Image Filter Design with Interactive Evolutionary Computation

Jaksa, Rudolf
Kyushu Institute of Design

Nakano, Shota
Kyushu Institute of Design

高木, 英行
Department of Art and Information Design, Kyushu Institute of Design

出版情報：pp.1-, 2003-08-29
バージョン：
権利関係：

Image Filter Design with Interactive Evolutionary Computation*

Rudolf Jakša†
Kyushu Institute of Design
4-9-1 Shiobaru, Minami-ku,
Fukuoka, 815-8540
Japan
jaksa@neuron.tuke.sk

Shota Nakano
Kyushu Institute of Design
4-9-1 Shiobaru, Minami-ku,
Fukuoka, 815-8540
Japan
nakano@artemis.aid.kyushu-id.ac.jp

Hideyuki Takagi
Kyushu Institute of Design
4-9-1 Shiobaru, Minami-ku,
Fukuoka, 815-8540
Japan
takagi@kyushu-id.ac.jp

Abstract – We apply an interactive evolutionary computation technique to processing images in medical and other fields, analyze the performance of the image enhancement, and discuss its capabilities and possibilities for further improvement. We first show our system scheme and the design of the user interface and then show experimental results of the image enhancement. Next, we also propose an idea for a new interactive genetic programming method that, in addition to the arithmetic operators, also has image processing operators in generated mathematical equations to increase its performance for image processing.

I. INTRODUCTION

Signal processing's areas of application are widely spread. With this increase in the range and the number of applications, the number of cases where the application user is not a signal processing expert has also increased. As most conventional signal processing is based on mathematics, it is difficult for non-signal processing experts to design signal processing systems. At the same time, signal processing experts may not possess the domain knowledge of the target application fields. Medical image processing is typical of such cases.

Interactive evolutionary computation (IEC) is one solution which allows signal processing application users to design signal processing systems based on user hearing or vision [4, 5, 6].

In this paper, we focus on IEC-based image processing and evaluate how visual-based image filter design works well without any image processing knowledge. This is a suitable approach for image processing where the users are not signal processing experts. Our approach is to specify the input-output characteristics of a coloring filter with linear spline functions for each of the RGB colors, apply IEC to optimize the parameters of the spline functions based on the users' visual inspection, and interactively design the coloring filters accordingly.

First we describe the generic approach of IEC-based image processing and image filters in sections II. Next we show several experimental results of the IEC-based image processing in section IV. Finally, we introduce a novel approach, using image processing operators, to create powerful filters with genetic programming (GP) in section V.

*This work is supported by Japan Society for the Promotion of Science.
†Author is on leaving from Technical University of Košice, Slovakia.

II. INTERACTIVE EVOLUTIONARY COMPUTATION

A. Interactive Evolutionary Computation Framework

IEC is a form of EC where a human is used instead of a *fitness function* [5]. A basic schematic of an IEC-based image processing application is illustrated in Fig. 1.

Interaction between a human user and a computer, indicated on the right and left sides of Fig. 1 respectively, is conducted through a user interface. The user interface displays candidate images to the IEC user and collects the user evaluations of these images.

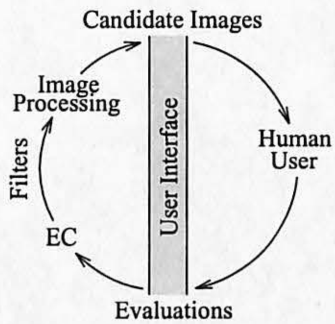

Fig. 1: IEC-based image processing loop

The computer part of IEC consists of an EC part and an image processing part. User-inputed evaluations are used by the EC to optimize individual image processing filters. The role of the image processing part is to generate (the next generation of) images using image filters created by EC. This interactive optimization process loops iteratively.

The advantage of IEC image processing is:

a) if a user is looking for particular filter or features, the user can develop or find it in this process, and
b) if the user wants to find some new unknown features of a certain image – to explore the image, the user can accomplish that through this process as well.

