A Fuzzy Method for Pixel Classification and its Application to Print Inspection

Peter Bauer, Ulrich Bodenhofer, Erich Peter Klement
Fuzzy Logic Laboratorium Linz–Hagenberg
Institut für Mathematik, Johannes Kepler Universität
A-4040 Linz, Austria
E-mail: ip@flll.uni-linz.ac.at

Abstract

In this paper a fuzzy method for a certain kind of image pixel classification is introduced. It is the most important result of the development of an inspection system for a silk-screen printing process. The algorithm computes a fuzzy segmentation of a given image into four different types of areas which are to be checked by applying different criteria. Furthermore, it is discussed how this algorithm is integrated in the inspection system.

1 INTRODUCTION

The main goal of this project was to design an automatic inspection system which does not sort out every print with defects, but only those which have visible defects. It is intuitively clear that the visibility of a defect depends on the structure of the print in its neighborhood. While little spots can hardly be recognized in very chaotic areas, they are disturbing in rather homogeneous areas. So, the first step towards a sensitive inspection is to extract areas from the print which should be checked differently.

The following four types were specified by experts of our partner company. For certain reasons, which can be explained with the special principles of the silk-screen printing process, it is sufficient to consider only these types:

Homogeneous area: uniformly colored area;

Edge area: pixels within or close to visually significant edges;

Raster: area which looks rather homogeneous from a certain distance, but which is actually obtained by printing small raster dots of two or even more colors;

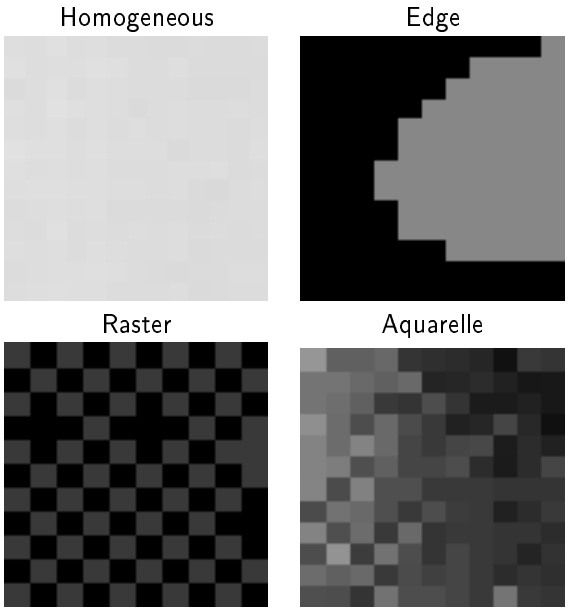

Figure 1: Magnifications of typical representatives of the four types

Aquarelle: rastered area with high chaotic deviations (e.g., small high-contrasted details in picture prints).

The magnifications in Figure 1 show how theses areas typically look like. Of course, transitions between two or more of these areas are possible, hence a fuzzy model is recommendable.

First of all, we should define precisely what an image is:

Definition 1 An $N \times M$ matrix of the form

$$\big((u_r(i,j), u_g(i,j), u_b(i,j))\big)_{i=1,\ldots,N}^{j=1,\ldots,M} \tag{1}$$

with three-dimensional entries

$$(u_r(i,j), u_g(i,j), u_b(i,j)) \in \{0,\ldots,255\}^3$$

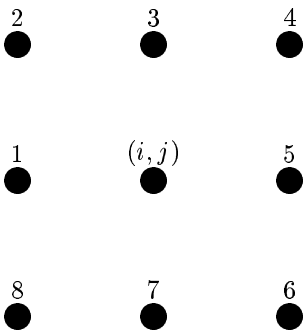

Figure 2: Enumeration of the neighborhood of a pixel

is called a 24-bit color image of size $N \times M$. A coordinate pair (i, j) is called a pixel; the values $(u_r(i,j), u_g(i,j), u_b(i,j))$ are called the gray values of pixel (i, j).

It is near at hand to use something like the variance or an other measure for deviations to distinguish between areas which show only low deviations, such as homogeneous areas and rasters, and areas with high deviations, such as edge areas or aquarelles.

