s c c h
software competence center
hagenberg

FUZZY.LOGIC.LAB.LINZ

Abstracts of the FLLL/SCCH Master and PhD Seminar

Rittersaal, Schloss Hagenberg

February 27, 2006

Software Competence Center Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz-Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

Session 1 (Chair: Ulrich Bodenhofer) 14:00–15:30

14:00 Edwin Lughofer:
Process Safety Enhancements for Data-Driven Evolving Fuzzy Models

14:30 Leila Muresan, Erich Peter Klement:
Denoising Microscopy Image Sequences with Fine Structure Preservation

15:00 Bettina Heise, Heimo Wolinski, Leila Muresan, Sepp Dieter Kohlwein, Erich Peter Klement:
DIC-Image Restoration with Integrational Methods and Phase Filtering for automated Analysis of Yeast Cell Scans

15:30 Coffee Break

Session 2 (Chair: Susanne Saminger) 15:45–16:45

15:45 Stefan Larndorfer, Peter Haslinger:
Support Vector Machines – A Case Study in Computer Vision

16:15 Ulrich Bodenhofer:
Lexicographic Composition of Fuzzy Orderings

Process Safety Enhancements for Data-Driven Evolving Fuzzy Models

Edwin Lughofer
Fuzzy Logic Laboratorium Linz-Hagenberg
Johannes Kepler University Linz
A-4040 Linz, Austria
E-mail: edwin.lughofer@jku.at

***Abstract*— In this paper several improvements towards a safer processing of incremental learning techniques for Takagi-Sugeno fuzzy models are demonstrated (applicable also partially to neuro-fuzzy systems or radial basis function networks). The first group of improvements include stability issues for making the evolving scheme more robust against faults, steady state situations and extrapolation occurrence. In the case of steady states or constant system behaviors a concept of overcoming the so-called 'unlearning' effect is proposed by which the forgetting of previously learned relationships can be prevented. The concepts regarding fault omittance are demonstrated, as usually faults in the training data lead to problems in learning underlying dependencies. An improvement of extrapolation behavior in the case of fuzzy models when using fuzzy sets with infinite support is also highlighted. A discussion on the convergence of the incremental learning scheme to the optimum in the least squares sense is included as well. The second group deals with interpretability and quality aspects of the models obtained during the evolving process. An online strategy obtaining better interpretable models is presented. This strategy is feasible for online monitoring tasks, as it can be applied after each incremental learning step, that is without using prior data. Interpretability is important, whenever the model itself or the model decisions should be linguistically understandable. The quality aspects include an online calculation of local error bars for Takagi-Sugeno fuzzy models, which can be seen as a kind of confidence intervals. In this sense, the error bars can be exploited in order to give feedback to the operator, regarding fuzzy model reliability and prediction quality. Evaluation results based on experimental results are included, showing clearly the impact on the improvement of robustness of the learning procedure.**

***Keywords*—Takagi-Sugeno fuzzy models, incremental learning, evolving process, unlearning effect, convergence, extrapolation, faults, online interpretability, local error bars**

I. Introduction

Nowadays automatic adaptation techniques and evolving schemes for data-driven models become more and more an essential point in industrial processes. This is because real-time processes should be identified, which possess a dynamically changing behavior over time (also called time-variant processes). Furthermore, some dependencies between certain system variables may change smoothly or abruptly whenever new operating conditions or new system states are triggered. Hence, it is indispensable to adjust the already trained data-driven models to this newly arising situations. For online identification tasks, this requires an adaptation of some model parameters in form of incremental learning steps with new data. This is because a complete rebuilding of the models from time to time with all so far recorded measurements would yield an unacceptable computational effort. Other requirements to incremental learning includes refinement of already existing knowledge-based models with data, preventing a virtual memory overload in case of huge databases (the data needs to be processed blockwise then) and an automatic improvement of the accuracy and generalization capability of models initially trained on only a handful of data. Quite often a complete evolving mechanism is demanded, as the inner structure of a model needs to be changed as well. In these sense, automatic self-evolving data-driven models can be seen as a step towards computational intelligence [1]. In order to meet these requirements algorithms for an incremental and evolving learning scheme of fuzzy models are proposed in literature, such as *DENFIS* [2], *eTS* [3] or *FLEXFIS* [4].

In this paper several approaches for significantly improving the process safety when using data-driven evolving fuzzy systems for online operation modes are demonstrated. In this sense, some aspects not treated in the approaches mentioned above may serve as potential extensions in order to obtain more accurate, more transparent and more stable fuzzy models. This can be essential in order to yield better or more correct decisions in online prediction, fault detection or control tasks based on these models. Moreover, some of the approaches could be also applicable for other related types of models such as neuro-fuzzy systems or radial basis function networks. In fact, for the latter it is demonstrated in [5] that they are similar to Takagi-Sugeno fuzzy systems. In Section II maintaining the stability of the models during online training is the central point to be discussed, in Section III some techniques for omitting faulty recorded data to be incorporated into the online trained models are described. Section IV describes a possibility how to prevent a bad extrapolation behavior, especially in the case of fuzzy sets with infinite support. Section V deals with an online improvement of interpretability and transparency of the models, whereas Section VI treats the quality aspect by demonstrating how to process local confidence statements about fuzzy models in incremental manner.

First of all, the definition of a Takagi-Sugeno-Kang fuzzy system is given as referenced in the sections below. A Takagi-Sugeno-Kang fuzzy system with multiple input variables

$(x_1, ..., x_p)$ and a single output variable y can be generally defined in the following way:

$$\hat{f}(\vec{x}) = \hat{y} = \sum_{i=1}^{C} l_i \Psi_i(\vec{x}) \quad (1)$$

where

$$\Psi_i(\vec{x}) = \frac{\mu_i(\vec{x})}{\sum_{j=1}^{C} \mu_j(\vec{x})} \quad (2)$$

are called *normalized membership functions*, which normalize the degrees of rule fulfillment by using a t-norm, i.e.

$$\mu_i(\vec{x}) = T_{j=1}^{p} \mu_{ij}(x_j) \quad (3)$$

where x_j is the j-th component in current data vector, hence reflecting the value of the j-th variable and μ_{ij} the membership degree of x_j to the fuzzy set describing the j-th premise part of the i-th rule. The symbol T denotes a t-norm in general. The l_i's are the so-called consequent functions of the C rules and are defined by multi-dimensional polynomials to an arbitrary but fixed degree with unknown coefficients. In the case of Takagi-Sugeno fuzzy systems as a specific form of Takagi-Sugeno-Kang fuzzy systems the polynomials are reduced to hyper-planes in the p dimensional space:

$$l_i = w_{i0} + w_{i1}x_1 + w_{i2}x_2 + ... + w_{ip}x_p \quad (4)$$

Note that two types of parameters appear in the Takagi-Sugeno(-Kang) fuzzy systems: nonlinear ones in the antecedent parts (fuzzy sets) of the rules and linear ones in the rule consequent functions (4). Both as well as the rule structure need to be adaptively changed in incremental as well as evolving manner in order to cope with flexible online learning demands.

II. Improving the Stability of the Incremental Learning Phase

A. Overcoming the Unlearning Effect

When performing incremental learning steps of linear rule consequent parameters in Takagi-Sugeno(-Kang) fuzzy models [6] [4] (both compared in [7]) respectively neural-fuzzy inference systems [2] with the help of *recursive weighted least squares* [8] it may happen that the learning process get instable over while. This would be the case if the newly loaded data points or recently recorded measurements stay (almost) constant for a certain period. To demonstrate this on a simple practical example see Figure 1, where the first 200 data samples (light dots) are indeed well distributed over the whole input space, but the next 1300 samples (concentrated in one big dark dot) are concentrated around the specific point $(1400, 2)$.

Note: the x-axis denotes rotation speed, where the y-axis denotes the pressure of the oil. Obviously, the car motor was steered with a constant rotation speed, the slight noise variance can be explained by sensor inaccuracies during recording. This example represents a situation, whenever a process is in steady state staying at one specific operating condition for a long time. When doing an initial learning phase with the first 200 points (no matter if in incremental or batch mode) and performing an adaptation of the fuzzy model with the later 1300 points (in steady state), an undesired 'unlearning' effect of already learned relationships outside this small constant region occurs. This can be recognized in the left image in Figure 1, where right to the small constant region the shape of the adapted model (dotted line) tends to be different to the shape of the original one (solid line), even though no new measurements were recorded for that area. This is even worse, when adapting with *recursive weighted least squares* in the case of incorporating a forgetting factor in order to track time dependent dynamic relationships, see the left images in Figure 2. The lower row represents a shift in the trajectory (dark dots) due to a dynamic evolution of the process, for which a forgetting of the older learned relationship on the right part is indispensable (otherwise the model would be inbetween the light and dark dots due to least squares estimation). The older learned relationships are unlearned in the case when all rules are adapted with each sample represented as dark dots. The reason for these effects is that the parameters of all linear consequent functions are adapted for each incoming data sample, no matter which firing degree the rules have. This means that the consequent parameters are forced to minimize the error at $(1400, 2)$. This causes the linear consequent hyperplanes to go through the point $(1400, 2)$, see upper left image of Figure 2: obviously, the consequent functions (shown as straight lines) all point to $(1400, 2)$.

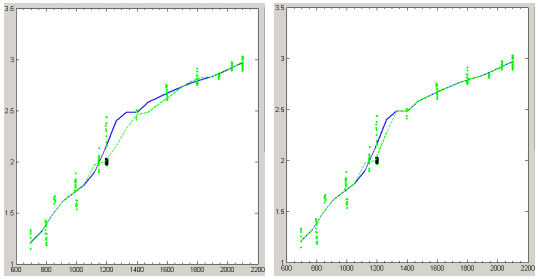

Fig. 1. Left: Slight unlearning effect when adapting all rule consequents' for each sample at $(1400, 2)$ (dark dot) without forgetting; right: the unlearning effect prevented when only adapting the significantly firing rules

A promising strategy to circumvent this problem lies in the adaptation only of those parameters which corresponds to significantly active rules, i.e. rules which possess normalized membership function values $\Psi(\vec{x}_{act})$ higher than a certain threshold near 0. This guarantees that rules which represent areas not lying near constant process state remain unchanged. This is demonstrated in the right images in Figures 1 and 2, where a threshold of 0.1 was used for a reasonable approximation behavior. In fact, setting this threshold is a crucial point for a well performing adaptation: if setting it too close to 0, the 'unlearning' effect remains, if setting it too far away from 0, too less new input states are incorporated and the fuzzy system is not able to represent a reliable prediction model. Especially for the incremental learning variant *FLEXFIS* [4] it could be shown empirically with a lot of various high-dimensional data

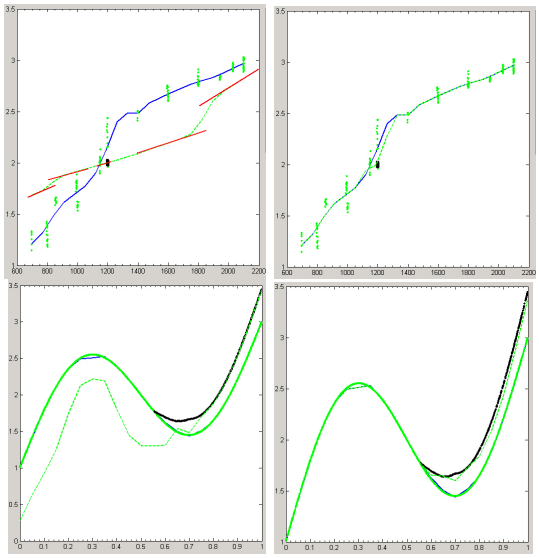

Fig. 2. Left: Significant unlearning effect when adapting all rule consequents' for each sample in steady state (dark dot) with forgetting; right: the unlearning effect prevented when only adapting the significantly firing rules; the lower row represents a shift in the trajectory (dark dots) due to a dynamic evolution of the process; in all figures the dotted light (green) lines represent the updated models

sets, that a threshold of 0.1 did not spoil the convergence to optimality (see Section II-B) and therefore did not worsen the prediction accuracy of the fuzzy models significantly.

B. A Note on Convergence to Optimality

With convergence to optimality it is meant that an optimal or even near optimal parameter solution in the (weighted) least squares sense should be achieved throughout the learning phase. Otherwise, the incremental learning approach becomes quite heuristic. When exploiting the *recursive least squares* respectively the *recursive weighted least squares* approach for incremental learning of linear consequent parameters, the convergence to optimality in the least squares sense is guaranteed as long as no adaptation of the nonlinear rules' premise parts takes place. This is a direct consequence of the convergence of RLS when choosing appropriate starting values for the parameter vector (i.e. the zero vector) and the inverse Hesse matrix (i.e. α times identity matrix) [9]. However, in most evolving fuzzy systems approaches the premise parts are adapted as well in order to cope with new operating conditions and system states. If using the *recursive weighted least squares* for locally training the rules' consequents, the adjoining of new rules for more flexibility of the fuzzy system does not violate the convergence to the weighted least squares optimum. Whenever fuzzy sets are shifted or newly set within a premise part of one or more rules, it could disturb the convergence of the consequent parameters, depending on the degree of the shifts. In the case of exploiting a modified (incremental) version of vector quantization for premise part adaptation (as carried out in *FLEXFIS*) it could be shown in [10] that this disturbance has an intensity, such that a convergence to a

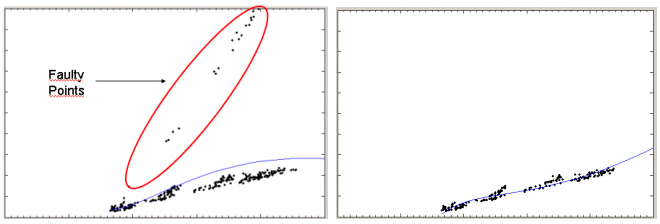

Fig. 3. Left: approximation obtained by incorporating faulty points into the training algorithm (solid line), right: faulty points omitted

near optimality which is close to the real optimality can be achieved. Furthermore, it could be evaluated empirically that the more data samples an online data buffer contains, the closer the optimum can be reached.

It should be noticed that in fact the least squares measure on the training set is not the best performing one with respect to approximizing the true generalized prediction error on unseen data. But, it triggers not too complicated optimization problems to be solved, especially for incremental learning techniques. Moreover, for linear parameters even an analytical solution exists. When adding some penalty terms incorporating the model complexity [11] or extending it by the so-called *in-sample error* [12], the optimization problem immediately becomes quite complex. Therefore, the least squares optimization function is a quite commonly used function which is going to be optimized when learning from data, especially in the case for regression respectively function approximation tasks. This is also true for the most common evolving variants of neural networks, neuro-fuzzy models and fuzzy systems. However, this should not prevent us to develop incremental learning techniques for these kind of models which act on optimization functions giving a better representation of the generalized prediction error. This would be a promising starting point for future improvements of evolving mechanisms.

III. Towards Outlier and Fault Omittance

Opposed to noise in the data, which usually occurs whenever sensors are used for recording measurements (sensor inaccuracies white noise), process or system faults have a much more intense appearance in the measurement. Faults can arise for instance due to interface defects, sensor overheatings or broken parts at a test object (e.g. broken pipes) and is usually characterized by an unpermitted deviation of at least one variable of the system from acceptable or standard behavior. Hence it is obvious that the incorporation of faulty measurement into the offline as well as online data-driven model training process can spoil the model in a way such that it leads to an incorrect approximation of the real underlying dependency, see Figure 3: the left image shows a wrongly approximated relationship (solid line) when faulty data points are incorporated into the training process, the right image shows a correct approximation (again solid line) achieved on the basis of filtered data, i.e. data where the faults were filtered in a preliminary step. In the batch modelling (offline) case several strategies exist for outlier and fault omittance,

such as *robustfit* [13], pre-processing with a *Mahalanobis-distance* based [14] or a *principal component* based [15] classifier. They all underlie the assumption that data samples representing faulty situations can be clearly distinguished from data samples representing non-faulty ones (usually with respect to their density and distribution), when looking at the complete amount of available data at once.