The last filter obtained is the result in case a), while it is an interactive process as whole in case b); case a) represents an exploitative behavior, while case b) the explorative one.

B. User Interface

The user interface is an component with IEC, a respect in which it is unlike other EC algorithms. Several types of user interface for IEC-based image processing can be configured. The most basic role of the user interface is to display the sets of images to user. However, the impression of the GUI interface is dramatically different between cases where 10 images are displayed and where 100 images are displayed at once. Therefor, thumbnails or groupings of images are sometimes used for interface design.

The trade-off here lies between the ability to see details in the image and ability to see the overall appearance of image. This might be balanced by usage of zooming and/or cropping of the image (displaying only a part of it). We are constrained by the need to display several images in parallel. If we reduce the the size of a member of the population too much, the optimization performance may be limited. If we display only a portion of population at once, the ability of user to compare particular individuals may be limited, as he/she must recall non-displayed images. Optimal setting of these parameters of IEC-based image processing will be application or task dependent.

The image-evaluation part of the interface might also be configured in several ways. The user might search and pick out interesting images, select best images or precisely evaluate them. Furthermore, there is a difference between continuous and discrete evaluation, as well as between 3-value and 5-value ones. If the user is looking for any interesting features of the image and wants to see a lot of different views of the image, he/she will be frustrated by the need to precisely evaluate every single image. In a different scenario, however, where the user knows the target filter behavior, the user will want to precisely evaluate slightly different images in order to fine tune the resulting filter.

User interface handles two information flows: one from computer to human and the other from human to computer. The total bandwidth of the information flow through the user interface is then the sum of these two flows. We may assume that a higher bandwidth means a deeper interaction. The information exchanged in interactions may, however, differ dramatically in quality. Using more expressive image presentations, the user evaluations may be more accurate and may lead to faster convergence of the search. To measure this quality of information, the sum of future evaluations might be used.

We may thus determine that a user interface is better if:

a) with the same type of presented images and the same type of evaluation, the bandwidth of information flow through user interface is higher,
b) with the same bandwidth, the overall sum of evaluations through time is better.

Note that overall performance of an algorithm depends not only on the performance of the user interface, but also on the performance of the EC algorithm, the performance of the image processing part of algorithm and the abilities of a particular user as well.

III. IMAGE PROCESSING FILTERS

A. Algebraic Filters

Algebraic filters compute the output pixel value using input pixel information and an arbitrary mathematical function to describe the input/output relation. To evolve such a filter the natural choice is to use GP methods, since they can build the structure of the function and also optimize its parameters. However, it's possible to build such a filter with predefined structuretype using a simple GA for its optimization.

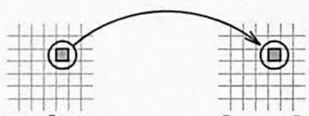

Fig. 2: An arbitrary mathematical function may be used as an image processing filter to map input pixel values to their corresponding output pixel values.

The following mathematical function is an example of an algebraic filter as discussed:

$$r_{out} = 100\sin(r_{in}/20) + 50\log(g_{in}) - 120 + r_{in}/b_{in} \quad (1)$$

Here, r_{out} is the intensity of the output pixel in its red channel and r_{in}, g_{in} and b_{in} are its intensities in the red, green and blue channels. This filter realize a one-to-one mapping of pixels in the images. To be able to achieve edge detection or other operations that rely on relationships between different pixels in the input image, we must include several input pixels in the filter function.

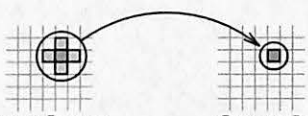

Fig. 3: Many-to-one pixel mapping filter

An algebraic filter function for implementing a many-to-one pixel mapping may take the following form:

$$r_{out} = 0.2r_{-1,0} + r_{0,0} + 0.1g_{0,-1} + 0.2g_{0,0} - 0.1g_{0,1} \quad (2)$$

Where $r_{-1,0}$ is the red channel intensity of the pixel at the position $(-1,0)$, (left) of the current pixel.