On the contrary, it is intuitively clear that such a measure can never be used to separate edge areas from aquarelles, because any geometrical information is neglected. Experiments have shown that well-known standard edge detectors, such as the Laplacian and the Mexican Hat filter mask, cannot distinguish sufficiently if deviations are chaotic or anisotropic. Another possibility we also took into consideration was to use wavelet transforms (see [3] or [5]). Since the size of the image is approximately 1 Megabyte and the segmentation has to be done in at most three seconds, it is obvious that such highly advanced methods would require too much time. Finally, we found a fairly good alternative which is based on the discrepancy norm. This approach uses only, as filter masks like the Laplacian or the Mexican Hat also do, the closest neighborhood of a pixel. Figure 2 shows how the neighbors are enumerated for our algorithm. For an arbitrary but fixed pixel (i, j) we can define the enumeration mapping l with the following table:

k	$l(k)$
1	$(\ i\ , j-1)$
2	$(i-1, j-1)$
3	$(i-1,\ j\ )$
4	$(i-1, j+1)$
5	$(\ i\ , j+1)$
6	$(i+1, j+1)$
7	$(i+1,\ j\ )$
8	$(i+1, j+1)$

(2)

If we plot one color extraction with respect to this enumeration, i.e $(u_x(l(k))_{k\in\{1,\ldots,8\}}$, where $x \in \{r, g, b\}$, we

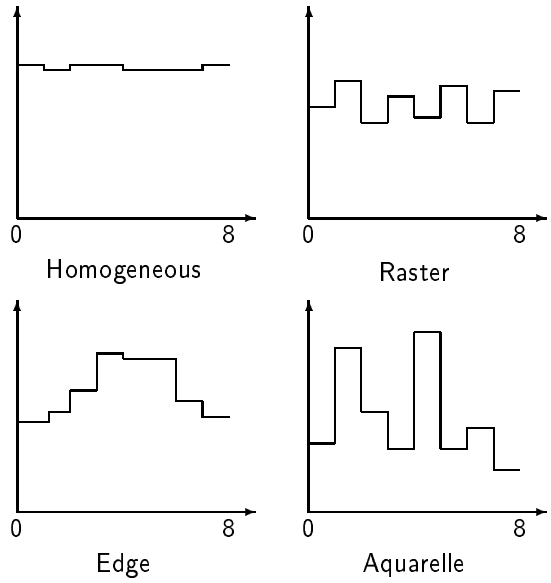

Figure 3: Typical gray value curves of the form $(u_x(l(k)))_{k\in\{1,\ldots,8\}}$

typically get curves like those ones shown in Figure 3. From these sketches it can be seen easily that a measure for the deviations can be used to distinguish between homogeneous areas, rasters, and the other two types. On the other hand, the most eyecatching difference between aquarelles and edge areas is that edge areas show long connected peaks while aquarelles typically show chaotic, mostly narrow peaks. So, a method which judges the shape of the peaks should be used in order to separate edge areas from aquarelles. A simple but effective method for this purpose is the so-called discrepancy norm.

2 THE DISCREPANCY NORM

Definition 2

$$\begin{aligned} \|.\|_D : \quad \mathbb{R}^n &\longrightarrow \mathbb{R}^+ \\ \vec{x} &\longmapsto \max_{1\le\alpha\le\beta\le n} \left| \sum_{i=\alpha}^{\beta} x_i \right| \end{aligned} \tag{3}$$

It can be proven easily that this mapping is a norm, a proof is provided, e.g., in [2].

In measure theory the discrepancy between two measures μ and ν on $\mathbb{R}$ is normally defined as

$$\mathcal{D}(\mu,\nu) := \sup_{a\le b} |\mu([a,b]) - \nu([a,b])| .$$

If we have two discrete measures $\overline{\mu}$ and $\overline{\nu}$ on the set $\{1,\ldots,n\}$, where $\overline{\mu}(i) =: x_i$ and $\overline{\nu}(i) =: y_i$, then $\mathcal{D}(\overline{\mu},\overline{\nu})$ and $\|\vec{x}-\vec{y}\|_D$ are equal (see [6] and [4]). Thus, it is reasonable to call $\|.\|_D$ discrepancy norm in $\mathbb{R}^n$.