In the case of online learning, the fault problem is more complicated. This is because it is very hard to differ between a new operating condition and a fault, whenever measurements are newly recorded or loaded sample-wise or block-wise. In *FLEXFIS* [10] a strategy for procrastination of rule evolution is demonstrated: whenever a new sample comes in which lies far away from the previously estimated clusters (representing local areas), a new cluster is born immediately, but not a new rule and hence no new fuzzy sets. Is is waited for more data points appearing in the same region until a new rule is set. This is based on the assumption that the more data points in a new region occur the more likely it gets that they denote a new operating condition and no fault. With this strategy the incorporation of isolated outliers into the fuzzy model can be prevented. In *eTS* [3] this outlier omittance is done implicitly, as long as no new rule is set, when a data point has not enough potential, i.e. is not surrounded by a more or less dense data cloud. The precarious thing is that the assumption mentioned above is not always true, especially when a systematic error within the system occurs as for example in the case as visualized in Figure 3. In such cases the following possible strategies may help:

- The newly gathered local area is compared with respect to its range of influence, density and number of data points with the other already obtained local areas so far. If completely different, a fault is more likely than a new operating condition, as various operating conditions usually possess more similar ranges, densities etc. In this case no new rule (or neuron) would be incorporated into the (neuro)-fuzzy model, as long as the newly gathered area does not fit to the other ones.
- Filtering of dynamic measurements with signal analysis methods in intelligent sensors, which also effects a filtering or 'cleaning' of stationary measurements which are elicited through averaging of dynamic ones: this is quite promising, as usually faulty points cause a more distinct anormal behavior in the one-dimensional signal curves as new operating conditions.
- The exploitation of analytical and knowledge-based models in a complete fault detection framework [16] for delivering a fault detection statement for any loaded or online recorded data point. This statement can be taken as input for the incremental learning method and reacted upon it such that points classified as faulty are not incorporated into the adaptation process.

IV. Improving the Extrapolation Behavior

When doing adaptation of premise parts in Takagi-Sugeno fuzzy models with the usage of fuzzy sets with infinite support

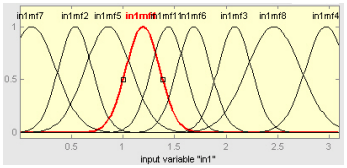

Fig. 4. A fuzzy partition where inner sets are broader than outer ones, especially compare the two right most sets

(especially with the most common Gaussian sets) it usually happens that not all fuzzy sets possess the same width. This is due to the nature of online data streams or measurements, where local areas (which are projected onto the axes to form the fuzzy sets) reflect different system behaviors with different ranges. This can lead to nasty extrapolation effects, when the outer most set is not the broadest one, as a kind of reactivation of an inner membership function in the extrapolation region is caused. This is due to the nature of the Gaussian fuzzy sets which tend faster to zero if the width is tinier For demonstrating this on the basis of an example, consider the two right most fuzzy sets in the partition visualized in the right image in Figure 4, i.e. the sets described by

$$e^{-\frac{1}{2}\left(\frac{x-2.46}{0.361}\right)^2} \qquad e^{-\frac{1}{2}\left(\frac{x-2.97}{0.2}\right)^2}$$

so Gaussian fuzzy sets with center=2.46 and width=0.361 respectively center=2.97 and a smaller width of 0.2. Then the membership degree of the first fuzzy set gets obviously higher than the second one if $\left(\frac{x-2.46}{0.361}\right) \leq \left(\frac{x-2.97}{0.2}\right)$. After resolution it turns out that this is the case when x greater or equal 3.6, so an x lying in the 'extrapolation region' of the fuzzy system. So concluding, inner sets can be reactivated, which is a precarious thing, as an outer most set usually stands for the most confidential information in the case of extrapolation as it is the nearest one. In order to ensure a safe extrapolation the outer most sets can be simply kept at the membership degree one for the second half of the Gaussian fuzzy set. In this sense, the two outer most fuzzy sets for input dimension j are described in the following way:

$$\mu_{(right)j} = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-c_{(right)j}}{\sigma_{(right)j}}\right)^2} & x \leq c_{(right)j} \\ 1 & otherwise \end{cases}$$

for the right most fuzzy set and

$$\mu_{(left)j} = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-c_{(left)j}}{\sigma_{(left)j}}\right)^2} & x \geq c_{(left)j} \\ 1 & otherwise \end{cases}$$

for the left most fuzzy set, where $c_{(right)j}$ resp. $c_{(left)j}$ denotes the center of the right resp. left most fuzzy set and $\sigma_{(right)j}$ resp. $\sigma_{(left)j}$ the width of these sets. This triggers automatically a safe zero order extrapolation as stated in [17]. In order to overcome a partially defined fuzzy set with two parts (which is not always easy to handle), the right resp. left most set can be transferred to one-fold sigmoid fuzzy sets, defined by:

$$\mu_{ij}(x_j) = \frac{1}{1+e^{(-a_{ij}(x_j-b_{ij}))}} \tag{5}$$

This can be achieved by the following transfer formulas for the parameters a and b:

$$a_{(right)j} = -\frac{\ln 0.01}{\sigma_{(right)j}}$$
$$b_{(right)j} = c_{(right)j} - \sigma_{(right)j} \quad (6)$$

for the right most fuzzy set and

$$a_{(left)j} = \frac{\ln 0.01}{\sigma_{(left)j}}$$
$$b_{(left)j} = c_{(left)j} + \sigma_{(left)j} \quad (7)$$

for the left most fuzzy set, substituting a_{ij} and b_{ij} in 5. This kind of transformation ensures that the position of inflection point of the Gaussian fuzzy set is equal to those of the sigmoid function (formula for b in both equations) and that at the center position of the Gaussian set the value of the sigmoid function is almost 1 (namely 0.99). Note, that in the case of fuzzy sets with finite support (for instance triangular shaped sets) the reactivation of inner sets can not occur, but the problem of extrapolation still remains, if the outer ones are not extended appropriately (as they are not defined on new regions). Hence, also for these kind of sets the above transformation strategy is a quite promising approach for safer predictions.

V. A Note on (Online) Interpretability

To deliver interpretable models is usually an important issue, especially when the model should be readable and understandable for the experts in order to check the plausibility of the model. This guarantees a safer processing further on, as then the model becomes a grey box model (opposed to a pure black box model, where nothing can be interpreted inside). Another requirement is that the model decisions are understandable and can therefore be traced back: for instance consider a fault monitoring system, where the reason for a fault should be found. With the help of a transparent fuzzy model, a couple of linguistic rules can be extracted which were violated. Based on these violated rules a reason can be found much more easier. Whenever fuzzy models are trained from data, the interpretability usually suffers. This is because fuzzy sets and rules are flexibly adapted, changed and projected from the high-dimensional space to the one-dimensional one. This mostly leads to strongly overlapping fuzzy sets and redundant rules, as two rules can move together over while, especially in the incremental learning phase. Furthermore, an optimistic parameter setting (which has to be carried out in advance) may lead to too many fuzzy sets and rules for the actual process, such that the fuzzy model is quite difficult to understand.

An important issue is that the rule consequents are interpretable. For Takagi-Sugeno fuzzy systems this means that the linear hyperplanes should snuggle along the surface. This yields a good interpretation in the field of control and dynamic systems, as from the steepness of the hyperplanes the current behavior of the dynamic (control) systems can be gathered (e.g. when all linear parameters except the intercept are statistically not different from 0 someone can conclude that the system is in constant state). The snuggling along the surface can be achieved when using *recursive weighted least squares* (for local learning), but not with usual *recursive least squares* (for global learning), see Figure 5. There, the function $\sin(x^2) + x + \epsilon_1$, with ϵ_1 a normally distributed random noise with $\sigma = 0.1$ (left image), is approximated once with global approach (middle image), once with the local one (right image): in the case of the global approach a 'break out' of the consequent functions can be observed, whereas for the local one more or less interpretable local behaviors can be obtained directly from the consequent functions. It should be noticed that this snuggling in the case of RWLS is only guaranteed, when choosing α times identity matrix with α big for the starting value of the inverse Hesse matrix.

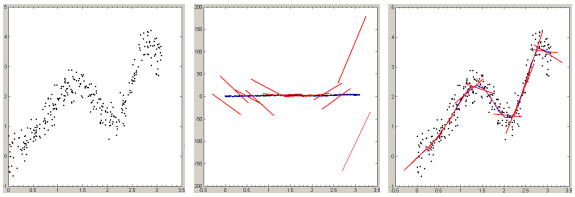

Fig. 5. A sinusoidal relationship (left) approximated with global approach (middle) and local one (right), the consequent functions as straight lines

With fuzzy set merging and rule base simplification techniques it is possible to reduce superfluous information and make the fuzzy system more transparent. Various approaches for that task exist in literature, such as *orthogonal least squares* [18], [19], *singular value decomposition and QR with column pivoting method* [20], [21] for detecting and eliminating redundant rules, merging of narrow and close fuzzy sets due to their model values [22] or fuzzy set merging and rule simplification through similarity measures [23] [24]. However, most of the proposed techniques (especially for rule reduction) require the complete data set at once and hence are not applicable in fast online identification processes. In [23] after fuzzy set merging and rule simplification the consequent parameters of the linear hyper-planes need to be re-estimated with the complete data set. This can be prevented, when exploiting the fact that when using *recursive weighted least squares* the consequent functions tend to snuggle along the surface and hence reflect local linear approximations which are very close to the real approximation (e.g. see right image in Figure 5). In this case, a merging of q rules consequent functions can be reliably carried out by:

$$w_{(new)j} = \frac{w_{1j}k_1 + ... + w_{qj}k_q}{k_1 + ... + k_q}, \qquad \forall j = 0, ..., p \quad (8)$$

where w_{ij} is the linear parameter in the consequent of the ith redundant rule with respect to the j dimension and k_i is the amount of data points belonging to the cluster corresponding to the ith redundant rule (local area). Thus, the parameters of the new rule consequent are defined as a weighted average of the consequent parameters of the redundant rules, with the weights representing the relevancy of the rules. This merging strategy can obviously be applied in online mode as it does not

require any prior training data. It should be noticed, that from the fuzzy logical point of view the merging formula (8) should be only carried out for those rules, whose consequent functions are not significantly more dissimilar than their (redundant) antecedent parts (otherwise contradicting rules are present). A dissimilarity measure between a pair of consequent functions can be obtained through a normalized angle between two linear hyperplanes (normalized with the degree of $\pi/2$ as maximal non-similar):

$$\phi = \frac{2}{\pi} \arccos\left(\left|\frac{\vec{a}^T\vec{b}}{|\vec{a}||\vec{b}|}\right|\right) \tag{9}$$

with a and b the normal vectors of the hyperplanes, i.e. $a = (w_{h1}\ w_{h2}\ ...\ w_{hp}\ -1)^T$ and $b = (w_{i1}\ w_{i2}\ ...\ w_{ip}\ -1)^T$ for the hth and ith rules. A dissimilarity measure between the antecedent parts of two rules can be obtained through one minus the minimal Jaccard index [23] between two corresp. fuzzy sets (i.e. fuzzy sets for the same input variable) in the antecedent parts of the rule pair. If the rules are conflicting i.e. the dissimilarity degree in the consequent parts is higher than in the antecedent parts, either no merging is performed or the more reliable rule is kept and the other deleted. A merging of the inverse Hesse matrices belonging to the redundant rules is required, too, in order to not disturb the convergence to the optimum significantly (see Section II-B). This is quite a difficult task and could not be solved till now. Indeed, a setting back of the inverse Hesse matrix of the new (merged) rule to αI is possible, but this would lead to a much worse convergence of the algorithm. A possibility to circumvent this problem would be the 2-layer fuzzy model building strategy: the original fuzzy model is kept as it is (so nothing is modified) and processed further through the incremental learning process, whereas the linguistically improved fuzzy model is just for the purpose of visualization to the operator.

VI. Local Error Bars for Takagi-Sugeno Fuzzy Systems

Local error bars are for the purpose of processing confidence regions surrounding the trained models. These confidence regions provide an insight, how trustful a model within a certain region is. In this sense, useful and not so useful dependencies between certain system variables can be gathered automatically from data and shown to the operators. Error bars are also quite often applied in industrial processes, when models' decisions are weighted with their confidence for online monitoring tasks. For instance in fault detection applications, a residual between a new incoming point and already trained models can be compared against the error bars. This deviation is then used in order to gain a final decision if the current point is faulty or not [16].

For the online evolving mechanism of data-driven models, the error bars need to be updated as well, as models quite often change too strongly over time (see Section I). It is well-known [25] that error bars for linear models, i.e. models possessing only linear parameters, which were obtained by solving the least squares optimization function can be calculated due to the following formula:

$$\hat{y} \pm \sqrt{diag(cov\{\hat{y}\})} \tag{10}$$

where $\hat{y}$ the estimated output value and

$$cov\{\hat{y}\} = X_{act} cov\{\hat{w}\} X_{act}^T \tag{11}$$

with

$$cov\{\hat{w}\} = \sigma^2 (X^T X)^{-1}$$

where the noise variance σ can be estimated by

$$\hat{\sigma}^2 = \frac{2\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N - deg}$$

where N the number of data points, deg the number of degrees of freedom (=the number of parameters in the linear case), X the regression (training data) matrix, X_{act} the matrix containing the actual data points for which the error bars should be calculated and w the estimated linear parameters. If X_act contains only one single point the $diag$ operation in (10) is not necessary. Note that the smaller the number of data points and/or the higher the noise in the data, the higher the noise variance gets. Obviously, when adapting linear models with *recursive (weighted) least squares* approach, the error bars can be updated as well after each adaptation step. For this, σ is updated with m new points by

$$\hat{\sigma}^2(new) = \frac{(N - deg(old))\hat{\sigma}^2(old) + \sum_{i=N}^{N+m}(y_i - \hat{y}_i)^2}{N - deg(new)} \tag{12}$$

(where $deg(new)$ the new number of parameters, if changed in the model) and the new covariance of the linear parameters, $cov\{\hat{w}\}$, calculated due to this updated σ and due to the inverse matrix $(X^T X)^{-1}$ updated by *recursive (weighted) least squares*. For new incoming points stored in X_{act} up-to-date error bars fitting to the actual updated model can be elicited then.

In case of Takagi-Sugeno fuzzy systems the usage of the error bars as in (10) can be exploited if the linear consequent functions are trained with *recursive weighted least squares* approach (for local learning and more flexibility during online evolution [6] [4]). This is underlined due to the fact that the linear hyperplanes then snuggle along the surface as also pointed out in Section V and hence yield a good approximation of the model with linear pieces. In this sense, it is reliable to calculate error bars for each rule consequent function separately and then connect them with weights to form an overall error bar for the fuzzy model. Obviously, the membership degrees of each rule are a feasible choice for a rule weight as first, the degrees are getting smaller at the borders of each rule and second the same weighting of linear consequent actions is done when inferencing through Takagi-Sugeno fuzzy systems, see (1). Thus, in case of C rules the error bar of a Takagi-Sugeno fuzzy model at a specific position $\vec{x}_{act}$ can be calculated by

$$\hat{y} \pm \frac{\mu_1(\vec{x}_{act})\sqrt{cov\{\hat{y}_1\}} + ... + \mu_C(\vec{x}_{act})\sqrt{cov\{\hat{y}_C\}}}{\mu_1(\vec{x}_{act}) + ... + \mu_C(\vec{x}_{act})} \tag{13}$$

where $\hat{y}_i$ the estimated value of the ith rule consequent function, for which cov is calculated as in (11) by using the linear parameters and inverse weighted matrix $(X_i^T Q_i X_i)^{-1}$ corresponding to the ith rule (which were both updated in incremental mode by *recursive weighted least squares* before), and $\mu_i(\vec{x}_{act})$ denotes the membership degree of the actual point to the ith rule.

VII. Conclusion and Outlook

In this paper several issues for improving the evolving mechanisms of fuzzy systems by process safety enhancements were treated. All of the aspects were discussed and it was clearly pointed out, where the enhancements lie. The feasibility of some techniques and approaches were underlined based on data examples. It should be noticed that some of these approaches, i.e. convergence to optimality (Section II-B), some parts of outlier and fault omittance (Section III) and most of the interpretability aspects discussed in Section V are already integrated in the incremental training variant for fuzzy systems *FLEXFIS* [4] and hence could be also validated for high-dimensional real-recorded measurement data in industrial projects. A special attention was given to Takagi-Sugeno fuzzy systems, but a lot of the aspects should be also applicable to kindred types of data-driven models such as other types of fuzzy systems (e.g. Mamdani fuzzy systems or fuzzy classification models), neuro-fuzzy systems or radial basis function networks. In this sense, it was intended to show potentials for increasing process safety when using data-driven evolving models.