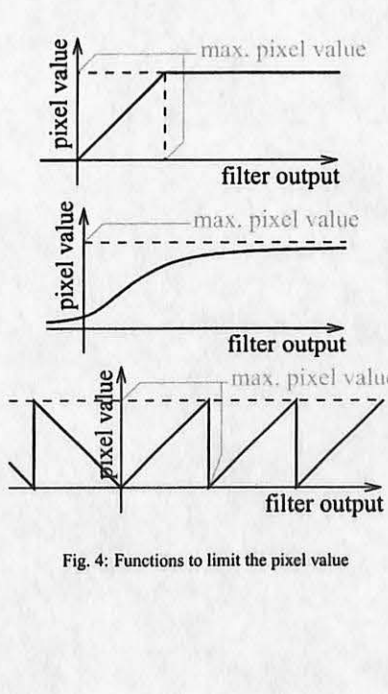

Fig. 4: Functions to limit the pixel value

The maximal value of pixel intensity is usually limited in images to 8, 12, 16, etc. bits per pixel. When using arbitrary mathematical functions to compute pixel value this maximal value of pixel might be easily exceeded. To deal with this problem we must limit the output of the filter back to the dynamic range of image. We may use limiting functions to do this, such as those shown in Fig. 4. All these functions are nonlinear and may have a dramatic impact on the function of the filter.

An autoscale method may be used instead of these limiting functions to overcome this nonlinear behavior. However, the autoscale method is specific to a given filter and may be image specific. This may prohibit the use of given filter with other images and/or may affect the dynamics of the IEC search process. Also, the dynamic range of the output image will be fixed when using autoscale method; the brightest and darkest pixels will be always of the same intensity.

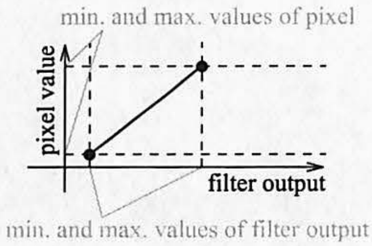

Fig. 5: Autoscaling of pixel value

Why not employ an IEC optimization method also to optimize the limiting function? In the ideal situation, the output of the filter function will never exceed the maximal value of pixel, and thus we wouldn't need a limiting function. However, the initial filters in IEC image processing are not optimal (usually random) and the optimal solution is found through the optimization process. Therefore, if we use filters which are able to exceed the permitted range, in order for the process to proceed correctly we must explicitely limit the output pixel values to the correct range.

For more details on algebraic filters in IEC image processing see [6].

B. Parametric Filters

Parametric filters with fixed structure are based on parametric functions, and IEC is used to optimize only their parameters. If the number of parameters is fixed, a simple GA algorithm will be sufficient for the optimization of such a filter. An example of this type of filter might be one consisting of three functions for adjusting pixel intensities in each of the red, green and blue (R,G,B) channels, where every of these functions is a linear spline function with a fixed number of "joints".

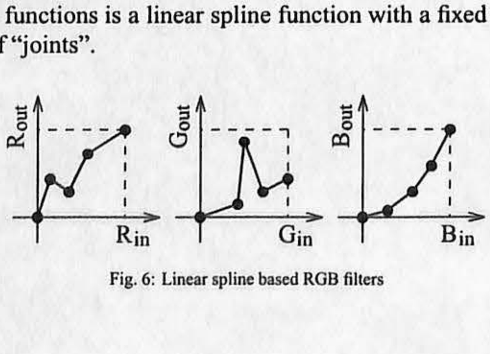

Fig. 6: Linear spline based RGB filters

The filters in Fig. 6 are severely limited in function compare to the algebraic filter of the type in equation (1). The filters in equation (1) are of the type:

$$r_{out} = f_r(r_{in}, g_{in}, b_{in}) \tag{3a}$$

$$g_{out} = f_g(r_{in}, g_{in}, b_{in}) \tag{3b}$$

$$b_{out} = f_b(r_{in}, g_{in}, b_{in}) \tag{3c}$$

The spline based RGB example filters (Fig. 6) are of type:

$$r_{out} = f_r(r_{in}) \quad g_{out} = f_g(g_{in}) \quad b_{out} = f_b(b_{in}) \tag{4}$$

