# Towards an Universal Approach to Background Removal in Images of Bankchecks

**Katrin Franke, Mario Köppen**

*Fraunhofer Institute for Production Systems and Design Technology IPK,*
*Pascalstr. 8-9, D-10587 Berlin, Germany,*
*Fon: +49 30 39006194, Fax: +49 30 3917517,*
*E-mail: {katrin.franke, mario.koeppen}@ipk.fhg.de*

This paper is concerned with an approach to the automated design of filters for background removal of bankchecks. The approach is derived from a simplified version of the morphological subtraction and its interpretation, using the 2D-Histogram. From an empty bankcheck, a mask is computed, which can be added to every filled-in bankcheck of the same kind, filtering the user entered information parts within the bankcheck image. This approach works well for many types of bankchecks, but it fails for EC bankchecks. By considering the 2D- Histogram of empty and filled-in EC bankcheck images, a means is given for improving the result for EC checks, using the so-called 2D-Lookup algorithm. Moreover, this algorithm allows for the use of the filled-in image only, if there is a way to specify two filter operations, which process the filled-in image in order to get a separable 2D-Histogram. An evolutionary algorithm, the genetic programming (GP), is used for performing this adaptation task. The evolutionary generated filters shows very good performance for EC bankcheck background removal. A detailed study gives the importance of a family of image processing operations, the OWA operations, for this removal.

## 1 Introduction

Automatic bankcheck processing is an established subject of research and development activities. Besides of the handwriting recognition itself, the extraction of user entered information is an important processing stage for providing accurate recognition results [10, 9]. Some algorithms have been proposed for the extraction of user entered information [5]. These algorithms can be grouped into algorithms for layout analysis, like automatic location of text blocks; algorithms for background texture and/or image cleaning, as histogram and threshold based techniques; algorithms for foreground cleaning, as guideline and pre-printed data removal; and algorithms for handwriting reconstruction. There are different strategies for the cleaning of background and foreground. Some approaches uses separated cleaning steps [2, 9, 3, 1] others use combined cleaning methods for fore- and background [10, 8]. However, traditional automatic bank check processing is based on the assumption that the background cleaning is only a particular task of the check processing, and some simple algorithms are used for background removal. Recently, several researchers have criticized this assumption and proposed alternative and enhanced approaches [10 5 9 8].

Usually for the removal of nearly homogenous backgrounds different kind of histograms and threshold techniques are used. Against it, is was pointed out by Okada and Shridhar [10], that enhanced inter-image subtraction, the so-called morphological subtraction, provides acceptable cleaning results for textured and/or image check backgrounds. In this paper we propose a method for background cleaning which provides results which are comparable to the morphological subtraction [10]. In contrary to the approach by Okada and Shridhar [10], a blank reference check image is filtered once in an off- line phase and is only used for grayvalue comparison during the background cleaning process. For the comparison, a segmented 2D-Histogram is used for looking-up. This way it will be possible to provide qualitatively comparable cleaning results by using a smaller amount of computational time. Moreover, it will be shown, that the 2D-Histogram-Lookup can also be derived as a abstraction of the former inter-image subtraction, which gives a more flexible approach to inter-image filtering. This is caused by the fact, that for the graylevel lookup a blank check image is not mandatory. It is also possible to use two different filtered images of one filled-in check image.

In this manner, the difficult procedure for positional adjustment between filled-in and blank check image can be avoided. The fundamental problem for a practically and efficient usage of the 2D-Histogram is the selection of filter operations, which provide significant differences between background and user entered information. For solving this problem, the Lucifer II framework for filter design is used. Lucifer II was developed for textural filter design in the context of surface inspection, and it uses evolutionary algorithms for its adaptation [6]. The feasibility of using the Lucifer II framework for the design of check background filters will be presented in this paper. This paper is organized as follows. The revised approach of morphological subtraction [10] and the derivation of the 2D-Histogram-Lookup will be outlined in section 2. The Lucifer II framework and its exploitation for check background filter design will be shortly described in section 3. Results and the practical application of the designed check background filter operation will be presented in section 4, followed by a summary and the reference.

