s c c h

software competence center
hagenberg

FUZZY.LOGIC.LAB.LINZ

Advances in Knowledge-Based Technologies

Proceedings of the
Master and PhD Seminar
Winter term 2007/08, part 2

Softwarepark Hagenberg
SCCH, Room 0/2
March 3, 2008

Software Competence Center Hagenberg
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

13:30–15:00 Session 1 *(Chair: Roland Richter)*

13:30	Bernhard Moser: *Misalignment Measure Based on Hermann Weyl's Discrepancy*
14:00	Stefan Raiser: *Object extraction from binary images with graph-based and spectral clustering*
14:30	Werner Groißböck: *Image and Texture Classification based on Image Metrics*

15:00 Coffee Break

15:15–16:45 Session 2 *(Chair: Bernhard Moser)*

15:15	Frank Bauer: *Choosing Regularization Parameters in an Optimal Way without Knowing the Noise Level*
15:45	Peter Sarkoci: *Web-Geometric Aspects of Triangular Norms*
16:15	Bettina Heise: *3D-Micro-Material Inspection by Interferometic and OCT Techniques*

Misalignment Measure Based on Hermann Weyl's Discrepancy[1)]

Bernhard Moser and Thomas Hoch
Software Competence Center Hagenberg, Austria
bernhard.moser@scch.at, www.scch.at

Abstract:
Similarity measures for translationally misaligned image patterns are studied. It turns out that for measures based on standard concepts like cross-correlation, L_p-norm and mutual information monotonicity with respect to the extent of misalignment can not be guarantueed. In this paper we introduce a novel distance measure for images based on Hermann Weyl's discrepancy concept which relies on the evaluation of partial sums. In contrast to standard concepts in this case monotonicity, positive-definiteness and a homogenously linear upper bound with respect to the extent of misalignment can be proven. It is shown that this monotonicity property is not influenced by the image's frequencies or other characteristics which makes this new similarity measure predestinated for similarity-based registration, tracking and segmentation.

1 Introduction

In this paper we focus on measuring the extent of misalignment between translationally transformed gray-valued (textured) image patches. Such similarity measures are commonly used in the context of video tracking of moving objects, registration purposes and also texture analysis. In literature one can find a series of concepts of similarity measures for images. Exemplarily, let us single out the following concepts: cross-correlation, L_p-norms, mutual information, Bhattacharyya coefficient as most popular measures in this context, see, e.g., [4, 5, 6, 7, 8, 11, 13, 16]. We postulate that such a measure should satisfy at least some compatibility criterions with respect to the translational transformations under consideration. Therefore, let us consider a pattern P and its rigid spatial transform P' then — having in the field of image registration and video tracking in mind — usually one expects from such a distance measure between P and P' that it reflects the extent of misalignment performed by the rigid transformation T that carries P over to P'. Roughly spoken, this compatibility criterion means that

[C1] a vanishing distance entails a vanishing extent of misalignment and vice versa (*positive*

[1)] This work was supported by Austrian COMET program.

definiteness)

[C2] the distance measure behaves continuously at least with respect to arbitrary small misalignments (*continuity*)

[C3] an increasing extent of misalignment implies an increasing distance measure and vice versa (*monotonicity*)

As it will be discussed later on and demonstrated by means of examples the standard concepts just mentioned do not meet the criterions **[C1]-[C3]** simultaneously.

To put the conditions **[C1]-[C3]** more formally, let us consider the family

$$\mathcal{T}^N = \{T_t : \mathbb{R}^N \to \mathbb{R}^N | \, T_t(x) = x - t, t \in \mathbb{R}^d\}$$

of translations on $\mathbb{R}^N$, $N \in \mathbb{N}$, respectively.

In order to formalize the criterions **[C1]-[C3]**, let us think of an image pattern P modeled by a discrete function $f : \mathbb{Z}^N \to \{0, 1, \ldots, m\}$ with finite support, or, more generally by a non-negative Lebesque-integrable function $f \in \mathbb{R}^N, d\mu$, $f \geq 0$ with respect to the measure μ. For convenience let us denote $\mathcal{F}^N$ the set of such functions.

Then, let us consider distance measures $d : \mathcal{F}^N \times \mathcal{F}^N \longmapsto \mathbb{R}_0^+$ and its induced misalignment functions $\Delta_d[f] : \mathbb{R}^N \to \mathbb{R}_0^+$,

$$\Delta_d[f](t) = d(f, f_t)$$

where

$$0 < \sigma(f) = \sup\{d(f, f \circ T) | \, T \in \mathcal{T}^N\} < \infty.$$

Then, d induces the corresponding similarity measure s_f

$$s_f : \{f\} \times \{f \circ T | \, T \in \mathcal{T}\} \longmapsto [0, 1], s_f(f, f \circ T) = 1 - \frac{d(f, f \circ T)}{\sigma(f)} \tag{1}$$

for the pattern f and its transform $f \circ T$, $T \in \mathcal{T}^N$.

While the positive-definiteness property **[C1]** can be modelled by requiring

$$\Delta_d[f](t) = 0 \Longleftrightarrow t = 0, \tag{2}$$

the continuity condition **[C2]** ends up with

$$\Delta_d[f](t) \to 0 \Longleftrightarrow t \to 0, \tag{3}$$

and, finally, we have the monotonicity condition **[C3]**

$$0 \leq \lambda_1 \leq \lambda_2 \Longleftrightarrow \Delta_d[f](\lambda_1 t) \leq \Delta_d[f](\lambda_2 t), \tag{4}$$

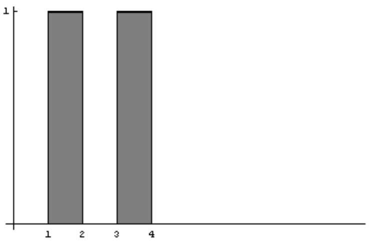


Figure 1: illustration of pattern of eq. (5)

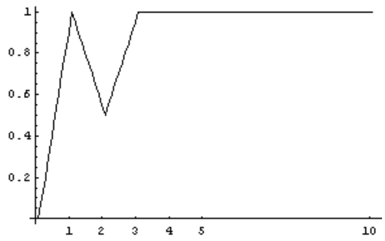


Figure 2: cross-correlation induced misalignment

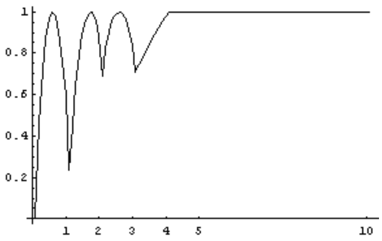


Figure 3: mutual information induced misalignment

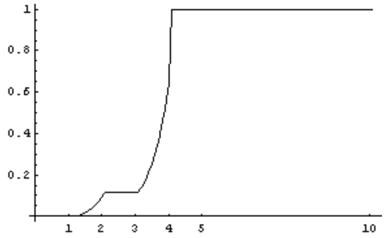


Figure 4: Bhattacharyya induced misalignment

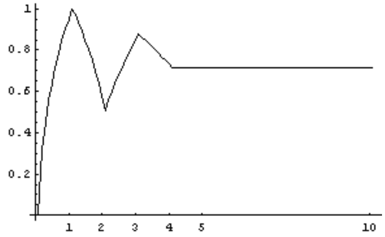


Figure 5: Euclidean distance induced misalignment

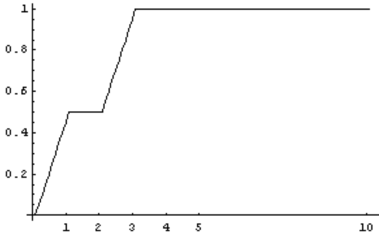


Figure 6: misalignment based on Weyl's discrepancy

where $t \in \mathbb{R}^N$. For simplicity let us look at an illustrative example on $\mathbb{R}$ given by

$$f(x) = \begin{cases} 1 & \text{if } x \in [1,2[\cup[3,4[\\ 0 & \text{else,} \end{cases}$$

as illustrated in Figure 1.

The cross-correlation of f and $f \circ T_t$, which in fact is an auto-correlation, $\chi_f(t) = \int_{-\infty}^{\infty} f(x)(f \circ T_t)(x)dx$, induces the distance measure $d_{\mathbf{corr}}(f, f \circ T_t) = 1 - \chi_f(-t)$. As Figure 2 demonstrates, if the pattern f is shifted from left to right by a shift of t the corresponding misalignment function $\Delta_{\mathbf{corr}}[f]$ does not increase monotonically. In particular, already for $t = 1$ the induced distance becomes maximal which is somewhat counterintuitive as for $t = 1$ the patterns f and $f \circ T_1$ are still close together. This means that the distance measure $d_{\mathbf{corr}}$ is not fully compatible with respect to translations as it does not reflect spatial separation properly. As illustrated in Figure 3 this non-monotonicity applies also to the misalignment function

$$\Delta_{\text{mut}}[f](t) = 1 - I_{\mathbf{mut}}(f, f \circ T_t)$$

based on mutual information

$$I_{\mathbf{mut}}(f, g) = \sum_i \sum_j H[f, g](i, j) \log \frac{H[f, g](i, j)}{H[f](i)H[g](j)}$$

where $H[f, g]$ denotes the joint histogram of f and g and $H[f]$, $H[g]$ denotes the histograms of f, g, respectively. As well for L_p-norms as demonstrated in Figure 5 for the Euclidean distance monotonicity can not be guaranteed. In contrast to the distance and similarity concepts just mentioned the Bhattacharyya similarity coefficient between histograms,

$$B(f,g) = \sum_i \sqrt{H[f](i)\, H[g](i)},$$

induces a misalignment function

$$\Delta_{\textbf{Bhattacharyya}}[f](t) = 1 - B(f, f \circ T_1)$$

that is monotonic but does not satisfy the positive-definiteness condition **[C1]** as demonstrated in Figure 4. Finally, Figure 6 shows the misalignment function based on the so-called discrepancy norm which is defined in Section 2 for one-dimensional functions and vector data. Later on in Section 4 the concept of discrepancy is carried over to Lebesque-integrable functions on arbitrarily finite real domains. Corollary 5 is the central statement of this paper which claims that this discrepancy norm satisfies all the conditions **[C1]**-**[C3]** simultaneously for arbitrary dimensions.

