

Accepted Manuscript

Semiautomatic Identification of Print Layers from a Sequence of Sample Images: A Case Study from Banknote Print Inspection

Daniel Soukup, Ulrich Bodenhofer, Markus Mittendorfer-Holzer, Konrad Mayer

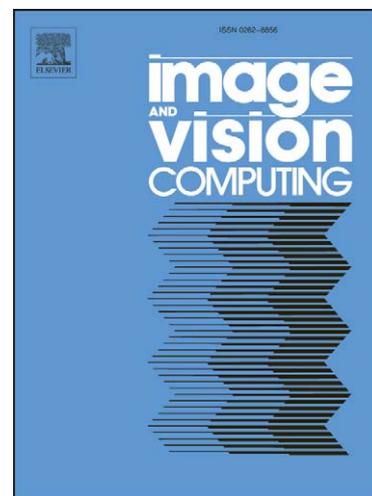
PII: S0262-8856(08)00182-0
DOI: [10.1016/j.imavis.2008.08.008](https://doi.org/10.1016/j.imavis.2008.08.008)
Reference: IMAVIS 2768

To appear in: *Image and Vision Computing*

Received Date: 5 October 2005
Revised Date: 30 May 2008
Accepted Date: 18 August 2008

Please cite this article as: D. Soukup, U. Bodenhofer, M. Mittendorfer-Holzer, K. Mayer, Semiautomatic Identification of Print Layers from a Sequence of Sample Images: A Case Study from Banknote Print Inspection, *Image and Vision Computing* (2008), doi: [10.1016/j.imavis.2008.08.008](https://doi.org/10.1016/j.imavis.2008.08.008)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



Semiautomatic Identification of Print Layers from a Sequence of Sample Images: A Case Study from Banknote Print Inspection[☆]

Daniel Soukup^a, Ulrich Bodenhofer^{b,*}, Markus Mittendorfer-Holzer^c, Konrad Mayer^a

^aAustrian Research Centers, ARC smart systems Division, A-2444 Seibersdorf, Austria

^bInstitute of Bioinformatics, Johannes Kepler University Linz, A-4040 Linz, Austria

^cCOMNEON electronic technology, A-4040 Linz, Austria

Abstract

This article presents an approach for finding displacements of print layers from sequences of sample images semi-automatically with the aim to simplify and shorten the setup of inspection systems for printing processes in which the perfect alignment of print layers cannot be guaranteed. The basic idea behind the proposed approach is to identify pixels which are likely to have the same displacements for a given pair of images. This relatively coarse information is computed for several pairs of sample images and aggregated in order to identify regions that tend to have the same displacement over a large proportion of image pair comparisons. This idea is motivated and justified in detail. The test cases considered in this study are data from banknote print inspection. We use these data to illustrate the steps of the algorithm. The examples demonstrate the method's capability to sensibly identify print layers, even if they overlap partially. Although the paper concentrates on a particular case study, the method can be used in any print inspection process with similar characteristics.

Key words: Block matching; Local correlation; Print inspection; Separation of print layers

1. Introduction

A vast majority of visual print inspection systems are based on comparisons between a given reference and the samples to be checked. A commonly accepted and widely used procedure is to align the sample image to the reference image such that a simple pixel-to-pixel comparison is possible. If the print consists of several layers, this procedure of aligning images globally is only applicable as long as the alignments/displacements of the layers relatively to each other are constant or almost constant. This assumption is mostly true for printing processes in which all layers are printed in one

printing machine (e.g. rotary printing processes, sheet-feed printing processes where the carrier is appropriately fixed). There are indeed printing processes for which this assumption cannot be guaranteed. Consider a complex print that is applied in several entirely different process steps that cannot be managed by one single printing machine. If an exact alignment of the carrier to be printed cannot be preserved when the carrier is transferred from one process step to the next one, the relative positioning of print layers is not constant anymore, but subject to variations the magnitude of which is determined by maximum displacement errors that may occur between two process steps.

For processes in which the displacements of print layers are varying, a robust and sensitive pixel-to-pixel comparison is only possible if the individual layers can be aligned and checked independently. This requires full knowledge about which part of the print image belongs to which layer. This knowledge may come from explicit pre-press information. If such information is not available, however, the separation of print layers has to be done manually in a tedious and time-consuming process.

This paper investigates an alternative to manual print

[☆]The research presented in this paper was carried out while the first three authors were affiliated with Software Competence Center Hagenberg GmbH, A-4232 Hagenberg, Austria. Therefore, support by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the *K_{plus}* Competence Center Program is gratefully acknowledged.

*Corresponding author. Tel. +43 732 2468 9552.

Email addresses: daniel.soukup@arcs.ac.at (Daniel Soukup), bodenhofer@bioinf.jku.at (Ulrich Bodenhofer), Markus.Mittendorfer-Holzer@comneon.com (Markus Mittendorfer-Holzer), konrad.mayer@arcs.ac.at (Konrad Mayer)

Preprint submitted to Image and Vision Computing

layer separation. We attempt to utilize the displacements that can be observed from a set of sample images in order to identify the print layers in a semi-automatic manner. Note that we do not aim at separating all colors (even if printed in one process), but only at separating print areas that have been applied in completely separate process steps and the displacements of which are subject to changes. The basic idea is that we try to identify regions for which we can observe that they tend to move into the same direction. For demonstrating the method and for validating the results, we consider an important real-world application as a case study, namely banknote print inspection.

Section 2 gives a detailed overview of the sample data from banknote print inspection, while Section 3 highlights the final goal of print layer separation and its special requirements in the light of the banknote print inspection application. It is worth to mention, however, that the method presented in the following is in no way specific to this particular application, but applicable to any printing process in which the displacements of print layers are varying—only with the additional requirement that a sufficient quantity of sample data needs to be available. In Section 4, we give an overview of related topics and motivate the need for a local correlation-inspired approach as presented in this paper. Section 5 describes this method in detail along with examples and interpretations. Section 6 finally presents some hints how the method is integrated into the real inspection system.

2. Sample Data and Setup

This study is mainly motivated by the special requirements of the banknote print inspection process [1, 2]. Modern banknotes (e.g. Euro bills) are produced in a complex process that consists of several distinct steps. Beside different kinds of prints (intaglio print and offset print), different security features have to be placed on the bills. These are entirely different steps with varying displacements occurring between the different layers. Therefore, we concentrate on this application, although, as noted above, the considerations of this paper can be transferred to any other printing process in which varying displacements between print layers occur.

In the banknote inspection process, images are taken by high-performance line scanning color cameras with 1024 pixels resolution at speeds of several meters per second (similar to the competitor’s system described in [1]). The raw images have a resolution of approximately 300 ppi (pixels per inch) and are down-sampled to around 100 ppi for noise reduction.