Acknowledgements

The author thanks Dr. Plamen Angelov for valuable discussions and comments on the manuscript of this paper.

References

[1] P. Angelov and N. Kasabov, "Evolving computational intelligence systems," in *Proceedings of the 1st International Workshop on Genetic Fuzzy Systems*, Granada, Spain, 2005, pp. 76–82.

[2] N. K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[3] P. Angelov and D. Filev, "Flexible models with evolving structure," *International Journal of Intelligent Systems*, vol. 19, no. 4, pp. 327–340, 2004.

[4] E. Lughofer and E. Klement, "FLEXFIS: A variant for incremental learning of Takagi-Sugeno fuzzy systems," in *Proceedings of FUZZ-IEEE 2005*, Reno, Nevada, U.S.A., 2005, pp. 915–920.

[5] L. Kóczy, D.Tikk, and T. Gedeon, "On functional equivalence of certain fuzzy controllers and RBF type approximation schemes," *International Journal of Fuzzy Systems*, 2000.

[6] P. Angelov and D. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Trans. on Fuzzy Systems*, vol. 34, no. 1, pp. 484–498, 2004.

[7] P. Angelov, E. Lughofer, and E. Klement, "Two approaches to data-driven design of evolving fuzzy systems: eTS and FLEXFIS," in *Proceedings of NAFIPS 2005*, Ann Arbor, Michigan, U.S.A., 2005, pp. 31–35.

[8] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, New Jersey 07458: Prentice Hall PTR, Prentic Hall Inc., 1999.

[9] K. Aström and B. Wittenmark, *Adaptive Control - Second Edition*. Addison-Wesley ISBN-0-201-55866-1, 1995.

[10] E. Lughofer, "Data-driven incremental learning of Takagi-Sugeno fuzzy models," Ph.D. dissertation, Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, February 2005.

[11] V. Vapnik, *Statistical Learning Theory*. New York: Wiley and Sons, 1998.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York, Berlin, Heidelberg, Germany: Springer Verlag, 2001.

[13] W. DuMouchel and F. O'Brian, "Integrating a robust option into a multiple regression computing environment," in *Computer Science and Statistics: Proceedings of the 21st Symposium on the Interface, American Statistical Association*, 1989.

[14] P. Deer and P. Eklund, "A study of parameter values for a mahalanobis distance fuzzy classifier," *Fuzzy Sets and Systems*, vol. 137, pp. 191–213, 2003.

[15] L. Chiang, E. Russell, and R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London, Great Britain: Springer Verlag London Berlin Heidelberg, 2001.

[16] E. Lughofer, E. Klement, J. Lujan, and C. Guardiola, "Model-based fault detection in multi-sensor measurement systems," in *Proceedings of IEEE IS 2004*, Varna, Bulgaria, 2004, pp. 184–189.

[17] A. Piegat, *Fuzzy Modeling and Control*. Heidelberg, New York: Physica Verlag, Springer Verlag Company, 2001.

[18] L. Wang and J. Mendel, "Fuzzy basis functions, universal approximation and orthogonal least-squares learning," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.

[19] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 29, no. 1, pp. 13–24, 1999.

[20] G. Mouzouris and J. Mendel, "A singular-value-qr decomposition based method for training fuzzy logic systems in uncertain environments," *Journal of Intelligent and Fuzzy Systems*, vol. 5, pp. 367–374, 1997.

[21] J. Yen and L. Wang, "An svd-based fuzzy model reduction strategy," in *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, New Orleans, LA, USA, 1996, pp. 835–841.

[22] J. Espinosa and J. Vandewalle, "Constructing fuzzy models with linguistic intergrity from numerical data - afreli algorithm," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 5, pp. 591–600, 2000.

[23] M. Setnes, "Simplification and reduction of fuzzy rules," in *Interpretability Issues in Fuzzy Modeling*, ser. Studies in Fuzziness and Soft Computing, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds. Berlin: Springer, 2003, vol. 128, pp. 278–302.

[24] M. Setnes, R. Babuska, U. Kaymak, and H. Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. SMC-B*, vol. 28, no. 3, pp. 376–386, 1998.

[25] N. Draper and H. Smith, *Applied Regression Analysis. Probability and Mathematical Statistics*. New York: John Wiley & Sons, 1981.

Technical Report
FLLL–TR–0210

Denoising Microscopy Image Sequences with Fine Structure Preservation

Leila Muresan, Erich Peter Klement
Fuzzy Logic Laboratorium Linz-Hagenberg
e-mail leila.muresan@jku.at

Johannes Kepler Universität Linz

Institut für Wissensbasierte Mathematische Systeme
Johannes Kepler Universität Linz
Altenbergerstrasse 69
A-4040 Linz
Austria

Fuzzy Logic Laboratorium Linz-Hagenberg

FLLL
Softwarepark Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Austria

Abstract

Detecting the trajectories of single proteins in a living cell is a challenging problem in wide-field microscopy. The task of manually identifying the proteins in the image sequence is tedious and hard to replicate. The automation of the task has to deal with the low signal-to-noise ratio, the autofluorescence of the cell, the small size of tracked features, the specific Poisson noise etc. A pre-processing denoising step might considerably improve the detection of single proteins and their tracking.

In this paper we describe and compare denoising algorithms that preserve fine details. Detection of fine details is based on the $\grave{a}\ trous$ wavelet decomposition and the performance of the algorithm is improved via wavelet coefficient denoising. Results on simulated and real image sequences are presented.

Introduction

One approach to understand biological processes at the cellular level is based on the study of protein trajectories in the living cell. Fluorophores are attached via an antibody to the proteins of interest and the detection of the photons emitted by each fluorophore offers information on the position of the corresponding protein. In order to collect statistically meaningful data, the detected protein trajectories have to be long and reliable. To achieve these goals a compromise between the quality of the signal and the bleaching time of the fluorophores has to be reached. The compromise is controlled via the illumination time. A shorter illumination time leads to lower signal-to-noise ratio in the image sequence, on the other hand it also means that the fluorophore bleaches slower. In order to be able to visualize the moving protein widefield microscopy techniques have to be used. Although the quality of images is lower than in the confocal case, the probability that the protein stays within the field of view is considerably higher. Details on the imaging of single molecules can be found in [6, 11, 9].
The tracking of proteins is performed manually, but the process is tedious and hard to replicate, raising the need for the automatization of the task.

1 Description of the problem

Typically the images contain on one hand a background profile due to the autofluorescence of the cell and on the other hand the signal due to the fluorophore-tagged proteins. Only the signal due to single proteins is of interest.
Due to the small size s of one fluorophore, $s \ll p$ where p is the size corresponding to one pixel, (in our setting $p = 200$nm), the fluorophore is equivalent to a point source. Hence the pattern of interest in these image sequences is the image of a point source through the optical system, the point spread function (PSF) of the system. Although the exact form of the PSF of a diffraction limited optical system with circular aperture is the Airy pattern:

$$\text{PSF}(r) = 2\left(\frac{J_1(\pi f_c r)}{\pi f_c r}\right) \tag{1}$$

where J_1 is the Bessel function of the first kind and f_c the cutoff frequency, a good approximation can be achieved with a two-dimensional Gaussian [4]:

$$G(r) = e^{-\left(\frac{r^2}{2\sigma^2}\right)} \tag{2}$$

A pre-processing denoising step can be used to improve the single protein detection and therefore tracking. Since the interest lies in finding trajectories of single molecules, the fine structures, corresponding to point source fluorophores, have to be preserved during denoising. These structures are small, bright, circular spots (representing quantized Gaussians with $\sigma \in [0.8, 1.4]$ pixels). The motion of the single molecules (Brownian, directed, etc.) makes the task even more difficult. However the motion between to successive frames is assumed small (usually less than two pixels).

2 Spot detection and denoising

Due to the quantum nature of light the images are corrupted by Poisson noise. Hence the Anscombe transform was applied as a pre-processing step. After applying the Anscombe transform the intensity values appear as corrupted by Gaussian noise:

$$T\{f(x,y)\} = 2\sqrt{f(x,y) + \frac{3}{8}} \tag{3}$$

2.1 The $\grave{a}\ trous$ wavelet transform

A robust spot detection is based on the $\grave{a}\ trous$ wavelet decomposition described in [10]. It was successfully used in [7] to detect fluorescent spots in microscopy images. It is a simple, non-decimated wavelet transform based on successive convolutions with B-spline-based kernels, resulting in low-passed filtered images $f_i(x,y)$. The wavelet coefficients W_i correspond to the difference of two successive low-pass filtered images. The algorithm is briefly described in Appendix A.

After performing N steps of the transform, the following relation holds:

$$f(x,y) = f_N(x,y) + \sum_{i=1}^{N} W_i(x,y) \tag{4}$$

The wavelet coefficients corresponding to smaller scales ($W_i, i < k$, $k = 3 \vee 4$) detect the single molecules. Since $f_N(x,y)$ corresponds roughly to the background due to autofluorescence, ignoring this level is equivalent to removing the background.

The coefficients in a wavelet plane are mainly due to noise, so the robust estimation of the mean and standard deviation of these coefficients gives the estimated parameters of the noise. The median m of $w_{i,j}$, where $w_{i,j}$ are the wavelet coefficients corresponding to W_i, robustly estimates μ and the median absolute deviation MAD

$$\text{MAD} = \frac{\text{median}\,|w_i - m|}{0.6745} \tag{5}$$

is the robust estimator of σ.

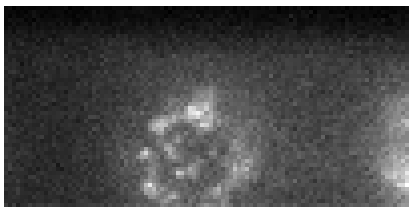

Figure 1: One frame of the protein image sequence

In order to detect spots, the coefficients are thresholded in the following way:

$$W_i(x,y) = 0 \text{ if } \frac{W_i(x,y) - m}{\text{MAD}} < 3. \tag{6}$$

Spots are found in the locations for which $W_i(x,y) > 0, i = 1,2,\ldots p$, where p is chosen depending on the scale of the features to be detected. The result of the thresholding in a real protein image is shown in figure 2.

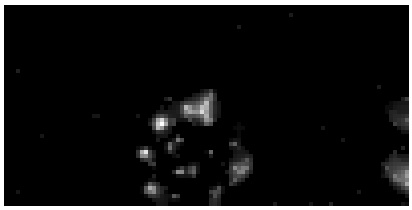

Figure 2: Recombination of denoised and thresholded wavelets

The performance of the algorithm was tested on artificial sequences and on microscopy images containing bright static structures imaged at decreasing illumination time (the number of structures is equal, but unknown in each frame and for each illumination time). The results are presented in table 1. The robustness of the results is remarkable, however in the challenging case of protein tracking 3D information might improve the 2D spot detection.

Table 1: Result of spot detection via $\grave{a}\ trous$ wavelet method. The ground truth is the same for all illumination times. The results are computed for ten frames/illumination time. The wavelet coefficients of the smallest scale W_i were discarded.

Illumination time (in ms)	Mean spot number	Variance in spot number
50	18.9	1.655
10	18.5	1.611
1	16.7	4.455
0.5	17.3	3.344
0.3	17.7	8.011
0.15	14.9	6.766

2.2 Denoising the wavelet coefficients

In order to achieve an improvement of the spot detection algorithm we try to exploit correlations between the wavelet coefficients of succesive frames, based on the assumption that signal is correlated in successive scales and frames, while noise is not.

The simplest approach to denoise wavelet coefficients is by averaging in time if their value is similar. If their value is not similar, then no averaging is performed. This behaviour is achieved by a modified version of the bilateral filtering [12], called spatial-tonal convolution [13]. It is well known that in the case of Gaussian noise the least squares error is minimized by the average of the sample. By selecting the sample via a Gaussian window, the problem becomes minimizing the functional

$$F(u(x)) := \int_{\Omega} \left(f(y) - u(x)\right)^2 G^s(x-y)\, dy. \tag{7}$$

The solution is simply a convolution with the d-dimensional Gaussian:

$$u_0(x) = \frac{\int_{\Omega} f(y) G^s(x-y)\, dy}{\int_{\Omega} G^s(x-y)\, dy} = \left(f * G^s\right)(x) \tag{8}$$

where u is the original, noiseless image, f the input image, G^s is a d-dimensional ($d \in 2, 3$) Gaussian window. In the case of windows containing spatial structure as well as movement over time of the small structures (fluorescent proteins), averaging might not hold the best results. The method described in [13], replaces the windowed least squares by robust estimators, which are less sensitive to outliers (in this case, the fine structure). The problem becomes minimizing:

$$E(u) := \int \rho\left(f(y) - u(x)\right) G^s(x-y)\, dy \tag{9}$$

The minimum is achieved when

$$\int \phi\left(f(y) - u(x)\right) G^s(x-y)\, dy = 0 \tag{10}$$

where $\phi = \rho'$.

Choosing $\rho^\tau(p) = 1 - \exp(-\frac{p^2}{2\tau^2})$, where τ controls the similarity of two intensity values whose difference is p, results in

$$\phi^\tau(p) = \frac{d\rho^\tau(p)}{d\rho} = \frac{p}{\tau^2} \exp\left(-\frac{p^2}{2\tau^2}\right) \tag{11}$$

and substituting ϕ^t in 10 yields:

$$u(x) = \frac{\int_{R^d} f(y) G^\tau(f(y) - u(x)) G^s(x-y)\, dy}{\int_{R^d} G^\tau(f(y) - u(x)) G^s(x-y)\, dy} \tag{12}$$

which can be solved using fixed point iteration: $u^{i+1} = F(u^i)$ until convergence or a maximum number of iterations is reached. However the method is computationally expensive and the results are often too blurred.

A related but more efficient method is anisotropic diffusion [8, 2, 1]. As described in [5], the transformed image u is the solution of the diffusion equation

$$\partial_t u = \operatorname{div}(D(|\nabla u|^2)\, |\nabla u|) \tag{13}$$

with initial condition:

$$u(x, 0) = f(x).$$

The diffusivity D is chosen as:

$$D(|\nabla u|^2) = \frac{1}{1 + |\nabla u|^2 / k^2}$$

which has small values for large gradients. In order to cope with the motion of proteins one can apply it either as a first denoising step before the $\grave{a}\ trous$ wavelet transform or alternatively each wavelet scale can be diffused and then thresholded as described above. The latter combines the isotropic feature detection in $2D$ (the $\grave{a}\ trous$ wavelet transform) with the detection of motion of small features in time (through diffusion). Results for both approaches are considered in the next section.

3 Results

The algorithm was tested on artificial images, with known ground truth as well as on real sequences. Artificial sequences were created based on 20 Brownian motion trajectories with a diffusion coefficient $D = 2$. Each sequence contains 20 frames and each frame k consists of a smooth background profile and 20 particles. The image of a particle i is obtained via a spatial Gaussian distribution of N_i photon counts (in the most challenging sequence N_i is drawn from a Gaussian distribution with mean 150 and standard deviation 20). The images are corrupted by Poisson noise and Gaussian noise (of mean 0 and standard deviation 0.008).

The results of the spot detection algorithm without any wavelet coefficient averaging is presented in table 2. Since the wavelet coefficients for the smallest scale, W_1 are the most contaminated by noise, it is possible to ignore the whole scale as a denoising step. However, if only the small support in W_1 has to be corrected (often the coefficients have a small shift compared to the other wavelet planes) the problem is solved by a morphological dilation of W_1 .

Table 2: Result of spot detection via $\grave{a}\ trous$ wavelet method in synthetic sequence (20 frames with 20 sots/frame) . W_1 represents the smallest scale.

Method	Mean spot number	Variance in spot number
With W_1	33.2	14.168
Without W_1	8.65	4.976
Dilated W_1	23.75	17.144

For wavelet coefficient correction, two kinds of error measures were considered: the mean square error as a measure for denoising and the number of false positives and false negatives, respectively, for the error in protein detection.

