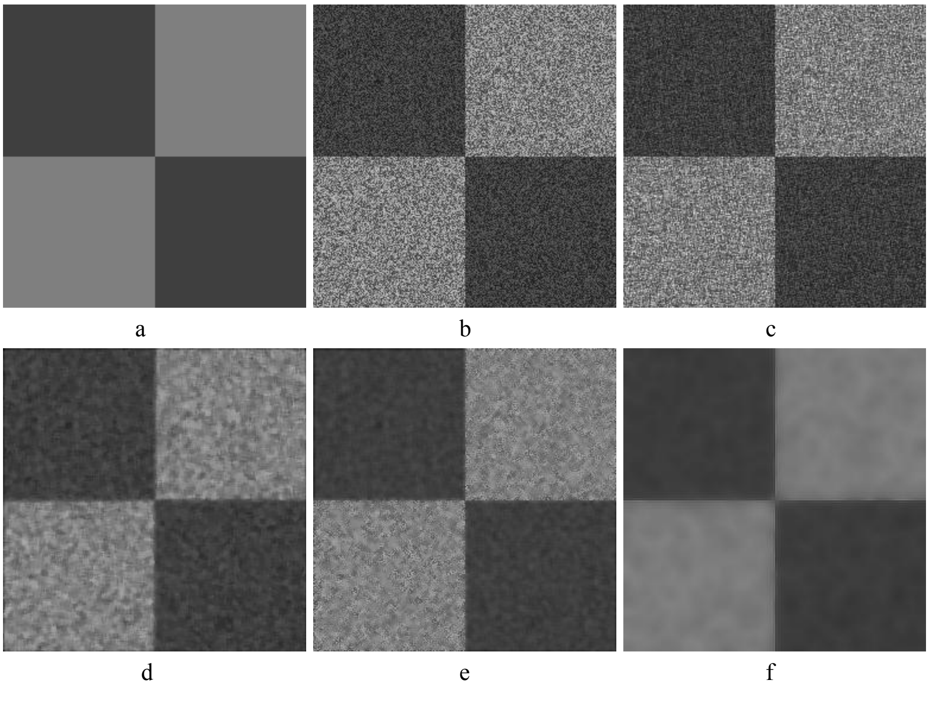


Fig. 2. a) Synthetic image; b) image (a) affected by speckle noise; c) median filtering d) Wiener filter e) wavelet filtering; f) the presented method (images from [10]).

3.2. EDGE DETECTION

Edges determine image brightness discontinuities that occur on the boundary between different regions. The pixels located on the edge are detected by determining the local extremes of the first derivative or by determining the zero-crossing of the second derivative. Another method used in edge detection is the gradient magnitude which provides information about the strength and direction of the edge [9, 24].

$$magn(\nabla f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = \sqrt{M_x^2 + M_y^2} \tag{3.25}$$

$$dir(\nabla f) = arctg\frac{M_y}{M_x} \tag{3.26}$$

The gradient is approximated using finite differences:

$$\frac{\partial f}{\partial x} = \frac{f(x+h_x, y) - f(x,y)}{h_x} = f(x+1, y) - f(x,y) \text{ if } h_x = 1 \tag{3.27}$$

$$\frac{\partial f}{\partial y} = \frac{f(x, y+h_y) - f(x,y)}{h_y} = f(x, y+1) - f(x,y) \text{ if } h_y = 1 \tag{3.28}$$

3.3. CANNY EDGE DETECTOR

The Canny filter [9, 12, 24] is the most used edge detector, offering a good detection (stronger response at the edge location than to noise), good localization (exact edge location) and low false edges (there should be only one maximum in a reasonable neighborhood of the real edge pixel).

The Canny edge detector is based on the fact that the first derivative of the Gaussian closely approximates the operator that optimizes the product of signal-to-noise ratio and localization. The analysis is based on “step-edges” corrupted by “additive Gaussian noise”.

