Intelligent Image Processing for Liquid Etching Processes

Matthias Koch, Jens Knobloch
Fraunhofer-Institute for Microelectronic Circuits and Systems
Grenzstr. 28, D-01109 Dresden, Germany
Phone: +49-351-8823279, Fax: +49-351-8823266
email: {koch,knobloch}@imsdd.fhg.de

ABSTRACT: To determine the exact duration of liquid etching processes on spin-etchers an add-on device was developed which analyses optical properties of the wafer surface through a camera. The image processing is performed on normal PC hardware. A modular software structure is described, and the results of applying fuzzy methods in various parts of the algorithm are presented.

KEYWORDS: fuzzy logic, image processing, end-point detection

1 PROBLEM DESCRIPTION

As wafer diameters rise costs for raw wafers increase more than proportionally. Therefore reclaiming of wafers after defective preparation becomes increasingly important. This is usually performed with spin etchers. An etching agent is dispersed over a rotating wafer until all traces of the layer to remove have vanished. Because of the rising wafer diameters spin etchers replace step-by-step conventional etch baths. To minimize usage of the etching agent which is not only very expensive but also strains the environment, and to maximize throughput which reduces production costs it is important to determine the duration of the etching process exactly. Up till now a process engineer watches the progress of the etching of a test wafer and establishes the desired duration based on visual effects on the wafer. For a fixed number of wafers the etching agent is re-used during each etching process which leads to continually decreasing concentrations. This is corrected by rising the etching time by a fixed amount for each processed wafer. After the predefined number of wafers has been processed the recycling of the etching agent is switched off and a constant etching time is applied. This procedure enforces high stability of temperature, concentration, and flow-rate of the etching agent. To compensate random variations all wafers have to be etched longer than necessary. This not only requires more time and etching agent thus increasing cost, but also makes liquid etching not suitable for processes with sensitive stopping layers.

PROTOTYPE STRUCTURE

To implement the prototype the analysis of image data from a digital camera was chosen. As a result the end-point detection system can imitate the actions of a process engineer and use already acquired process knowledge. Additionally direct observation of the current process is possible by displaying the camera signals on the computer monitor. This is especially useful when using etching machines for larger wafer diameters.

In fig. 1 a typical camera picture of an unmoving wafer on the chuck of an etching machine is shown. In addition to the dispenser hiding part of the wafer surface and the prominent mirror effects we encounter distortions from motion blur and uneven as well as changing distributions of the etching agent during the etching process.