The best algorithms proved to be the combination of anisotropic diffusion with the $\grave{a}\ trous$ wavelet transform. In table 3 a comparison of the mean square error (per pixel, averaged over the 20 frames) is presented, in the case of denoising via anisotropic diffusion and denoising by diffusion of each wavelet coefficient. The second approach produces better mean square error values. This is due mainly to a better denoising of the background, separated from the details of the image via the wavelet transform.

Table 3: Mean square error ($k = 30, \lambda = 0.1$)

Method	2 iterations	3 iterations	4 iterations
Anisotropic diffusion and wavelet denoising	7.1223	5.7551	6.1351
Diffusion of wavelet coefficients	7.4492	3.5985	2.6590

The difference between the original wavelet and the diffused wavelet is presented in figure 3. As expected, the variance in the wavelet plane is decreasing, since the variance of the mean is inversely proportional with the number of samples used and averaging the coefficients reduces the range as well. As can be seen in the tables 4 and 5 an improvement in the detection (decrease in the number of undetected spots) can be achieved by an increase in computation. The computational complexity of the second algorithm is the complexity of the first multiplied by the number of wavelet scales.

In the case of the algorithm consisting of denoising the wavelet coefficients via spatial-tonal convolution the mean and variance of the detected spot numbers were 19.71 and 6.2198, respectively (comparable to the results obtained via diffusion). The drawbacks of the method were its high computational complexity (even if the considered neighborhood was only in time and not in space), the high number of false positives (on average 4 per frame) and the shift which appeared in the position of the detected spot compared to the original location.

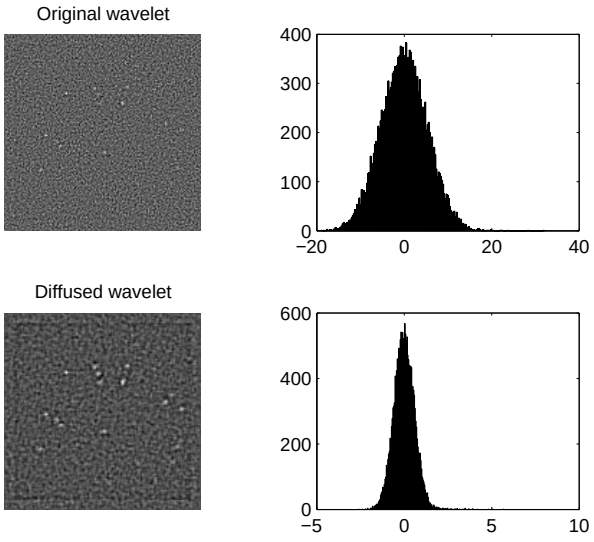

Figure 3: Original W_1 and diffused W_1 and the corresponding histograms

Results for real images for the $3D$ anisotropic diffusion, a $3D$ version of adaptive mean filter [3], $3D$ spatial tonal convolution are presented in Figures 4 and 5.

Table 4: Detection error. W_1 ignored (Wavelet coefficient threshold = 3σ) FP means false positives, FN means false negatives.

Method	FP mean	FP variance	FN mean	FN variance
Diffusion followed by wavelet detection	2.95	2.2475	3.15	3.0275
Diffusion of wavelet coefficients	0.75	1.0875	3.5	3.95

Table 5: Detection error. W_1 dilated (Wavelet coefficient threshold = 2.5σ) FP means false positives, FN means false negatives.

Method	FP mean	FP variance	FN mean	FN variance
Diffusion followed by wavelet detection	0.8	0.66	4.85	4.72
Diffusion of wavelet coefficients	1.3	1.31	3.25	2.78

4 Conclusion and future work

In the paper several methods for denoising microscopy images with preservation of fine structure were presented. The best performance was achieved for the diffusion and the subsequent thresholding of $\grave{a}\ trous$ wavelet coefficients. The parameters of the algorithms (the number of iteration for the diffusion equation, k, λ, as well as the number of wavelet scales used) were adjusted empirically. It would be desirable to study the effect of these parameters on the performance of the algorithms and if possible tune them automatically. Further performance comparisons are planned with other denoising and detection algorithms.

Appendix A

The kernel K_0 used for the low-pass filtering is generated by the B-spline of order 3:

$$K_0 = \begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}$$

K_i is obtained from K_{i-1} by inserting a line and column of zeros between each line and column of the previous kernel.

```
// A trous wavelet transform
// Input: image f0, number of computed wavelet levels N
function Atrous(f0, N)
```

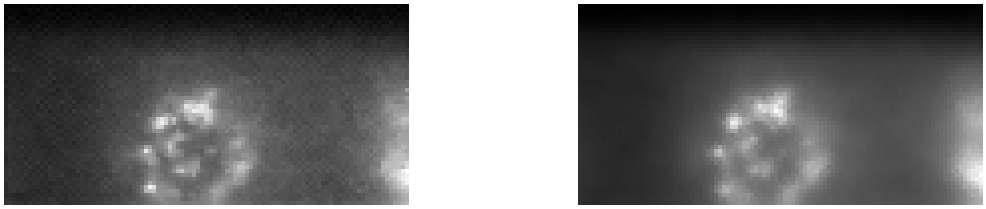

Figure 4: Denoising results on real images: Anisotropic diffusion and adaptive mean filter

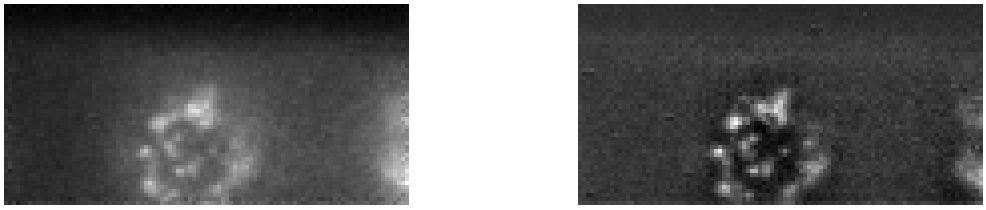

Figure 5: Denoising results on real images: Spatial-tonal convolution and spatial-tonal convolution applied to wavelet coefficients

```
  f[0] = f0
  for i = 1 to N
    f[i] = f[i-1] * K[i-1]
    W[i] = f[i-1] - f[i]
end
return W
```

Note

This work will be presented at the 30th annual workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR), 2006.

References

[1] D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, June 2002.

[2] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11(10):1141–1151, 2002.

[3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[4] S. B. Howell. *Handbook of CCD Astronomy*. Cambridge University Press, 2000.

[5] Bernd Jähne and Horst Haußecker. *Computer Vision and Applications*. Academic Press, 2000.

[6] M.J.Saxton and K. Jacobson. Single-particle tracking: Applications to membrane dynamics. *Annual Rev. Biophys. Biomol. Struct.*, 26:373–99, 1997.

[7] J.-C. Olivo-Marin. Extraction of spots in biological images using multiscale products. *Pattern Recognition*, 35:1989–1996, 2002.

[8] P. Perona and J. Malik. Scale -space and edge detection using anisotropic diffusion. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[9] A. Sonnleitner, G. J. Schütz, and Th. Schmidt. Free brownian motion of individual lipid molecules in biomembranes. *Biophysical Journal*, 77(5):2638–2642, 1999.

[10] Jean-Luc Starck, Fionn Murtagh, and Albert Bijaoui. *Image and Data Analysis: The Multiscale Approach*. Cambridge University Press, 1998.

[11] Russell E. Thompson, Daniel R. Larson, and Watt W. Webb. Precise nanometer localization analysis for individual fluorescent probes. *Biophysical Journal*, 82(5):2775–2783, 2002.

[12] C. Tomasi and Manduchi R. Bilateral filtering for gray and color images. In *Conference on Computer Vision, Bombay, India*, 1998.

[13] Rein van den Boomgaard and Joost van de Weijer. On the equivalence of local-mode finding, robust estimation and mean-shift. 2002.

DIC-Image Restoration with integrational Methods and Phase Filtering for automated Analysis of Yeast Cell Scans

Bettina Heise * Heimo Wolinski † Leila Muresan
Sepp Dieter Kohlwein Erich Peter Klement

Abstract

Automatic analysis of Differential Interference Contrast (DIC) images is often afflicted with difficulties due to the pseudo 3D- profile of the objects. A previous computational image transformation and restoration based on the modelling of the interference image formation can turn them into an appearance allowing a processing with conventional segmentation algorithms. We demonstrate the advantages of the use of phase-only filter, especially spiral phase filter against integrational approaches as an essential pre-processing step for further automated cell population analysis. The final assignment of fluorescently labelled subcellular structures to the corresponding DIC imaged yeast cells can be eased and automated by the developed software tool.

1 Introduction

Live cell imaging provides important information for functional analysis of proteins and organelles in living cells and have been applied to a wide range of biological specimens including the yeast, Saccharomyces cerevisiae, [12]. Fluorescence microscopy has got a key position and lots of sophisticated cutting edge techniques provide excellent images of the marked structures. However, avoiding multiple staining the registration of the cells and sub-cellular structures the imaging is partly realized by a combination of fluorescence and classical microscopic techniques. Due to the low absorbance of the unstained cells, phase contrast microscopy is still a method of choice producing illustrative images. For our application, the entire yeast cell populations are automatically registered by Differential Interference Contrast microscopy (DIC) and simultaneously, a confocal fluorescence microscope images the fluorescently marked substructures. Image processing for

*Department of Knowledge-Based Mathematical Systems, FLLL Linz-Hagenberg, Johannes Kepler University Linz, A- 4040 Linz, Austria

†SFB Biomembrane Research Center Institute of Molecular Biosciences, Department of Biochemistry, University Graz, Schubertstrasse 1, A- 8010 Graz, Austria

these purposes has to fulfil various tasks: automated cell recognition in DIC scans and their morphological description, the detection of sub-cellular structures, the mutual assignment of these structures to their related cells as well as the characterization of pattern formation, [13]. For the automated analysis of fluorescence images numerous software tools exist, however, they often fail for a direct application on phase contrast images which are very sensitive to the optical settings of the microscope and background inhomogeneities which often results in halos and other disturbing artefacts. By this reason the analysis of phase contrast images, especially DIC images, is complicated and the cell counting and mutual assignment of subcomponents to DIC imaged cell populations is often still done manually by the biologists till nowadays. In this paper we mainly consider the task of automated analysis of DIC cell scans, based on a previous DIC image transformation. This image conversion step enables an easier segmentation by standard segmentation methods in the following and further influences a correct assignment and characterization of fluorescently labelled particle patterns inside these cells. Although in this special application the specimens are given by yeast cells, Fig. 1, and a shape model could be additionally used under circumstances, we don't want to restrict us to a particular shape of the objects in scope to be of any use for biological varying cell types and modifications.

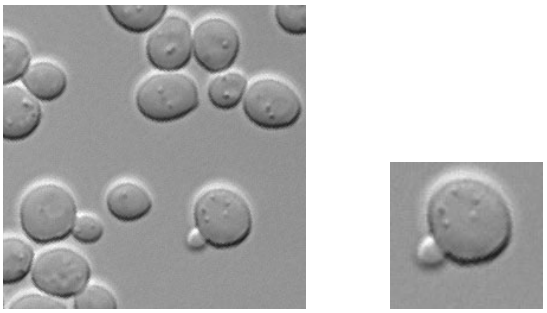

Figure 1: DIC image of Saccaromyces cerevisiae cell population

2 Transformation Methods

2.1 DIC Image Formation

DIC microscopy [11] is a type of phase contrast microscopy, which allows visualizing transparent or weak absorbing objects as commonly assumed for cells. Zernikes phase contrast microscope [3] measures the phase difference $\Delta\varphi$ between the diffracted and the propagating wave. In contrast to that, in DIC microscopy the phase difference between two waves, which are slightly displaced by a shear τ propagating the biological object under a distance $|\tau|$, gives a measure for the gradient $\nabla\theta_z$ of the optical path length θ projected onto the shear direction $\mathbf{e}_\tau$. (Assumed a propagating of the both waves parallel to the optical

axis $\mathbf{e}_z$ we will further denote $\theta = \theta_z$). The DIC shear direction $\mathbf{e}_\tau$, which is commonly diagonal, parallel or orthogonal to the scanning movement, determines the direction of the resulting intensity gradient, and $|\tau|$ gives the resolution of the DIC imaging.

$$\Delta\varphi(x,y) = \frac{2\pi}{\lambda} \langle \nabla\theta(x,y), \tau \rangle \tag{1}$$

Usually, an additional bias retardation φ_0 between the two waves is introduced by optical components, which modifies the intensity distribution. A bias $\varphi_0 = \pi/2$ results in a by human eye easily interpretable pseudo 3D-profile but leads to difficulties for automated image processing. Hence, the shadow-like appearance of a biological object imaged by a DIC microscope is a result of interference between the two waves and not of illumination. Standard image processing segmentation methods able to detect edges [16] or threshold- based methods [15] tends to fail directly applied to the DIC images due to the very nature of the image formation process resulting in no contrast perpendicular to the shear direction and in the same mean intensity as the background for homogeneous area.

2.2 Line Directed Methods

The measured intensity I in the DIC image, (s.App.A),

$$I(x,y) = 2I_0(1 - \cos\varphi) = 2I_0(1 - \cos(\varphi_0 + \Delta\varphi(x,y))), \tag{2}$$

can be linearized. Assuming weak phase objects $\Delta\varphi \ll \pi$ and a bias retardation of $\varphi_0 = \pi/2$, we get a relative intensity difference ΔI to the background intensity

$$\Delta I(x,y) \sim \Delta\varphi(x,y) \sim \langle \nabla\theta(x,y), \tau \rangle . \tag{3}$$

Integration of (3), i.e. summing up the relative intensity changes in DIC shear direction, could theoretically be used for retrieving θ, but leads to an accumulation of noise and scattering artefacts. An iterative line integration method by exponential averaging is proposed by [10] to reduce these artefacts. In [1] the 1D-Hilbert Transform (HT), (s.App.B), is applied for better visualisation of biological structures. For 2D-signals this transform can be performed in shear direction as partial Hilbert Transform, [4]. In [8] we tested a method based on an Iterative Hilbert Transform (IHT) algorithm for further contrast enhancement, mentioning that all these iterative integration methods require finding a compromise between object enhancement and noise amplification, which is in particular inherent in all the tested line-directed transformations. Hence, as stopping criteria for the number of iterations we used the energy reduction.

2.3 Isotropic Phase Filters

The Radial Hilbert Transform (RHT), which can be expressed by a spiral phase filter (SPF) in Fourier domain (u, v) as

$$H(u,v) = \exp(i\phi(u,v)) \text{ with } \phi(u,v) = \arctan(u,v), \tag{4}$$

is described for four DIC- images recorded with two different shear and bias values for exact phase reconstruction in [2]. However, in case of living biological objects a repeated scanning is mostly impossible. Because SPF can be used [9] to produce a shaded appearance of homogenous intensity objects, (s.App C), we apply the SPF inversely to our DIC-images to transform them for a more homogenous appearance of the entire cell. Based on the complex relation $H^{-1} = H^*$ in Fourier space between RHT and Inverse RHT we perform the spiral phase filtering reversely by a multiplication with $\exp(-i\phi)$ on the Fourier transform (FT) of DIC image. By inverse Fourier transform and considering the real part only, we get the reconstructed image.

3 Results

3.1 Line Directed Methods

Although the linearization of (2) and the straight forward line integration of (3)is a mathematically easy way it gives insufficient results, Fig. 2a. Further it can not reflect the nonlinearity of the DIC image formation. A previous correction of the DIC image intensity by the inverse function to (2) results in a slight contrast improvement, Fig. 2b, however, the heavy streaking artefacts are still obvious.