Note, however, that while the "simplicity" of this filter type will limit it in function, it will also reduce the search space for the IEC and may accelerate the search. Furthermore, this type of limitation of filter abilities might be useful in that it already prohibits the IEC from changing the image too much.

To overcome the limitation of the given simple filter we might build a more capable parametric filter from modules of the type $r_{out} = f_r(r_{in}, g_{in})$, as in Fig. 7.

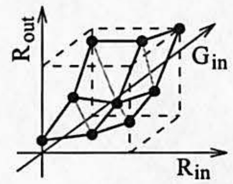

Fig. 7: Two-dimensional linear spline filter

Finally, to build a parametric filter fully compatible with those in equation (3) we must use three-dimensional spline functions. Such a 3D parametric filter may still not be fully equivalent to the algebraic filter in equation (1). Linear spline functions have a rough shape compared to functions, such as $x2$ or $\sin(x)$, which may be used in an algebraic filter. To smooth the shape of the spline, cubic splines may be used or the number of "joints" may be increased. Instead of splines, other functions may be used for parametric filters as well, such as Taylor series or Fourier series functions.

The structure in Fig. 8 represents a many-to-one pixel parametric filter. However, the sum in Fig. 8 is not general enough, so for a truly general case we have to use multidimensional splines. Therefore, for 5 pixels the in input image and 3 channels (red, green, blue), we have to use a 15-dimensional spline function.

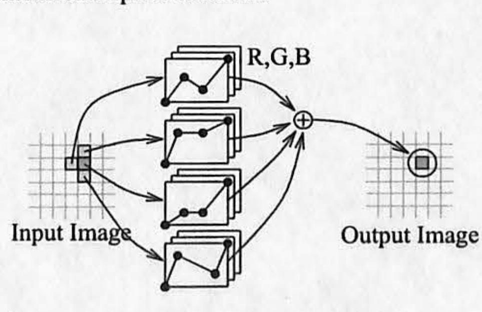

Fig. 8: Multiple pixel spline filter

C. Structured Filters

Structured filters are filters built from a set of parametric or algebraic filters. This allows us to achieve the characteristics of algebraic filters while using parametric filters as building blocks. See Fig. 9 for example.

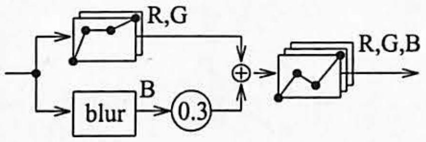

Fig. 9: Structured filter

This type of filter mimics the conventional approach, where image processing is accomplished by combining several predefined types of filters. Thus, the advantage of this type of filter is its similarity to traditional approaches and this might be welcome to many potentials users. Furthermore, the dynamics of the IEC search might be controlled when using structured filters by enabling or disabling particular building blocks.

To evolve this type of filter, a GP like algorithm is necessary to adapt the structure of the filter.

IV. EXPERIMENTAL STUDIES

A. Setup of Experiments

In our experiments we use the schematic shown in Fig. 10 with parametric filters (algebraic filters are discussed in [6]).

The input image is filtered by an initial set of random filters (F1,F2,...F9) and filtered images are displayed for evaluation. After the user evaluates those images, an EC algorithm is used to generate a new set of filters (F10,F11,...F18). These are constructed by the application of crossover and mutation operations on the previous set of filters according to user evaluations of them. New filters are again applied to the input image and displayed for further evaluation. This process is repeated until the user is satisfied with the result and, in general.