Figure 1: Blue channel of a 10 Euro banknote sample image

We consider sample images of 10 Euro bills as a case study. We were given a set of 95 color images (RGB model) of 10 Euro bills processed as mentioned above. A higher resolution could have been achieved by recording images off-line with a surface camera or a flatbed scanner. In order not to introduce additional transformations, however, we used exactly the same imaging technology that is employed in the inspection process.

We only consider the blue channel for our studies. This simplification does not pose a severe limitation, as 10 Euro bills are mostly red and orange, therefore, the blue channel shows good contrasts. Note that the methods presented in this paper can easily be transferred to other color channels or models anyway. Figure 1 shows one such input image as we consider it in the following.

The coordinate system of the sample images we consider is normalized with respect to the border of the paper, i.e. the positions of the borders of the banknotes are almost constant in these images. Correspondingly, the positions of the print layers are varying from image to image.

To illustrate the displacements of print layers, Figure 2 shows an overlay of two images, where one image has been kept in its original position. The second image has been transformed such that its intaglio print is aligned to the intaglio print of the first image.¹ Observe that the intaglio print consists of three main blocks: (1) the big red Romanesque gate (plus the vertical stripe and the large “10” attached to it); (2) the ECB copyright notice in the upper middle of the banknote; (3) the “10 EURO” label in the lower left corner. It is clearly visible that the other print areas and security features are significantly shifted to each other.

¹This was done by choosing three linearly independent tie points and transforming the second image according to the unique solution of the six-parameter 2D pose estimation problem that takes translation, rotation, scaling, and skew into account [3, 4, 5, 6].



Figure 2: Overlay of two sample images illustrating displacements between the intaglio print and other layers

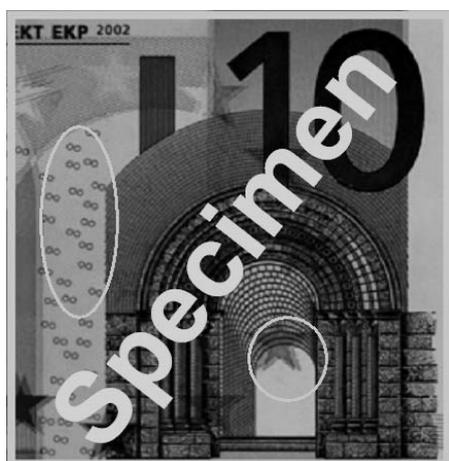


Figure 3: Closeup of displacements between intaglio print and the offset print

The frame in Figure 2 marks a representative 540×600 pixel clipping that contains the two most important print layers—intaglio print and offset print. We restrict to this clipping in the following. In this part of the image, all print details not belonging to the intaglio print belong to the offset print layer that consists of the following main elements: (1) the stars; (2) the orange circles to the left of the gate; (3) the hatching around and overlapping with the gate. Figure 3 shows a closeup of the displacements shown in Figure 2 with the displacements of orange circles and stars highlighted.

3. The Task of Print Layer Separation in Detail

Our ultimate goal is the following: *to establish a method that is able to identify potentially displaced print layers from a sequence of sample images. The result should be a pixel-by-pixel labeling of the image*

that assigns each pixel to one or several² print layers. Furthermore, we require that the method needs to be generic in the sense that it is not restricted to a particular kind of data or application and that it does not rely on additional explicit information (e.g. from the pre-press step). Let us assume that each print layer does not consist of dispersed single pixels, but of a set of spacious, connected, sufficiently large regions. Finally, we make the assumption that the method only needs to separate regions for which a significant displacement can be observed in the sample data—otherwise we may assume that the two regions belong to one print layer and can be aligned and inspected together.

The practical conditions and requirements imposed above are fulfilled by the sample data from banknote print inspection. Before we attempt to solve the problem, let us discuss an issue that is fundamentally important for a successful solution of the problem—distortions. We have observed that one cannot rely on the fact that a perfect alignment between the regions belonging to one print layer is possible. The left image in Figure 4 shows the difference of two images that have been aligned with respect to three tie points in the intaglio print area as in Figures 2 and 3. Note that a lot of artifacts are visible in textured areas such as the Romanesque gate, although the alignment has been done in sub-pixel accuracy. The middle image in Figure 4 shows an analogous difference image, but one image has been moved by 0.8 pixels to the right and 0.2 pixels up. Obviously, the alignment in the lower left part of the gate (as highlighted) is improved while the difference in other areas of the intaglio prints are increased. In the right image in Figure 4, the alignment has been chosen such that the differences in the highlighted area of the offset print area are minimized. One can observe that the hatching in the upper part of the images shows high differences, i.e. it is badly aligned. Even worse, there seems to be no fundamental difference in the magnitude of the alignment discrepancies in the hatching around the big “10” if we compare the left picture in which the two images are aligned with respect to the intaglio print and the right picture in which the two images are aligned with respect to the offset part (a part of which the hatching is).

We may draw the following conclusion from the examples above: even if the alignment is done with respect to tie points that fully belong to one print layer and even if this is done with sub-pixel accuracy, a perfect global

²As print layers may overlap, which is the case for the 10 Euro banknote as well, the assignment of a pixel to at most one print layer appears inappropriate.

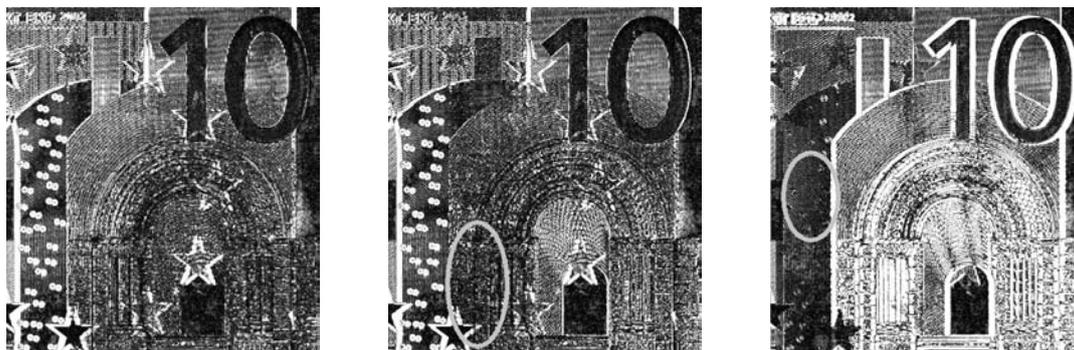


Figure 4: Three difference images demonstrating distortions

alignment of this print layer cannot be found. If one seeks for an optimal alignment in one particular region, worse alignments have to be expected in dislocated areas of the same print layer. The method we are finally aiming at, therefore, may not rely on the existence of perfect alignments, but needs to be prepared for distortions. In the banknote example, the lateral distortions are in the range of around one pixel or less, which is still manageable for a pixel-by-pixel comparison if we take certain measures. As demonstrated above, however, this magnitude is enough to abandon any approach that tries to guess print layers from discrepancies between images that are aligned with respect to one print layer—no matter whether the necessary tie points are set manually or determined by some kind of more advanced methodology.