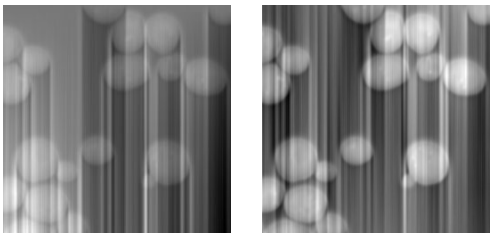

Figure 2: a: Integrated DIC image, b: Integrated DIC image corrected for the non-linear characteristic of the DIC microscope

Applying the Hilbert Transform these artefacts are reduced compared to the previous integration, with higher number of iterations in case of the iterative realisation they are slightly amplified, Fig. 3a,b. Under circumstances this can reduce the performance of further image segmentation

3.2 Phase Filtering

In compare to the integrational methods we investigated several phase filters, applying them directly to the FT of the recorded DIC cell scans. The arguments of these complex filter functions are shown in Fig. 4.

Compared to directional HT method, Fig. 2, it can be seen that spiral phase filtering prevents streaking artefacts, Fig. 5. On the other hand, the contrast to the background is lowered in case of spiral phase filtering for the registered yeast cells. That could be improved by

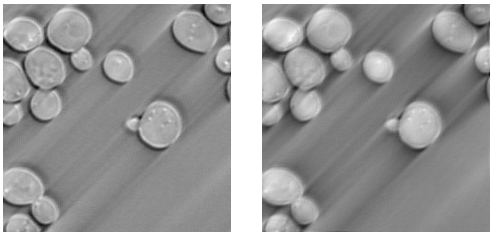

Figure 3: a: DIC image after applying a 1D- Hilbert Transform, b: DIC image after Iterative Hilbert Transform, 4 iterations

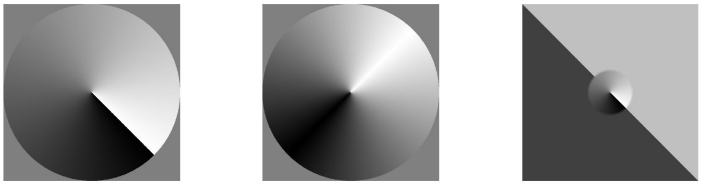

Figure 4: a: Spiral phase filter mask with an angular transfer function between 0 and 2π, b: Modified spiral phase filter with an mirrored angular phase transfer function, c: Combined phase filter mask with a spiral phase region in the centre

using two DIC images (Img_1, Img_2) recorded simultaneously in two orthogonal shear directions for reconstruction. Constructing a complex image of both $Img = Img_1 + iImg_2$ and applying on it the spiral phase filter would improve the contrast and reduce the slight halo effects. Such a DIC microscope has been patented recently [17], but is to the authors' knowledge not yet commercially available.

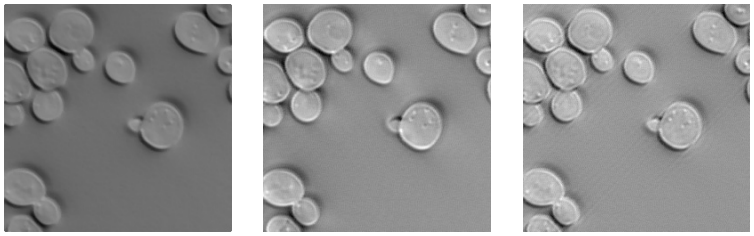

Figure 5: a: DIC image after filtering with the spiral phase mask , s. Fig. 4a, b: DIC image after filtering with the modified spiral phase mask, s. Fig. 4b, c: DIC image after filtering with combined phase mask, s. Fig. 4c

Meanwhile, as a compromise a modified spiral filter by mirroring the half of the spiral filter, Fig. 4b, and a combined filter, with an angular phase transfer function in the central part and a linear phase step in the outer region, Fig. 4c, are suggested. In case of the combined filter a smooth transition between the regions must be guaranteed to

prevent ringing artefacts. Furthermore, it needs a careful tuning of the filter range according to the spatial frequencies of the objects. The orientation of the spiral filter is independent of the shear, but as a complex filter is applied an orientation orthogonal to the DIC shear direction enables us to consider the real part of the filtered result only. Here we performed the filter in 45° direction according to the fixed diagonal shear direction.

3.3 Image Segmentation and Assignment of subcellular Components

By application of a spiral phase filter the histogram of the image is changed to a bimodal distribution, Fig. 6. Hence, a threshold-based segmentation can be performed.

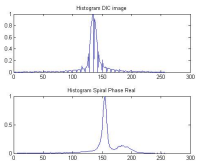

Figure 6: Histogram of the original and the spiral filtered images

We further tested segmentation by local adaptive thresholding, Fig. 7, wavelet decompositions or region growing methods.

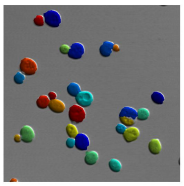

Figure 7: Result after segmentation by a local adaptive threshold applying a phase filtering previously. Cell separation was performed by watershed algorithm

Finally, after detection and localization of the fluorescently labelled sub-cellular components by a connected component algorithm using the Amira software package [14], the subcellular structures are assigned to the related yeast cells by mutual matching of positions, Fig. 8.

4 Conclusions

Phase filters are used as a fast and easily realizable transformation of the DIC imaged cell scans applying them as filter in Fourier domain. Spiral phase filter prevent streaking artefacts in the background

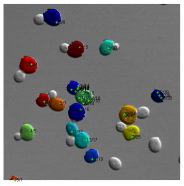

Figure 8: Assignment of fluorescently marked sub-cellular structures to yeast cells in the DIC scan

compared to previously tested, line-directed methods. Because for an exact image and path length resoration at least two DIC images with different shears or three images with different bias retardations are necessary the reconstruction with phase filters is not a quantitative one. But the following automated processing of the transformed DIC images can be eased by applying phase filters previously, as directional phase filters or as spiral filters. Finally, standard methods can be used for segmentation of these transformed images.

5 References

[1]Arnison MR, Cogswell CJ, Smith NI, Fekete PW, Larkin KG. 2000. Using the Hilbert transform for 3D visualization of differential interference contrast microscope images. J Microsc 199: 79-84.

[2]Arnison MR, Larkin KG, Shepphard CJR, Smith NI, Cogswell CJ. 2004. Linear phase imaging using differential interference contrast microscopy. J. Microsc 214:7-12.

[3]Born M, Wolf E. 2005. Principles of Optics-7th ed. Cambridge University press. p.472

[4] Bülow T, Sommer G. Das Konzept einer zweidimensionalen Phase unter Verwendung einer algebraisch erweiterten Signalrepräsentation, Mustererkennung 1997. Tagungsband, 19. DAGM-Symposium. p. 351-358.

[5]Davis JA, McNamarra DE, Cottrell DM. 2000. Image processing with the radial Hilbert transform: theory and experiments. Optics Letters. 25(5): 99-101

[6]Fürhapter S, Jesacher A, Bernet S, Ritsch-Marte M. 2005. Spiral phase contrast imaging in microscopy. Optics Express 13(3): 689-694

[7]Fürhapter S, Jesacher A, Bernet S, Ritsch-Marte M. 2005. Interference imaging with a spatial spiral phase filter. Optical Design and Engineering. Proc. SPIE 5962: 59621F-1-5

[8]Heise B, Sonnleitner A, Klement EP.2005. DIC Image reconstruction on Large Cell Scans. J Microscopy Research and Technique. 66:312-320

[9]Jesacher A, Fürhapter S, Bernet S, Ritsch-Marte M. 2005. Shadow effects in Spiral Phase Contrast Microscopy. Physical Review Letters. June: 233902-1-4

[10]Kam Z. 1998. Microscopic differential interference contrast image processing by line integration (LID) and deconvolution. Bioimaging 6: 166-176.

[11]Murphy D.2001. Differential interference contrast (DIC) microscopy and modulation contrast microscopy. Fundamentals of Light Microscopy and Digital Imaging. Wiley-Liss, New York, p. 153-168.

[12]Natter K, Leitner P, Faschinger A, Wolinski H, McCraith S, Fields S, Kohlwein SD. 2005. The spatial organization of lipid synthesis in the yeast Saccharomyces cerevisiae derived from large scale green fluorescent protein tagging and high resolution microscopy. Mol Cell Proteomics.4 (5): 662-72.

[13]Ohtani M, Saka A, Sano F, Ohya Y, Morishita S. 2004. Development of image processing program for yeast cell morphology. J Bioinform Comput Biol 1(4): 695-709.

[14]Software Package Amira 3.1 Mercury Computer Systems. Ltd. USA., www.amiravis.com

[15]Young D, Gray AJ. 1997. Semi-automatic boundary detection for identification of cells in DIC microscope images. IEE 6th Int. Conf. on Image Processing and its Applications, Dublin, Ireland, 1:346-350. IEEE Conference Publication No. 443.

[16]Young D, Gray AJ. 1996. Cell identification in Differential Interference Contrast microscope images using edge detection, British Machine Vision Conference, BMVA Press, 1, p.133-142.

[17]0050152030, United States Patent Application

A Phase difference

For a small shear distance, we can approximate the optical path difference $\Delta\theta$ between the two waves by the gradient projected to the shear τ:

$$\Delta\theta(x,y) = \theta(x,y) - \theta(x-\tau_x, y-\tau_y) = \langle \nabla\theta(x,y), \tau \rangle \tag{5}$$

which correspnds for a wavelength λ to a phase shift $\Delta\varphi$, Fig. 9,

$$\varphi(x,y) = \varphi_0(x,y) + \Delta\varphi(x,y) \text{ with } \Delta\varphi(x,y) = \frac{2\pi}{\lambda} \langle \nabla\theta(x,y), \tau \rangle \tag{6}$$

and results finally in an intensity distribution $I(x,y)$ by the interference of the two propagating waves

$$I(x,y) = 2I_0 \left[1 - \cos(\varphi(x,y))\right] \tag{7}$$

where I_0 is the intensity of each of both incident waves.

B Hilbert Transform

The 1D- Hilbert Transform of a function $f(x)$ in the spatial domain is defined by

$$f_H(x) = H\left\{f(x)\right\} = f(x) * \frac{1}{\pi x} = \frac{1}{\pi} \int_{\infty}^{\infty} \frac{f(x')}{(x-x')} dx' \tag{8}$$

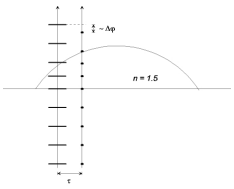

Figure 9: Illustration of phase shift by path length difference

A 2D-iterative realization of the directional HT in Fourier domain applied in diagonal direction to the image Img can be performed by

$$H_\alpha^p = F^{-1}\left\{ -i\,\mathrm{sgn}(u-v)\,\frac{\alpha}{\alpha + p\left(|u-v|^2 + |u+v|^2\right)^{1/2}}\,\mathcal{F}\{Img\}\right\} \tag{9}$$

where α denotes a smoothing parameter, p the number of iteration, (u, v) the spatial frequencies.

C Radial Hilbert Transform

The RHT can be seen as a 2D- generalization of the 1D- HT, [5]. Spiral phase filters are used both, in image processing for computational Fourier filtering, and as the optical counterpart in Fourier optics for the visualization of phase objects, e.g. realized by a Spatial Light Modulator (SLM), [6, 7]. There a SLM is inserted at the focal plane in the light path of a Spiral Phase Contrast microscope to create a shadowing effect on the phase objects, [9].

Support Vector Machines – A Case Study in Computer Vision

Stefan Larndorfer
Peter Haslinger
Software Competence Center Hagenberg
e-mail *Stefan.Larndorfer@scch.at*
Peter.Haslinger@scch.at

Kompetenzzentren-Programm

1 Introduction and Motivation

Support Vector Machines show a very promising performance when classifying difficult-structured multi-dimensional data. Recently, Support Vector Machines have been applied to many challenging problems like face recognition, text categorization, hand-written character recognition, image classification, object tracking [2], face detection [9], or biosequences analysis. The intention of this paper is to provide results on utilization of support vector machine technology to raw image data analysis.

Object classification/recognition in digital image processing is a very challenging process where many various methods can be applied to achieve a respectable performance. However, most of these techniques require a kind of feature selection and a model-based evaluation of these features. Based on this evaluation the final classification/recognition is made. In contrast to that, the performance of Support Vector Machines (SVMs) is not necessarily dependent on a correct selection and implementation of feature extraction algorithms, as the complete input data (e.g. image data) is used during classification/recognition. This is in particular helpful if it is very difficult to select meaningful features or if lots of object types have to be differentiated.

2 Case Study Environment

The Support Vector Machine will be used with raw image data in a concrete case study: the detection of a soccer ball in video sequences. The tasks of the SVM in this case study are defined by the following points:

- localize the soccer ball in the full image
- or find the best candidate among a set of preselected objects

The images possibly containing a soccer ball have a resolution of 352x288 pixels, consist of 3 color channels (24bit RGB) and are either grabbed from a real-time frame grabber or extracted from video data. The approximate size of the soccer ball in the complete image is about 12x12 pixels (ranging from 8x8 when it is further away to 15x15 when the ball is very close to the camera). In general the observed target object (i.e. the soccer ball) can be described as follows:

- surrounded by a rather constant light environment
- rigid object with a view angle invariant shape (a ball)
- object often is subject to occlusion
- the small object can appear anywhere in the full image

Before we are going into detail in how to solve the problem described above, we will give an introduction on support vector machines:

2.1 Support Vector Machines

Support vector machines are a machine learning tool for binary classification. Multi-class variations exists, but in its basic form it is used for two classes. Now, let us consider such a binary classification problem. We are given empirical data $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \{\pm 1\}$, where $\mathcal{X}$ is some non-empty set (*domain*) from which *pattern* x_i are taken. The y_i are the so-called *labels*, which determine the affiliation of the pattern to one of the two classes. In the task of learning we want to *generalize* to unseen data points. Given a pattern $x \in \mathcal{X}$ we want to predict the corresponding $y \in \{\pm 1\}$, therefore estimating a function $f : \mathcal{X} \to \{\pm 1\}$. One attempt to solve this problem is to introduce a hyperplane that optimally separates the two data point classes.

Vapnik et al. [3] considered the class of hyperplanes in some dot product space$\mathcal{H}$,

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \quad \mathbf{w} \in \mathcal{H}, b \in \mathbb{R} \tag{1}$$

corresponding to the decision function

$$f(\mathbf{x}) = sgn(\langle \mathbf{w}, \mathbf{x} \rangle + b), \tag{2}$$

which will be the basis for our support vector learning algorithm. If the dataset is separable then the data will be correctly classified if $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0 \quad \forall i$ with data points $\mathbf{x}_i \quad (i = 1, \ldots, m)$ and corresponding labels $y_i = \pm 1$. The relation is invariant to a positive rescaling , so we define a canonical hyperplane with $\langle \mathbf{w}, \mathbf{x} \rangle + b = 1$ for the closest point on one side and $\langle \mathbf{w}, \mathbf{x} \rangle + b = -1$ on the other. The normal vector for the separating hyperplane is $\frac{\mathbf{w}}{||\mathbf{w}||}$, and the margin is given by the projection of $\mathbf{x}_1 - \mathbf{x}_2$ onto this vector (see Figure 1). Which leads to a margin of $\frac{1}{||w||}$. To maximize the margin leads to the quadratic problem:

$$minimize \quad g(\mathbf{w}) = \frac{||w||^2}{2} \tag{3}$$

subject to the constrains:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 0 \quad \forall i \tag{4}$$

Such a constrained optimization problem can be dealt with by introducing *Lagrange multipliers* $\alpha_i \geq 0$ and a minimization of the primal Lagrangian [7]:

$$L(\mathbf{w}, b, \alpha) = \frac{||w||^2}{2} - \sum_{i=1}^{m} \alpha_i (y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1). \tag{5}$$

With the Karush-Kuhn-Tucker (KKT) complementarity conditions and taking the derivatives with respect to $\mathbf{w}$ and b we receive the Wolfe dual [12]:

$$maximize \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{6}$$

subject to the constrains

$$\alpha_i \geq 0 \tag{7}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{8}$$

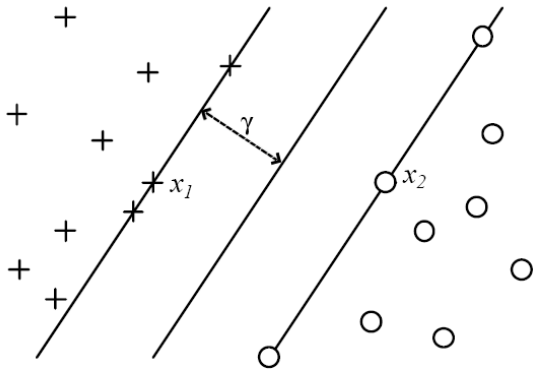

Figure 1: The *margin* γ is the perpendicular distance between the separating hyperplane and the closest data points (support vectors). x_1 and x_2 are such support vectors.