B. IEC-based Contrast Enhancement

Our objective in the first experiment is to enhance details in an input image and the results obtained were from the 7th generation. The input image itself already had enhanced contrast using the histogram equalization method, which means that the distribution of grey-scale levels in it is somewhat optimal and further enhancement may not be trivial. To further enhance contrast and details in a particular area of image, manual manipulation of color intensity curves might be used. However, to maintain good contrast in several areas of the image might be difficult.

Parametric linear spline filters of the type shown in equation 4 were used to create the image in Fig. 11. The input image is grey-scale, and because the curves of the splines for the red, green and blue channels are different, the resulting filtered image is in color. Because of this "colorization", the appearance of the image is slightly worse in grey-scale print than it is on a color display. The detail in the grey-scale image may only be encoded by intensity differences between pixels, whereas in the color image, the detail might be also be encoded using color information.

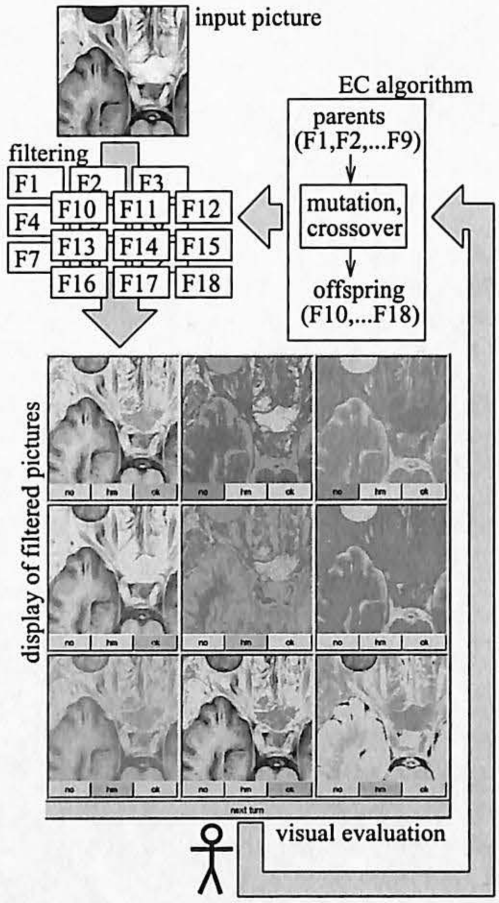

Fig. 10: Experimental setup of IEC image processing application

The primary advantage of IEC-based Contrast Enhancement is the IEC idea itself – the feedback is based on subjective visual evaluation and requires no expert level knowledge in the image processing area. Using expert knowledge, results similar to those in Fig. 11 might be achieved by application of a locally adaptive contrast enhancement filter. However, to do this, the user must know which filter to apply, what the behavior of this filter is and how it will match to his impression of a good result.

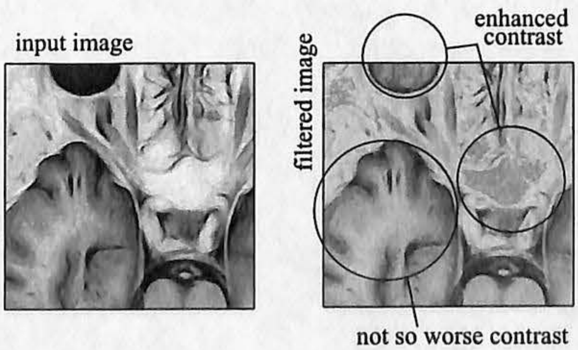

Fig. 11: Example result

Furthermore, in the case of "image exploration" the user does not know what exactly he/she is looking for in advance. In such a case, the user might just try random filters and random combinations of filters. Using IEC image processing brings some sophistication to this process. Also note that in contrast with the images produced using our IEC image processing approach, images prepared using contrast enhancement will not introduce color information to the image.