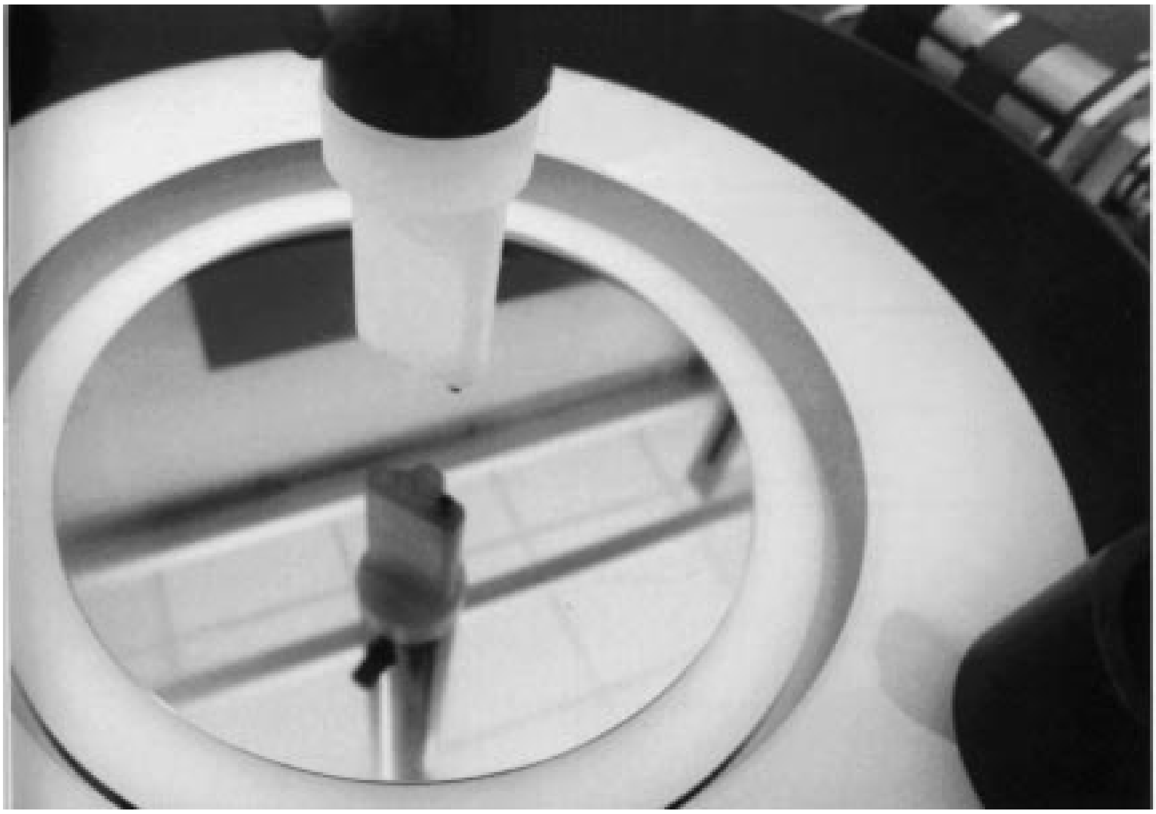

Figure 1: Still image of a wafer inside the etching machine

REQUIREMENTS FOR THE END-POINT DETECTION

Independently of the used etching agent the end point should be detected for a possibly wide spectre of processes. Though the etching process varies from wafer to wafer the accuracy of the end-point detection should lie below 0.2 seconds. The complete algorithm should be really fast to handle as many images per second as possible to ensure continuous supervision even for rapid etching processes. Regardless of disturbances on the wafer surface or deviations in the process course the end point has to be determined without failure.

2 BASIC SYSTEMS DESIGN

Our target was to develop an add-on device for a spin-etcher which would allow the detection of the end point of a liquid etching process by image processing. The fundamental structure of the prototype is shown in fig. 2. The camera signal is fed into a frame grabber and can be displayed directly on the computer monitor or is available for software manipulation. Communication with the controller of the spin etcher is performed through a serial port. The image processing step extracts the relevant wafer area from a single camera picture. The pixels in this area serve as raw material to calculate the features significant for the actual etching process. In the fuzzy recognition stage current values of several features and their history are analysed to determine the exact time of the end point.

