Automated inspection of microlens arrays

James Mure-Dubois and Heinz Hügli

University of Neuchâtel - Institute of Microtechnology, 2000 Neuchâtel, Switzerland

ABSTRACT

Industrial inspection of micro-devices is often a very challenging task, especially when those devices are produced in large quantities using micro-fabrication techniques. In the case of microlenses, millions of lenses are produced on the same substrate, thus forming a dense array. In this article, we investigate a possible automation of the microlens array inspection process. First, two image processing methods are considered and compared: reference subtraction and blob analysis. The criteria chosen to compare them are the reliability of the defect detection, the processing time required per frame, as well as the sensitivity to image acquisition conditions, such as varying illumination and focus. Tests performed on a real-world database of microlens array images led to select the blob analysis method. Based on the selected method, an automated inspection software module was then successfully implemented. Its good performance allows to dramatically reduce the inspection time as well as the human intervention in the inspection process.

Keywords: industrial inspection, microlens, blob analysis, automated inspection

1. MICROLENS ARRAYS INSPECTION

Inspection of microlens arrays produced through parallel microfabrication techniques borrowed from semiconductor technology is a task for which a convenient solution has not yet been developed, as pointed out by researchers[1] or industries[2] active in this field. In this paper, we investigate the automation of the inspection process through image processing techniques.

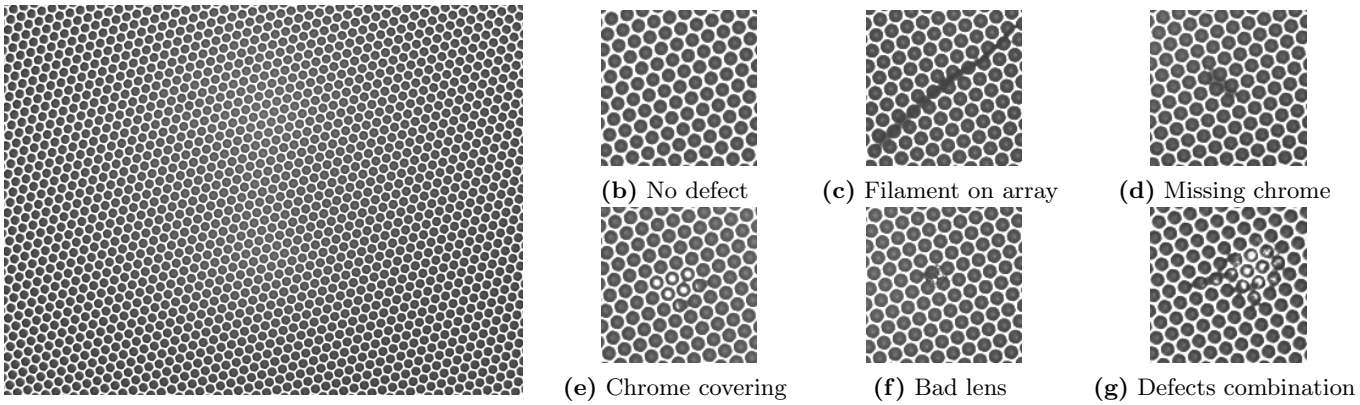
1.1 Inspection task

Microlens arrays are optical devices formed by a large number of small lenses, and are used in many applications including collimating, illuminating and imaging[3]. The work presented here concerns the inspection of devices with more than 2 millions lenses, with the specificity that gaps between lenses are coated with a reflective metal. In the inspection configuration considered here, the array is observed in reflection under a low magnification microscope, with an attached video camera, and the goal of the inspection is to spot defective lens or defects in the metal coating. Figure 1 shows a typical test image. The typical diameter of a lens is a few tens of μm, so that the microscope field of view covers more than 2000 lenses. Since a macroscopic device must be inspected, an xy platform is used to move the device under the microscope, and one image is acquired for each position. The total number of images acquired for each device is larger than 1800 (a certain amount of overlap between neighbor positions allow to ensure complete coverage). In the standard inspection procedure, a human operator examines each image, trying to identify and count the defects in the microlens array, in order to ascertain its quality.

We investigated the automation of this inspection task. The motivations are first to relieve the human operator from the strain of watching the series of images, then to increase the reproducibility of the inspection procedure, and finally to reduce the inspection time. More specifically, we focused on developing an automated defect detection process. This process allows a semi-automated inspection, where the human operator needs only to examine images containing defects.

1.2 Overview of defective samples

A small set of defects observed on real microlens devices is now presented. Those defects range from foreign particles stuck on the array (1**c**) to incomplete metal coating (1**d**), metal coating on the lenses, (1**e**), bad lens shape (1**f**), or combinations of all those defects (1**g**).