2 Hermann Weyl's Discrepancy Measure

In [14] Hermann Weyl studies pseudorandomness of sequences of numbers from the unit interval. As a measure of discrepancy from uniformly distributed sequences he introduces the so-called discrepancy measure. Recall that a sequence $(x_k)_k \subset (0,1)$ is uniformely distributed if and only if for all $a, b \in (0,1)$, $b < a$, the fraction $\frac{N(a,b)}{N}$ converges uniformly to $(b-a)$, where $N(a,b)$ denotes the number of the first N elements of the sequence (y_k) which are contained in the subinterval (a,b), that is $N(a,b) = |\{k \leq N |\, y_k \in (a,b)\}|$. From this it is quite natural to define

$$D_N = \sup_{0 \leq a < b \leq 1} \left| \frac{N(a,b)}{N} - (b-a) \right| \tag{5}$$

as a notion of discrepancy that reflects to which degree a sequence deviates from being uniformly distributed. In terms of Weyl's definition (5) uniformly distributed sequences can be characterized by $\lim_N D_N = 0$. If we think of $\nu(a,b) = \int_a^b 1dx$ and $\mu(a,b) = \frac{N(a,b)}{N}$ as the measures (in the sense of measure theory) induced by the unifom and the empirical distribution given by the sequence $(y_k)_k$, respectively, we come to the general concept of discrepancy for arbitrary measures

$$D(\mu,\nu) = \sup_{A \in \mathcal{A}} |\mu(A) - \nu(A)|\,, \tag{6}$$

where $\mathcal{A}$ is a σ-algebra of measureable sets over the domain $\mathcal{X}$ and μ and ν are measures defined on the measure space $(\mathcal{X}, \mathcal{A})$.

2.1 Discrepancy for vector data and one-dimensional functions

In this section we focus on discrete data and on measures given by $\mu = \sum_{i=1}^n \alpha_i \delta_{x_i}$ where $x_i \in \mathcal{X}$. For example, x_i represents integer points on the real line or pixel coordinates of an image. For vectors, therefore, let us introduce

$$\|.\|_D : \mathbb{R}^n \to \mathbb{R}^+,\ (\alpha_1, \dots, \alpha_n) \mapsto \max_{1 \le a \le b \le n} |\sum_{i=a}^{b} \alpha_i| \tag{7}$$

which is called the discrepancy norm on $\mathbb{R}^n$.

Applications of Weyl's concept of discrepancy can be found in the field of numerical integration, especially in the context of Monte Carlo methods in high dimensions (e.g., [12, 15]), in computational geometry (e.g., [1]) and in pattern recognition (e.g., [10]). There are also applications to image processing in the context of pixel calssification [2, 3].

The mapping (7) yields the maximal absolute sum of values α_i over index intervals. In contrast to p-norms $\|.\|_p = (\sum_i \alpha_i^p)^{(1/p)}$ the mapping $\|.\|_D$ strongly depends on the sign and also the order of the entries. However, the mapping (7) turns out to fulfill the axioms of a norm: the properties of positive-definiteness, that is $\vec{\alpha} = 0$ if and only if $\|\vec{\alpha}\|_D = 0$ and of positive-homogeneity, $\|\lambda \cdot \vec{\alpha}\|_D = \lambda \cdot \|\vec{\alpha}\|_D$ where $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$, immedeately follow from definition (7). The triangle inequality can also easily be proven by observing that $|\sum_i \alpha_i + \beta_i| \le |\sum_i \alpha_i| + |\sum_i \beta_i|$ implies

$$\max_{1 \le a \le b \le n} |\sum_{i=a}^{b} \alpha_i + \beta_i| \le \max_{1 \le a \le b \le n} \{|\sum_{i=a}^{b} \alpha_i| + |\sum_{i=a}^{b} \beta_i|\} \le \max_{1 \le a \le b \le n} \{|\sum_{i=a}^{b} \alpha_i|\} + \max_{1 \le a \le b \le n} \{|\sum_{i=a}^{b} \beta_i|\}.$$

It is interesting and illustrative to compare $\|.\|_D$ with conventional p-norms. As it is well known from vector space theory all norms of a finite vector space are topologically equivalent, therefore, for any $p \in [1, \infty)$ there are constants $\lambda_1, \lambda_2 > 0$ such that there holds $\lambda_1 \|\vec{\alpha}\|_p \le \|\vec{\alpha}\|_D \le \lambda_2 \|\vec{\alpha}\|_p$ for all $\vec{\alpha} \in \mathbb{R}^n$. This means that the unit balls of the corresponding norms can be nested by scaling. By the way, it turns out that $\lambda_1 = n^{-1/p}$ and $\lambda_2 = n^{1-1/p}$, which can be proven by the fact

$$\|\vec{\alpha}\|_\infty = \max_{1 \le i \le n} |\alpha_i| \le \|\vec{\alpha}\|_D \le \sum_{i=1}^{n} |\alpha_i| = \|\vec{\alpha}\|_1,$$

the inequality

$$\sum_{i=1}^{n} |\alpha_i|^p \le \sum_{i=1}^{n} \|\vec{\alpha}\|_\infty^p = n \|\vec{\alpha}\|_\infty^p,$$

hence, $\|\vec{\alpha}\|_p \le n^{1/p} \|\vec{\alpha}\|_\infty$, and by applying Hoelder's inequality

$$\|\vec{\alpha}\|_1 = \sum_{i=1}^{n} |\alpha_i \cdot 1| \le \|\vec{\alpha}\|_p \|(1, \dots, 1)\|_{1-1/p} = \|\vec{\alpha}\|_p n^{1-1/p}.$$

It is interesting to observe that the unit ball of $\|.\|_D$ is less symmetric than the unit balls of the p-norms. For $n = 2$ we can illustrate the unit balls graphically, see Figure 7 which

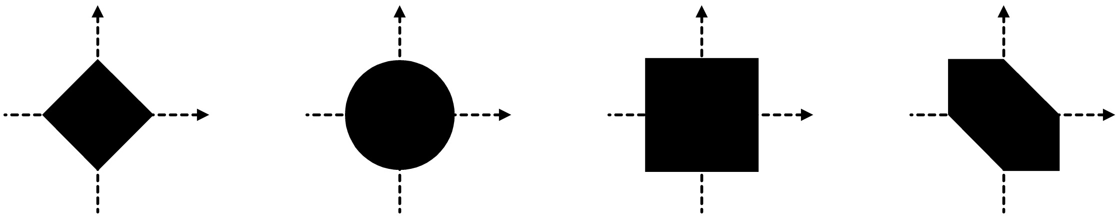


Figure 7: unit balls of $\|.\|_1$ and $\|.\|_2$, $\|.\|_\infty$ and $\|.\|_D$

demonstrates that the $\|.\|_D$ unit ball is biased and that this norm is not isotropic. Gernerally, we have that $\vec{\alpha} = (\alpha_i)_i$ with $\alpha_i \geq 0$ entails the relation $\|\vec{\alpha}\|_D = \|\vec{\alpha}\|_1$ and $\vec{\alpha} = ((-1)^i)_i$ the equality $\|\vec{\alpha}\|_D = \|\vec{\alpha}\|_\infty$, respectively, indicating that the more there are alternating signs of consecutive entries the lower the value of the discrepancy norm. To get a geometric understanding of the discrepancy norm let us consider a contour generated by a sequence of vectors $\mathbf{v} = (\vec{v}_1, \ldots, \vec{v}_n)$ as shown in Figure 8. Observe that the diameter

$$\Theta(\mathbf{v}) = \max_{1 \leq a \leq b \leq n} \| \sum_{i=a}^{b} \vec{v}_i \|_2$$

can be understood as a generalized discrepancy extended to vector entries. The diameter with respect to a certain direction $\vec{w}$ is given by

$$\Theta(\mathbf{v}, \vec{w}) = \max_{1 \leq a \leq b \leq n} | \sum_{i=a}^{b} < \vec{v}_i, \vec{w} > |$$

where $< .,. >$ denotes the usual inner product. Note that we have

$$\Delta(\mathbf{v}, \vec{w}) = \|(< \vec{v}_i, \vec{w} >)_i\|_D. \tag{8}$$

Consequently, let us extend the vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$ to a sequence $\alpha = ((1, \alpha_1), \ldots, (1, \alpha_n))$ of vectors by setting $\vec{v}_i = (1, \alpha_i)$, then we regain the discrepancy by choosing the direction $\vec{w} = (0, 1)$, which means that the contour α is projected onto the y-axis, hence

$$\|(\alpha_1, \ldots, \alpha_n)\|_D = \max_{1 \leq a \leq b \leq n} \left| \sum_{i=a}^{b} \left\langle (1, \alpha_i)^T, (0, 1)^T \right\rangle \right|$$

As illustrated in Figure 8 this means that the discrepancy of a vector $(\alpha_1, \ldots, \alpha_n)$ can be understood as diameter of the contour generated by $(1, \alpha_i)$ in the direction of the y-axis. The contour can also be looked at as the graph of a piecewisely linear function f with slopes α_i, respectively, where $f|_{x=i} = \sum_{i=1}^{j} \alpha_i$. As this diameter equals the difference between the maximal and the minimal value of f we obtain the relationship

Proposition 1 *Denote $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$ then*

$$\|\vec{\alpha}\|_D = \max_{1 \leq i \leq n} \sum_{j=1}^{i} \alpha_j - \min_{1 \leq i \leq n} \sum_{j=1}^{i} \alpha_j. \tag{9}$$

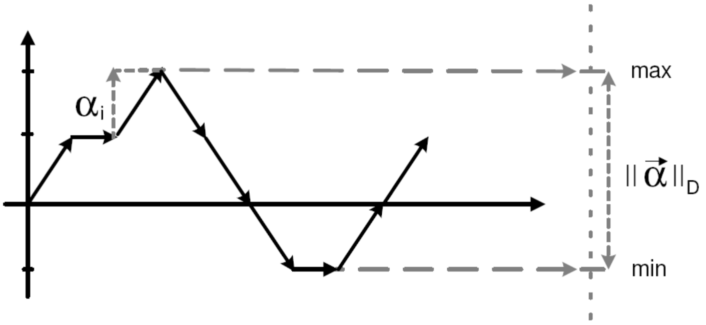


Figure 8: Interpretation of the discrepancy norm $\|(\alpha_1, \ldots, \alpha_n)\|_D$ as diameter

Proposition 1 is crucial for performance reasons. Note that equation (9) allows to compute the discrepancy of a vector of length n with $O(n)$ operations instead of $O(n^2)$ number of operations resulting from the original definition.