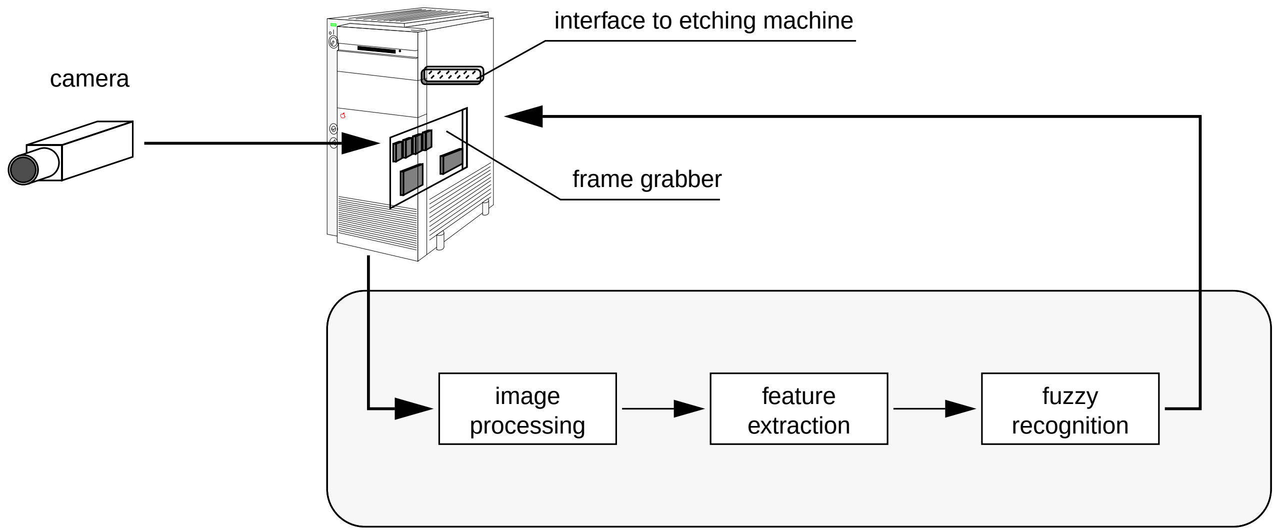

Figure 2: Systems overview

EXAMPLE PROCESS DATA

Process data for several etching steps were acquired. As a typical example fig. 3 shows the etching of an aluminium layer on a glass substrate. During a start phase the signal can assume irregular values. After the 97th second the fluctuations decrease below a certain level and a so-called plateau phase begins. During this phase the first derivation of the feature over time is approximately constant. The begin of the end phase during which the structures on the wafer are opening is characterized by a strong change in the first derivative (140th second). The end point is reached approximately in the 147th second. During a short over-etching phase the feature signal is largely constant. In the 150th second the wafer is raised to an upper processing level inside the spin-etcher to clean it with water.

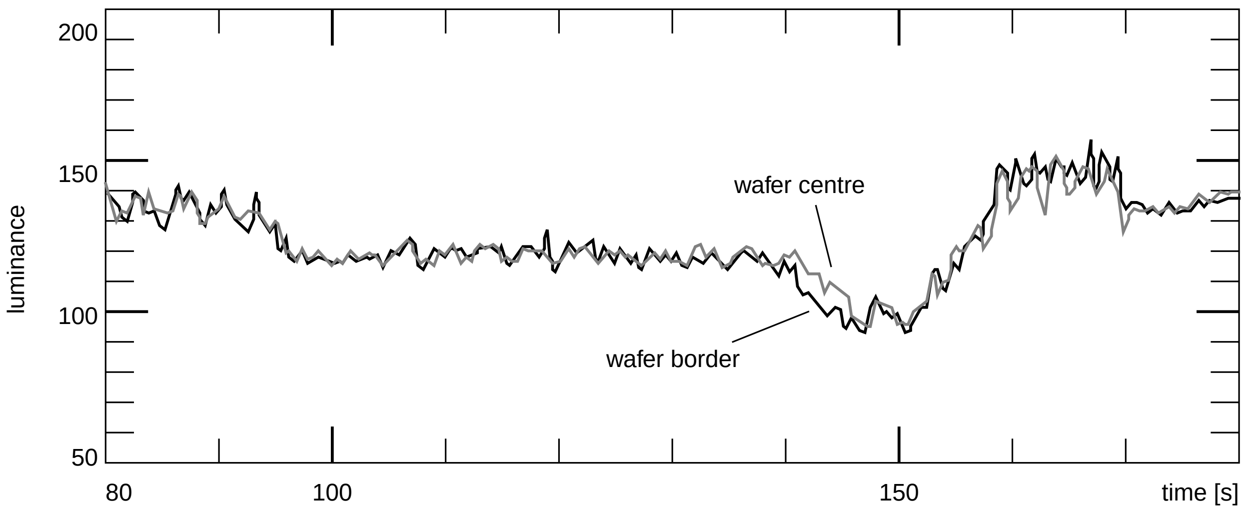

Figure 3: Exemplary curve of an etching process

PHASE MODEL

Generalizing these data we assumed that each etching process feature can be represented by a phase model according to fig. 4. After an erratic signal course during the start phase the 1st derivative of the signal over time assumes a constant positive or negative value with only small fluctuations (plateau phase). During the end phase (which may not be present in every process) the derivative is rather large. The end point is reached if the derivative reaches the null point (with an adjustable error margin). Signals with other characteristics can be transformed to match the required characteristic.