4. Related Work and First Steps in Motion Detection

Our first step towards a better understanding of the problem was to put the sample images together into a kind of movie. Watching this movie, it is relatively easy for a human to determine the print layers as they can be perceived as areas that always “move” together. Thus, it was near at hand to have a careful look at methods available in the field of motion detection.

Motion detection is concerned with identifying moving objects in a sequence of images. A large set of methods is available for this task. We will investigate the most prominent approaches in closer detail in the following.

4.1. Feature-based matching

Feature-based matching is a standard methodology in computer vision. It consists of two basic steps: (1) feature extraction, that is, the isolation of relevant features

from the image (e.g. edge pixels, corner points, etc.); (2) matching, that is, the determination of correspondences between features from one image and features from another.

In motion detection, most methods are based on tracking feature points, i.e. finding paths of point correspondences in consecutive images. The methods available in motion detection are mostly based on defining a certain motion model and using an optimization technique to maximize (minimize) a gain (cost) function based on that motion model [7, 8, 9, 10, 11, 12, 13, 14]. No meaningful motion model is available for the print layer separation task, because the displacements from one frame to the other are absolutely random, as the positioning of the print layers does not obey any model. Moreover, we have no influence on the order of images, and the method we are aiming at should be independent of the order of images. Therefore, no approach that assumes a certain model-like behavior of the motion/print layer displacements is feasible.

Anyway, point pattern matching algorithms are available that are able to find correspondences in pairs of images without assuming any motion model (e.g. [3, 15]). Therefore, we made experiments with different feature point extraction algorithms [3, 16, 17, 18, 19] and tried to find correspondences with a point pattern matching algorithm similar to the one presented in [15]. Doing so, we faced the following major problems: the point pattern matching algorithm we tried, like all similar algorithms, is not able to identify two or more sets of independent correspondences. We are quite convinced that this task could have been solved by an appropriate modification of a method from feature-based motion detection (e.g. [9]), but even if we had managed to solve this task, it would still not be the full solution, as our goal is not to identify sets of features, but to identify

spacious regions. The more or less obvious idea would be to compute the corresponding displacement vectors for each considered pair of images and evaluate for each pixel for how many pair comparisons of aligned images the alignment is sufficiently good. As noted in Section 3 (and illustrated by the examples in Figure 4), there is no sufficiently reliable way to get an estimate for the quality of local fit. This is the reason why we finally abandoned the feature-based matching approach.

4.2. Block Matching and Optical Flow

Block matching approaches [5, 20, 21, 22] and optical flow approaches [23, 24] are two completely different classes of motion estimation methods. What they have in common, yet, is the fact that both use image data more or less directly instead of extracting specific features first.

Given a pair of images, the basic idea of *block matching* is to find best-fitting displacement vectors for a block around each pixel: For each pixel, a certain block relative to the pixel's coordinate is considered (e.g. a square symmetrically around the pixel under consideration). Then that displacement vector is determined for which the fit between the shifted block from the first image and the non-shifted block from the second image is best. A simple variant could be the minimization of absolute or Euclidean distances. A commonly used, more flexible method is the maximization of normalized cross correlation [5]. The final result for each pair of images is a displacement vector for each pixel or block of pixels.

We have tried to apply block matching to the print layer separation task. In the first step, we took pairs of images and computed the displacement vectors for all pixels. The naive assumption was that, for a given pair of images, each print layer would be characterized by a common displacement vector and that consequently the print layer separation task could be solved by segmentation of displacements. Beside the computational effort caused by computing normalized cross correlation for all blocks several times, we encountered two major difficulties:

1. The block matching approach works locally and can only handle shifts. If the images are rotated or skewed, there is no optimal displacement vector that is valid for the whole print layer. Moreover, we figured out that the block matching approach is sensitive to the distortions discussed above.
2. There are cases where no valid optimal displacement vector can be found, in particular along

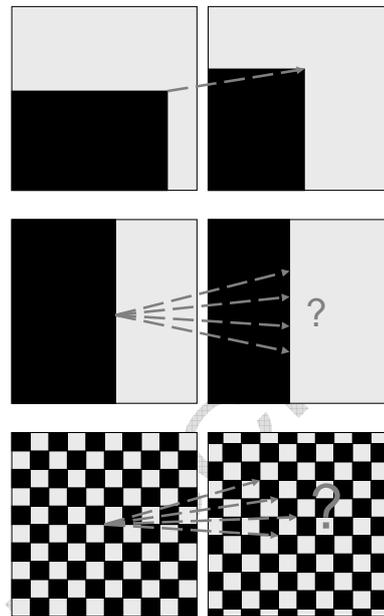


Figure 5: A case in which a perfect block alignment can be found (top); no unique optimum, however, can be found along straight edges (middle) or in isotropic textures (bottom)

straight edges or in textures like hatchings. Figure 5 demonstrates this difficulty. Note that Jähne refers to this issue as the *aperture problem* [5].

One approach to handle the aperture problem, as the name already suggests, is to change the window size. We investigated several different window sizes and even adaptive selection of window sizes, but did not find a satisfactory solution. Therefore we finally abandoned the block matching approach. More details can be found in [25] (available from the corresponding author freely on request).

The optical flow is the apparent motion at the image plane based on visual perception and has the dimension of a velocity [5]. If we consider a gray value image and consider the gray values as a differentiable function g over the continuous image plane, then the correspondence between the changes of gray values over time and the optical flow—let us denote it with $\mathbf{f} = (f_x, f_y)$ —is given by the *brightness change constraint equation (BCCE)* [5, 24]:

$$\frac{\partial g}{\partial t} + \mathbf{f} \nabla g = 0 \quad (1)$$

It is well-known that the optical flow cannot be determined from pairs of images in a stable way. The first term $\frac{\partial g}{\partial t}$ is a temporal derivative which assumes the existence of a temporarily continuous motion, i.e. that the

sample images are only snapshots of such a temporarily continuous motion. This is not the case in the print layer separation task. Davis, Karu and Freeman have proved that block matching and optical flow methods, although these approaches are based on completely different ideas, generate numerically equivalent estimates for sub-pixel displacement [26]. This fact can also be regarded as a well-founded explanation for the difficulties we encountered with block matching.³ Moreover, it tells us that no solution can be expected from optical flow-based approaches either.

5. The Difference Descent Method

In this section, we describe the approach that has finally led to a satisfactory solution of the print layer separation task. Let us start from the point where the block matching approach failed. We have seen that block matching provides us with very detailed, but unreliable, information about the displacements. Our approach is still inspired by block matching, but follows a slightly different strategy: we consider a certain number of pair comparisons. In each comparison, we produce coarser, but more reliable, information. Finally we try to get maximum information out of all pair comparisons. As we will see later, the latter step is not fully automatic; instead it requires some user interaction.