## 2 Revised morphological subtraction and the 2D-Histogram

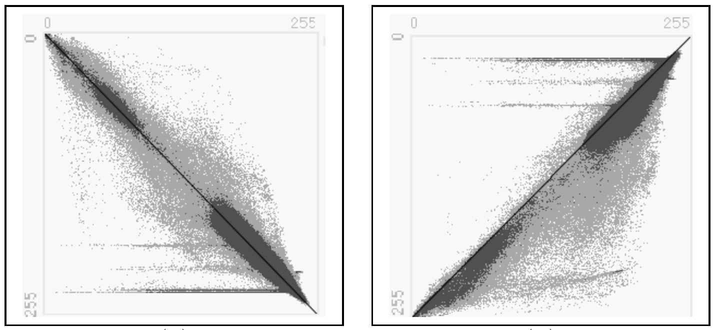
Usually, background removal is considered as a task of somehow "subtracting" the given reference image of an empty bankcheck from the image of the bankcheck, including the user entered information. Due to ambiguities in the scanning process (misalignment, skew, different preprocessing of the scanning device, color variations in the printing process), the direct subtraction of empty and filled-in bankcheck image would not lead to an exact match of all regions, which contains no user entered information, as it would be the hypothetical case.

Therefrom, the concept of "subtraction" was extended in order to relax a little bit from the strong demand of an exact match between empty and filled-in image. The decay from an exact match to the real case of mismatching goes on over at least two levels. Firstly, for the intensity values it is assumed that they remain constant. The only source for differences between reference and filled-in image is geometric misalignment. Therefrom, it can be expected for each pixel position within the images to find the corresponding pixel value within the local neighborhood of the pixel. This case is considered in [10], where so-called morphological subtraction is introduced, and in [3], where the concept of pixel tolerance is directly exploited. However, the second level of mismatching is based on the assumption, that even the pixel values may be different to a reasonable degree. Unfortunately, this case has to be considered as the usual one in the context of practical applications.

In the following, a rather simple approach is presented, which copes with the problem of small intensity and positional variations at once.

From the reference image of the bankcheck, a mask image is constructed as follows: the image is eroded (effectively broadens dark regions) and after that, it is inverted. This mask image can be applied to every filled-in image by simply pixelwise addition. Clearly, the mask sweeps out all corresponding positions of pattern and reference, leaving only the user-entered information and some easyly-removable debris within the result image. Two things might be unexpected for this operation: it is far more simple than the computational effort e.g. taken in [10], and it is an addition of a mask instead of a subtraction of a reference. However, the basic idea underlying this operation can be easily illustrated, if the 2D-Histogram of grayvalue pairs at corresponding image positions of filled-in image and reference image is considered.

Figure 1a shows such an 2D-Histogram of an empty reference and a filled-in bankcheck. Ideally, there would be only entries along the main diagonal of the matrix. Due to the named effects, the true activations are spread along the main diagonal. Now, the morphological erosion is applied, and the eroded image is inverted, with the effect, that every activation is moved towards the lower-diagonal half of the matrix (see figure 1b). All the positions within this triangle have a common property: the sum of the corresponding grayvalues exceeds 255. If the mask is added to the reference, from which it originated, or to another pattern, the result image is nearly completely white. By the same




(a) (b)

Figure 1: 2D-Histogramm of an empty reference and a filledin banckcheck before (a) and after (b) reference filtering.

procedure, the upper-diagonal triangle of the histogram matrix is "emptied," leaving space for activations caused by the user-entered information. The tolerance for such activations is limited by the diagonal of the matrix, i.e. if user-entered information at a given place has grayvalue $g$, the underlying grayvalue of the reference must not exceed $255 - g$ in order to be separable by this approach.

From this, the presented mask approach may be reformulated: if the mask image is given, the matrix of the 2D-Histogram of mask and filled-in image is computed. In the result image, only activations are kept, which corresponds to positions of the upper-diagonal triangle of the matrix. These are the same positions as the set of positions, for which the sum of mask and filled-in image does not exceed 255. But what has changed is the manner, which is used to register user entered information. Instead of an arithmetic property of its pixel values, a positional property of the matrix of the 2D-Histogram is considered.

Both manners become quite more different, if more complicated bankchecks are considered. The quality of separation is worse, by considering all activations in the upper-diagonal triangle. However, the result can be remarkably improved, if the case is considered from the "viewpoint" of 2D-Lookup matrix segmentation. Manually, a segment can be crafted, which registers much better for the grayvalue appearance of the image parts containing user entered information. Therefrom, the question arises, if there is a more general approach to background removal based on 2D-Histogram matrix segmentation. The answer is positive, and a possible approach will be presented in the following sections.