For the input image I and a Gaussian G with zero mean and standard deviation σ the Canny filter is implemented as follows:

1. $J = I * G$ (convolution);

2. For each pixel (i, j) of J:

– Compute the image gradient: $\nabla J(i,j) = (J_x(i,j), J_y(i,j))'$;

– Estimate edge strength: $e_s(i,j) = (J_x^2(i,j) + J_y^2(i,j))^{1/2}$;

– Estimate edge orientation $e_0(i,j) = \arctan(J_x(i,j)/J_y(i,j))$;

– Consider the four directions set $D = \left\{0^\circ, 45^\circ, 90^\circ, 135^\circ\right\}$;

3. For each pixel (i,j) of e_s and e_0:

– find the direction $d \in D$ so $d \cong e_0(i,j)$;

– if { $e_s(i,j)$ is smaller than at least one of its neighbors along d}

then $I_N(i,j) = 0$;

else $I_N(i,j) = e_s(i,j)$;

4. Considering a low threshold value L and a high threshold value H for each pixel (i,j) of I_N do:

– Locate the next unvisited pixel with $I_N(i,j) > H$;

– Starting from $I_N(i,j)$ follow the chains of connected local maxima, in both directions perpendicular to edge normal, as long as $I_N > L$ and mark all visited points and save the location of the contour points. This list will be the output of the Canny filter.

4. EDGE FUSION

Different information fusion techniques were used for edge detection. Li proposed in [15] an edge detection method based on decision-level information fusion to classify image pixels into edge and non-edge categories. Giannarou and Tania Stathaki described in [8] a framework for the quantitative fusion of edge maps that arise from both different preselected edge detectors and multiple image realizations.

4.1. EDGE FUSION PROCEDURE

In this section, a fusion procedure is proposed to combine features obtained from different spectral bands using the same edge filter. We must note that the source images must be registered.

The fusion method is based on the weighted average of the pixel intensity in the corresponding source images. The weights are determined using the trust level of the edges to which the pixels belongs.

– Let $I_k, k = \overline{1,n}$ be the input spectral images, already registered. The Canny filter is applied to all the spectral images, creating a set of output images $\widetilde{I}_k, k = \overline{1,n}$.

– For each image $\widetilde{I}_k$ a labeling procedure is applied to obtain the connected components and evaluate their area (number of 8-connected pixels).

– Let $a_{\min}^k, a_{\max}^k$ be the extreme values of areas for the connected components in the image I_{k},

– For each image $\widetilde{I}_k$, the range $[a_{\min}^k, a_{\max}^k]$ is divided in p+1 partitions $P_i, i = \overline{0, p}$, and each connected component is assigned to a partition depending on its area. All the components with a reduced number of pixels are assigned to partition P_0 and will be eliminated from future processing steps.

– For each component (edge) C in the remaining partitions $P_i, i = \overline{1, p}$, the trust level TL_C is defined by the index of the partition to which the component belongs:

$$TL_C = t, \quad a_{\min} + t(a_{\max} - a_{\min}) / p \le area(C) \le a_{\min} + (t+1)(a_{\max} - a_{\min}) / p$$

– In images $\widetilde{I}_k$, the pixel values are replaced by the trust level of the corresponding edge or 0 if the pixel is not part of an edge.

$$\widetilde{I}_k(i.j) = \begin{cases} TL_C & if\ \widetilde{I}_k(i,j) \in C \\ 0 & if\ \widetilde{I}_k(i,j) \notin any\ edge \end{cases}$$

– The image I_{edge} (edge) is computed as the weighted average of images $\widetilde{I}_k$. The value of each pixel is the membership trust level to an edge.

– In the last step, an edge validation is performed using an user defined threshold.

if $I_k(i,j) \ge threshold$ then the pixel (i,j) is part of an edge,
else the pixel (i,j) is not part of an edge.

4.2. EXPERIMENT. MULTISPECTRAL IMAGE SEGMENTATION

A multispectral image is actually represented by a set of images obtained for certain frequencies of the electromagnetic spectrum. The separation is achieved using various filters or devices that are sensitive to certain wavelengths. In the usual panchromatic images the sensor records the complete intensity of the radiation.

The proposed algorithm allows multispectral segmentation by identifying the edges in a number of images of the same scene taken in different spectral bands, followed by combining the results. The images are assumed to be aligned.