5.1. Pair Comparisons

Given a pair of images, block matching provides us with the best-matching displacement vector for each pixel. That is computationally expensive and unreliable due to the sensitivity to distortions and the aperture problem. What we are interested in is to identify image regions that have a common shift behavior. As we have enough data to aggregate several pair comparisons, it is sufficient to reduce the information block matching gives us to a reliable essence. This reliable essence is not the exact displacement vector as in traditional block matching, but only the expected direction of displacement.

We consider a relatively small neighborhood of 3×3 or 5×5 pixels symmetrically around each pixel under consideration. Assume that $N(x, y)$ is the set of pixel coordinates in the neighborhood of position (x, y) . The gray value at position (x, y) in the first image is denoted with $G_1(x, y)$ and the gray value at position (x, y) in the

second image is denoted with $G_2(x, y)$. It is clear that, given a position (x, y) , the value

$$\sum_{(x', y') \in N(x, y)} |G_1(x', y') - G_2(x', y')| \quad (2)$$

is nothing else but the sum of absolute differences of gray values in image no. 1 and no. 2 in the neighborhood $N(x, y)$. Correspondingly, given a shift vector (Δ_x, Δ_y) ,

$$\begin{aligned} D(x, y, \Delta_x, \Delta_y) \\ = \sum_{(x', y') \in N(x, y)} |G_1(x', y') - G_2(x' - \Delta_x, y' - \Delta_y)| \end{aligned} \quad (3)$$

is the sum of absolute differences of gray values in the neighborhood $N(x, y)$ in image no. 1 and gray values in the same neighborhood if image no. 2 is shifted by $(-\Delta_x, -\Delta_y)$. The value $D(x, y, \Delta_x, \Delta_y)$, therefore, is a measure how well the two images match in the neighborhood of (x, y) if we shift image no. 1 by (Δ_x, Δ_y) . The smaller $D(x, y, \Delta_x, \Delta_y)$ is, the better the two neighborhoods match.

We finally compute for which displacement the value $D(x, y, \Delta_x, \Delta_y)$ is minimal. However, we do not consider a wide range of possible displacements, but only offsets of -1, 0, and 1 in each direction. So the result is a displacement that is computed as

$$\Delta(x, y) = \operatorname{argmin}_{\Delta_x = -1, 0, 1; \Delta_y = -1, 0, 1} D(x, y, \Delta_x, \Delta_y). \quad (4)$$

This displacement direction is computed for every pixel (x, y) .

So far, this idea pretty much coincides with the block matching approach, except in three fundamental aspects:

1. We consider sums of absolute differences instead of normalized cross correlation. The reason for this is to save computational effort. As images for industrial quality control must be taken under stable illumination conditions (otherwise pixel-to-pixel comparisons would not be possible), this simplification is not a severe limitation.
2. As the magnitude of displacements is beyond the magnitude of one pixel, we do not obtain the best-fitting displacement vector, but only an estimate in which direction the best fit is expected to be. That is the reason why we call this method *difference descent method*—because we compute a direction in which the sum of absolute difference is decreasing most. The result for each pixel can only take nine different values. So this is rather coarse information. This coarseness, however, guarantees

³We did not use sub-pixel accuracy, but the result by Davis, Karu and Freeman suggests that even sub-pixel accuracy would not have led to a satisfactory solution.

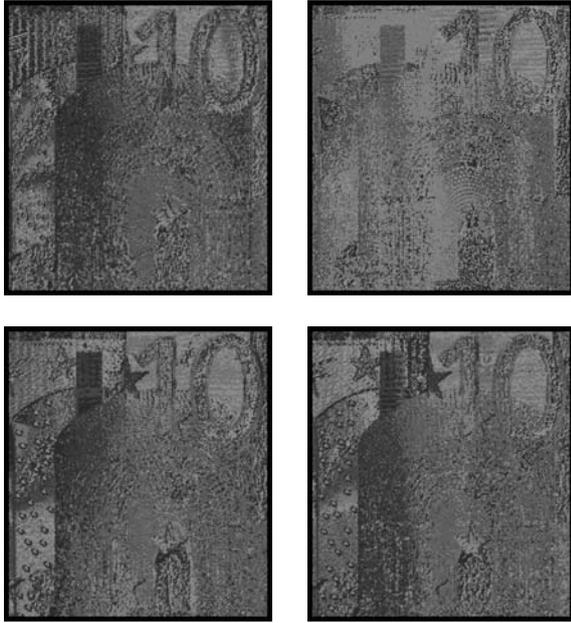


Figure 6: Four examples of pair comparisons with a 3×3 neighborhood (displacement vectors coded as gray values)

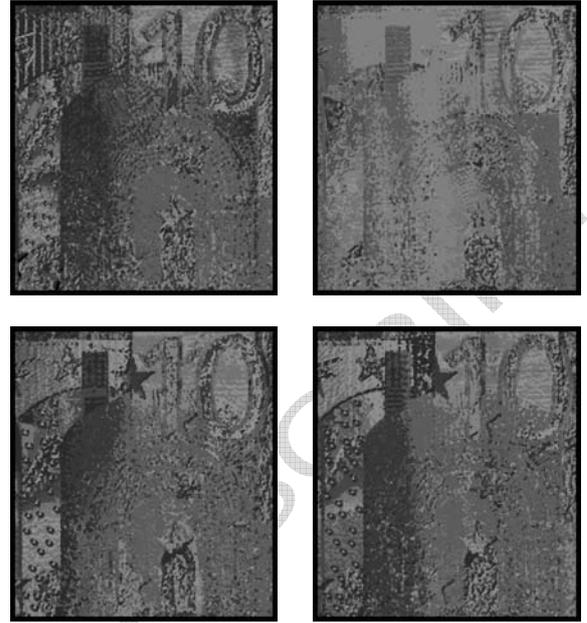


Figure 7: Four examples of pair comparisons with a 5×5 neighborhood (displacement vectors coded as gray values)

that the result is robust with respect to distortions, small rotations and skew. As the prints we consider here are aligned according to the paper border, only small rotation and minimal skewness may occur.

3. We are considering much smaller block sizes/neighborhoods (3×3 or 5×5) than one would typically use in traditional block matching. This saves computational effort on the one hand and, on the other hand, ensures that good matches can be found even for small details.

Figures 6 and 7 show some examples of such pair comparisons. The nine possible displacements are coded as gray values. It is clearly visible that different pair comparisons produce completely different images. This is clear, because the displacements are different for each pair of images. What is more interesting is the fact that the major details of the prints are visible, although the images have a rather speckled structure. It is also worth to note that a neighborhood size of 3×3 produces finer structures (see Figure 6), while a neighborhood size of 5×5 leads to slightly larger segments (see Figure 7).