Everything so far was formulated within a dot product (inner product) space, so we will perform a substitution with symmetric kernels of the shape

$$k(x, x') := \langle \mathbf{x}, \mathbf{x}' \rangle = \langle \Phi(x), \Phi(x') \rangle , \tag{9}$$

where the kernel k is a function

$$\begin{aligned} k: \quad \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (x, x') &\mapsto k(x, x'). \end{aligned} \tag{10}$$

and the map Φ, representing the patterns as vectors in some dot product space $\mathcal{H}$ (*feature space*), is given by

$$\begin{aligned} \Phi: \quad \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \mathbf{x} := \Phi(x). \end{aligned} \tag{11}$$

The name kernel is derived from integral operator theory, which underpins much of the theory of the relation between kernels and their corresponding feature spaces. We will give now 2 examples of commonly used kernels, the Gaussian radial basis function(RBF) kernel:

$$k(x, x') = e^{-\gamma ||x - x'||^2},$$

where $\gamma > 0$, and the homogeneous polynomial kernel:

$$k(x, x') = \langle x, x' \rangle^d,$$

with $d \in \mathbb{N}$.

The substitution with (9) is referred to as *kernel trick* – one of the most important steps within SVMs. With such a choice of kernel the data can become separable in feature space despite being non-separable in the original input space. The Wolfe dual can be reformulated now:

$$maximize \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{12}$$

with the new constrains from (7) and (8). After the optimal values of α_i have been found the decision function is based on the sign of:

$$f(\mathbf{z}) = \sum_{i=1}^{m} y_i \alpha_i k(\mathbf{z}, \mathbf{x}_i) + b, \tag{13}$$

with bias b given by solving the primal constrains.

Only the points closest to the found maximal margin hyperplane have $\alpha_i > 0$ and these points are called the *support vectors*. All other points have $\alpha_i = 0$. This means that the evaluation of the decision function depends solely on the points closest to the hyperplane. They are the most informative patterns of the data.

2.2 Soft Margin SVM

To deal with the non-separable case (for example noise in the training data) the so-called *soft margin* is introduced to reduce the effect of outliers. The so-called slack variables $\xi_i, i = 1, \ldots, m$ are used within the constrains [6], which then become

$$\begin{aligned} \langle \mathbf{w}, \mathbf{x_i} \rangle + b \geq +1 - \xi_i \quad for \quad y_i = +1 \\ \langle \mathbf{w}, \mathbf{x_i} \rangle + b \leq -1 + \xi_i \quad for \quad y_i = -1 \\ \xi_i \geq 0 \quad \forall i. \end{aligned} \tag{14}$$

The sum $\sum_i \xi_i$ is an upper bound for the number of training errors. To assign an extra cost for errors, the objective function is changed to minimize $||w||^2/2 + C \sum_i \xi_i$, where C is a to be chosen parameter. The resulting difference in the Wolfe dual is only the new box constraint $0 \leq \alpha_i \leq C$, the α_i have an upper bound.

3 Basic Approach

One thing to keep in mind is that SVMs are supposed to work on data that is based on record sets that all have a constant dimension. This implies that a SVM ideally is used to classify images that all have the same size. In order to provide this input data to the SVM, a simple but effective window-shift algorithm was utilized that cuts out lots of small image parts (SVM thumbnails) of the current image and passes this data on to the SVM which then can decide whether the SVM thumbnail contains a soccer ball or not. In order to keep the amount of the SVM thumbnails small and improve performance, only the parts of the image that have a significant difference to the background image are taken as a basis for the window-shift algorithm (moving objects – the foreground). In figure 2 the window-shift algorithm can be seen in action.

3.1 The window-shift algorithm

As the average size of the soccer ball ranges within 8-15 pixel in diameter, a SVM thumbnail size of 16x16 pixels was chosen. The window-shift algorithm creates all these SVM thumbnail images by shifting a 16x16 pixel wide window over all foreground objects. If the foreground object is

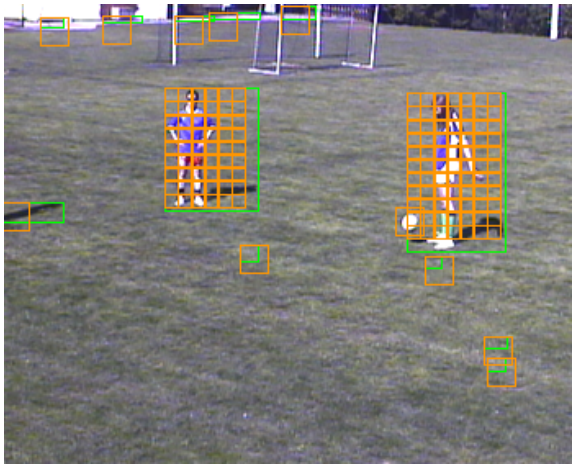

Figure 2: Every part of the image that is in the foreground is treated with the window-shift algorithm producing lots of tiny SVM thumbnail images that then can be analyzed by the Support Vector Machine.

bigger than 16x16 pixels, the algorithm shifts this window only by 8 pixels, creating overlapping SVM thumbnail images as a consequence. This is important in order to be able to recognize a ball that otherwise sometimes may be distributed over two different thumbnail images and, as a result would not be detected in any of the created thumbnail images.

Other SVM applications that process raw images (e.g. [9]) often apply additional image processing algorithms such as histogram equalization or blob-scaling to the SVM thumbnails to further improve the overall performance of the SVM. In order not to distort the original performance of the underlying SVM, no additional image processing algorithms were applied on the SVM thumbnail images in this case study.

4 Case Study in Soccer Ball Detection

The used testing framework for our case study was the support vector framework LIBSVM [5], an implementation widely in use.

The following steps have been performed to test SVMs on the problem of soccer ball detection (see also Figure 3):

1. **Collection of samples for soccer ball and non-soccer ball images**.
2. **Conversion of images to raw data:** the gray scale images are converted to normalized one dimensional data vectors.
3. **Determination of optimal parameters:** a grid search is performed to find the optimal value for the soft margin (parameter C) and the kernel (in our case RBF Kernel with parameter γ).
4. **Training of SVM on training data with optimal parameters**.

5. **Testing of resulting SVM on test images/video with unknown soccer ball positions**

A detailed description of the most important steps is given in the following sections.

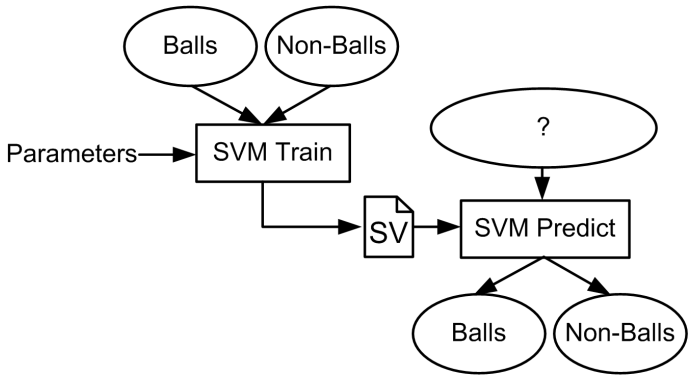

Figure 3: The case study workflow.

4.1 Creating the Learning Data Set

Due to the nature of the object that had to be detected in this case study (i.e. the soccer ball), only gray scale image data was used for SVM classification. This also had the advantage that the dimensionality of the input model file decreased by the factor of 3 (from RGB data to gray scale images). The data vector was rescaled/normalized from $[0, 255]$ to a $[0, 1]$, as SVMs work best on normalized data.

4.1.1 Creating non-soccer ball SVM thumbnails

For training purposes, we have selected 50 video frames that did not contain a visible soccer ball and stored theses frames as a sequence of bitmap images. This sequence of bitmap images was the input data for the window-shift algorithm as described in section 3.1. We additionally incorporated some code that caused the window-shift algorithm to store each of the created SVM thumbnail images as a tiny bitmap. With this configuration we immediately received a large data basis that could server as non-soccer ball examples for the SVM training process.

4.1.2 Creating soccer ball SVM thumbnails

Efficiently creating many suitable soccer ball SVM thumbnails is a little bit more time consuming than creating non-soccer ball SVM thumbnails. This time we could not use the convenient window-shift algorithm that stores each of the created SVM thumbnails as this would have implied to manually select about 20-30 soccer balls out of a SVM thumbnail data basis of about 7000. This clumsy approach additionally would not have ensured that enough *different* soccer ball SVM thumbnail images would have been generated.

A more systematic way of generating soccer ball SVM thumbnails is depicted in figure 4. This method, that will be referred to as "Snake-Windowing" from now on, generates lots of SVM

Figure 4: With a snake-like movement of the 16x16 thumbnail window over a single image several independent SVM thumbnail images can be generated. This procedure ensures that the SVM finally is trained to be rather *location invariant*, so that it does not matter whether the soccer ball appears at the lower right or at the upper left in a SVM thumbnail image.

thumbnail images out of one single image that contains the object to be detected. As the position of the target object within the SVM thumbnail window should not be a decisive factor when the SVM classifies input data, the SVM also has to be trained that way: This means that as many as possible soccer balls that are located at all possible locations within the SVM thumbnail window should be learned. Snake-Windowing exactly provides this data basis in an efficient way.

4.2 Finding optimal parameters

Choosing the right SVM kernel with the right parameters has a major influence on the overall performance of the trained SVM. In our test case we concentrated on the RBF kernel (which are a good choice to start with, as it has the best generalization behavior). there are basically two parameters: C and γ. As the value of these parameters is not known in advance, some kind of parameter search must be done. A exhaustive "grid-search" on C an γ was performed using cross-validation. The SVM was trained on the give soccer ball data set with various different parameter settings (pairs of (C, γ)), and the resulting SVM with the best cross-validation accuracy rate was chosen for the test cases. Basically are tried and the one with the best cross-validation accuracy is picked. It is a very straightforward approach, but gives good results after a reasonable time. The parameter search needs a lot of computational power. It took about 10 hours to find the optimal parameters for our SVM training data set which consisted of about 7500 record sets. Figure 5 shows a diagram of such a grid search.

5 Results and Conclusion

To our surprise, the quality of the obtained SVM dramatically exceeded our expectation. The final SVM showed a very good generalization behavior, robustly classified soccer balls even when the circumstances were tricky. Considering that up to now not a lot of time was spent in order to optimize the SVM, the results are very promising: The classification performance mostly is better than 99%. Figures 6-7 show a few example images.

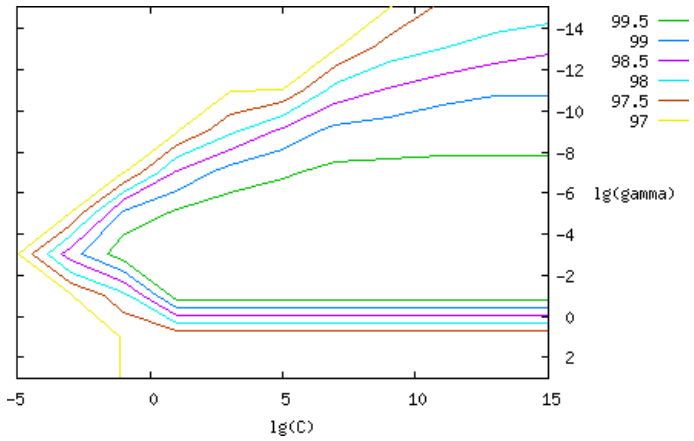

Figure 5: The generated output image after the grid search for optimal parameters for C and γ in the soccer ball training data set.

The performance concerning speed of the current SVM implementation is not overwhelming, but still acceptable. In our case study the SVM had to classify about 120 SVM thumbnail images per image. When processing a live video the SVM could process about 7 frames per second (using an Intel Pentium D820 with 2x2.8 GHZ). As tuning the SVM implementation concerning speed was not a relevant aspect of this research, there is still plenty of room to incorporate further speed improvements.

5.1 Further Work

This case study showed a very straight forward approach in using SVMs in image processing. A lot of improvements in terms of computational speed and detection rate are possible.

To improve the performance a combination of tracking methods with the detection of soccer balls is thinkable. This will decrease the search area for the SVM classifier. It might also be possible that the detection need not be called on every frame, but only after a given number or when the tracking seems to be unsecure.

A performance boost and an increase of the detection rate can also be achieved by feature selection for soccer images (for example wavelet transformation) and image preprocessing (histogram equalization,...).

Tests with other kernels should to be performed. The polynomial kernel is know for its good behavior in image processing.

Modifying the training data itself (removing the background) can increase the detection rate. Irrelevant data within images (background behind object) is replaced by a pair of images with white and black background (see [10]).

Techniques from face detection and detection of humans could be applied. For example in [8] a method is applied that, for a given SVM, creates a set of so-called reduced set vectors (RSVs) that approximate the decision function (introduced in [4]). A special decomposition algorithm is shown that guarantees global optimality and allows to train SVMs on very large data sets. In [11]

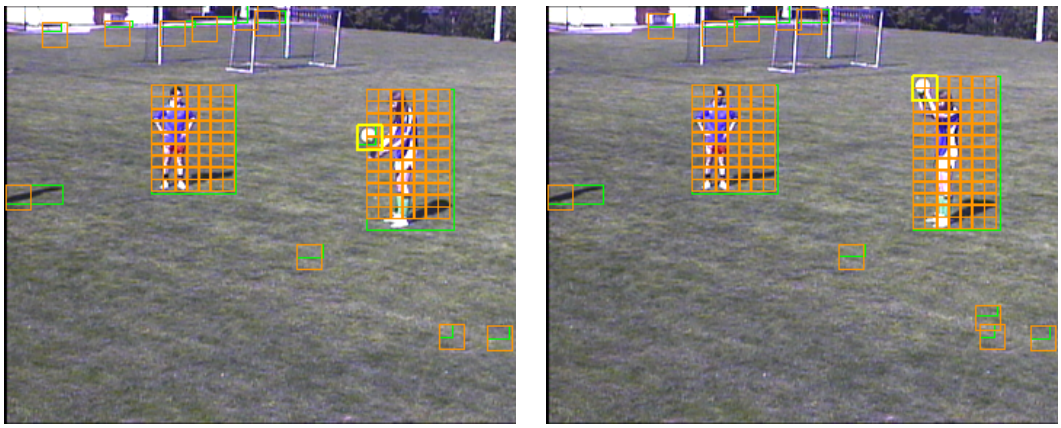

Figure 6: 2 samples of a successful ball detection. The orange rectangles mark all investigated SVM thumbnail images. The yellow rectangle shows the SVM thumbnail image with the highest soccer ball reliability. On the right: even with major distortions (hand touching the ball) a correct classification is found.

SVMs are trained with dense optical flow patterns to detect humans in outdoor surveillance. A lot of unpredictable influences (light, clothing, size, etc.) complicate detection of a humans, therefore patterns of human motion are to be discovered. Another optic-flow based tracker is presented in [1]. These improvements could also be applied to soccer ball detection.

References

[1] S. Avidan. Support vector tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, pages 1064–1072, August 2004.

[2] Shai Avidan. Support vector tracking, mobileye vision technologies.

[3] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

[4] Christopher J. C. Burges. Simplified support vector decision rules. In *International Conference on Machine Learning*, pages 71–77, 1996.

[5] Chihi-Jen Lin Chih-Wei Hsu, Chihi-Chung Chang. A practical guide to support vector classification. Technical report.

[6] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[7] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, second edition, 1987.

[8] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. *CVPR'97*, 1997.