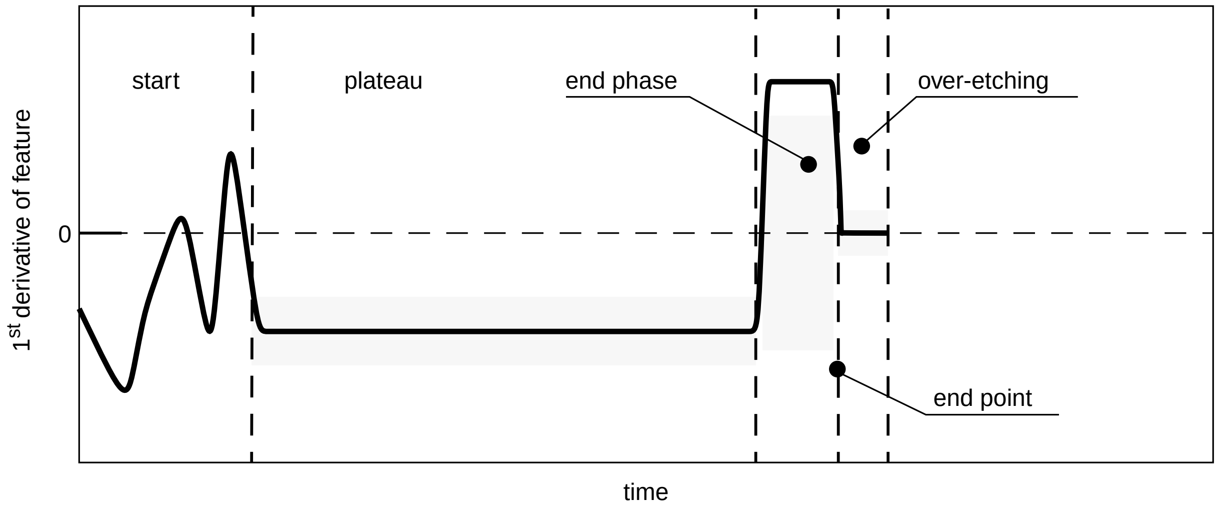

Figure 4: General phase model for etching processes

3 THE GENERAL ALGORITHM

Fig. 5 shows a part of the software representing the flow graph to process a single image during the etching process. Distortions in the camera image are excluded by overlaying a mask constructed with image processing algorithms. The remaining pixels enable the generation of a wide number of features (e. g. brightness, difference between red and blue channels, chrominance distribution over the wafer radius). Because of unified interfaces to preceding and following software routines the user can choose the most suited features from a modular system of methods. Each feature is passed to an analyser where independently of other features the membership degree to the fuzzy set “the end point is reached” is determined based upon the current value of the feature and its history. At the same time the reliability of this membership is estimated. The results of the different analysers are passed to a kind of majority operator where they are merged into an unambiguous decision whether the end point was reached or not.