Contominant to the introduction of 2D-Histogram matrix segmentation, which will be referred to as 2D-Lookup in the following, is the basic question about the origins of the two images, which are used for the look-up. So far, they have been the mask image, which was derived from the reference image, and the filled-in image, usually untreated. But, the 2D- Lookup can be applied to either pair of grayvalue images. This opens the possibility to refrain from using reference images at all. Hence, the goal of employing a 2D-Lookup is extended to use the filled-in image only, what opens the possibility of a universal, reference-free approach to background removal in bankcheck images.

## 3 Lucifer II framework

### *3.1 General Overview*

The framework is composed of (user-supplied) original image, filter generator, filter output images 1 & 2, result image, (user-supplied) goal image, 2D-Lookup matrix, comparing unit and filter design signal.

An evolutionary algorithm maintains a population of individuals, each of which specifies a setting of the framework. By applying the resulting 2D-Lookup and measuring the quality of coincidence of goal and result image, a fitness value can be assigned to each individual. These fitness measures are used for the standard genetic operations of an evolutionary algorithm. The 2D-Lookup algorithm, the fitness measure, the node and terminal functions of the individual's expression trees and the setting of the 2D-Lookup matrix will be shortly described in the next subsections. A more comprehensive introduction to the framework is given elsewhere [6].

### *3.2 2D-Lookup Algorithm*

The 2D-Lookup algorithm stems from mathematical morphology [11] [12]. The algorithm was generalized to apply for two grayvalue images. For starting off the 2D-Lookup algorithm, the two filter images 1 & 2, which are of equal size, need to be provided. This is achieved by the filter design signal, which is in control of the individuals of the evolving population. The filter design signal causes the filter generator to determine two image processing operations, which are applied to the original image. The 2D-Lookup algorithm goes over all positions of the filter images. For each position, the two pixel values at this position in filter images 1 & 2 are used as indices for looking-up the 2D-Lookup matrix. The matrix element, which is found there, is used as pixel value for this position of the result image. If the matrix is bi-valued (as for the goal image), the result image is a binary image.

### *3.3 Fitness function*

A fitness measure is given by the degree of coincidence of goal image and result image. Both are binary images. The fitness measure, which is computed by the comparing unit, is a weighted sum of three single measures: the quota of white pixels in the result image, which are also white in the goal image (**whitegoalok**), the quota of black pixels in the result image, which are also black in the goal image (**blackgoalok**) and the quota of black pixels of the goal image, which are also black in the result image (**blackresok**). Note, that **blackgoalok** and **blackresok** are different. The multiple objective here is to increase these measures simultaneously. After performing some experiments with the framework, it was decided to use the following weighted sum for these three objectives:

$$\mathrm{f} = 0.1\,\mathbf{blackresok} + 0.5\,\mathbf{whitegoalok} + 0.4\,\mathbf{blackgoalok}.$$

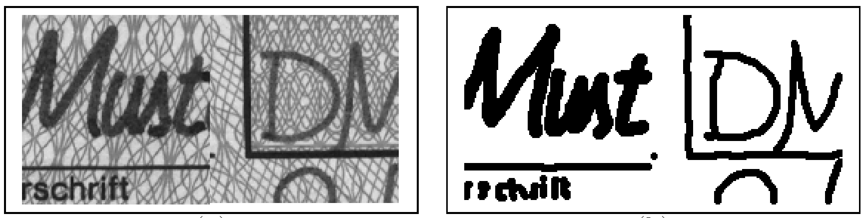
This fitness function was designed in order to direct the genetic search according to the schemata theorem, by which higher weighted objectives are fulfilled first.

### *3.4 Genetic Design of Filter Operations*

One essential part of the framework are the two image processing operations. In order to generate them, genetic programming is used [7]. Genetic programming maintains a population of expression trees (shortly: trees). Every tree equals a program by its structure. For the design of the trees, the following operators were used as terminal functions: Move, Convolution, Ordered Weighted Averaging (OWA) [14], Fuzzy integral [4 13], Texture Numbers. The node functions were out of the set of the operations minus, minimum, maximum, square, evaluate. In all tests, a maximum number of 50 generations was used. For details, and also for the relaxation-based technique for the setting of the 2D-Lookup matrix, consider [6].