5.2. Exploitation of Pair Comparisons

The question arises how we can exploit the information contained in the displacement vectors determined by a set of several pair comparisons.

It is worth to note, yet a trivial fact, that each pair comparison induces an equivalence relation on the pixel set if we consider two pixels as equivalent if they have the same displacement vector. This means that, if we fix two pixel coordinates (\bar{x}, \bar{y}) and (x, y) , we can easily extract those pixels for which we have determined the same displacement vector. Now assume that we have N such pair comparisons in total and denote the displacement vector identified for positions (\bar{x}, \bar{y}) and (x, y) in the i -th pair comparison with $\Delta_i(\bar{x}, \bar{y})$ and $\Delta_i(x, y)$, respectively (computed as in Eq. (4); for each $i \in \{1, \dots, N\}$). Then the relative frequency of pair comparisons in which the displacement vectors $\Delta_i(\bar{x}, \bar{y})$ and $\Delta_i(x, y)$ are equal is defined as

$$C((\bar{x}, \bar{y}), (x, y)) = \frac{1}{N} \cdot |\{i = 1, \dots, N \mid \Delta_i(\bar{x}, \bar{y}) = \Delta_i(x, y)\}|. \quad (5)$$

Obviously, C maps pairs of pixel coordinates to values from the unit interval $[0, 1]$. A value $C((\bar{x}, \bar{y}), (x, y)) = 1$ obviously means that the displacement vectors associated with positions (\bar{x}, \bar{y}) and (x, y) have been equal for all N pair comparisons, whereas a value of $C((\bar{x}, \bar{y}), (x, y)) = 0$ means that $\Delta_i(\bar{x}, \bar{y}) \neq \Delta_i(x, y)$ for all $i = 1, \dots, N$. If $\Delta_i(\bar{x}, \bar{y})$ and $\Delta_i(x, y)$ were uniformly distributed random values, the expected value of $C((\bar{x}, \bar{y}), (x, y))$ would be $\frac{1}{9}$. That means, if

$C((\bar{x}, \bar{y}), (x, y))$ is significantly larger than $\frac{1}{9}$, there is statistical evidence that positions (\bar{x}, \bar{y}) and (x, y) belong to the same print layer.

We may also view the value C as the arithmetic average of the equivalence relations associated with the N pair comparisons. More specifically, if we denote the i -th equivalence relation with

$$E_i((\bar{x}, \bar{y}), (x, y)) = \begin{cases} 1 & \text{if } \Delta_i(\bar{x}, \bar{y}) = \Delta_i(x, y), \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

then the equality

$$C((\bar{x}, \bar{y}), (x, y)) = \frac{1}{N} \cdot \sum_{i=1}^N E_i((\bar{x}, \bar{y}), (x, y)) \quad (7)$$

holds trivially. It can be proved that the arithmetic average of classical equivalence relations is a fuzzy equivalence relation with respect to the Łukasiewicz t-norm $T_L(a, b) = \max(a + b - 1, 0)$ [27, 28, 29]. This means that for an arbitrary triple of pixel coordinates $((x, y), (x', y'), (x'', y''))$, the following holds:

1. $C((x, y), (x, y)) = 1$ (reflexivity)
2. $C((x, y), (x', y')) = C((x', y'), (x, y))$ (symmetry)
3. $T_L(C((x, y), (x', y')), C((x', y'), (x'', y''))) \leq C((x, y), (x'', y''))$ (T_L -transitivity)

Given a certain pixel coordinate (\bar{x}, \bar{y}) , it is theoretically justified to interpret the value $C((\bar{x}, \bar{y}), (x, y))$ as the degree to which pixel coordinate (x, y) belongs to the equivalence class of the pixel coordinate (\bar{x}, \bar{y}) (with respect to the fuzzy relation C) [30, 31].

Figure 8 shows two examples where $C((\bar{x}, \bar{y}), (x, y))$ is displayed for all (x, y) with respect to a given coordinate (\bar{x}, \bar{y}) , where this particular coordinate is marked with a white spot. In the left image, a point in the Romanesque gate (intaglio print) is chosen and, in the right image, a point in the hatching (offset print) is taken. It is easy to see that large parts of the intaglio print are already covered in the left image. In the right image, we see that there is high evidence that pixels around the reference point belong to the same print layer, but the evidence that other areas belong to the same print layer is rather low (yet still visible).

Now assume that we have a selection of K pixel coordinates (\bar{x}_j, \bar{y}_j) ($j = 1, \dots, K$) for which we know that they can all be attributed to the same print layer. Let us call these points *reference points* in the following. Then we can interpret the value

$$A(x, y) = \max_{j=1, \dots, K} C((\bar{x}_j, \bar{y}_j), (x, y)) \quad (8)$$

as the degree to which pixel (x, y) belongs to the union of equivalence classes of reference points (\bar{x}_j, \bar{y}_j) . This means that, by aggregating equivalence classes of a certain number of reference points, we can put together different parts of the print layer. The maximum is a standard disjunction in fuzzy logic. In this context, we can vaguely interpret it as follows: a point belongs to the print layer under consideration if it is contained in at least one of the equivalence classes induced by the reference points. Figure 9 shows a closeup that demonstrates the result of using two reference points in the intaglio print area and two in the offset print area.

What we have sketched here is the fundamental idea of the aggregation procedure: given a set of N pair comparisons, a human expert determines a set of reference points for each print layer and the algorithm determines the degrees to which pixels belong to the print layer associated with the respective print layer.

5.3. Summary

Let us now summarize Subsections 5.1 and 5.2 and formulate the overall procedure.

Algorithm 1.

Input: set of input images G_1, \dots, G_M with equal sizes

(I) Pair comparisons:

- Fix one reference image G_j (by an expert's choice or randomly) and choose block size
- According to Subsection 5.1, compute $N = M - 1$ pair comparisons of G_j with images G_i ($i \neq j$)

(II) Selection of reference points:

- An expert determines the number of print layers R and chooses appropriate reference points for each print layer

(III) Computation of evidence degrees:

- For all print layers $k = 1, \dots, R$ and all pixels (x, y) , compute the degree of evidence that pixel (x, y) belongs to the k -th print layer; taking the reference points associated with the k -th layer into account, this is accomplished by Eq. (8)

Output: for each print layer, an "image" of evidence degrees to which pixels belong to it

A flowchart-like visualization of Algorithm 1 is depicted in Figure 10.