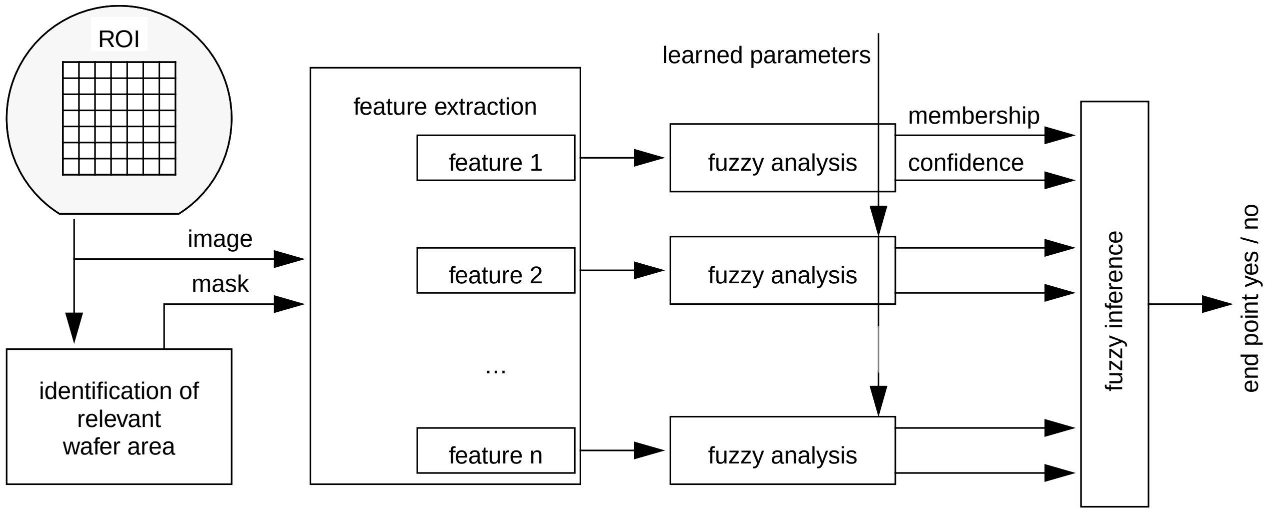

Figure 5: Software structure

Besides the part described above the software implements a learning and supervising system and a graphical user interface.

IDENTIFICATION OF RELEVANT WAFER AREA

A simple possibility to identify the relevant wafer area consists of a partition of the region of interest into rectangular sub-areas. Each of this sub-areas can be classified with a minimal number of features whether it is disturbed or not. A fuzzy classifier constructed from sample data is shown in fig. 6. Though the separation between the classes is not very good a rather high classification rate was achieved.

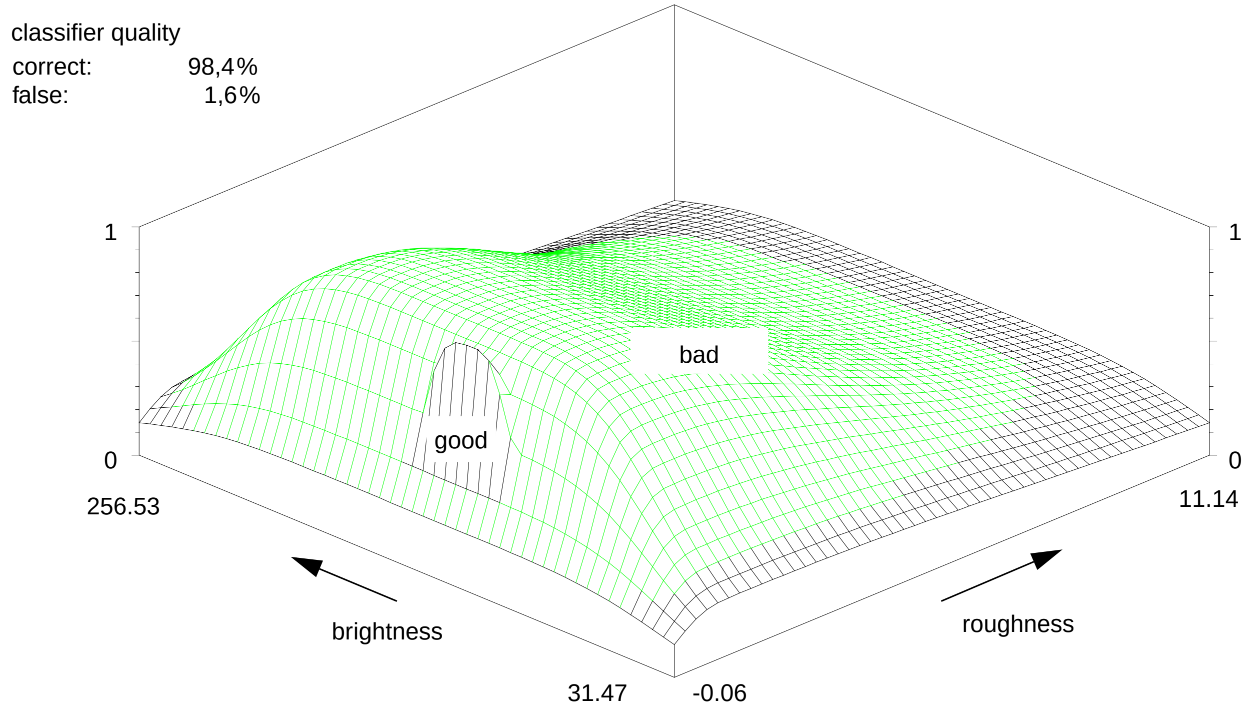

Figure 6: Class view of the constructed classifier

Unfortunately this method is process-dependent to a high degree, so it had to be discarded.