Obviously, the computation of $\|.\|_D$ primitively using the definition requires $\mathcal{O}(n^2)$ operations. The following theorem allows us to compute $\|.\|_D$ with linear speed:

Theorem 3 *For all $\vec{x} \in \mathbb{R}^n$ the equation*

$$\|\vec{x}\|_D = \max_{1\le\beta\le n} X_\beta - \min_{1\le\alpha\le n} X_\alpha \qquad (4)$$

holds, where the values $X_j := \sum_{i=1}^{j} x_i$ denote the partial sums.

Proof: If we assign 0 to x_0 and x_{n+1} we can conclude

$$\begin{aligned}
\|\vec{x}\|_D &= \max_{1\le\alpha\le\beta\le n+1} \left|\sum_{i=\alpha}^{\beta} x_i\right| \\
&= \max_{1\le\beta\le n+1} \max_{1\le\alpha\le n+1} \left|\sum_{i=1}^{\beta} x_i - \sum_{i=1}^{\alpha-1} x_i\right| \\
&= \max_{1\le\beta\le n} \max_{1\le\alpha\le n} \left|\sum_{i=1}^{\beta} x_i - \sum_{i=1}^{\alpha} x_i\right| \\
&= \max_{1\le\beta\le n} \max_{1\le\alpha\le n} |X_\beta - X_\alpha| \\
&= \max_{1\le\beta\le n} X_\beta - \min_{1\le\alpha\le n} X_\alpha.
\end{aligned}$$

■

3 THE FUZZY SYSTEM

For each pixel (i,j) we consider the nearest eight neighbors enumerated as described in (2). Then we can use

$$\begin{aligned}
v(i,j) \;:=\; & \sum_{k=1}^{8} \left(u_r(l(k)) - \overline{r}\right)^2 \\
& + \sum_{k=1}^{8} \left(u_g(l(k)) - \overline{g}\right)^2 \qquad (5) \\
& + \sum_{k=1}^{8} \left(u_b(l(k)) - \overline{b}\right)^2
\end{aligned}$$

as a measure for the size of the deviations in the neighborhood of (i,j) and

$$\begin{aligned}
e(i,j) \;:=\; & \|\, u_r(l(.)) - (\overline{r},\ldots,\overline{r})\,\|_D \\
& + \|\, u_g(l(.)) - (\overline{g},\ldots,\overline{g})\,\|_D \qquad (6) \\
& + \|\, u_b(l(.)) - (\overline{b},\ldots,\overline{b})\,\|_D
\end{aligned}$$

as a measure whether the pixel is part of or lying adjacent to a visually significant edge, where $\overline{r}$, $\overline{g}$ and $\overline{b}$ denote the mean values

$$\begin{aligned}
&\overline{r} := \tfrac{1}{8} \sum_{k=1}^{8} u_r(l(k)),\ \overline{g} := \tfrac{1}{8} \sum_{k=1}^{8} u_g(l(k)), \\
&\overline{b} := \tfrac{1}{8} \sum_{k=1}^{8} u_b(l(k)).
\end{aligned}$$