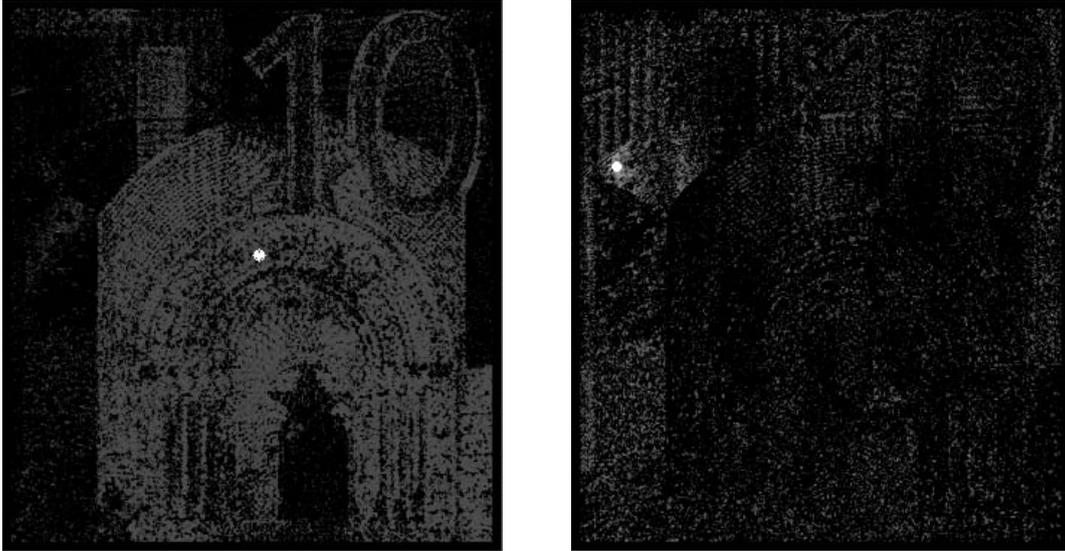


Figure 8: Two examples with one reference point in the intaglio print area (left) and the offset print area (right)



Figure 9: Closeup of experiment with two reference points. The left image shows two reference points in the intaglio print area (white) and two reference points in the offset print area (black). The two other images visualize the degree to which a pixel belongs to the intaglio print area (middle) and the offset print area (right)

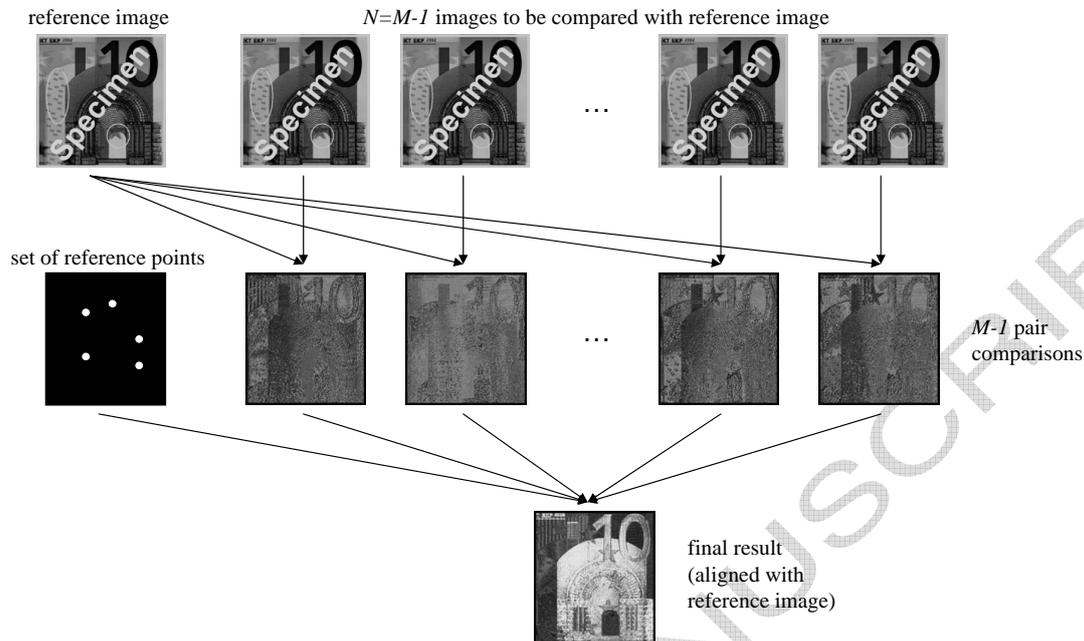


Figure 10: A flowchart-like visualization of Algorithm 1

That we start each pair comparison from the same reference image ensures that the coordinate systems of the pair comparisons are aligned to each other. Consequently, the resulting print layer separation is aligned to the reference image.

Let us remark that Algorithm 1 is not a one-way street. Step (I) is computationally more expensive (e.g. for 94 pair comparisons, it takes a few minutes on state-of-the-art PC hardware), but requires no user interaction; therefore it can be done in an off-line process. Steps (II) and (III), however, can be done interactively. We have implemented a graphical user interface in which the human expert can set reference points and within fractions of a second he/she obtains the result. In this way, reference points can be added and removed interactively and the human expert is able to determine an appropriate selection of reference points and the resulting degrees of evidence in a quick and efficient way.

Further note that we treat the print layers completely independently of each other. In particular the algorithm does not have to make a unique choice to which print layer a pixel belongs. Instead it computes degrees of evidence to which pixels belong to print layers. Therefore the algorithm presented here is able to handle overlaps of print layers (at least as long as one print layer does not entirely occlude the other. Figure 11 shows an example, where we have chosen 7 reference points in

the intaglio print area and 7 reference points in the offset print area. It is remarkable how well the overlaps are discovered (stars and hatching that are offset-printed behind the Romanesque gate that is intaglio-printed). The only eyesore in this example is that the bar to the left of the large “10” on top of the Romanesque gate is not satisfactorily assigned to the intaglio print area (although it is definitely intaglio-printed). This problem can easily be solved by a larger number of reference points, where also one or a few are chosen in the bar. Figure 12 shows another example, this time with a block size of 5×5 , where we have chosen 24 reference points in the intaglio print area and 34 reference points in the offset print area.

Finally, let us remark that the gray values in images in Figures 8, 9, 11 and 12 have been contrast-enhanced for easier visibility in black and white reproduction. The original result images actually obtained from the algorithm presented in this paper are darker and have less contrast.

5.4. Discussion

The encouraging results obtained by the algorithm presented in this paper (see Figures 11 and 12 in particular) are remarkable in the light of the fact that the basis is only very coarse information—the difference descent directions obtained from the pair comparisons.