In the end an image processing algorithm was implemented which by threshold discrimination identifies the position of the dispenser in the image and selects one of a set of predefined masks. In comparison with the fuzzy approach this method achieves a similar speed but cannot suppress disturbances from the etching agent on the wafer surface.

4 CLASSIFICATION STEPS

Each feature analyser consists of three fuzzy classifiers (one for each of the start, plateau, and end phases) which independently determine the membership of the signal to one of the four phases. Each classifier assesses its own phase as well as the preceding and following ones.

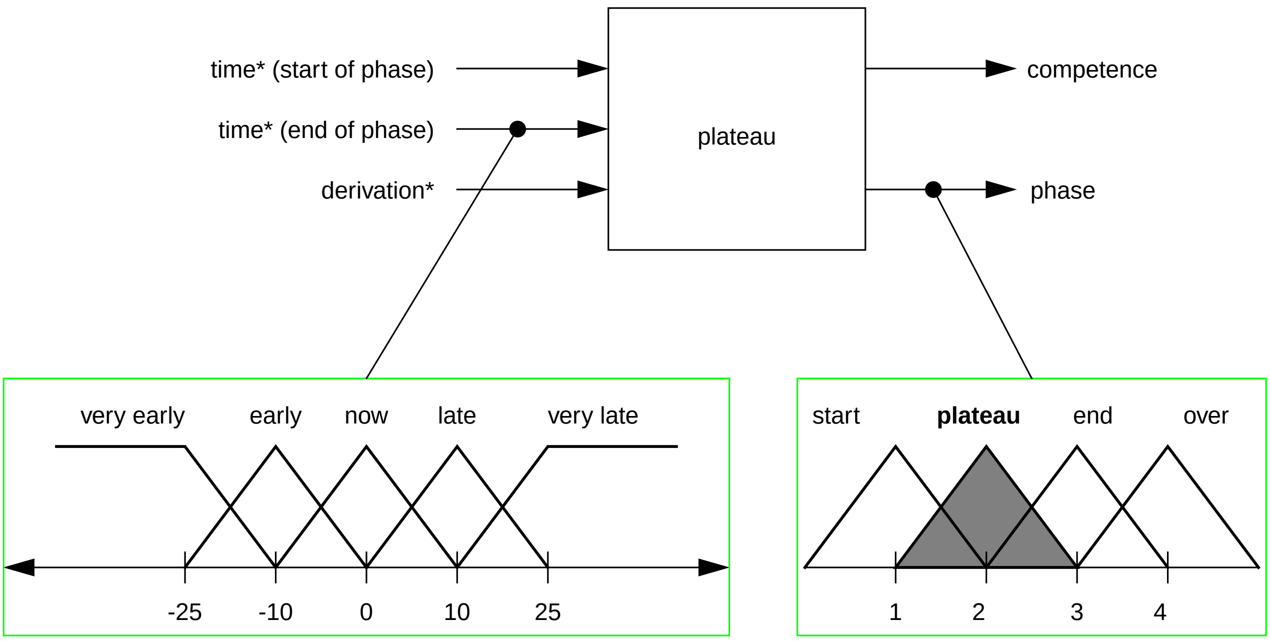

Figure 7: Fuzzy classifier for the plateau phase

Fig. 7 shows the fundamental classifier structure at the example of the plateau phase. One of the boundary conditions was the flexible mapping of discrete values to the linguistic variables without the aid of special fuzzy development tools. Therefore to the inputs of the classifier transformed signals are applied which are scaled by expected values for phase-change time and derivation and the interval size for a "small" divergence. The output variables of all classifiers are structured in the same way. For each term of the linguistic variable "phase" a membership between 0 and 1 is calculated. Additionally each classifier delivers a value between 0 and 1 describing its competency for the current phase.