To test the described method the multispectral data collection available in „Multispectral Image Database" [34] was used. Each collection contains images in the visible spectrum, wavelength between 400 and 700 nm in steps of 10 nm. All

31 images are in '.png' format, 16-bit gray levels. The figure bellow shows the original scene illuminated with a neutral light in full color format.

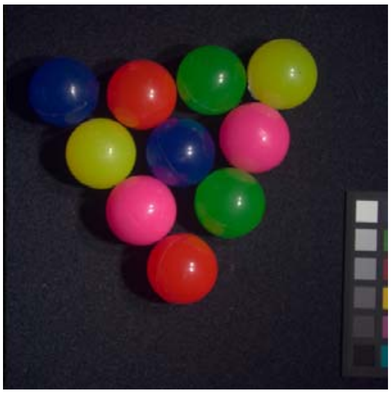
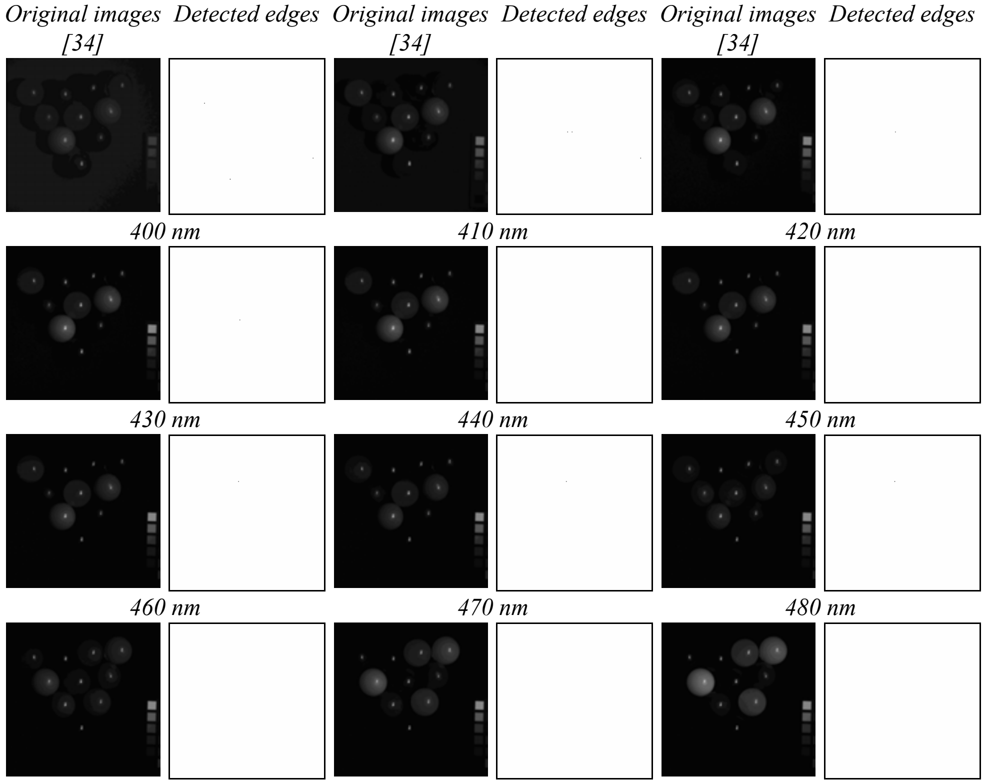


Fig. 3. Original image, full color [34].

The pictures in Figure 4 ilustrate the results obtained using the algorithm described above. The figure contains pairs of images: original image and extracted edges for each of the 31 spectral bands.