## 4 Case study for EC bankchecks

The LUCIFER II framework was applied to a patch image of a filled-in EC bankcheck. EC bankchecks are considered to be the most important family of bankchecks in Europe, and also to comprise the most challenging background filtering problem. The patch image and its binary goal image are shown in figure 2. The GP runs were configured as follows: 20 parent trees, 50 children trees, a maximum initial tree depth of 2 and of 3 during the run. For mating selection, tournament selection was used.




(a) (b)

Figure 2: Original (a) and goal (b) image used for the study.

Most runs gave promising results. Two examples with the corresponding filters and 2D- Lookup matrices are given in figures 3 and 4. However, the generated filters have to be redesigned in order to be practically applicable. For this reason, the case was explored in more detail.

At first, ten runs of the framework were performed. The proportion of every operator in the final population was determined, and the results were averaged over the number of runs. The results (in percent) were as follows:

| OWA | Fuzzy integral | Move | Convolution | Texture numbers |
|---|---|---|---|---|
| 41.6 | 11.8 | 23.2 | 23.4 | 0 |

These results indicate, that the OWA operator is the most important operator for background filtering of EC bankchecks. Next are convolution and move (including logical) operations. The Fuzzy integral performs fair, and texture numbers are useless in the context of EC bankchecks.
Referring to the remarks given in section 2, both, movement and OWA operations cope with the main sources of differences between two images of bankchecks of the same kind, i.e. geometric misalignment (move operations) and grayvalue differences (OWA). Hence, a combination of both operations should perform best. The result presented next is the single run for the OWA operation alone (see figure 5).

## 5 Summary

A framework was presented, which allows for the design of filters for background removal of bankcheck images. Using genetic programming, the filters are adapted to extract the user entered information parts within the bankcheck image. The framework is based on the 2D- Lookup algorithm. The feasibility

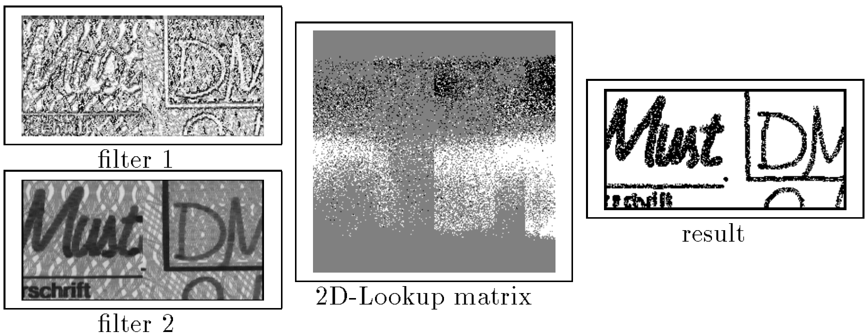



Figure 3: Example result 1

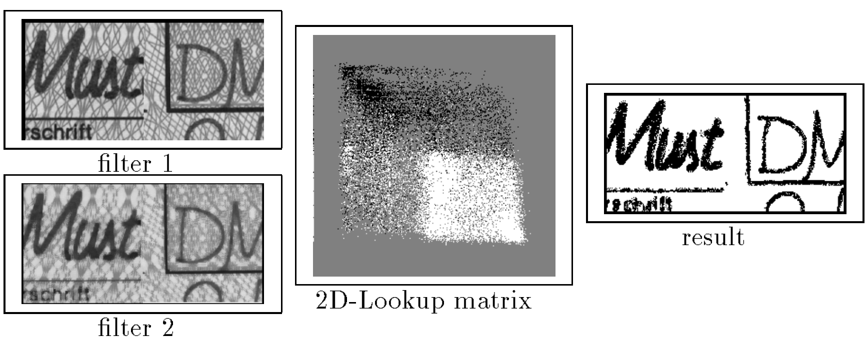



Figure 4: Example result 2

of using it for background removal was motivated by the discussion of a simplified version of morphological subtraction and the failure of this approach for more complicated bankcheck backgrounds (e.g. the EC bankcheck). The results could be improved by manually crafting a segmentation of the 2D-Histogram matrix and using this segmentation for 2D-Lookup. The task of the Lucifer II framework, which is presented here, is to design filter operations, which act on the filled-in image only in order to generate the two images necessary for computing a 2D-Histogram. The fitness of a GP individual expression tree is computed from the separation ability of the fully-specified 2D-Lookup. For expressing filter operations, the GP individual is designed using superoperators as terminals and a small set of node functions. The setting of the 2D-Lookup matrix is specified by a relaxation-based technique. From