3 Properties of the misalignment function based on the discrepancy norm, the one-dimensional case

In this section we turn to the more general setting of Lebesque integrable function space $\mathcal{L}_1(\mathbb{R}, \mu)$ on the real line $\mathbb{R}$ equipped with a Lebesque measure μ. Then, in analogy of Definition 7 let us set

$$\|f\|_D = \sup_{X=[a,b]\subset\mathbb{R}} |\int_X f d\mu| \tag{10}$$

Recall that

$$\omega(\delta; g) := \sup_{|h|<\delta} \sup_x |g(x+h) - g(x)| \tag{11}$$

is called the modulus of continuity of the continuous function g. Theorem 2 demonstrates that H. Weyl's discrepancy meets the conditions **[C1]**-**[C3]** of Section 1. The proof can be found in [9].

Theorem 2 *Let $f \in \mathcal{L}_1(\mathbb{R}, \mu)$, let $F(x) = \int_{[-\infty,x]} f d\mu$, $\delta_\mu(t) = \sup\{\mu([a,b])|\, |b-a| \leq t\}$, let $f_t(x) = f(x-t)$ and let $\Delta_D[f](t) = \|f - f_t\|_D$ denote the misalignment function, then:*

1. *If f is non-trivial, i.e., $\int |f| d\mu > 0$, then*

$$\Delta_D[f](t) = 0 \Longleftrightarrow t = 0.$$

2. *The misalignment function satisfies the Lipschitz continuity condition at $t = 0$, i.e.,*

$$\Delta_D[f](t) \leq 2\,\omega(t; F) \leq 2\,\delta_\mu(t)\,\|f\|_\infty \tag{12}$$

3. The misalignment function is monotone in the sense that for any $t \in \mathbb{R}$

$$0 \leq \lambda_1 \leq \lambda_2 \Longrightarrow \Delta_D[f](\lambda_1 t) \leq \Delta_D[f](\lambda_2 t) \tag{13}$$

4 Extension to multivariate functions

A natural way to extend the concept of discrepancy norm of vector data to image data or, more generally, to Lebesque integrable multivariate functions $f \in \mathcal{L}(\mathbb{R}^N, d\mu)$, $N \in \mathbb{N}$, is by replacing intervals in (10) by connected sets, i.e.,

$$\|f\|_{\mathcal{C}}^{(N)} = \sup_{\omega \in \mathcal{C}} |\int_C f d\mu| \tag{14}$$

where $\mathcal{C}$ refers to a system of connected subsets of $\mathbb{R}^N$. In this paper we concentrate on systems $\mathcal{C}$ which are made up by Cartesian products of intervals. Particularly, we study

$$\|f\|_{\mathcal{B}^N}^{(N)} = \sup_{B \in \mathcal{B}^N} |\int_B f d\mu| \tag{15}$$

where $\mathcal{B}^N$ denotes the set of N-dimensional open boxes $I_1 \times I_2 \times \cdots \times I_N$ with open intervals I_i from the real line, and the variant

$$\|f\|_{\widetilde{\mathcal{B}}^N}^{(N)} = \sup_{B \in \widetilde{\mathcal{B}}^N} |\int_B f d\mu| \tag{16}$$

where $\widetilde{\mathcal{B}}^N \subset \mathcal{B}^N$ is composed by Cartesian products of intervals of the form $]-\infty, x[$, $]x, \infty[$.

However, formula (14) is not the only possibility for an extension. Particularly, (9) gives reason to construct a measure based on integral images and analogous concepts for higher dimensions, respectively. Therefore, let us define the set of integrals

$$I_f^{(i_1,\ldots,i_N)} = \left\{ \int_{T_s(Q_{(i_1,\ldots,i_N)})} f d\mu \,\middle|\, s \in \mathbb{R}^N \right\} \tag{17}$$

where

$$Q_{(i_1,\ldots,i_N)} = \{(\lambda_1 i_1, \ldots, \lambda_N i_N)^T | \, \lambda_1 \geq 0, \ldots, \lambda_1 \geq 0\},$$

$T_s(A) = \{x - s | \, x \in A\}$ and $i_j \in \{-1, 1\}$. For example, the sets $Q_{(i,j)}$, $i, j \in \{-1, 1\}$ describe the four quadrants of the two-dimensional plane. Observe that (17) coincides with the well-known construction $I_{i_0,j_0} = \sum_{i \leq i_0, j \leq j_0} f(i,j)$ of an integral image in case of $N = 2$, $i_1 = i_2 = 1$ and the discrete finite measure $d\mu$.

For symmetry reasons we propose the following alternative definition of discrepancy for multivariate functions $f \in \mathcal{L}(\mathbb{R}^N, d\mu)$, $N \in \mathbb{N}$

$$\|f\|_{\mathbf{I}}^{(N)} = \max_{(i_1,\ldots,i_N) \in \{-1,1\}^N} \left\{ \sup I_f^{(i_1,\ldots,i_N)} - \inf I_f^{(i_1,\ldots,i_N)} \right\}. \tag{18}$$

Next proposition states that (15), (16) and (18) extend the 1-dimensional case 10 in the sense that they coincide for the one-dimensional case with (10) and that they are norms. Its proof can be found in [9].

Proposition 3 *Let $N \in \mathbb{N}$, then*

1. $\|.\|_{\mathcal{B}^N}^{(N)}$, $\|.\|_{\widetilde{\mathcal{B}}^N}^{(N)}$ *and* $\|.\|_{\mathbf{I}}^{(N)}$ *are norms on* $\mathcal{L}(\mathbb{R}^N, d\mu)$.
2. $\|f\|_D = \|f\|_{\mathcal{B}^1}^{(1)} = \|f\|_{\widetilde{\mathcal{B}}^1}^{(1)} = \|f\|_{\mathbf{I}}^{(1)}$ *for all* $f \in \mathcal{L}(\mathbb{R}^1, d\mu)$.

However, the 2-dimensional example $f_0 : \mathbb{Z}^2 \to \mathbb{R}$, given by

$$(f_0(i,j))_{i,j} = \begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & 3 & 3 & -1 \\ -1 & 3 & 3 & -1 \\ -1 & -1 & -1 & -1 \end{pmatrix}$$

yields $\|f_0\|_{\widetilde{\mathcal{B}}^2}^{(2)} = 7$, $\|f_0\|_{\mathbf{I}}^{(2)} = 11$ and $\|f_0\|_{\mathcal{B}^2}^{(2)} = 12$ showing that in general the constructions (15), (16) and (18) do not coincide.

Surprisingly, however, next lemma shows that on the reduced function space $\{f - f \circ T | \, f \in \mathcal{L}(\mathbb{R}^N, \mu), f \geq 0\}$ the discrepancy concepts $\|.\|_{\widetilde{\mathcal{B}}^N}^{(N)}$ and $\|.\|_{\mathcal{B}^N}^{(N)}$ coincide, and that the misalignment function based on (15) can be expressed in terms of integral images and its higher dimensional variants, respectively.

Lemma 4 *For all $N \in \mathbb{N}$ and non-negative $f \in \mathcal{L}(\mathbb{R}^N, d\mu)$, $f \geq 0$, $T \in \mathcal{T}^N$, there holds*

$$\|f - f \circ T\|_{\mathcal{B}^N}^{(N)} = \|f - f \circ T\|_{\widetilde{\mathcal{B}}^N}^{(N)} \tag{19}$$

$$= \max_{(i_1,\ldots,i_N)\in\{-1,1\}^N} \left\{ \sup I_{f - f\circ T}^{(i_1,\ldots,i_N)} \right\} \tag{20}$$

$$\tag{21}$$

Finally, corollary 5 provides the generalization of theorem 2 for $\|.\|_{\mathcal{B}^N}^{(N)}$ and $\|.\|_{\widetilde{\mathcal{B}}^N}^{(N)}$.

Corollary 5 *Let $N \in \mathbb{N}$, let $f \in \mathcal{L}(\mathbb{R}^N, \mu)$, $f \geq 0$, and let denote $\Delta_{\mathcal{C}}[f](t) = \|f - f \circ T_t\|_{\mathcal{C}}$ the misalignment function, $t \in \mathbb{R}^N$, furher, let*

$$\delta_\mu[f](t) = \sup_{\widetilde{B}\in\mathcal{B}^N} \max \left\{ \int_{\widetilde{B}\setminus T_t(\widetilde{B})} f d\mu, \int_{T_t(\widetilde{B})\setminus\widetilde{B}} f d\mu \right\}$$

then for $\mathcal{C} = \mathcal{B}^N = \widetilde{\mathcal{B}}^N$ we have

1. *If f is non-trivial, i.e., $\int |f| d\mu > 0$, then*

$$\Delta_{\widetilde{\mathcal{C}}}[f](t) = 0 \Longleftrightarrow t = 0$$

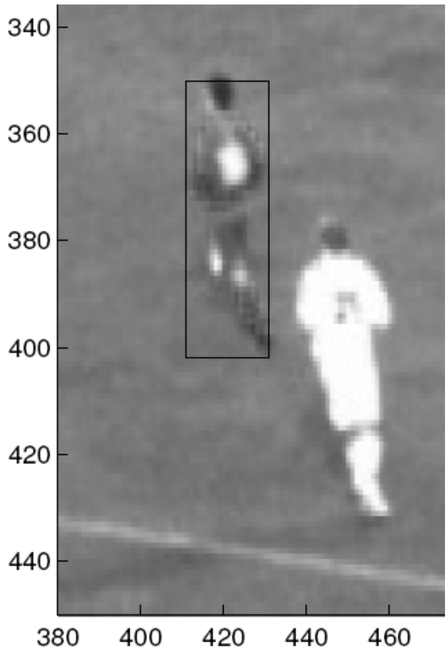


Figure 9: Example for 2D misaligment; the box is shifted

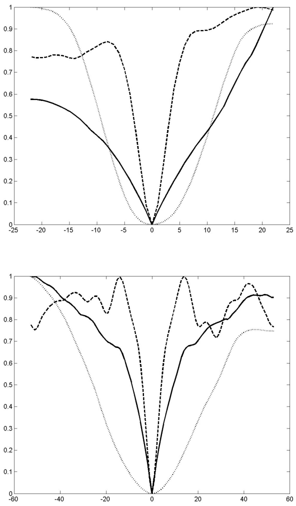


Figure 10: Misalignment w.r.t. shift in x (top) and y (below); gray: Bhattacharyya, solid: discrepancy, dashed: correlation

2. *The misalignment function satisfies the inequality*

$$\Delta_{\widetilde{C}}[f](t) \leq \delta_{\mu}[f](t). \tag{22}$$

3. *The misalignment function is monotone in the sense that*

$$\lambda_1 \leq \lambda_2 \Longrightarrow \Delta_{\widetilde{C}}[f](\lambda_1 t) \leq \Delta_{\widetilde{C}}[f](\lambda_2 t) \tag{23}$$

for arbitrary $t \in \mathbb{R}^N$.