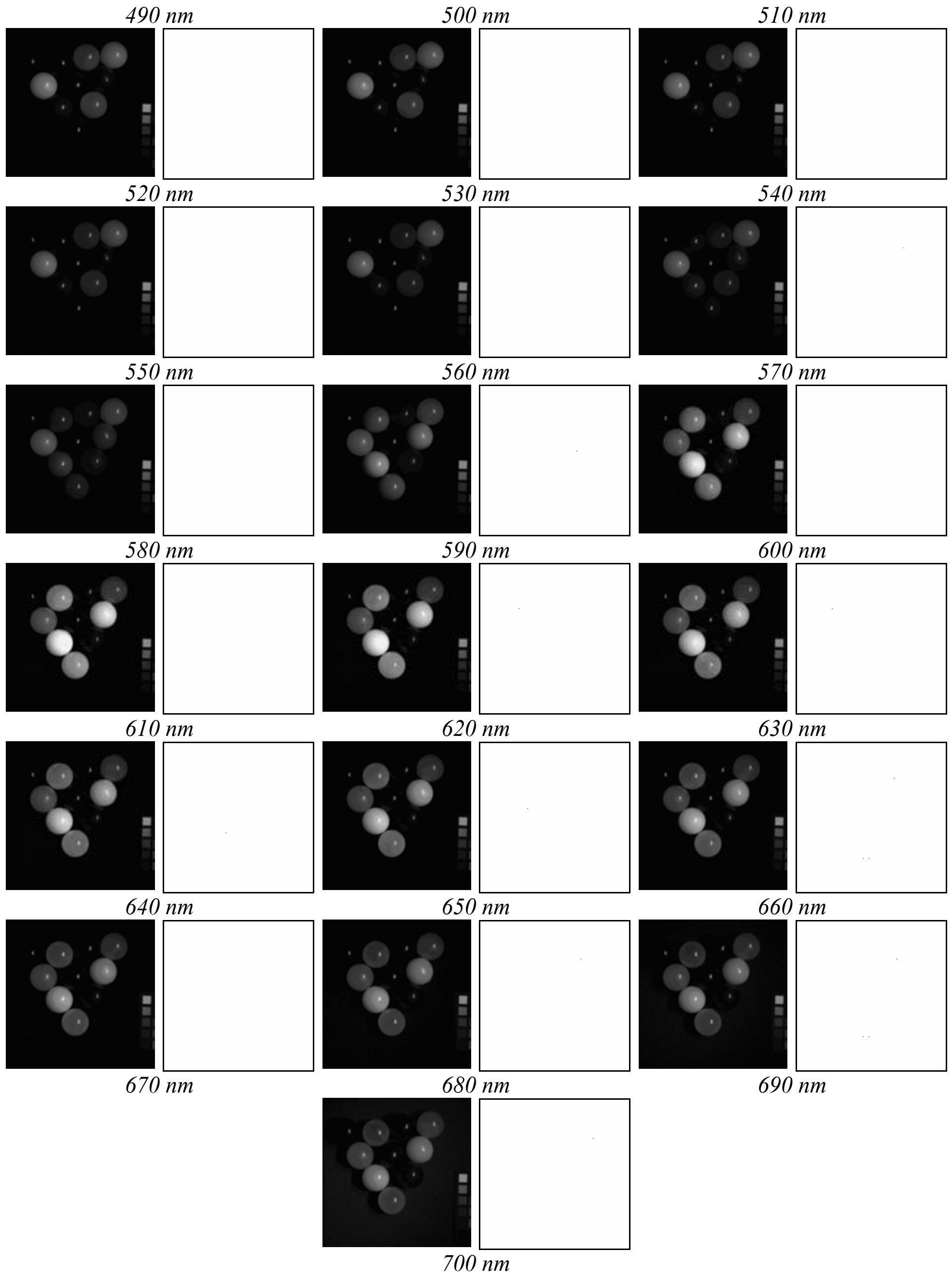


Fig. 4. Processing results for all 31 spectral bands. The original images are from [34]; the result images are obtained by the authors.

It should be noted that in none of the resulting images were not detected more than 6 contours. The result of the fusion procedure is presented in the Figure 5 – all the contours are detected. The fused edges set was obtained using the described method with the following parameters: $p = 8$, $threshold = 0.08$.

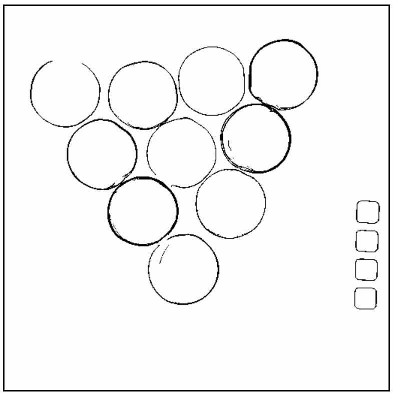


Fig. 5. Result of the fusion procedure for edges detection in multispectral images (image obtained by authors).