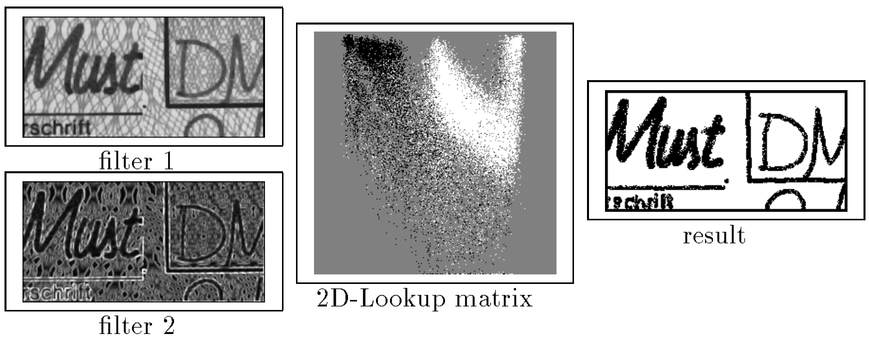



Figure 5: Result for running OWA alone. Fitness was 0.0118. The best result with a plain segmentation of the 2D-Lookup matrix.

that, goal-fulfilling filters are evolved. The results are considered for the case of EC bankchecks. The best results were achieved for expression trees, which mainly contains OWA operations. This operation and the simple move operation together seems to comprise the most promising constituents for EC bankcheck background filter. This approach can be applied to every case, where the simple approach of mask addition, which was presented in section 2, fails to fulfill its goal. Further studies will concentrate on a better treatment of cluttered 2D-Histograms, which causes for example overemphasizing the role of the convolution operation for EC background removal.

## References


1. Cheriet, M., Said, J. N., Suen, C. Y., "A Formal Model for Document Processing of Business Forms", *Proc. of the 3rd International Conference on Document Analysis and Recognition (ICDAR)*, Montreal, Canada, August 14-16, vol. 1, pp. 210-213 (1995).
2. Dimauro, G., Impedove, S., Prirlo, G., Salzo, A., "Automatic Bankcheck Processing: A New Engineered System", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4 pp. 467-504 (1997).
3. Djeziri, S., Nouboud, F., Plamondon, R., "Extraction of Items From Checks", *Proc. of the 4th International Conference on Document Analysis and Recognition (ICDAR)*, Ulm, Germany (1997).
4. Grossert, S., Köppen, M., Nickolay, B., "A new approach to fuzzy morphology based on fuzzy integral and its application in image processing", *Proc. ICPR'96*, Vienna, Austria, Vol. II, pp. 625-630 (1996).

5. Impedovo, S., Preface," Special Issue: Automatic Bankcheck Processing", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4 pp. 463-466 (1997).
6. Köppen, M., Teunis, M., Nickolay, B., "A Framework for the Evolutionary Generation of 2D-Lookup Based Texture Filters", *Proc. IIZUKA'98*, Iizuka, Japan, pp. 965-970 (1998).
7. Koza, J., "Genetic programming - on the programming of computers by means of natural selection", *MIT Press*, Cambridge, London (1992).
8. Lee, L. L., Lizarraga, M. G., Gomes, N. R., Koerich, A. L.," A Prototype for Brazilian Bankcheck Recognition", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4 pp. 549-570 (1997).
9. Liu, K., Suen, C. Y., Cheriet, M., Said, J. N., Nadal, C., Tang, Y. Y., "Automatic Extraction of Baselines and Data from Check Images", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4 pp. 675-697 (1997).
10. Okada, M., Shirdhar, M., "Extraction of User Entered Components from a Personal Bankcheck Using Morphological Substacltion", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 5 pp. 699-715 (1997).
11. Serra, J., "Image analysis and mathematical morphology", *Academic Press*, London (1982).
12. Serra, J., ed., "Image analysis and mathematical morphology. Volume 2: theoretical advances", *Academic Press*, London (1988).
13. Sugeno, M.," Theory of fuzzy integral and its applications", Tokyo Inst. of Technology, *Ph.D. thesis* (1974).
14. Yager, R.R.," On ordered weighted averaging aggregation operators in multi-criteria decision making", *IEEE Trans. on SMC 18*, pp. 183-190 (1988).