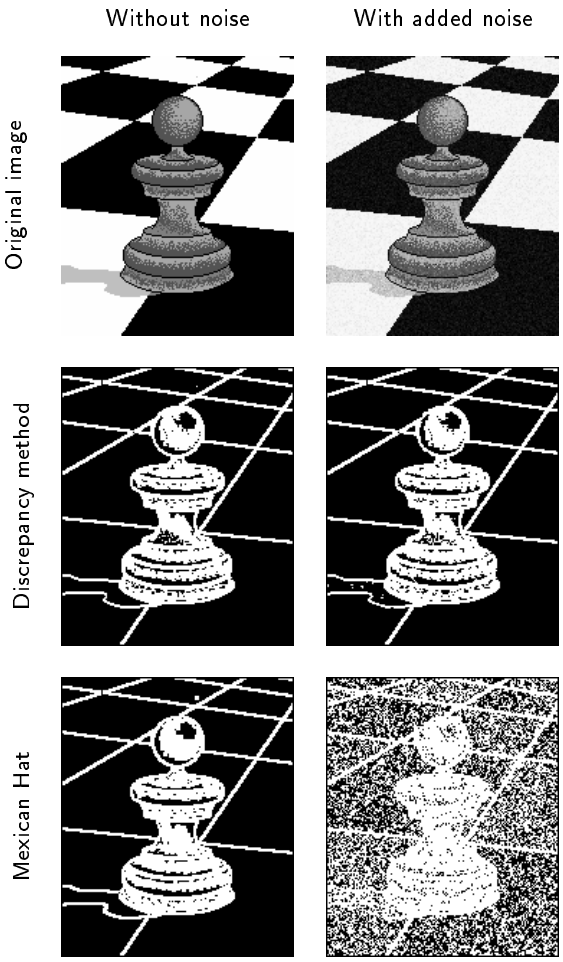

Figure 4: Comparison between e and a standard 3×3 filter mask

Of course, e itself can be used as an edge detector. Figure 4 shows how good it works compared with the commonly used Mexican Hat filter mask.

The fuzzy decision is then done in a rather simple way: We have to compute the degrees of membership to which the pixel belongs to the four types of areas. Hence, the output of the fuzzy system is a vector $t(i,j) = (t_{\mathsf{H}}(i,j), t_{\mathsf{E}}(i,j), t_{\mathsf{R}}(i,j), t_{\mathsf{A}}(i,j))$ with $t_{\mathsf{H}}, t_{\mathsf{E}}, t_{\mathsf{R}}, t_{\mathsf{A}} \in [0,1]$. Since the parameterization of the fuzzy systems is independent from the coordinates in our case we just write v and e for the two inputs $v(i,j)$ and $e(i,j)$ which are treated as linguistic variables in the following. Experiments have shown that $[0,600]$ and $[0,200]$ are appropriate universes of discourse for v and e, respectively. We used simple fuzzy partitions for the fuzzy decomposition of the input space. Their typical shape can be seen in Figure 5.

Five rules, which cover all the possible cases, complete the fuzzy system:

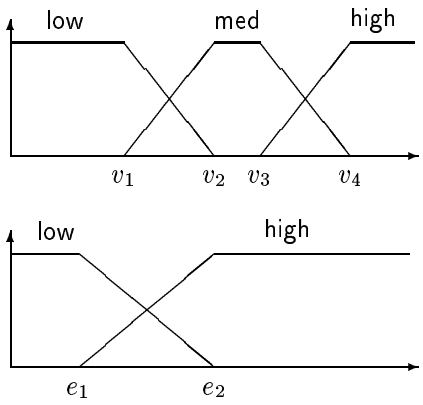

Figure 5: The linguistic variables v and e

IF	v is low			THEN	$t = \mathsf{H}$
IF	v is med	AND	e is high	THEN	$t = \mathsf{E}$
IF	v is high	AND	e is high	THEN	$t = \mathsf{E}$
IF	v is med	AND	e is low	THEN	$t = \mathsf{R}$
IF	v is high	AND	e is low	THEN	$t = \mathsf{A}$

The classification $t(i,j)$ is then computed for each pixel (i,j) by evaluating the rulebase above with respect to the inputs $v(i,j)$ and $e(i,j)$. In this case traditional Mamdani inference is applied with the operations min and max.

Experimental results: The algorithm does not even take three seconds for an image with approximately 250000 pixels on a workstation with a 150MHz CPU.

In [1] and [2] methods for optimizing the fuzzy sets of the two variables (see Figure 5) with genetic algorithms are presented.

4 THE INSPECTION PROCEDURE

As already mentioned, the size of the images is approximately 1MB. The output frequency of the printing machine is one print per second. Therefore, it might be clear that it is impossible to check print after print independently, because this requires a profound analysis of the contents of the picture. Since external references are not available, the first three prints of an order, which typically consists of a few thousand prints, are used as a reference. Consequently, the operator has to take special care of these prints in order to avoid that the following prints are checked against erroneous ones.

Figure 6 shows how the whole system is configured. The inspection program runs autonomously. No additional operators are necessary. The checks and the recognition of order changes are completely automatized.

The following algorithm provides a schematical outline of how the inspection of one single order is accomplished. While the segmentation and the tolerance intervals are computed, which takes approximately three seconds, the printing machine is halted in order to avoid unchecked prints.