Finally, to check the quality of the determined contours, a circle detection procedure was applied on the result image. The circles are approximated using the modified version of the Hough transform. The implementation available in the Open CV library was used [35].

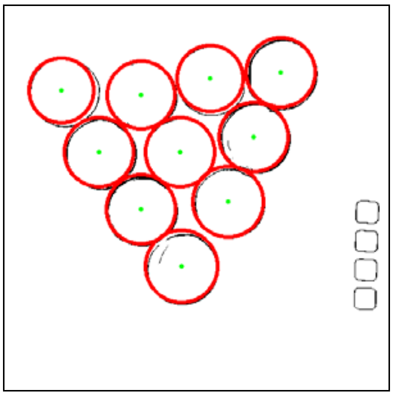


Fig. 6. Circles detected using the modified Hough transform (image obtained by authors).

The quality of the results is underlined by the radius of determined circles. The values are in range [43.1, 47.8] (pixels). Most of them are 46 pixels length. The erroneous values are determined for balls located on the limit of the group and are caused by the existing shadows.

5. CONCLUSIONS

In spite of the large amount of work in the field, an ideal scheme able to detect and localize edges with precision in many different contexts has not yet been produced. Some edge detectors are robust to noise, but the generated contours are not complete, without corners and junctions.

It is proposed a new segmentation approach for edge detection in multispectral images using image fusion techniques. First, the input images are pre-processed to reduce the noise and then the Canny operator is applied to detect the edges in each image. The results are combined using a fusion method based on the weighted average of the pixel intensity to obtain a more accurate description of the objects in the input image. The output of the fusion procedure is more complete than any of the input data sets. The proposed method was implemented in Matlab and tested on a large number of multispectral images obtaining satisfactory results.

Our future research will be focused on fusion of texture information and fusion of geometric features.

Authors' contributions: S. Bejinariu made the description of data and image fusion concept, designed and implemented the segmentation procedure. F. Rotaru made the speckle noise filtering and applied the features extraction methods. C.D. Nita tested and experimented the multispectral image segmentation procedure. The authors consider that they equally contributed to the paper.

REFERENCES

1. BERGHOLM F., *Edge focusing*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, **9**, *6*, 726–741.
2. BLUM R., LIU Y. (Editors), *Multi-Sensor Image Fusion and Its Applications*, Signal Processing and Communications, CRC Press, 2005, 1–36.
3. BURT P.J., KOLCZYNSKI R.J., *Enhanced image capture through fusion*, Proceedings of the 4th International Conference on Computer Vision, 1993, Berlin, Germany, 173–182.
4. BURT P.J., ADELSON E.H., *The Laplacian pyramid as a compact image code*, IEEE Trans. Commun., 1983, **31** (4), 532–540.
5. CANNY J.F., *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, **8**, *6*, 679–698.
6. CHIPMAN L.J., ORR T. M., LEWIS L. N., *Wavelets and image fusion*, Proceedings IEEE International Conference on Image Processing, Washington D.C., USA, 1995, 248–251.
7. DERICHE R., *Using Canny's criteria to derive a recursive implemented optimal edge detector*, International Journal of Computer Vision, 1987, **1**, *2*, 167–187.
8. GIANNAROU S., STATHAKI T., *Fusion of edge maps using statistical approaches*, Image Fusion – Algorithms and Applications, Academic Press, Elsevier, 2008, 173–298.
9. GONZALES R.C., WOODS R.E., *Digital Image Processing*, Prentice-Hall, Inc., 2002, 568–584.
10. GUO Y., *Computer-Aided Detection of Breast Cancer Using Ultrasound Images*, PhD Thesis, Utah State University, 2010; HALL D.L. and LLINAS J. (Editors), *Handbook of multisensor data fusion*, CRC Press LLC, 2001, 26–44.
11. JAIN R., KASTURI R., SCHUNCK B. G., *Machine Vision*, McGraw-Hill, 1995, 140–185.