Figure 9 shows an example of a soccer player. The image patch marked by the black box is shifted around the player in order to generate the 2-dimensional misalignment function which is illustriated in the Figures 11, 12 and 13 for Bhattacharyya, cross-correlation and dsicrepancy, respectively. In order to make the various similarity measures visually better comparable, the x and y cuts of the surfaces of Figures 11, 12 and 13 are depicted in Figure 4. As expected from theory the misalignment function induced by the discrepancy $\|.\|_{\widetilde{\mathcal{B}}^N}$ shows monotonicity and positive-definiteness.

5 Conclusion

In this paper we have discussed similarity of images from an axiomatic point of view concentrating on positive-definiteness and monotonicity properties of the corresponding misalignment

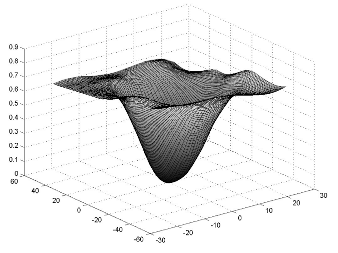


Figure 11: Bhattacharyya induced misalignment

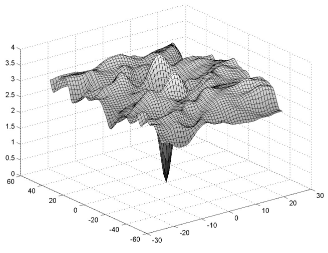


Figure 12: cross-correlation induced misalignment

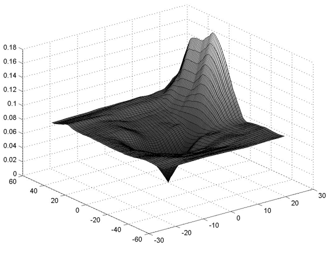


Figure 13: discrepancy induced misalignment

functions. It could be demonstrated that a similarity concept based on Hermann Weyl's discrepancy meets these criterions in contrast to the standard concepts like mutual information and cross-correlation. Both properties make this similarity measure predestinated for tracking and registration applications which remains to be investigated in depth in future work.

Acknowledgment

The authors would like to thank Ulrich Bodenhofer for his original idea to apply Hermann Weyl's discrepancy norm to pixel classification which inspired the authors to the investigations outlined in this paper. This work was supported by the Austrian COMET Program.

References

[1] J. R. Alexander, J. Beck, and W.W.L. Chen. Geometric discrepancy theory and uniform distribution. pages 185–207, 1997.

[2] P. Bauer, U. Bodenhofer, and E. P. Klement. A fuzzy method for pixel classification and its application to print inspection. In *Proc. 6th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume 3, pages 1301–1305, Granada, July 1996.

[3] 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*, volume 1, pages 285–290, Vienna, April 1996. Austrian Society for Cybernetic Studies.

[4] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distributions. *Bull. Calcutta Math*, 35:99–109, 1943.

[5] R. Jain, S.N.J. Murthy, P.L.J. Chen, and S. Chatterjee. Similarity measures for image databases. volume 3, pages 1247–1254, 1995.

[6] W. Jiang, G. Er, Q. Dai, and J. Gu. Similarity-based online feature selection in content-based image retrieval. *IEEE Transactions on Image Processing*, 15(3):702–712, 2006.

[7] M.S. Khalid and M.B. Malik. Biased nature of Bhattacharyya coefficient in correlation of gray-scale objects. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 209–214, 2005.

[8] O. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the Bhattacharyya gradient flow. *IEEE Transactions on Image Processing*, 16(11):2787–2801, 2007.

[9] B. Moser. Similarity measure for image and volumetric data based on Hermann Weyl's discrepancy measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* (submitted).

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

[11] J.P.W. Pluim, J.B.A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.

[12] G. Takhtamysheva, B. Vandewoestyneb, and R. Coolsb. Quasi-random integration in high dimensions. *Image Vision Comput.*, 73(5):309–319, 2007.

[13] N. Vasconcelos and A. Lippman. A unifying view of image similarity. In *Proceedings. 15th International Conference on Pattern Recognition*, volume 1, pages 38–41, August 2001.

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

[15] S.K. Zaremba. The mathematical basis of Monte Carlo and Quasi-Monte Carlo methods. *SIAM Review*, 10(3):303–314, 200.

[16] B. Zitov and J. Flusser. Image registration methods: a survey. *Image Vision Comput.*, 21(11):977–1000, 2003.

Technical Report
FLLL–TR–2008-01

FUZZY.LOGIC.LAB.LINZ

Object extraction from binary images with graph-based and spectral clustering

Stefan Raiser
Fuzzy Logic Laboratorium Linz-Hagenberg
e-mail `stefan.raiser@jku.at`

Institut für Wissensbasierte Mathematische Systeme
Fuzzy Logic Laboratorium Linz

Johannes Kepler Universität Linz
Altenberger Str. 69
A-4040 Linz
Austria

softwarepark hagenberg

Softwarepark Hagenberg
Softwarepark 21
A-4232 Hagenberg
Austria

Abstract

In this paper clustering methods are used to perform object extraction from binary images as they occur in industrial surface inspection processes. Graph-based and spectral clustering algorithms, which are both capable of detecting arbitrary and non-connected shapes of "white" pixels, are presented.

Keywords: object extraction, graph-based clustering, spectral clustering

1 Introduction

One of the major steps in image processing is image segmentation, which means to extract regions from the image that correspond to the objects we are interested in. In surface inspection these interesting objects are all deviations from the "perfect" part, because they may represent an unacceptable imperfection or even a severe defect. So in many cases the segmentation takes a test image from a part to be inspected and a master image of a "perfect" workpiece as inputs. It's obvious that these two images have to be congruent (at least as good as possible) and have to be aquired under similar illumination conditions in order to make them comparable. The first step now is to calculate the absolute difference between master and test image. The resulting values make up a difference image, where all non-zero pixels represent a deviation from the ideal master. As next step a thresholding, which is the simplest segmentation algorithm, is applied on the difference image. The threshold operation sets all pixels, which have a grayvalue above a certain level to one and the remaining pixels to zero. So thresholding produces a binary image with a black (0) background and a white (1) foreground, which contains the objects of interest. Generally the binary image contains multiple (interesting) objects that should be returned individually. In other words the white pixels have to be somehow grouped together in order to extract distinct regions.

Clustering methods have been used in a variety of disciplines leading from statistics and numerical analysis to data mining and machine learning. Generally speaking clustering can find "natural" groupings (clusters) in data. Each of these clusters consist of datapoints that are similar between themselves and dissimilar to those of other groups with respect to a previously choosen similarity/dissimilarity measure.

These characteristics make clustering a candidate for solving the object extraction task described before. The basic idea is to consider the white pixels in the binary difference image as datapoints, which serve as input for the clustering algorithm. In this case the original task, namely the extraction of objects, becomes equivalent to the problem of finding clusters in the set of datapoints. The question, which pixels belong together or to the same object respectively, is now addressed by the clustering algorithm and on how it partitions the datapoints.

In the next sections two different clustering approaches, which could be used for the object extraction task, are described in detail.

2 Graph-Based Clustering - Reduced Delaunay Graph

Generally graph-based clustering starts by constructing a (neighborhood) graph, which vertices correspond to data points and weighted edges represent (dis-)similarities between them. Then an algorithm is applied to partition the graph by deleting edges according to some criterion. The result is a set of connected components, where each connected component represents a single cluster. In this sense clustering can be seen as a problem of cutting graphs into "good" pieces.

A short review on the most important terminology of graph theory needed in this section is given in the following (see also [1]):

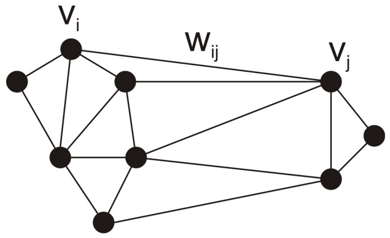


Figure 1: Graph example.

- A *graph* is a set of vertices $V = \{v_1, ..., v_n\}$ and edges E which connect various pairs of vertices. A graph can be written $G = \{V, E\}$. Each edge can be represented by a pair of vertices (v_i, v_j), that is $E \subset V \times V$.
- A *directed graph* is one in which edges (v_i, v_j) and (v_j, v_i) are distinct.
- An *undirected graph* is one in which no distinction is drawn between edges (v_i, v_j) and (v_j, v_i).
- A *weighted graph* is one in which a weight w_{ij} is associated with each edge.
- If the edges are not weighted, the graph is called *unweighted graph.*
- Two vertices are said to be *connected* if there is a sequence of edges starting at the one and ending at the other;
- A *connected graph* is one where every pair of vertices is connected.
- A graph that is not connected is called *disconnected graph.*
- Every graph consists of a disjoint set of *connected components*, that is $G = \{V_1 \cup V_2 ... V_m, E_1 \cup E_2 ... E_m\}$, where $\{V_i, E_i\}$ are all connected graphs and there is no edge in E that connects an element of V_i with one of V_j for $i \neq j$.
- A graph is described by its *adjacency matrix (also called weight matrix)*

$$W = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & w_{ij} & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix}.$$

  Its elements w_{ij} are zero if the vertices v_i and v_j are not connected. If they are connected, w_{ij} exhibits the weight of the edge.