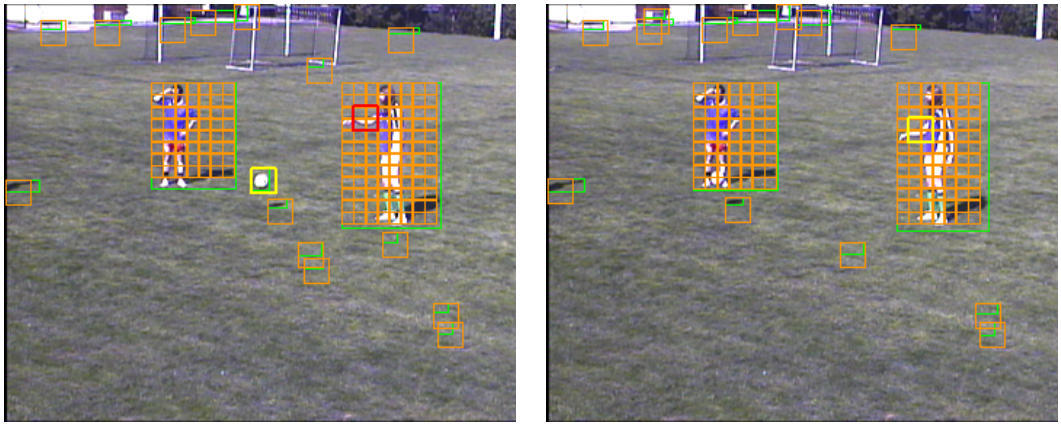

Figure 7: 2 samples with classification problems. On the left: the red rectangle indicates a second soccer ball candidate, but its reliability is not as high as the yellow rectangle. On the right: the white legs of the player with red shorts cause a too high distortion, therefore the best candidate found is the white arm of other player.

[9] Edgar Osuna, Robert Freund, and Federico Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, 1997.

[10] D. Roobaert. Improving the generalization of linear support vector machines: an application to 3d object recognition with cluttered background. In *Proc. SVM Workshop at the 16th International Joint Conference on Artificial Intelligence*, pages 857–862, 1999.

[11] Hedvig Sidenbladh. Detecting human motion with support vector machines. *IAPR International Conference on Pattern Recognition*, 2:188–191, 2004.

[12] P. Wolfe. A duality theorem for nonlinear programming. *Quarterly of Applied Mathematics*, 19:239–244, 1961.

s c c h
software competence center
hagenberg

Technical Report
SCCH–TR–0538

Lexicographic Composition of Fuzzy Orderings

Ulrich Bodenhofer
Software Competence Center Hagenberg
ulrich.bodenhofer@scch.at

Abstract — The present paper introduces an approach to construct lexicographic compositions of similarity-based fuzzy orderings. This construction is demonstrated by means of non-trivial examples. As this is a crucial feature of lexicographic composition, the preservation of linearity is studied in detail. We obtain once again that it is essential for meaningful results that the t-norm under consideration induces an involutive (strong) negation.

Key words — *fuzzy equivalence relations, fuzzy orderings, lexicographic composition, strict fuzzy orderings.*

Submitted to *11th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Paris, France, July 2–7, 2006.

Kplus
Kompetenzzentren-Programm

1 Introduction

Lexicographic composition is a fundamental construction principle for ordering relations. The most important feature of this construction is that the composition of two linear orderings again yields a linear ordering. Given two orderings $\leq_1$ and $\leq_2$ on non-empty domains X_1 and X_2, respectively, the lexicographic composition is an ordering $\leq'$ on the Cartesian product $X_1 \times X_2$, where $(x_1,x_2) \leq' (y_1,y_2)$ if and only if

$$(x_1 = y_1 \wedge x_2 \leq_2 y_2) \vee (x_1 \neq y_1 \wedge x_1 \leq_1 y_1). \tag{1}$$

Rewriting $x_1 \neq y_1 \wedge x_1 \leq_1 y_1$ as $x_1 <_1 y_1$ (i.e. the strict ordering induced by $\leq_1$) and taking into account that $x_1 = y_1 \vee x_1 \neq y_1$ is a tautology and that $\leq_1$ is reflexive, we obtain that (1) is equivalent to

$$(x_1 \leq_1 y_1 \wedge x_2 \leq_2 y_2) \vee x_1 <_1 y_1. \tag{2}$$

The study of fuzzy orderings can be traced back to the early days of fuzzy set theory [12, 17, 18, 22]. Partial fuzzy orderings in the sense of Zadeh [22], however, have severe shortcomings that were finally resolved by replacing the crisp equality by a fuzzy equivalence relation, thereby maintaining the well-known classical fact that orderings are obtained from preorderings by factorization [1, 3, 4, 11, 14].

In [2, 4], several methods for constructing fuzzy orderings are presented, including Cartesian products. How to transfer lexicographic composition to the fuzzy framework, however, remained an open problem. The reason why this remained an open issue for a relatively long time is that there was no meaningful concept of strict fuzzy ordering in the similarity-based framework so far. As this issue is solved by [5] now, we are able to give a solution in this paper. Detailed proofs are omitted, as they are long and technical. Details are available from the author upon request (and the reader is also referred to upcoming publications).

2 Preliminaries

For simplicity, we consider the unit interval $[0,1]$ as our domain of truth values in this paper. Note that most results, with only minor and obvious modifications, also hold for more general structures [11, 13, 14, 15]. The symbols T, $\tilde{T}$, etc., denote left-continuous t-norms [16]. Correspondingly, $\vec{T}$ denotes the unique residual implication of T. Furthermore, we denote the residual negation of T with $N_T(x) = \vec{T}(x,0)$.

Definition 1. A binary fuzzy relation $E : X^2 \to [0,1]$ is called *fuzzy equivalence relation*[1] with respect to T, for brevity *T-equivalence*, if the following three axioms are fulfilled for all $x,y,z \in X$:

1. Reflexivity: $E(x,x) = 1$
2. Symmetry: $E(x,y) = E(y,x)$
3. T-transitivity:
$$T\big(E(x,y),E(y,z)\big) \leq E(x,z)$$

[1] Note that various diverging names for this class of fuzzy relations appear in literature, like similarity relations, indistinguishability operators, equality relations, and several more [7, 15, 20, 22]

Definition 2. A binary fuzzy relation $L : X^2 \to [0,1]$ is called *fuzzy ordering* with respect to T and a T-equivalence $E : X^2 \to [0,1]$, for brevity *T-E-ordering*, if it fulfills the following three axioms for all $x, y, z \in X$:

1. E-reflexivity: $\quad E(x,y) \leq L(x,y)$
2. T-E-antisymmetry:
$$T\big(L(x,y), L(y,x)\big) \leq E(x,y)$$
3. T-transitivity:
$$T\big(L(x,y), L(y,z)\big) \leq L(x,z)$$

Definition 3. A fuzzy relation $R : X^2 \to [0,1]$ is called *strongly complete* if $\max(L(x,y), L(y,x)) = 1$ for all $x, y \in X$ [6, 12, 17]. R is called T-linear if $N_T(L(x,y)) \leq L(y,x)$ for all $x, y \in X$ [6, 14].

Definition 4. A binary fuzzy relation $S : X^2 \to [0,1]$ is called *strict fuzzy ordering* with respect to T and a T-equivalence $E : X^2 \to [0,1]$, for brevity *strict T-E-ordering*, if it fulfills the following axioms for all $x, x', y, y', z \in X$:

1. Irreflexivity: $\quad S(x,x) = 0$
2. T-transitivity:
$$T\big(S(x,y), S(y,z)\big) \leq S(x,z)$$
3. E-extensionality:
$$T(E(x,x'), E(y,y'), S(x,y)) \leq S(x',y')$$

As already mentioned above, it is of vital importance for lexicographic composition how to "strictify" a given fuzzy ordering. The following theorem summarizes the most important facts.

Theorem 5. [5] *Consider a T-equivalence $E : X^2 \to [0,1]$ and a T-E-ordering $L : X^2 \to [0,1]$. Then the following fuzzy relation is a strict T-E-ordering on X:*

$$S(x,y) = \min(L(x,y), N_T(L(y,x)))$$

If T does not have zero divisors, the equality $S(x,y) = \min(L(x,y), N_T(E(y,x)))$ holds. Moreover, S is monotonic w.r.t. L in the following sense (for all $x, y, z \in X$).

$$T(L(x,y), S(y,z)) \leq S(x,z)$$
$$T(S(x,y), L(y,z)) \leq S(x,z)$$

S is the largest strict T-E-ordering contained in L that fulfills this kind of monotonicity.

For intersecting T-transitive fuzzy relations, the concept of domination between t-norms is of vital importance [9, 16, 19].

Definition 6. A t-norm T_1 is said to *dominate* another t-norm T_2 if, for every quadruple $(x, y, u, v) \in [0,1]^4$, the following holds:

$$T_1\big(T_2(x,y), T_2(u,v)\big) \geq T_2\big(T_1(x,u), T_1(y,v)\big)$$

Lemma 7. [9] *Consider two t-norms T_1 and T_2. The T_2-intersection of any two arbitrary T_1-transitive fuzzy relations is T_1-transitive if and only if T_2 dominates T_1.*

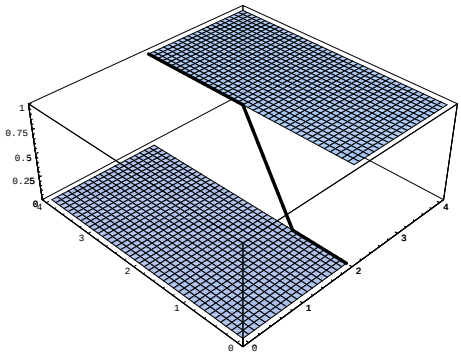

Figure 1: Cut view of a lexicographic composition of a crisp linear ordering and a fuzzy ordering according to Proposition 8

3 Starting the Easy Way: One Crisp and One Fuzzy Ordering

Let us first consider the case where the primary ordering is crisp and the secondary ordering is fuzzy. As the strict ordering is only needed for the primary ordering, we do not need to take any strict fuzzy ordering into account.

Proposition 8. *Let us consider a crisp ordering $L_1 : X_1^2 \to \{0,1\}$ and a T-E_2-ordering $L_2 : X_2^2 \to [0,1]$ (with $E_2 : X_2^2 \to [0,1]$ being a T-equivalence). Then the fuzzy relation $L : (X_1 \times X_2)^2 \to [0,1]$ defined as*

$$L((x_1,x_2),(y_1,y_2)) = \begin{cases} 1 & \text{if } x_1 \neq y_1 \text{ and} \\ & L(x_1,y_1) = 1, \\ L_2(x_2,y_2) & \text{if } x_1 = y_1, \\ 0 & \text{otherwise,} \end{cases}$$

is a fuzzy ordering w.r.t. T and the T-equivalence $\tilde{E} : (X_1 \times X_2)^2 \to [0,1]$ defined as

$$E((x_1,x_2),(y_1,y_2)) = \begin{cases} E_2(x_2,y_2) & \text{if } x_1 = y_1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that, if both components L_1 and L_2 are crisp orderings, then L as defined above is equivalent to the constructions (1) and (2).

Example 9. Consider $X_1 = X_2 = [0,4]$, let L_1 be the classical linear ordering of real numbers, and assume that L_2 is defined as follows:

$$L_2(x,y) = \max(\min(1-x+y,1),0)$$

It is well-known that L_2 is a fuzzy ordering with respect to the Łukasiewicz t-norm $T_{\mathbf{L}}(x,y) = \max(x+y-1,0)$ and the $T_{\mathbf{L}}$-equivalence $E_2(x,y) = \max(1-|x-y|,0)$. Figure 1 shows a cut view of the fuzzy ordering L that is obtained when applying the construction from Proposition 8. The cut view has been obtained by plotting the value $L((2,2),(y_1,y_2))$ as a two-dimensional function of y_1 and y_2.

The following proposition clarifies in which way linearity of the two component orderings L_1 and L_2 is preserved by the construction in the previous proposition.

Proposition 10. *Let us make the same assumptions as in Proposition 8. If L_1 is a crisp linear ordering and L_2 is strongly complete, then L is also strongly complete. If L_1 is a crisp linear ordering and L_2 is T-linear, then L is also T-linear.*

4 Lexicographic Composition of Two Non-Trivial Fuzzy Orderings

The results of the previous section have been known to the author since 1998, but they were not published so far, as they cannot be considered a full-fledged solution of the problem. So let us now consider the general case, where both components are fuzzy orderings without any further assumptions so far. The following theorem gives a general construction inspired by the classical construction (2).

Theorem 11. *Consider two T-equivalences $E_1 : X_1^2 \to [0,1]$, $E_2 : X_2^2 \to [0,1]$, a T-E_1-ordering $L_1 : X_1^2 \to [0,1]$, and a T-E_2-ordering $L_2 : X_2^2 \to [0,1]$. Moreover, let $\tilde{T}$ be a t-norm that dominates T. Then the fuzzy relation $\mathrm{Lex}_{\tilde{T},T}(L_1,L_2) : (X_1 \times X_2)^2 \to [0,1]$ defined as*

$$\begin{aligned}\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((x_1,x_2),(y_1,y_2)) = \\ \max\big(\tilde{T}(L_1(x_1,y_1),L_2(x_2,y_2)), \\ \min(L_1(x_1,y_1),N_T(L_1(y_1,x_1)))\big)\end{aligned}$$

is a fuzzy ordering w.r.t. T and the T-equivalence $\mathrm{Cart}_{\tilde{T}}(E_1,E_2) : (X_1 \times X_2)^2 \to [0,1]$ defined as the Cartesian product of E_1 and E_2:

$$\begin{aligned}\mathrm{Cart}_{\tilde{T}}(E_1,E_2)((x_1,x_2),(y_1,y_2)) = \\ \tilde{T}(E_1(x_1,y_1),E_2(x_2,y_2))\end{aligned}$$

Note that, if L_1 is a crisp ordering, then $\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)$ defined as in Theorem 11 coincides with the fuzzy relation L defined in Proposition 8. Consequently, if both components L_1 and L_2 are crisp orderings, then $L_{\tilde{T}}$ is equivalent to the constructions (1) and (2).

Example 12. Consider again the domain $X = [0,4]$ and consider the following three fuzzy relations on X:

$$\begin{aligned}L_3(x,y) &= \max(\min(1-\tfrac{1}{2}(x-y),1),0)\\ L_4(x,y) &= \min(\exp(y-x),1)\\ L_5(x,y) &= \min(\exp(3(y-x)),1)\end{aligned}$$

L_3 is a $T_{\mathbf{L}}$-E_3-ordering with $E_3(x,y) = \max(1-\frac{1}{2}|x-y|,0)$. L_4 is a $T_{\mathbf{P}}$-E_4-ordering[2] with $E_4(x,y) = \exp(-|x-y|)$ and, since $T_{\mathbf{L}} \leq T_{\mathbf{P}}$, a $T_{\mathbf{L}}$-E_4-ordering as well. L_5 is a $T_{\mathbf{P}}$-E_5-ordering with $E_4(x,y) = \exp(-3|x-y|)$ and a $T_{\mathbf{L}}$-E_5-ordering as well. Thus we can define the following fuzzy relations from the fuzzy orderings L_2 (from Example 9), L_3, L_4, and L_5:

$$\begin{aligned}\mathbf{L}_a &= \mathrm{Lex}_{T_{\mathbf{M}},T_{\mathbf{L}}}(L_2,L_2)\\ \mathbf{L}_b &= \mathrm{Lex}_{T_{\mathbf{L}},T_{\mathbf{L}}}(L_3,L_2)\\ \mathbf{L}_c &= \mathrm{Lex}_{T_{\mathbf{P}},T_{\mathbf{L}}}(L_4,L_2)\\ \mathbf{L}_d &= \mathrm{Lex}_{T_{\mathbf{P}},T_{\mathbf{L}}}(L_5,L_5)\end{aligned}$$

[2] with $T_{\mathbf{P}}$ denoting the product t-norm

Theorem 11 then ensures that all these four fuzzy relations are fuzzy orderings with respect to the Łukasiewicz t-norm $T_{\mathbf{L}}$ and $T_{\mathbf{L}}$-equivalences defined as the corresponding Cartesian products. Figure shows cut views of the four lexicographic compositions, where we keep the first argument vector constant (we choose $(x_1,x_2) = (2,2)$) and plot the value $\mathbf{L}_*((2,2),(y_1,y_2))$ as a two-dimensional function of y_1 and y_2.

Now the question arises whether the lexicographic composition of two linear fuzzy orderings is again linear. Note that there are several notions of linearity of fuzzy orderings [6]. Let us first consider strong completeness.