12. KOREN I., LAINE A., TAYLOR F., *Image fusion using steerable dyadic wavelet transforms,* Proceedings IEEE International Conference on Image Processing, Washington D.C., USA, 1995, **3**, 232–235.
13. LACROIX V., *The primary raster: A multiresolution image description*, Proc. 10th International Conference on Pattern Recognition, Atlantic City, New Jersey, USA, 1990, Vol. **1**, 903–907.
14. LI J., *Edge Detection Based on Decision-Level Information Fusion and its Application in Hybrid Image Filtering*, Proceedings of International Conference on Image Processingm 2004, ICIP '04, Singapore, Vol. **1**, 251–254**.**
15. LI H., MANJUNATH L.H., MITRA S.K., *Multisensor image fusion using the wavelet transform,* Graphical Models and Image Processing, Academic Press, Inc. Orlando, FL, USA, 1995, 51–55.
16. MARR D., HILDRETH E.C., *Theory of edge detection*, Proceedings of the Royal Society of London, UK, Series B, 1980, **207,** 187–217.
17. MATTHEWS J., *An introduction to edge detection: The Sobel edge detector*, available at http://www.generation5.org/content/2002/im01.asp, 2002
18. MOIGNE J.L., CROMP R.F., *The use of wavelets for remote sensing image registration and fusion*, Proceedings of SPIE, 1996, Vol. **2762**, 535–544.
19. NALWA V.S., BINFORD T.O., *On detecting edges*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **8**, No *6*, 1986, 699–714.
20. NIKOLOV S., HILL P., BULL D., CANAGARAJAH N., *Wavelets for Image Fusion,* in *Wavelets in Signal and Image Analysis*, Kluwer AP, 2001, 213–244.
21. PETROVIC V., XYDEAS C., *Cross band pixel selection in multi resolution image fusion*. Proceedings of SPIE 1999, Vol. **3719**, 319–326.
22. PIELLA G., HEIJMANS H., *A new quality metric for image fusion*. Proceedings of International Conference on Image Processing. Barcelona, Spain, 2003, Vol. **2**, 173–176.
23. PRATT W.K., *Digital Image Processing*, John Wiley & Sons, Inc., 2001, 443–508.
24. ROBERTS L.G., *Machine Perception of 3-D Solids*, Optical and Electro-Optical Information Processing, MIT Press, 1965, 159–197.
25. ROCKINGER O., *Pixel-level fusion of image sequences using wavelet frames* (K.V. Mardia, C.A. Gill and I. L Dryden, Eds.), Proceedings in Image Fusion and Shape Variability Techniques, Leeds, UK, 1996, 149–154.
26. ROCKINGER O., *Image sequence fusion using a shift invariant wavelet transform*, Proceedings of the IEEE International Conference on Image Processing, Washington DC, USA, 1997, Vol. **3**, 288–291.
27. ROSENFELD A., THURSTON M., *Edge and curve detection for visual scene analysis*, IEEE Transactions on Computers, C-20, 1971, 562–569.
28. ROTHWELL C.A., MUNDY J.L., HOFFMAN W., NGUYEN V.D., *Driving vision by topology*, International Symposium on Computer Vision, Coral Gables, FL, USA, 1995, 395–400.
29. WILSON T.A., ROGERS S.K., MYERS L.R., *Perceptual based hyperspectral image fusion using multiresolution analysis,* Optical Engineering, 1995, **34** (*11*), 3154–3164.
30. WITKIN A.P., TENENBAUM J.M., *On the role of structure in vision,* In *Human and Machine Vision* (J.Beck, B. Hope, and A. Rosenfeld, Eds.), Academic Press, New York, USA, 1983, 481–544.
31. XYDEAS C., PETROVIC, V., *Objective pixel-level image fusion performance measure*, Proceedings of SPIE, 2000, **4051**, 88–99.
32. ZHANG Z., BLUM, R. S., *A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application*, Proceedings of IEEE, 1999, **87** (*8*), 1315–1326.
33. ⁂ *Multispectral Image Database*, http://www1.cs.columbia.edu/CAVE/databases/multispectral/
34. ⁂ *Open Source Computer Vision Library, Reference Manual*, Copyright 1999–2001 Intel Corporation.

Received April 10, 2012