- The *degree of a vertex* is the sum of all adjacent edge weights: $d_i = \sum\limits_{j=1}^{n} w_{ij}$.
- The *degree matrix* D is a diagonal matrix with the degrees of the vertices as diagonal elements:

$$D = \begin{bmatrix} d_1 & \dots & 0 \\ \vdots & d_i & \vdots \\ 0 & \cdots & d_n \end{bmatrix}.$$

The graph-based data representation is ideally suited for encoding pairwise information of objects such as similarities, distances and relations. In this way the graph captures the local neighborhood relationships of the data points. Moreover, in most cases this leads to a sparse representation of the data, which has computational advantages.

In the object extraction task every white pixel x_i of the binary image is viewed as a vertex v_i of a weighted graph. To calculate the (spatial) similarity between two white pixels x_i und x_j the Gaussian similarity function

$$s(x_i, x_j) = e^{-\frac{d(x_i,x_j)^2}{2\sigma^2}}$$

is commonly used. As distance function $d(x_i, x_j)$ the Euclidian or some other distance measure can be applied. The extraction of objects in this case is accomplished by partitioning the graph according to the particular clustering algorithm and finding the resulting connected components.

Given a data set consisting of points $x_1, ..., x_n$ with pairwise simlilarities $s_{ij} = s(x_i, x_j)$ or distances $d_{ij} = d(x_i, x_j)$, a graph can be constructed in various ways. Some of the most popular constructions are listed in the following:

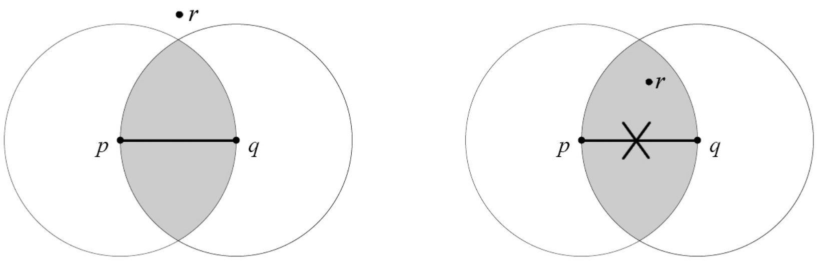
- **ϵ-Neighborhood Graph**. Here all vertices whose pairwise distances are smaller than ϵ are connected. As edge weights the distance values can be used, but as the distances between all connected points are roughly the same (at most ϵ), the ϵ-neighborhood graph is usually considered as an unweighted graph.

- **k-Nearest Neighbor Graph (k-NNG)**. In this case vertex v_i is connected to vertex v_j if v_j is among the k nearest neighbors of v_i or vice-versa. If k=1 then this type of graph is called simply *Nearest Neighbor Graph (NNG)*. The second possibility is to connect vertex v_i with v_j, if both v_i is among the k-nearest neighbors of v_j and v_j is among the k-nearest neighbors of v_i. The resulting graph is called the *Mutual k-Nearest Neighbor Graph*. The edges are weighted by either the pairwise similarity or distance values. The edges of the so-called *Shared Nearest Neighbour Graph* exhibit the number of nearest neighbors two vertices have in common.

- **Minimum Spanning Tree (MST)**. A spanning tree of a graph is a connected graph which connects all points of the original graph by only one path. A single graph can have many different spanning trees. Hence, the minimum spanning tree is a spanning tree with weight less than or equal to the weight of every other spanning tree.

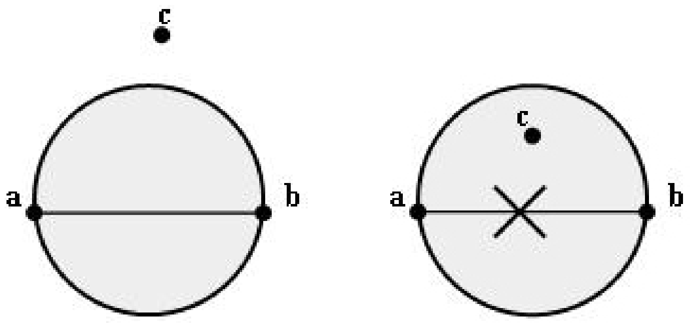
- **Relative Neighborhood Graph (RNG)**. Here the two points v_i and v_j are connected if and only if there is no other vertex from the original data set laying in a lune between v_i and v_j, which is defined as the disjoint intersection between two hyperspheres centered at v_i and v_j and whose radii are equal to the distance between them (see figure 2).

- **Gabriel Graph (GG)**. Here two vertices v_i and v_j are connected by an edge, if they form the endpoints of the diameter of an empty sphere. In the two-dimensional case v_i and v_j are connected, if the disk, having the line segment $\overline{v_i v_j}$ as its diameter, contains no other vertices from the original data set (see figure 3).



(a) (b)

Figure 2: a) Points p and q are connected, b) Points p and q are not connected, because of r inside the lune



(a) (b)

Figure 3: a) Points a and b are connected, b) Points a and b are not connected, because of c inside the disk

- **Delaunay Triangulation (DT)**. The DT (also called *Delaunay Diagramm*) is the dual of the *Voronoi Diagramm*. Both are discussed below in detail in the context of the *Reduced Delaunay Graph*.
- **Fully Connected Graph**. Here all points with positive similarity are connected with each other and the edges are weighted by the corresponding similarity values. This construction is usually only chosen if the similarity function itself already encodes mainly local neighborhoods.

In figure 4 different graphs (namely the NNG, MST, RNG, GG and DT) are visualized for a sample set of data points. It can be seen i.e. that the NNG already consists of a set of connected components (without any partitioning operation applied), so it can be directly used as clustering result. Obviously the number of edges increase from left to right, which lines up with the mathematical relationship between the shown graphs (see also [2]):

$$NNG \subseteq MST \subseteq RNG \subseteq GG \subseteq DT$$

One representative of a graph-based clustering algorithm discussed in more detail here is the *Reduced Delaunay Graph* (RDG). According to [3] it is able to find clusters of complex shape and groups data points simlilarly to human observers. The algorithm starts by constructing a *Delaunay*

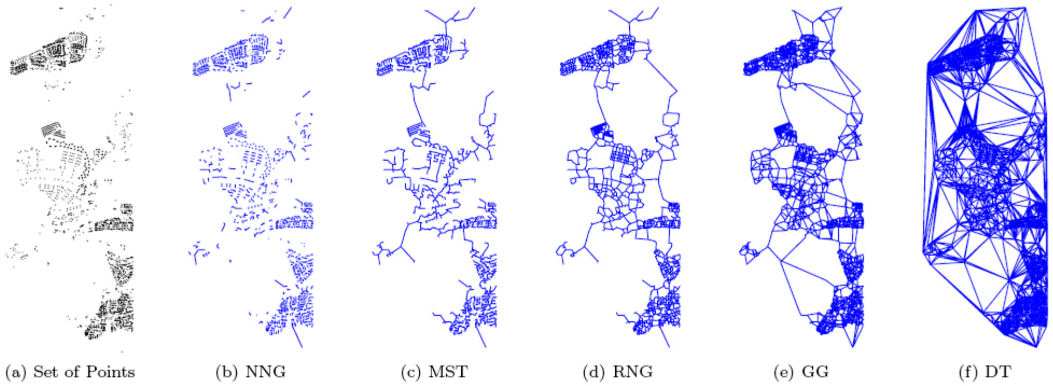

(a) Set of Points (b) NNG (c) MST (d) RNG (e) GG (f) DT

Figure 4: Different neighborhood graphs (taken from [2]).

graph DG from a set of data points S. Given the set $S = p_1, ..., p_N$ of N points in the plane, it is possible to partition the plane into cells $C_1, ..., C_N$ in such a way that the points which belong to cell C_j, associated with point $p_j \in S$, are closer to p_j than to any other point $p_k \in S, k \neq j$:

$$q \in C_j \quad \Leftrightarrow \quad d(q, p_j) \leq d(q, p_k), \quad \forall q, \quad \forall p_j, p_k \in S$$

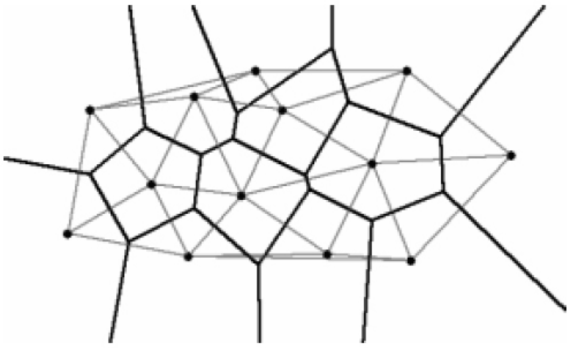


Figure 5: Voronoi diagramm and Delaunay graph of a set of data points.

The resulting partition is called *Voronoi diagramm*. By connecting all pairs of points of S, whose Voronoi diagramm cells share a boundary, the Delaunay graph is obtained. In order to determine the weight for each edge the distance between the corresponding vertices p and q is computed and normalized with the distance of p and q to their respective nearest neighbours according to the following formulas:

$$r_1(e) = \xi(p, q) = \frac{d(p, q)}{\min\limits_{x \in S} \{d(p, x)\}} \qquad r_2(e) = \xi(q, p) = \frac{d(q, p)}{\min\limits_{x \in S} \{d(q, x)\}}$$

So for each edge the two ratios $r_1(e)$ and $r_2(e)$ are obtained , which generally are not equal. These two are combined into a single number by calculating their geometric average:

$$r(e) = \sqrt{r_1(e) \cdot r_2(e)}$$

Clustering is now performed by removing all edges from the DG for which the weight $r(e)$ is larger than a fixed threshold. The remaining graph is called the *Reduced Delaunay Graph* (RDG). By applying a graph-traversing algorithm, i.e a recursive depth-search, the connected components of the graph can be found. Finally each connected component corresponds to a cluster.

One important question is how to choose the threshold value for removing the edges from the DG. When sorting all edge weights of the graph in a decending order and making a plot with the x-axis representing the numbers from one to the number of edges and the y-axis showing the sorted weight values, an L-shaped curve evolves. A good threshold value can be obtained by taking the edge weight at the "knee" of the curve. It can be determined automatically by normalizing the two axis and finding the point with the closest (euclidian) distance to the origin.