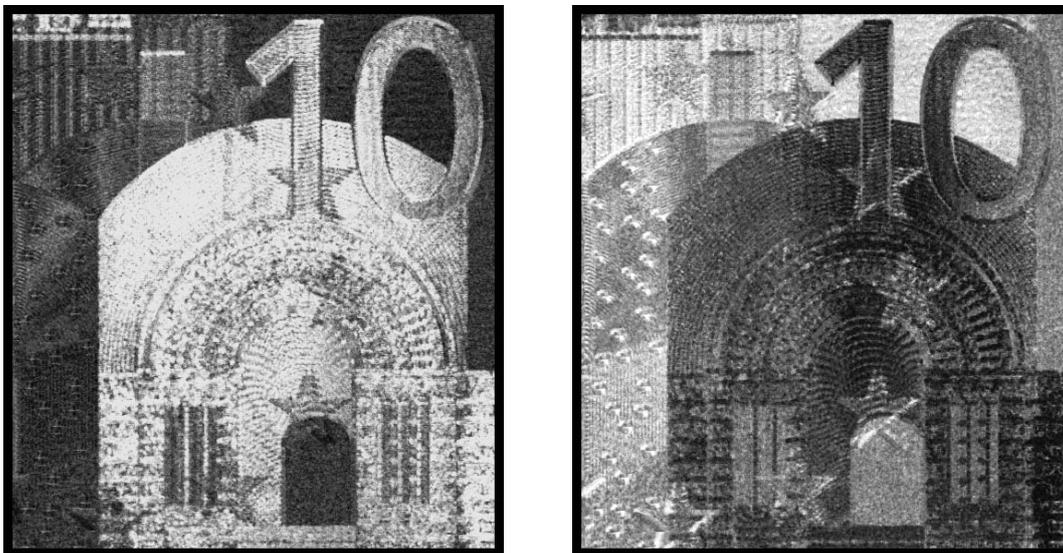


Figure 11: An example with 7 reference points in each print layer (block size 3×3); resulting evidence degrees for intaglio print area (left) and offset print area (right)

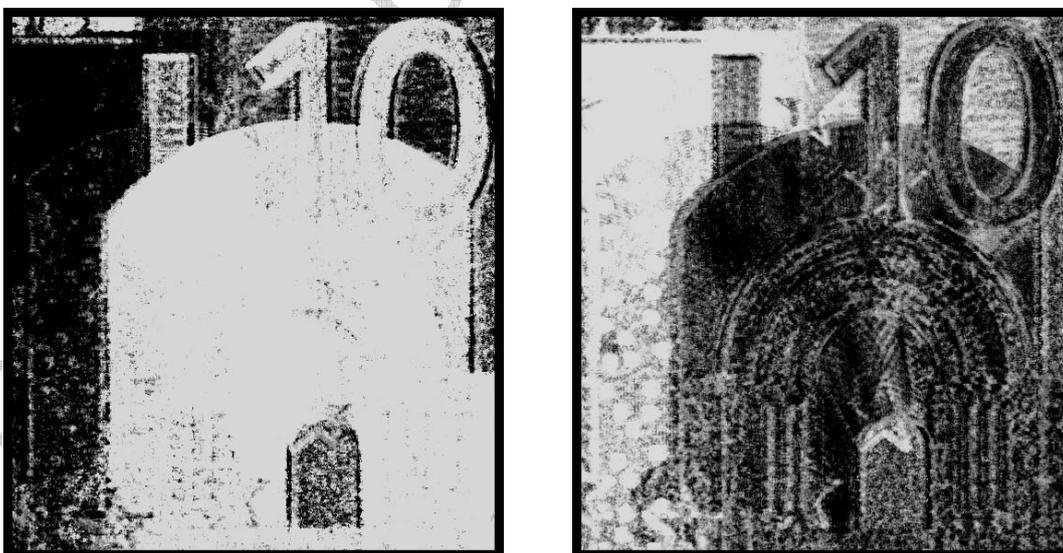


Figure 12: An example with 34 reference points in the offset print area and 24 reference points in the intaglio print area (block size 5×5); resulting evidence degrees for intaglio print area (left) and offset print area (right)

If the actual best-fitting displacement is larger than the size of one pixel, we cannot guarantee that the difference descent direction actually points to the right direction where the best-fitting displacement actually can be found. We have studied this phenomenon in detail and figured out that the displacements are normally distributed around a hypothetical average zero displacement with a standard deviation of slightly more than one pixel. This means that a significant proportion of best-fitting displacements is actually in the range of one pixel. Secondly, if the magnitude of the best-fitting displacement is larger than one pixel, the difference descent direction is still not a uniformly distributed random variable. In significantly more than $\frac{1}{9}$ of the observed cases, the difference descent direction actually pointed into the direction of the best-fitting displacement vector. As noted above, any value of evidence that significantly exceeds $\frac{1}{9}$ can be counted as a statistical evidence that two pixels belong to the same print layer.

An alternative solution of the same problem has recently been published in [32]. This approach is based on region matching using so-called *maximally stable extremal regions*. It yields well-defined contiguous image segments, but is not able to handle overlaps of print areas.

6. Integration Into the Print Inspection Procedure

We have pointed out that the algorithm is only semi-automatic and needs user interaction. Moreover, the result consists in degrees of evidence to which pixels belong to print layers (of course these degrees can be considered as images). This means that the algorithm as presented here is not fully ready for a real industrial process. In this section, we deal with the question how the proposed procedure can be integrated beneficially in an industrial inspection process.

The goal of the proposed algorithm is to support the operator of an industrial print inspection process in the setup phase of the print inspection system. Currently the operator has to determine the print layers with a free-hand drawing tool—a process that is tedious and time-consuming. From this point of view, the determination of some reference points in an interactive graphical user interface still means a significant improvement for the operator, at least as long as he/she has preliminary knowledge about the print layers (which is the case for banknote print inspection). So the need for user interaction is not a severe restriction.

The resulting images (as shown in Figures 11 and 12) are nothing else but degrees of evidence that cannot be used as print layer masks immediately. At least

the banknote print inspection system is not capable of dealing with degrees of membership, but requires a binary assignment of pixels to print layers (yet overlaps are possible). Thus it is necessary to threshold the degrees/images and to apply morphological operators [33, 34, 35] to eliminate small blobs and to close small holes. For confidentiality reasons, details cannot be made public.

7. Conclusion and Outlook

In this paper, we have presented a semiautomatic procedure for identifying print layers from a sequence of sample images. All examples were from banknote print inspection, but the method is applicable to any printing process in which random displacements of two or more print layers occur. The method is practically acceptable, although a fully automatic method that does not require user interaction would be more desirable.

The considerations presented in Subsection 5.2 give rise to an idea how the manual selection of reference points can be automated. We have noted that the values $C((\bar{x}, \bar{y}), (x, y))$, which can be considered as the degrees to which two pixels are belonging to the same print layer, define a fuzzy equivalence relation with respect to the Łukasiewicz t-norm. It is well-known that there is a one-to-one correspondence between fuzzy equivalence relations with respect to continuous Archimedean t-norms and pseudo-metrics [27, 30, 36, 37] (where the Łukasiewicz t-norm is a prominent representative of this class). Thus, agglomerative clustering methods, which suffice with distances only [38, 39, 40], can be used to group a pre-selection of points (e.g. selected by some interest operator) into sub-groups of reference points that are then likely to belong to the same print layer. To investigate this idea is left to future work.