C. Comparison with manual image enhancement

We compared the IEC-based image enhancement discussed in this paper with manual image enhancement. Photo retouch software has the same capabilities as our IEC-based image enhancement *if* its users have experience and skill. In this section, we evaluate how the IEC-based image enhancement is an effective support system for beginners.

An experimental evaluation was conducted with a non-professional subject using photo retouch software. The task was to increase the visibility of road parts in an aerial-photograph by controlling a density parameter in Photoshop® software and by applying the IEC-based method. Each image enhancement experiment was conducted for 10 minutes. As photo retouch software has no function to manipulate RGB densities individually, images made using the IEC approach were transformed to gray-scale images and compared with those made by Photoshop®.

Fig. 12: Images enhanced by the IEC-based approach (bottom) and by Photoshop® (upper).

The experimental results, Fig. 12, show that both results look similar, though comparison with one subject is not enough to conclude anything. However, we can say that it is difficult for non-experts to find the proper image processing parameters and optimize them within limited time in real applications, while the subject of this experiment knew that the density parameter was the correct parameter to adjust.

D. Image enhancement filter design by an expert for medical image diagnosis

Medical image diagnosis is a typical case where signal-processing users who want to design filters are not signal processing experts. As medical doctors do not have knowledge of signal processing in general, conventional filter design methodologies are not suited to them for design and tuning of image enhancement filters for their visual inspection.

To show the practical use of IEC-based image processing, a medical image diagnostic expert designed a filter, based on his own visual evaluation.

The experimental task used an ultrasonic image of lymph node in 16-bit DICOM format file as shown in Fig. 13. The task was to design a filter that increased the visibility of the lymph node part circled in the figure.

The enhanced image obtained after 20 IEC generations is shown in Fig. 13. The enhanced image is a color image, while the original image is monotone one. It took about 20 minutes to obtain the enhanced filter.

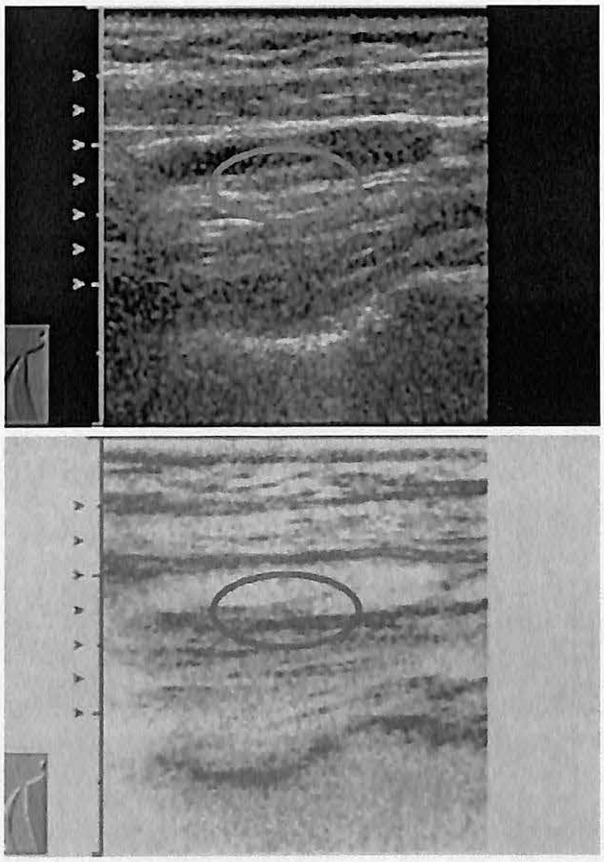

Fig. 13: Original ultrasonic image of lymph node (upper) and the IEC-enhanced image created by a medical image diagnostic expert (bottom).

From the view point of the satisfaction level of the medical doctor who designed this filter and the design time used, we can say that the IEC-based image enhancement filter design approach has potential for practical use in the medical field.