When applying the RDG algorithm to three artificial test images the following results are obtained:

Figures 6, 7 and 8 show the DG in the upper left, the RDG in the upper right, the clustering result in the lower left and the L-shaped curve with the sorted edge weights (in the plots called normalized distances) in the lower right corner.

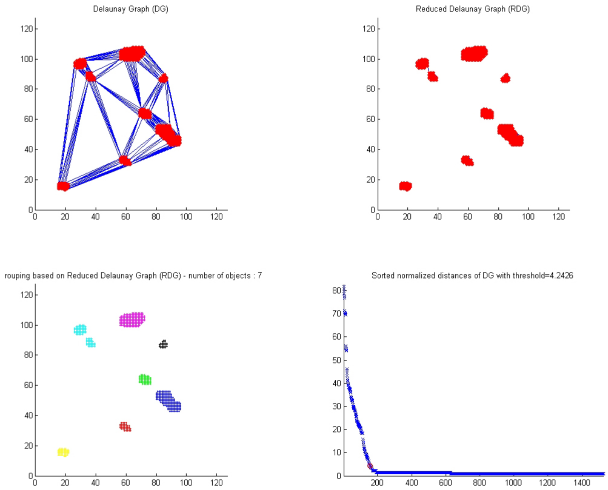


Figure 6: RDG results of the first artificial dataset.

In the first case of compact clusters the algorithm finds seven objects. As it can be seen in the upper right of figure 6 two compact clusters are still connected by an edge after thresholding and are therefore grouped together. Whether this is desirable or not depends on the application. Anyway, the two clusters are quite close, so grouping them into a single cluster is not completely wrong. In the lower right of the figure the determination of the threshold value is visualized. The small red circle represents the knee-point of the curve closest to the origin, which complies to the automatically found threshold value (in this case the threshold is around 4.24).

Figure 7 shows the results for the second test image with various shapes. Here the seven distinct objects are perfectly extracted.

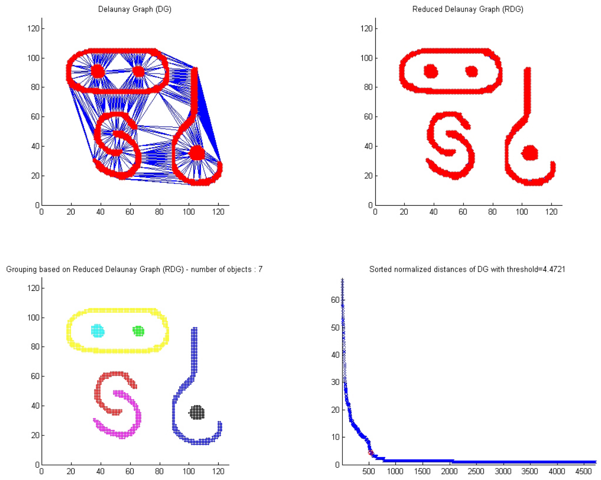


Figure 7: RDG results of the second artificial dataset.

Also in the last case of three objects made up by non-connected pixels the RDG algorithm works very well as it is shown in figure 8.

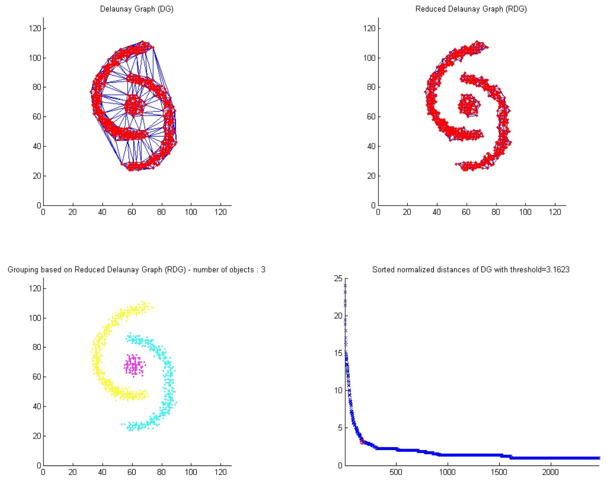


Figure 8: RDG results of the third artificial dataset.

Finally two other promising graph-based approaches suitable for object extraction should be mentioned here: the *Chameleon* algorithm [4] and the *Shared Nearest Neighbor Clustering* (SNN) [5]. The first one is a two-phase algorithm, which starts with partitioning a sparse graph representation of the data points (in this case a k-nearest neighbor graph) into several relatively small subclusters. In the second phase these subclusters are repeatedly merged according to two similarity measures, namely the relative interconnectivity (RI) and the relative closeness (RC). The latter approach represents a combination of graph- and density-based clustering. After constructing a shared nearest neighbor graph, an algorithm similar to DBSCAN is applied to group the data points. It is able to find clusters of varying shapes, sizes and densities.

3 Spectral Clustering - Normalized Cut

In recent years, spectral clustering has become one of the most popular modern clustering algorithms. It can be seen as a special kind of graph-based clustering as it also operates on a graph representation of the data set. There exist different point of views about spectral clustering, but basically all of them lead to the same equations and algorithms. In the following an approach derived from graph partitioning is pursued (as described by [6]). A throughout tutorial on spectral clustering can be found in [7].

Spectral clustering starts with a weighted, undirected graph $G = \{V, E\}$, where the weight of each edge $w(v_i, v_j)$ is a function of similarity between the vertices v_i and v_j. The grouping task now consists of partitioning the set of vertices into disjoint sets $V_1, ..., V_m$, where the similarity of the vertices within one of these sets is high and accross different sets is low.

The straightforward way to partition the graph into two pieces is to find the two disjoint sets A, B with $A \cup B = V$ and $A \cap B = \{\}$, which minimize the *cut* value:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

Computing the optimal bipartitioning (aka. the minimum cut) is a well-studied problem and there exist efficient algorithms for solving it. But the minimum cut criterium has one drawback as it favours cutting small sets of isolated vertices in the graph. Figure 9 illustrates that the cut which partitions out the vertices n_1 or n_2 has a smaller cut value than the vertical cut dividing the regions with different densities.

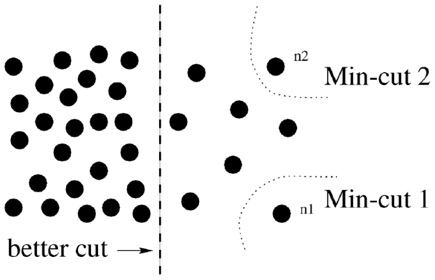


Figure 9: Here the minimum cut criterion leads to a bad partition.

In order to remove this bias a normalized measure called *normalized cut (Ncut)* is defined as:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

with

$$assoc(A,V) = \sum_{u \in A, t \in V} w(u,t) \quad , \quad assoc(B,V) = \sum_{u \in B, t \in V} w(u,t)$$

The normalized cut represents a dissassociation measure, which reflects the dissimilarity between the two sets A and B. Compared to the minimum cut this definition of the dissassociation between the two groups has the advantage, that isolated vertices don't get small cut values.

A measure for total normalized association within groups for a given partition is defined by:

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

where $assoc(A,A)$ and $assoc(B,B)$ are the total weights of edges connecting vertices within A and B, repectively. It reflects how tightly on average the vertices within the groups are connected to each other. These two measures for association and disassociation are related by the following formula:

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

This means that minimizing the disassociation between the groups and maximizing the association within the groups can be satisfied simultaneously. Hence, the clustering algorithm just has to minimize the normalized cut in order to find the "best" two-way partitioning of a graph. The criterion for doing so can be formulated mathematically in the following way:

Let x be an $N = |V|$ dimensional indicator vector, where $x_i = 1$ if vertex v_i is in A and -1 otherwise. Finding the minimum of $Ncut(A,B)$ can be rewritten as

$$\min_x Ncut(x) = \min_y \frac{y^T(D-W)y}{y^T Dy}$$

with the conditions $y(i) \in \{1, -b\}$ and $y^T D1 = 0$. D and W are representing the degree matrix and the adjacency matrix of the graph. The term $(D-W)$ is called the (unnormalized) *Laplacian matrix*. The values of y are used to distinguish between the vertices belonging to A and B after the bipartitioning. If $y(i)$ is 1 then the vertex v_i belongs to A and if $y(i)$ is $-b$ it belongs to B.

Finding the discrete valued vector y that minimizes the criterion is a very difficult non-deterministic polynomial time problem, but a solution can be found efficiently, when the problem is embedded in the real value domain. When y is allowed to take real values (also called relaxation) it can be shown that the minimization of the Ncut criterion leads to the following generalized eigenvalue problem:

$$(D-W)y = \lambda Dy.$$

It can be proven that the smallest eigenvalue is guaranteed to be zero and the second smallest eigenvector of the generalized eigensystem represents the real valued solution of the normalized cut problem. Consequently the third smallest eigenvector optimally subpartitions the first two pieces and so on. Finally the normalized cut clustering algorithm consists of the following steps [6]:

1. Set up a weighted graph $G = \{V, E\}$ and set the weight of an edge connecting two vertices to be a measure of similarity between the two vertices.
2. Solve $(D - W)y = \lambda Dy$ for eigenvectors with the smallest eigenvalues.
3. Use the eigenvector with the second smallest eigenvalue to bipartition the graph.
4. Decide if the current partition should be subdivided and recursively repartition the segmented parts if necessary.

When using the spectral clustering algorithm in practice for object extraction in binary images, some additional things have to be considered. First of all the graph edge weight has to be choosen according to a similarity measure. As mentioned in the previous chapter the Gaussian similarity function is commonly used for this task. In figure 10 a set of white pixels and the corresponding adjacency matrix based on Gaussian similarities are shown. Large entries of the matrix are displayed bright, small entries dark. The two bright diagonal blocks represent the high similarities of vertices inside the two clusters and the two nearly black off-diagonal blocks correspond to the low similarities between vertices of different clusters.

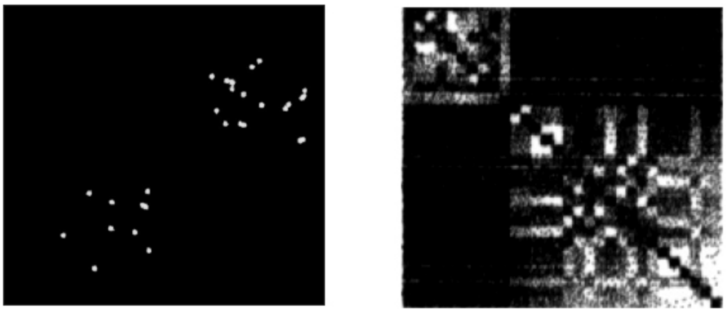


Figure 10: A set of white pixels (left) and the corresponding adjacency matrix (right).