References

- [1] G. Coldani, L. Cotrino, G. Danese, F. Leporati, M. Maneri, Notacheck: a parallel DSP-based architecture for real time high resolution inspection of bank-notes, in: Proc. 5th IEEE Int. Conf. on Computer Architectures for Machine Perception, Padova, 2000, pp. 163–169.
- [2] W. Hardin, Money in the bank: vision system inspects bank notes with color linescan camera, custom lens, and innovative lighting, Vision Systems Design 13 (1).
- [3] R. M. Haralick, L. G. Shapiro, Computer and Robot Vision, Vol. II, Addison-Wesley, Reading, MA, 1993.
- [4] W. Krattenthaler, K. J. Mayer, M. Zeiller, Point correlation: A reduced-cost template matching technique., in: Proc. 1st Int. Conf. on Image Processing, Austin, TX, 1994, pp. 208–212.
- [5] B. Jähne, Digital Image Processing: Concepts, Algorithms, and Scientific Applications, 4th Edition, Springer, Berlin, 1997.
- [6] B. Zitová, J. Flusser, Image registration methods: a survey, Image Vision Comput. 21 (11).

- [7] D. H. Ballard, C. M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, 1982.
- [8] S. T. Barnard, W. B. Thompson, Disparity analysis of images, *IEEE Trans. Pattern Anal. Mach. Intell.* 2 (4) (1980) 333–340.
- [9] R. Pless, T. Brodský, Y. Aloimonos, Detecting independent motion: The statistics of temporal continuity, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8) (2000) 768–773.
- [10] K. Rangarajan, M. Shah, Establishing motion correspondence, *Computer Vision, Graphics, and Image Processing. Image Understanding* 54 (1) (1991) 56–73.
- [11] V. Salari, I. K. Sethi, Feature point correspondence in the presence of occlusion, *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (1990) 87–91.
- [12] I. K. Sethi, R. Jain, Finding trajectories of feature points in a monocular image sequence, *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (1987) 56–73.
- [13] K. Shafique, M. Shah, A noniterative greedy algorithm for multiframe point correspondence, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (1) (2005) 51–65.
- [14] S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, MA, 1979.
- [15] S.-H. Chang, F.-H. Cheng, W.-H. Hsu, G.-Z. Wu, Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes, *Pattern Recognition* 30 (2) (1997) 311–320.
- [16] W. Förstner, A feature based correspondence algorithm for image matching, *Int. Arch. Photogramm. Remote Sensing* 26 (1986) 150–166.
- [17] W. Förstner, E. Gülch, A fast operator for detection and precise location of distinct points, corners, and centres of circular features, in: *Proc. Intercommission Conf. on Fast Processing of Photogrammetric Data, Interlaken, 1987*, pp. 281–305.
- [18] C. G. Harris, M. Stephens, A combined corner and edge detector, in: *Proc. 4th Alvey Vision Conference, Manchester, 1988*, pp. 189–192.
- [19] E. Sojka, A new approach to detecting the corners in digital images, in: *Proc. IEEE Int. Conf. on Image Processing, Vol. 2, Barcelona, 2003*, pp. 445–448.
- [20] H. Ghavari, M. Mills, Blockmatching motion estimation algorithms—new results, *IEEE Trans. Circuits Syst.* 37 (1990) 649–651.
- [21] T. Komarek, P. Pirsch, Array architectures for block matching algorithms, *IEEE Trans. Circuits Syst.* 36 (1989) 1301–1308.
- [22] Q. Tian, M. N. Huhns, Algorithms for subpixel registration, *Computer Vision, Graphics, and Image Processing* 35 (1986) 220–233.
- [23] J. K. Aggarwal, N. Nandhakumar, On the computation of motion from sequences of images—a review, *Proc. IEEE* 76 (8) (1988) 917–935.
- [24] K. P. Horn, B. G. Schunck, Determining optical flow, *Artificial Intelligence* 17 (1–3) (1981) 185–203.
- [25] D. Soukup, Separation of banknote print layers using a sequence of sample images, Master's thesis, Johannes Kepler Universität Linz (November 2004).
- [26] C. Q. Davis, Z. Z. Karu, D. M. Freeman, Equivalence of subpixel motion estimators based on optical flow and block matching, in: *Proc. IEEE Symp. on Computer Vision, Coral Gables, FL, 1995*, pp. 7–12.
- [27] D. Boixader, J. Jaças, J. Recasens, Fuzzy equivalence relations: advanced material, in: D. Dubois, H. Prade (Eds.), *Fundamentals of Fuzzy Sets, Vol. 7 of The Handbooks of Fuzzy Sets*, Kluwer Academic Publishers, Boston, 2000, pp. 261–290.
- [28] A. Pradera, E. Trillas, E. Castiñeira, On the aggregation of some classes of fuzzy relations, in: B. Bouchon-Meunier, J. Gutiérrez-Río, L. Magdalena, R. R. Yager (Eds.), *Technologies for Constructing Intelligent Systems 2: Tools, Vol. 90 of Studies in Fuzziness and Soft Computing*, Physica-Verlag, Heidelberg, 2002, pp. 125–136.
- [29] S. Saminger, R. Mesiar, U. Bodenhofer, Domination of aggregation operators and preservation of transitivity, *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems* 10 (Suppl.) (2002) 11–35.
- [30] U. Höhle, Fuzzy equalities and indistinguishability, in: *Proc. 1st European Congress on Fuzzy and Intelligent Technologies, Vol. 1, Aachen, 1993*, pp. 358–363.
- [31] S. Gottwald, *Fuzzy Sets and Fuzzy Logic*, Vieweg, Braunschweig, 1993.
- [32] R. Huber-Mörk, H. Ramoser, H. Penz, K. Mayer, D. Heiss-Czedik, A. Vrabl, Region based matching for print process identification, *Pattern Recognition Letters* 28 (2007) 2037–2045.
- [33] R. M. Haralick, L. G. Shapiro, *Computer and Robot Vision, Vol. I*, Addison-Wesley, Reading, MA, 1992.
- [34] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
- [35] J. Serra, P. Soille, *Mathematical Morphology and Its Applications to Image Processing*, Kluwer Academic Publishers, Dordrecht, 1994.
- [36] B. De Baets, R. Mesiar, Pseudo-metrics and T -equivalences, *J. Fuzzy Math.* 5 (2) (1997) 471–481.
- [37] L. Valverde, On the structure of F -indistinguishability operators, *Fuzzy Sets and Systems* 17 (3) (1985) 313–328.
- [38] M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, 1973.
- [39] R. S. Michalski, R. E. Stepp, Clustering, in: S. C. Shapiro (Ed.), *Encyclopedia of artificial intelligence*, John Wiley & Sons, Chichester, 1992, pp. 168–176.
- [40] J. H. Ward Jr., Hierarchical grouping to optimize an objective function, *J. Amer. Statist. Assoc.* 58 (1963) 236–244.