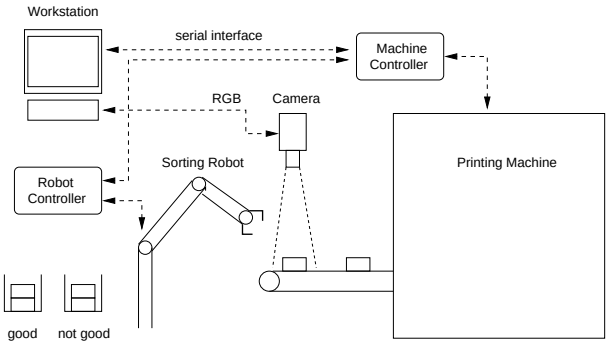

Figure 6: Schematical configuration of the inspection hardware

Algorithm 4

store the first three prints;
compute a reference from these;

compute the segmentation as proposed in 3;
compute tolerance intervals according to the segmentation;

WHILE *order not finished* **DO**
BEGIN
store actual print;
compute deviations from reference;
perform quality decision
END

The final quality decision is done with a fuzzy system whose schematical structure can be seen in Figure 7.

First of all, the number of pixels which exceed the tolerance intervals is computed for each area. Then a quality decision is performed for each area independently (variables $\mathsf{quality_H}$, $\mathsf{quality_E}$, $\mathsf{quality_R}$, and $\mathsf{quality_A}$). For this purpose, the afore said numbers of error pixels (variables $\mathsf{error_H}$, $\mathsf{error_E}$, $\mathsf{error_R}$, and $\mathsf{error_A}$) and the average sizes of the deviations (variables $\mathsf{cont_H}$, $\mathsf{cont_E}$, $\mathsf{cont_R}$, and $\mathsf{cont_A}$) are taken into account.

Finally, the decision, whether the prints are classified as good or not good, is carried out taking the qualities of the four areas and the geometrical shape (i.e., "distributed spots", "horizontal direction", "horizontal direction", "large blur", contained in variable geometry) of the error surface into account. The geometrical information is computed with techniques which are also based on the discrepancy norm.

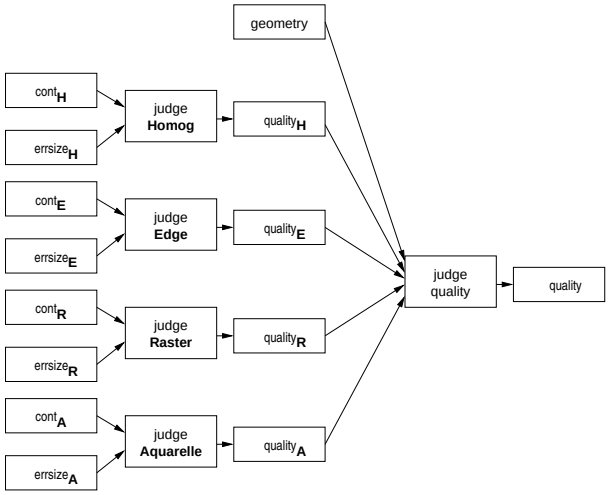

Figure 7: Sketch of the fuzzy system performing the quality decision

References

[1] P. Bauer, U. Bodenhofer, and E. P. Klement. A fuzzy system for image pixel classification and its genetic optimization. In R. Trappl, editor, *Cybernetics and Systems '96*, Vienna, April 1996. Austrian Society for Cybernetic Studies.

[2] U. Bodenhofer. Tuning of fuzzy systems using genetic algorithms. Master's thesis, Johannes Kepler Universität Linz, 1996.

[3] I. Daubechies. Orthonormal bases of wavelets with finite support-connection with discrete filters. In J. M. Combes, A. Grossmann, and P. Tchamitchian, editors, *Wavelets*. Springer, Berlin, Heidelberg, New York, 1989.

[4] H. Neunzert and B. Wetton. Pattern recognition using measure space metrics. Technical Report 28, Universität Kaiserslautern, Fachbereich Mathematik, November 1987.

[5] H. G. Stark. Multiscale analysis, wavelets, and texture quality. Technical Report 41, Universität Kaiserslautern, Fachbereich Mathematik, January 1990.

[6] H. Weyl. Über die Gleichverteilung von Zahlen mod. Eins. *Math. Ann.*, 77:313–352, 1916. in German.