Example 13. All fuzzy orderings considered in Examples 9 and 12 were strongly complete. Note, however, that none of the lexicographic compositions defined in Example 12 is strongly complete. To demonstrate that, consider the plots in Figure 3. These two plots show the values

$$\max\left(\mathbf{L}_a((2,2),(y_1,y_2)),\mathbf{L}_a((y_1,y_2),(2,2))\right)$$
$$\max\left(\mathbf{L}_d((2,2),(y_1,y_2)),\mathbf{L}_d((y_1,y_2),(2,2))\right)$$

as two-dimensional functions of y_1 and y_2. If $\mathbf{L}_a$ and $\mathbf{L}_d$ were strongly complete, these two functions would have to be the constant 1, which is obviously not the case. The same is true for the two other lexicographic compositions $\mathbf{L}_b$ and $\mathbf{L}_c$.

After this negative answer, let us relax the question a bit and attempt the question whether the lexicographic composition of two strongly complete fuzzy orderings is T-linear.

Proposition 14. *Let us make the same assumptions as for Theorem 11. If L_1 and L_2 are strongly complete fuzzy orderings and the residual negation N_T is involutive (i.e. $N_T(N_T(x)) = x$ holds for all $x \in [0,1]$), then the fuzzy ordering*

$$\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)$$

is T-linear.

Note that Proposition 14 also proves that all the four lexicographic compositions defined in Example 12 are $T_{\mathbf{L}}$-linear.

The proof of Proposition 14 does not work if we do not assume that N_T is an involution. The question arises, of course, whether this condition is not only sufficient, but also necessary. The answer is that this is the case, as the following example demonstrates.

Example 15. Consider a left-continuous t-norm for which a value $z \in]0,1[$ exists such that $N_T(N_T(z)) \neq z$. Since $N_T(N_T(z)) \geq z$ always holds, we can infer that, in this case, $N_T(N_T(z)) > z$ must hold. Now let us consider two simple strongly complete fuzzy orderings on the sets $X_1 = \{a,b\}$ and $X_2 = \{c,d\}$, respectively:

L_1	a	b
a	1	1
b	z	1

L_2	c	d
c	1	1
d	0	1

Then we can infer the following for any choice of $\tilde{T}$:

$$\begin{aligned}\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((a,d),(b,c)) &= N_T(L_1(b,a))\\ &= N_T(z)\\ \mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((b,c),(a,d)) &= L_1(b,a) = z\end{aligned}$$

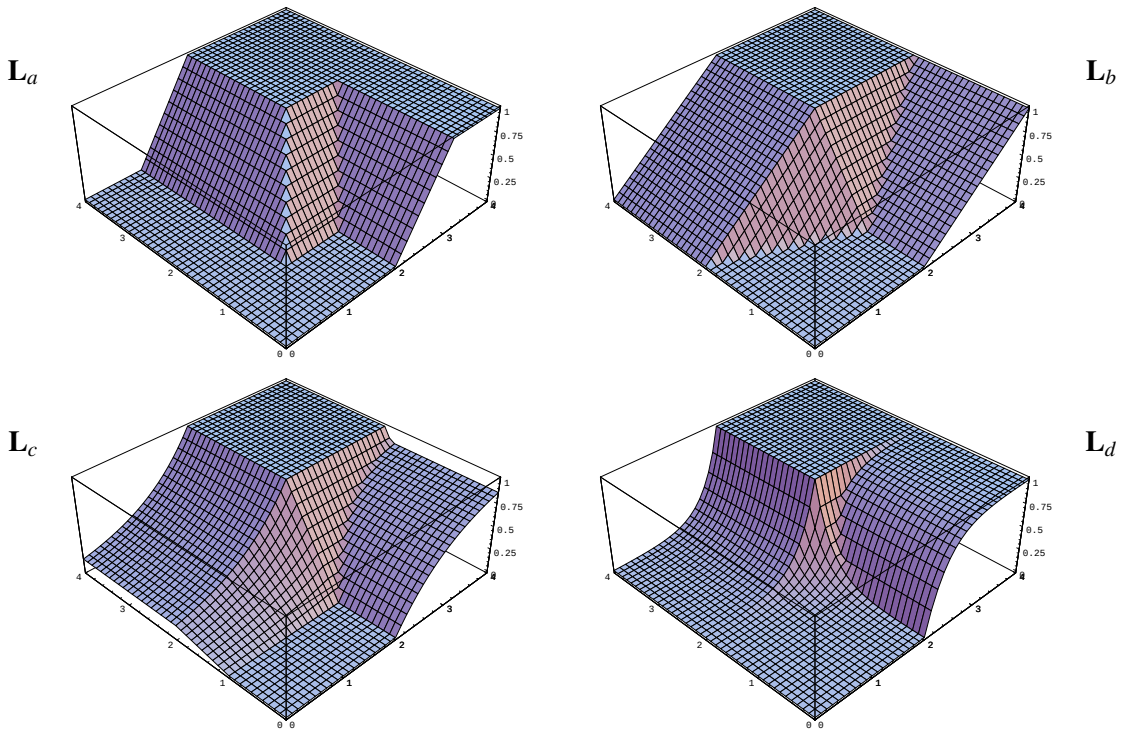

Figure 2: Cut views of the four lexicographic compositions from Example 12

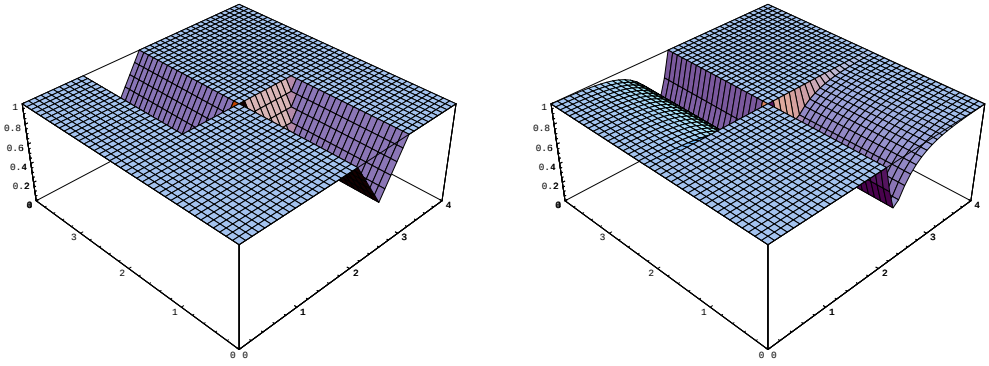

Figure 3: Plots of the functions $\max\left(\mathbf{L}_a((2,2),(y_1,y_2)),\mathbf{L}_a((y_1,y_2),(2,2))\right)$ (left) and $\max\left(\mathbf{L}_d((2,2),(y_1,y_2)),\mathbf{L}_d((y_1,y_2),(2,2))\right)$ (right)

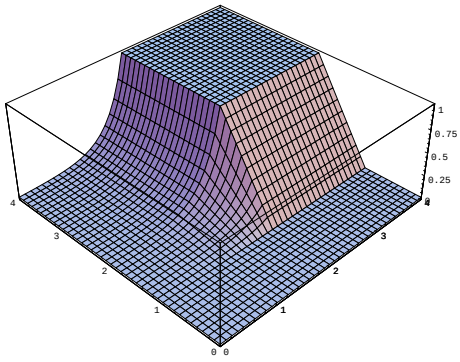

Figure 4: A cut view of $\mathrm{Lex}_{T_\mathbf{P},T_\mathbf{P}}(L_5,L_2)$

Hence we obtain that

$$\begin{aligned} &N_T\left(\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((a,d),(b,c))\right)\\ &\quad = N_T(N_T(z)) > z\\ &\quad = \mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((b,c),(a,d)), \end{aligned}$$

which shows that $\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)$ is not T-linear.

Note that the condition of involutiveness in particular excludes all t-norms without zero divisors. Therefore, lexicographic compositions of non-trivial (i.e. non-crisp) fuzzy orderings with respect to the popular minimum and product t-norms are problematic, if not meaningless. The reason for this is simple. As shown in [5], the only strict fuzzy ordering included in a fuzzy ordering that is strictly greater than zero (e.g. like L_4 and L_5 from Example 12) is the trivial zero relation. When it comes to lexicographic composition, the strict fuzzy ordering induced by the first component relation plays a crucial role. If it vanishes, no meaningful lexicographic composition that preserves linearity properties can be expected. As an example, see Figure 4. It shows a cut view of the fuzzy ordering $\mathrm{Lex}_{T_\mathbf{P},T_\mathbf{P}}(L_5,L_2)$. It is easy to see that $\mathrm{Lex}_{T_\mathbf{P},T_\mathbf{P}}(L_5,L_2)$ is nothing else but the Cartesian product of L_5 and L_2, which is of course not $T_\mathbf{P}$-linear.

The final and most important question is whether the lexicographic composition of two T-linear fuzzy orderings is again T-linear. Strong completeness always implies T-linearity [6], thus, strongly complete fuzzy orderings are a sub-class of T-linear fuzzy orderings (no matter which T we choose). If the involutiveness of N_T is a necessary condition for meaningful results in Proposition 14, there is no point in considering a t-norm that does not induce an involutive negation any further.

Theorem 16. *Let us again make the same assumptions as for Theorem 11. If L_1 and L_2 are T-linear fuzzy orderings and the residual negation N_T is involutive, then the fuzzy ordering*

$$\begin{aligned}&\mathrm{Lex}_{T_{\mathbf{M}},T}(L_1,L_2)((x_1,x_2),(y_1,y_2)) = \\ &\quad \max(\min(L_1(x_1,y_1),L_2(x_2,y_2)), \\ &\qquad \min(L_1(x_1,y_1),N_T(L_1(y_1,x_1))))\end{aligned}$$

is T-linear.

Obviously, Theorem 16 does not allow any choice of the aggregating t-norm $\tilde{T}$ as in the original construction in Theorem 11, but enforces the choice of the minimum t-norm (i.e. $\tilde{T} = T_{\mathbf{M}}$). This is not an arbitrary restriction, but a necessary condition, as the following example demonstrates.

Example 17. Consider an arbitrary left-continuous t-norm T that induces a strong negation N_T and assume that $\tilde{T} < T_{\mathbf{M}}$. Then there exists a $y \in]0,1[$ such that $\tilde{T}(y,y) < y$. Now let us consider the following two fuzzy relations:

L_1	a	b
a	1	y
b	1	1

L_2	c	d
c	1	y
d	$N_T(y)$	1

It is easy to see that L_1 and L_2 are T-linear fuzzy orderings with respect to T and some T-equivalences (the exact definition of them is not important at this point). Now we can compute:

$$\begin{aligned}&\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((a,c),(b,d)) \\ &\quad = \max(\tilde{T}(y,y),\min(y,N_T(1)) = \tilde{T}(y,y) \\ &\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((b,d),(a,c)) \\ &\quad = \max(\tilde{T}(1,N_T(y)),\min(1,N_T(y)) \\ &\quad = N_T(y)\end{aligned}$$

If $\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)$ was linear, the following inequality would be fulfilled:

$$\begin{aligned}&N_T\big(\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((b,d),(a,c))\big) \\ &\quad \leq \mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((a,c),(b,d))\end{aligned}$$

However, we obtain:

$$\begin{aligned}&N_T\big(\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((b,d),(a,c))\big) \\ &\quad = N_T(N_T(y)) = y > \tilde{T}(y,y) \\ &\quad = \mathrm{Lex}_{\tilde{T},T}(L_1,L_2)((a,c),(b,d))\end{aligned}$$

Therefore, $\mathrm{Lex}_{\tilde{T},T}(L_1,L_2)$ can never be T-linear if $\tilde{T} < T_{\mathbf{M}}$. This example, therefore, justifies the assumptions of Theorem 16.

5 Conclusion

In this paper, we have introduced an approach to lexicographic composition of similarity-based fuzzy orderings. This construction, in principle, works for all choices of t-norms. However, *the* essential property of lexicographic compositions—that the lexicographic composition of linear orderings is again a linear ordering on the product domain—is only maintained if the underlying t-norm T induces an involutive negation (in particular, including nilpotent t-norms and the nilpotent minimum). This once more confirms the viewpoint that such t-norms are most adequate choices in fuzzy relations theory, fuzzy preference modeling and related fields [5, 6, 8, 10, 21].

Acknowledgements

The author gratefully acknowledges 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. Support by COST Action 274 "TARSKI" is also gratefully acknowledged.

References

[1] R. Bělohlávek. *Fuzzy Relational Systems. Foundations and Principles*. IFSR Int. Series on Systems Science and Engineering. Kluwer Academic/Plenum Publishers, New York, 2002.

[2] U. Bodenhofer. *A Similarity-Based Generalization of Fuzzy Orderings*, volume C 26 of *Schriftenreihe der Johannes-Kepler-Universität Linz*. Universitätsverlag Rudolf Trauner, 1999.

[3] U. Bodenhofer. A similarity-based generalization of fuzzy orderings preserving the classical axioms. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 8(5):593–610, 2000.

[4] U. Bodenhofer. Representations and constructions of similarity-based fuzzy orderings. *Fuzzy Sets and Systems*, 137(1):113–136, 2003.

[5] U. Bodenhofer and M. Demirci. Strict fuzzy orderings in a similarity-based setting. In *Proc. Joint 4th Conf. of the European Society for Fuzzy Logic and Technology and 11 Recontres Francophones sur la Logique Floue et ses Applications*, pages 297–302, Barcelona, September 2005.

[6] U. Bodenhofer and F. Klawonn. A formal study of linearity axioms for fuzzy orderings. *Fuzzy Sets and Systems*, 145(3):323–354, 2004.

[7] D. Boixader, J. Jacas, and J. Recasens. Fuzzy equivalence relations: Advanced material. In D. Dubois and H. Prade, editors, *Fundamentals of Fuzzy Sets*, volume 7 of *The Handbooks of Fuzzy Sets*, pages 261–290. Kluwer Academic Publishers, Boston, 2000.

[8] B. De Baets and J. Fodor. Towards ordinal preference modelling: the case of nilpotent minimum. In *Proc. 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume I, pages 310–317, Paris, 1998.

[9] B. De Baets and R. Mesiar. T-partitions. *Fuzzy Sets and Systems*, 97:211–223, 1998.

[10] B. De Baets, B. Van de Walle, and E. E. Kerre. A plea for the use of Łukasiewicz triplets in the definition of fuzzy preference structures. (II). The identity case. *Fuzzy Sets and Systems*, 99(3):303–310, 1998.

[11] M. Demirci. A theory of vague lattices based on many-valued equivalence relations—I: general representation results. *Fuzzy Sets and Systems*, 151(3):437–472, 2005.

[12] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, 1994.

[13] P. Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 1998.

[14] U. Höhle and N. Blanchard. Partial ordering in L-underdeterminate sets. *Inform. Sci.*, 35:133–144, 1985.

[15] F. Klawonn and J. L. Castro. Similarity in fuzzy reasoning. *Mathware Soft Comput.*, 3(2):197–228, 1995.

[16] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000.

[17] S. V. Ovchinnikov. Similarity relations, fuzzy partitions, and fuzzy orderings. *Fuzzy Sets and Systems*, 40(1):107–126, 1991.

[18] S. V. Ovchinnikov and M. Roubens. On strict preference relations. *Fuzzy Sets and Systems*, 43:319–326, 1991.

[19] S. Saminger, R. Mesiar, and U. Bodenhofer. Domination of aggregation operators and preservation of transitivity. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 10(Suppl.):11–35, 2002.

[20] E. Trillas and L. Valverde. An inquiry into indistinguishability operators. In H. J. Skala, S. Termini, and E. Trillas, editors, *Aspects of Vagueness*, pages 231–256. Reidel, Dordrecht, 1984.

[21] B. Van de Walle, B. De Baets, and E. E. Kerre. A plea for the use of Łukasiewicz triplets in the definition of fuzzy preference structures. (I). General argumentation. *Fuzzy Sets and Systems*, 97(3):349–359, 1998.

[22] L. A. Zadeh. Similarity relations and fuzzy orderings. *Inform. Sci.*, 3:177–200, 1971.