In addition to the classifiers each analyser contains an arbitration logic which combines the results of the individual classifiers. Each analyser outputs three signals: membership to the class "end point reached", membership to the class "end point not reached" and a confidence degree for the membership values.

commutativity	$\mu_a \otimes \mu_b = \mu_b \otimes \mu_a$
associativity	$(\mu_a \otimes \mu_b) \otimes \mu_c = \mu_a \otimes (\mu_b \otimes \mu_c)$
neutral value	$0.5 \otimes \mu = \mu$
intensification	$\mu_a \otimes \mu_b \geq max(\mu_a, \mu_b)$, if $\mu_a > 0.5$ and $\mu_b > 0.5$ $\mu_a \otimes \mu_b \leq min(\mu_a, \mu_b)$, if $\mu_a < 0.5$ and $\mu_b < 0.5$
symmetry	$\mu_a \otimes \mu_b = 0.5$, for $\mu_a = 0.5 + x, \mu_b = 0.5 - x$ with $x \in [0, 0.5]$
propagation of total membership	$0 \otimes \mu = 0$, for $\mu \neq 1$ $1 \otimes \mu = 1$, for $\mu \neq 0$

Figure 8: Requirements for an aggregation operator

The results of the fuzzy analysers have to be transformed into an unambiguous decision. The requirements for an appropriate aggregation operator are shown in fig. 8.

5 CURVE CLASSIFICATION

In a later stage of the prototype development we acquired process data which cannot be so easily transformed to match the requirements of the phase model from fig. 4. As an example fig. 9 shows the etching of a nitride layer on a silicon substrate where as the nitride layer's thickness decreases its colour assumes repetitively the same values. In order to solve this problem and to allow for other unforeseen processes we decided to discard the phase model and implement a fuzzy classification of the feature curve.

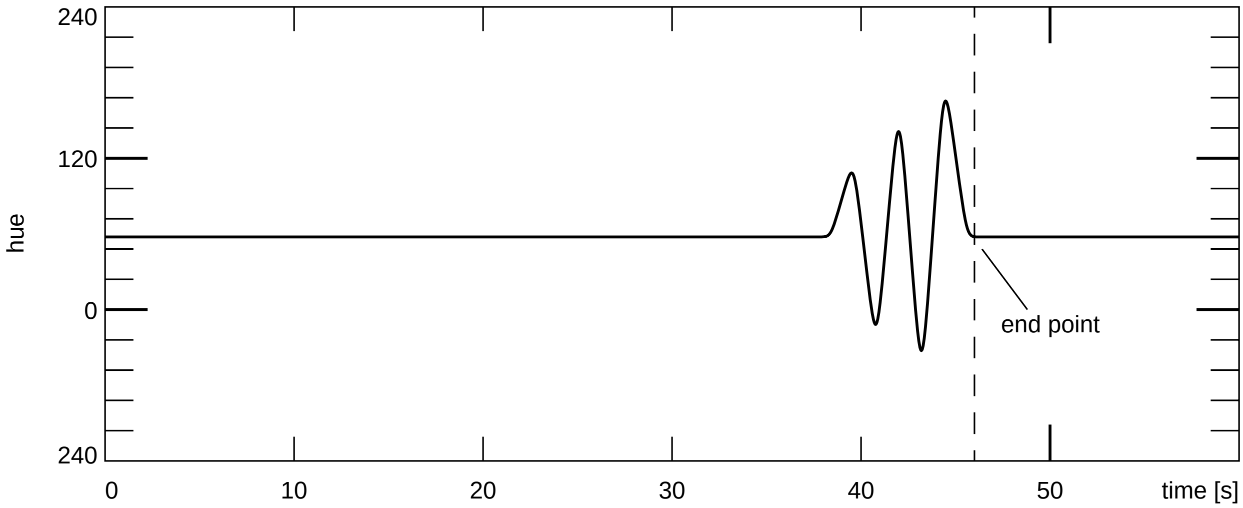

Figure 9: Basic characteristic for nitride etching

To detect the end point the feature curve is divided into several characteristic segments which can be compared with corresponding segments of a pattern curve. Before the actual classification can take place the raw data must be preprocessed to suppress disturbances. The consecutive processing stages are filtering, calculation of the 1st derivative, smoothing of the derivative, detection of the curve type and comparison with the pattern curve.