V. FUTURE EXPANSION WITH IMAGE PROCESSING OPERATORS

User fatigue in IEC is interesting topic. Using this method, the response of the EC algorithm is important for the user experience. The user wants to see the results of his evaluations reflected immediately in the next generation. The features provided by the EC algorithm may be important for enhancing this type of dynamic optimization. Also important is the choice of the type of filter, as it influences the characteristics of the EC search space. In future we want to use structured filters (see Fig. 9) as they allow great configuration flexibility, and are thus well suited for study of search space impact on the dynamics of IEC searches in image processing.

The dynamics of IEC image processing can best be analyzed by breaking the search process into three stages. In the initial stage, the user looks for any meaningful filters. In second stage, the user evolves a set of filters to meet his/her wishes or to find any interesting hidden features in the image. In third and final stage, the user fine-tunes the filters to obtain a perfect result. We observed difficulties in the third stage of the search, and hopefully with a better understanding of the dynamics of this process we may be able to improve it.

User interface is another interesting topic in IEC research. Considering the computer-to-human information flow, increasing the information bandwidth here may improve IEC performance. One method to accomplish this would be to use a client-server architecture for the IEC image processing software to use multiple displays for feeding information to the user. Another idea would be to employ motion video for the images and thereby increase the bandwidth of the information flow by time-encoding the image information. Further motivation for a client-server architecture is in the initialization of the IEC search. We used random initializations here, but good image processing filters from previous users' experiences could be stored by a server and used to initialize subsequent searches. Because the number of generations in the IEC search is limited, this increase of speed at the start of the search would be welcome.

On the human-to-computer side of the information flow, increasing the quantization of the evaluation to more than five values is probably not very interesting. It is, however, possible to increase user feedback by giving the user the ability to directly modify filter parameters. This ability doesn't make much sense when using algebraic filters or parametric filters of more than two dimensions, but it is a viable option when using structured filters and provides a compelling motivation to use structured filters in IEC image processing. Finally, when using complex algebraic or parametric filters, visualization of the search space, as in visualized IEC [1], may help to enrich user feedback.

VI. CONCLUSION

We analyzed the problem of filter type choice in IEC-based image processing. Application of algebraic, parametric or structured filters have several consequences on the behavior and performance of the system as whole. We discussed also the configuration of the user interface and the explorative versus exploitative mode of IEC operation. Our experiments with parametric filters based on the linear splines demonstrated the usefulness of our method. The next step after the application of parametric and algebraic filters in IEC-based image processing is to implement structured filters. We see several potential advantages of this type of filter and we plan to use them in our future work.

VII. REFERENCES

[1] Hayashida, N. and Takagi, H., "Acceleration of EC Convergence with Landscape Visualization and Human Intervention," *Applied Soft Computing*, Elsevier Science, Vol. 1, No. 4F, pp. 245-256 (2002).

[2] Mutoh, T. Komagata, N. and Ueda, Y., "An experimental study for automatically generating image filter sequence by using simulated breeding", in *Workshop on Interactive Evolutionary Computation*, Fukuoka, Japan, pp. 7-12, (Mar., 1998) (in Japanese).

[3] Otoba, K. Tanaka, K. and Hitafuji, M., "Image processing and interactive selection with Java based on genetic algorithms." in *3rd IFAC/CIGR Workshop on Artificial Intelligence in Agriculture* Makuhari, Japan, pp. 83-88, (Apr., 1998).

[4] Poli, R. and Cagnoni, S., "Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement," in *2nd Annual Conf. on Genetic Programming*, Palo Alto, CA, USA, pp. 269-277, (1997).

[5] Takagi, H., "Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation," *Proceedings of the IEEE*, vol.89, no.9, pp.1275–1296 (Sept., 2001).

[6] Takagi, H. and Hayashida, N., "Interactive EC-based Signal Processing," *4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL2002)*, Singapore, pp. 375-379 (Nov., 2002).