The value of σ in the similarity function has to be set according to the total range of the occuring distances. In case of object extraction it is fixed at some percentage (typically between 5 and 10 percent) of the image size. Figure 11 visualizes the shape of the Gaussian similarity function with $\sigma = 5$ for distances up to 128, which corresponds to the width and heigth of the artificial test images used in this chapter.

In the example shown here, vertices, which have a distance greater than 15, are considered to be nearly completely dissimilar, whereas vertices less than 5 pixels apart, exhibit a similarity value higher than 0.5.

Solving the generalized eigenvalue problem can be very time consuming. Due to the fact that the Laplacian matrix $(D - W)$ is sparse in most cases, only the top few eigenvectors are needed for graph partitioning and the precision required for the eigenvectors is low (often only the sign is required) specially optimized eigensolvers can be applied (like i.e the Lanczos method) in order to speed up the computation time.

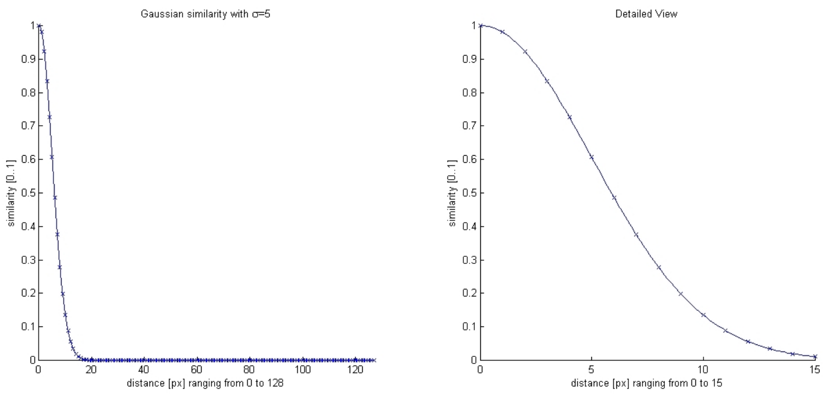
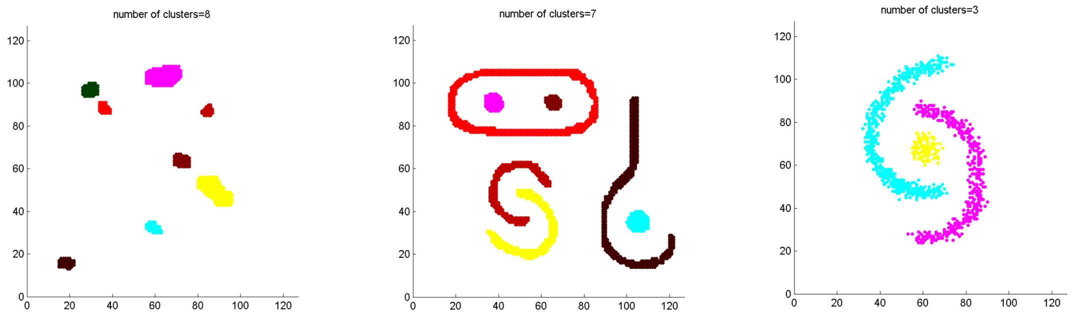


Figure 11: The Gaussian similarity function ($\sigma = 5$) with distances ranging from 0 to 128 (left) and from 0 to 15 (right).

To partite the graph into two pieces, the real valued eigenvector solution has to be thresholded. Basically there exist three methods for finding an appropriate threshold value: one can take 0 or the median value or can search for the threshold, which minimizes the Ncut.

Finally after the graph is bi-partitioned, the algorithm recursively splits the two parts again. In order to stop the recursion at some point, a second threshold value has to be introduced. If the Ncut becomes higher than this threshold, no more further partitioning is performed (and of course also not when there is only a single point left).

The results obtained when clustering the three test images with the normalized cut algorithm are shown in figure 12. In all cases the objects are extracted correctly. Here the σ value in the Gaussian similarity function was set to 2, the threshold for making the eigenvector discrete, was determined by finding the value, which minimizes the Ncut.

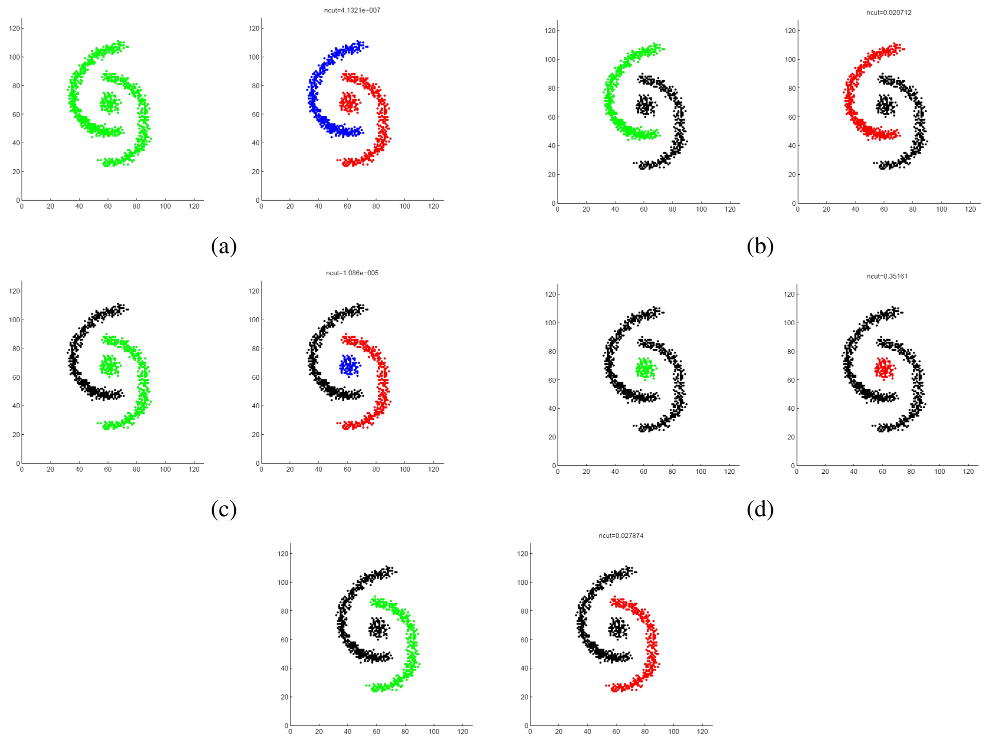


(a) (b) (c)

Figure 12: Normalized cut results of the a) first, b) second, (c) third artificial dataset.

As stopping criterion for the recursion a threshold for the Ncut was fixed at 0.01. In figure 13 the recursive nature of the computation is shown. The green areas show the pixels examined at the current stage, the blue and red areas stand for the partitions derived from the algorithm. In the case of the third test image the partitioning has been carried out in five steps:

1. All points are considered (all pixels are green in the left part of figure 13a) and a cut is made between the upper left arc and the rest. The Ncut value is approximately zero.
2. Only the upper left arc is considered and no further partitioning is performed, because it would lead to a minimum Ncut value of 0.02, which is higher than the threshold value set by the user (figure 13b).
3. The disc in the center and the lower right arc are processed and partitioned in two parts (figure 13c). The Ncut value is again nearly zero.
4. No more cutting of the center disc is performed, due to a minimum Ncut value of 0.035 (figure 13d).
5. Also the lower right arc is not further divided because of an NCut value approximately 0.028 (figure 13e).



(a) (b)

(c) (d)

(e)

Figure 13: Visualized recursion steps a) to e) performed by normalized cut algorithm on the third artificial dataset.

Another way to apply spectral clustering is using all k smallest eigenvectors without any recursion in order to cut the graph into multiple (in this case k) pieces in one step. Here the following procedure has to be carried out [7]:

1. Set up a weighted graph $G = \{V, E\}$ and set the weight of an edge connecting two vertices to be a measure of similarity between the two vertices.
2. Solve $(D - W)y = \lambda Dy$ for eigenvectors with the k smallest eigenvalues.
3. Build up a matrix U containing the k smallest eigenvectors.
4. Treat every row of U as a point z_i in a k-dimensional space.
5. Perform the k-means algorithm on the points z_i in order to find the clusters.

This algorithm changes the representation of the original data points into k-dimensional points z_i to enhance the cluster properties in the data. In this new representation the clusters are quite obvious and can easily been detected by a simple k-means algorithm.

Two problems arise, when using this approach for object extraction: the number of clusters (here k) has to be known in advance and, due to errors caused by the binarization of the real-valued eigenvectors, the method as described here is less robust than the recursive bi-partitioning. A more sophisticated algorithm for performing the multi-cut, starting with an oversegmentation followed by subsequent pruning, is described in [6].

4 Conclusion and Outlook

The two clustering approaches presented in this paper seem to be very useful for the extraction of objects from binary images as they are able to detect arbitrary and non-connected shapes. However they are computationally expensive and are therefore maybe not feasible for time-critical inspection applications. In the near future, further tests will be carried out in order to evaluate their performance on real images.

References

[1] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* New Jersey, USA: Prentice Hall, Inc., 2002.

[2] K.-H. Anders, “A hierarchical graph-clustering approach to find groups of objects,” in *Technical Paper, ICA Commission on Map Generalization, Fifth Workshop on Progress in Automated Map Generalization, IGN, Paris, April 28-30, 2003*, Institute of Cartography and Geoinformatics, University of Hannover, Germany, April 2003.

[3] G. Papari and N. Petkov, “Algorithm that mimics human perceptual grouping of dot patterns,” in *Proc. First Int. Symp. on Brain, Vision and Artificial Intelligence BVAI 2005, Naples, October 19-21, 2005, Lecture Notes in Computer Science*, M. D. G. et al., Ed., vol. 3704. Springer-Verlag Berlin Heidelberg, 2005, pp. 497–506.