SIGNAL CONDITIONING

Filtering is used to suppress signal variations caused by noise. Significant signal changes should nevertheless be visible at the filter output. These requirements contradict each other; each actual filter would be a compromise between both demands. As a further restriction for the end-point detection the signal delay should be as small as possible. The latter demand excludes classical low-pass filters. If we assume that the noise is significantly smaller than meaningful signal changes we can apply window filters. The implemented algorithm compares each new data sample with the preceding filtered value. If their difference remains below an adjustable threshold the filtered signal remains constant. Otherwise the window is shifted accordingly and the filter output becomes the middle of the window height. The filter works best if the signal/noise ratio is rather large. The delay time is equal to one sample interval. By choosing the appropriate window height small irrelevant signal changes can be totally suppressed. The limit of the filter principle is reached when the noise is nearly as large as the meaningful signal changes.

The curve classification differentiates between rising, falling, and constant curve segments. Therefore the 1st derivative of the curve over time has to be calculated. In our prototype we implemented a simple difference quotient with the current and the preceding filtered feature values to maximize processing speed.

The derivative requires smoothing for two reasons: once, there may appear singular constant values inside an otherwise continuously rising (or falling) signal edge caused by the filtering algorithm. On the other hand the derivative becomes spiky if the noise part in the signal increases. Such a torn derivative curve is not suited for classification. We have implemented a minimal smoothing by calculating the mean value of two consecutive sample points. For further smoothing a value of n points can be chosen by the process engineer. However, this leads to an increase of the filter delay time by $n/2$ sample points.

The classification of each point of the smoothed derivative is done applying a user-defined tolerance threshold. If the absolute value of the derivative lies within the threshold the curve is considered *constant*. For values outside this range the curve is considered *rising* for positive values and *falling* for negative ones.

When processing noisy signals one can work with lesser smoothing of the derivative and therefore shorter filter delay if the tolerance threshold is chosen appropriately.

PATTERN COMPARISON

All sample points of the same type build a curve segment. Curve segments are defined by their duration, additionally non-constant curve segments are characterized by the height difference between start and end points.

For end-point detection purposes the etching process is divided into several distinct phases: start phase, preconditions, end phase, and over-etch phase. The individual phases may not be present in every type of etching process.

During the start phase no signal processing is performed. Thereby irregular signal values and undetermined large signal deviations can be suppressed for a user-defined interval at the beginning of the etching process.

Preconditions are specified if the etching process must pass several characteristic curve segments (e. g. three large rising edges) before the end-point condition is reached. They may be utilized to enlarge the certainty of the end-point detection, or for complicated feature trajectories they allow, for instance, the detection of a small falling curve segment after two large falling ones. By specifying a repetition factor more than one equivalent precondition can be detected. The fulfilment of a precondition is tested at the end of each respective curve segment. If all of the requested preconditions have occurred the process passes to the end phase.

During the end phase the current curve segment is compared with similar criteria (segment type, duration, height) as throughout the preconditions. The single difference being that the testing for the end conditions is performed with each new sample. This is especially useful if the end point is defined by a curve segment without changes following various different curve segments.

During the over-etch phase the process is continued for a certain user-defined time to remove hardly recognizable layer traces certainly.

6 CONCLUSIONS

The prototype of the image processing algorithm was implemented on a PC under Windows NT. Approximately 10000 lines of C++ code were written. The feature-processing part of the algorithm needed only 0.35 ms per image for one feature on a Pentium-166 processor. The major part of the processing time is needed to transfer the image data into the computer storage.

The finally implemented curve-classification algorithm proved to be highly flexible and reliable. It showed good results for etching aluminium on glass substrates and nitride on silicon wafers. It should be well suited for other processes, too, as tests with a number of artificially generated data demonstrated. Even with a signal/noise ratio of up to 10% the end-point was successfully detected. The crucial element may be the selection of the most suited image features. But, with the modular concept new methods can be easily incorporated.