[4] G. Karypis, E.-H. S. Han, and V. Kumar, “Chameleon: Hierarchical clustering using dynamic modeling,” *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[5] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data." in *SDM*, D. Barbará and C. Kamath, Eds. SIAM, 2003.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[7] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, December 2007.

Image and Texture Classification based on Image Metrics

Werner Groißböck

Department of Knowledge-Based Mathematical Systems
Fuzzy Logic Laboratorium Linz-Hagenberg
Johannes Kepler University Linz, A-4040 Linz, Austria
werner.groissboeck@jku.at

1 Introduction

If machine learning methods are applied to data-sets of images or textures or even pieces of music, then the machine learning methods can not be applied directly. Instead, usually features are calculated for the visual data. If the used features are inappropriate, then the used machine learning method has no chance. As an alternative to features, image metrics can be used. Many classification methods, for example the k-nearest-neighbor-classification method do not need anything more than a metric. And sometimes it is much easier to calculate, if two images are similar or not similar, than to find reasonable features. This is especially true, if the images that we are dealing with, are textures, because in textures, no cars or animals have to be detected. Detecting cars or animals on images is a very challenging task, whereas calculating how similar two textures are, is much easier.

There two different approaches, how image metrics can be calculated:

1. The first possibility is to use algorithms that compare two images directly. For example, by comparing the histograms of the two images.
2. The second possibility is to calculate features for the images, and then to calculate metrics based on these features. Here we concentrate on the second possibility, because we want to have methods that are similar to what the brain does. Therefore we use a method, simulating the visual cortex, for calculating the features (see [4]).

We want to find out, which of the standard metrics lead to good texture classification rates. How can the qualities of different metrics be compared?

As we are dealing with textures, the answer is quite easy: When a texture is cut into four or nine pieces, then the resulting textures are very similar. So if the metric of two such textures is calculated, then the metric should have a small value. On the other hand, if we have two pictures that are very different, then the metric should be quite high. The more these two circumstances are fulfilled, the better is the quality of the metric.

The well known 'Brodatz' Dataset consists of 92 texture images. These images are cut into nine parts, and the classification task is to find out for each part, to which group it belongs.

References

1. Hastie, T., Tibshirani, R., and Friedman J. , "The Elements of Statistical Learning: Data Mining, Inference, and Prediction.", Springer Berlin, 2001.
2. J. R. Koza, "Genetic Programming", The MIT Press, Cambridge, Massachusetts, 1992.
3. Y. Rubner 'Perceptual Metrics for Image Database Navigation', Dissertation, Stanford University, 1999.
4. T. Serre, A. Oliva, T. Poggio, "A feedforward architecture accounts for rapid categorization", PNAS published online Apr 2, 2007.
5. H. Zhang, A. C. Berg, M. Maire, J. Malik, 'SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition', Computer Science Division, EECS Department, Univ. of California, Berkeley, CA 94720, 2006.
6. J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, 'Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study', Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop ,2006.

Choosing Regularization Parameters in an Optimal Way without Knowing the Noise Level – An Analysis of Quasi-Optimality –

Frank Bauer[1], **Stefan Kindermann**[2] **and Markus Reiß**[3]

[1] Fuzzy Logic Laboratorium
Linz-Hagenberg
Johannes Kepler
University Linz
Softwarepark 21
4232 Hagenberg
frank.bauer@jku.at

[2] Johann Radon Institute for
Computational and Applied
Mathematics (RICAM)
Austrian Academy
of Sciences
Altenbergerstrae 69
A-4040 Linz
stefan.kindermann@oeaw.ac.at

[3] Institute for
Applied Mathematics
University Heidelberg
Im Neuenheimer Feld 294
69120 Heidelberg
Germany
reiss@statlab.uni-heidelberg.de

Key Words: *Inverse Problems, Parameter Choice, Independence of Noise Level, Quasi Optimality*

ABSTRACT

Ill-conditioned or ill-posed problems occur in a big variety of applications ranging from image deconvolution to mathematical finance [EHN96]. In order to solve these one has to introduce a stabilization and a regularization parameter which controls this stabilization.

A short article by Bakushinskii in 1984 [Bak84] showed that in inverse problems with deterministic noise it is impossible to choose an optimal regularization parameter without knowing the noise level. The construction of particular examples is comparably easy. However, it could still be the case that these counterexamples are so rare that they do not affect the day to day business [Bau07].

Therefore the situation changes when one uses e.g. stochastic instead of deterministic noise, or imposes some non-degeneracy conditions on the noise. Similarly, depending on the regularization method minor conditions (in a Bayes case [KS05] similar to standard Hölder source conditions) on the solution need to be imposed. The smoothness of the solution and the error level stays unknown. For Tikhonov regularization one can use classical source conditions in certain cases.

Then one can show that a discretized version of quasi-optimality (which just needs regularized solutions as input) performs in an optimal way [BR07, BK07]. Numerical examples will be shown.

References

[Bak84] BAKUSHINSKY, A., *Remarks on choosing a regularization parameter using the quasioptimality and ratio criterion*, USSR Comput. Maths. Math. Phys. **24** (1984), no. 4, 181–182.

[Bau07] BAUER, F., *Some Considerations Concerning Regularization and Parameter Choice Algorithms*, Inverse Problems **23** (2007), 837 – 858.

[BK07] BAUER, F., AND KINDERMANN, S., *The Quasi-Optimality Criterion for Classical Inverse Problems*, Inverse Problems ((submitted 2007)).

[BR07] BAUER, F., AND REISS, M., *Regularisation independent of the noise level: an analysis of quasi-optimality*, Inverse Problems ((submitted 2007)).

[EHN96] ENGL, H., HANKE, M., AND NEUBAUER, A., *Regularization of Inverse Problems*, Kluwer Academic Publisher, Dordrecht, Boston, London, 1996.

[KS05] KAIPIO, J., AND SOMERSALO, E., *Statistical and computational inverse problems.*, Springer, New York, 2005 (English).

Web-Geometric Aspects of Triangular Norms

Milan PETRÍK
Center for Machine Perception
Department of Cybernetics
Czech Technical University
Karlovo náměstí 13
12135 Prague 2, Czech Republic

Peter SARKOCI
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University
A-4040 Linz, Austria

Abstract. Let us recall that a *triangular norm* (also *t-norm* for short) is any monotone commutative and associative binary operation on the *unit interval* $[0, 1]$ with neutral element 1. Being defined on a subset of the reals, it can be viewed also as a real function of two variables or equivalently as a surface in $\mathbb{R}^3$. Up to associativity, all the defining properties of the t-norms have transparent geometric counterparts which are easily visible on the corresponding surfaces. The associativity does not.

In our approach we adopt a level set view of triangular norms and we utilize tools and ideas from *web geometry.* Note that web geometry is mostly considered as a subbranch of differential geometry and deals with systems (mostly tripples) of foliations on a differential manifold. We adapt a simple web-geometric concept of the *Reidmeister closure condition* in such a way that it makes sense for more general structures than just foliations. Then we discuss relationships of the modified concept to the t-norms and we show that for many t-norms the Reidmeister closure condition is a geometric counterpart of associativity.

Finally we present several applications of the new approach. First of all, we provide a solution to an open problem about *convex combinations* of continuous nilpotent t-norms. In particular, we show that none nontrivial convex combination of such t-norms is again a t-norm. Further we provide several construction methods based on the Reidmeister closure condition and we show that in several cases we are able to generalize older construction methods. Finaly we give some ideas how to construct t-norms on more general lattice-ordered domains; here the main motivation are possible constructions of MTL-algebras.

3D- Micro-Material Inspection by Interferometic and OCT Techniques

B. Heise[*, **]; M. Wurm[**]; D. Stifter[**]

[*] *Department of Knowlegde-based Mathematical Systems, FLLL, Johannes Kepler University Linz, Austria*

[**] *Department of Contactless Sensor Technology, UAR, Linz, Austria*

February 28, 2008

Abstract

3D- micro-material inspection with respect to defect or deformation detection covers a wide application field in machine vision tasks. Optical testing methods are preferable techniques as they do not introduce any remarkable damages in the material. Depending on the kind of material and dimensions of the sample the wavelength and the appropriate imaging technique has to be chosen. In the micro- and sub-micrometer range of testing typically interferometric measurement techniques using UV, VIS or NIR light spectrum are well established. Image processing tasks for interferometric applications include e.g. demodulation, unwrapping, denoising, orientation estimation, topological profile reconstruction or recognition of artifacts and displacements in the (sub) micrometer range.

Optical Coherence Tomography (OCT) was developed for achieving additionally an inside view beyond the surface of the sample at a depth range of about 1 mm. In recent years OCT, originally introduced for the imaging of biological tissues [1], was established more and more in fields outside from biomedical applications [2]. The potential of OCT as a low coherence interferometric technique for contact-free and non-destructive testing was recognized meanwhile in material research. Due to improvements in increased resolution (Ultrahigh-resolution OCT) [3] or in higher acquisition speed and sensitivity (Fourier Domain OCT, FD-OCT) technical relevant processes can be considered.

Spectral Domain OCT (SD-OCT), as a variant of FD-OCT, has the important property that no movable parts are needed for a depth scan. The spectrometer as the crucial part replaces the movement of the mirror in the reference arm. Due to this, the acquisition speed only depends on the read out rate of the line detector. By inverse Fourier transform of the spectrogram the depth profile can be determined. The individual depth profiles can be combined to form cross-sectional images or 3D profiles of the investigated materials.

Due to its sensitivity and speed SD-OCT has a still developing potential for e.g. defect and crack recognition in ceramics or glasses, structure characterization in fiber composite materials, foams or multilayer foils.

Acknowledgments

The images were kindly provided by the UAR Department of Contactless Sensor Technology, Linz and Christian Maurer, Medical University, Innsbruck. This work has been financially supported by the Austrian Science Fund FWF (Project L126-N08 and Project P19751-N20).

References

[1] Huang, D. et al., *Optical coherence tomography*, Science, **254**(1991), p.1178-81

[2] Stifter, D., *Beyond biomedicine: a review of alternative applications and developments for optical coherence tomography*, Appl.Phys.B, **88**(2007), p.337-357

[3] Wiesauer, K. et al., *En face scanning optical coherence tomography with ultra-high resolution for material investigation*, Opt.Express, **13**(2005), p.1015-1024