(a) Typical input image

(b) No defect **(c)** Filament on array **(d)** Missing chrome

(e) Chrome covering **(f)** Bad lens **(g)** Defects combination

Examples of observed defects

Figure 1. Input images for the inspection task

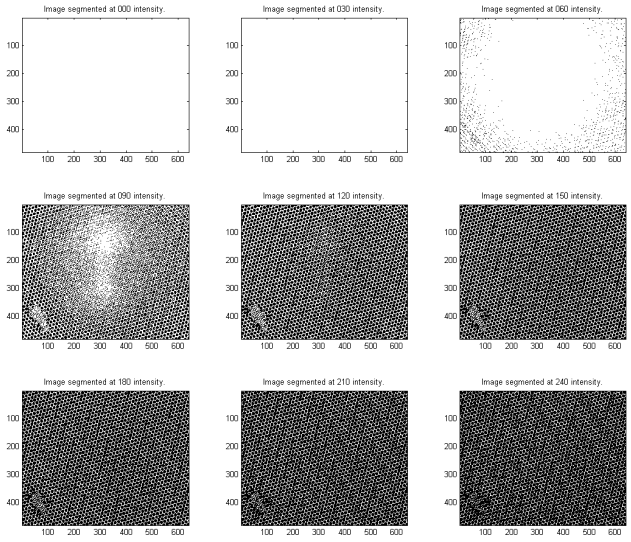


Figure 2. Illustration of vignetting : the input image is binarized with 9 different threshold values. The third and fourth image in the series clearly illustrate vignetting : for the same objects, intensity is lower in image corners than in the center.

Table 1. Main features for automated defect detection

Advantage	Challenge
⊕ Contrast is high (metal coating)	⊖ Illumination may vary (gradients + vignetting)
⊕ Reference image I_r available	⊖ No alignment between array lattice and image axes
⊕ Lens shape and aspect uniform throughout the array	⊖ Defects may vary greatly in size and intensity characteristics
⊕ Binary decision only (presence of defects)	⊖ Short processing time ($< 1\ s$)

1.3 Automated defect detection system

The defect detection system must spot all defects, independently of their size and intensity characteristics. Table 1 lists some key features to consider for this system. The main performance criterion for the detector is its *false negative rate* : devices where defects are present should be reliably reported. The defect detection stage operates a binary decision : either the submitted test image I_t contains no defect, in which case it is flagged as good, or one (or more) defect is found, and the image is added to the series of image presented to a human supervisor, for later inspection and defect characterization. The *false alarm rate* must be kept low in order limit the strain on the human supervisor. Finally, another desirable property would be for the detector to be easily *extensible*, in order to be used in different contexts (for example, lens arrays with different lens shapes, or lens arrays without metal coatings).

Different image processing options for the automated defect detection procedure were studied (section 2). Due to time constraints, only one of them was implemented (section 3). The resulting semi-automated inspection system based on this implementation is discussed in section 4, which includes a performance evaluation for the automated defect detection procedure.

2. INSPECTION METHODS

2.1 Reference subtraction

The first strategy considered is reference subtraction. In this approach, the absolute difference between the test image I_t and a previously recorded reference image I_r is computed :

$$I_d(i,j) = |I_t(i,j) - I_r(i,j)| \tag{1}$$

Any defect present in the test image is put in evidence by the difference operation. This method is economical in terms of processing time and memory requirements but, unfortunately, shows major drawbacks for the inspection task considered. First, using this method requires the test image I_t to be precisely aligned with the reference image I_r, both in translation and in rotation. This condition is very difficult to satisfy in the implementation of the inspection system. This issue is amplified by the fact that the digital image resolution is low for the objects observed. Therefore, effects related to coarse image sampling may be confused with genuine defects, leading to a high number of false alarms, as illustrated in fig. 3. The figure shows the difference image $I_t - I_r$ for 9 different translations of the reference image I_r. Ideally, the only active regions in this image should be caused by defects (such as those present in the lower left corner. However, the coarse sampling causes periodic structures to appear in the difference image, which becomes therefore cluttered with signals which do not correspond to lens defects. Finally, reference subtraction is sensitive to parasitic signals such as illumination gradients between different regions of the device under inspection. All these elements indicate that reference subtraction is not a promising candidate for this specific inspection problem.

2.2 Blob analysis

The second strategy considered is blob analysis. In the specific case of reflection images on a metal background, lens can be easily segmented. Moreover, each lens in the array satisfies precise constraints regarding its geometry (size, shape). Therefore, it is possible to spot defects by :

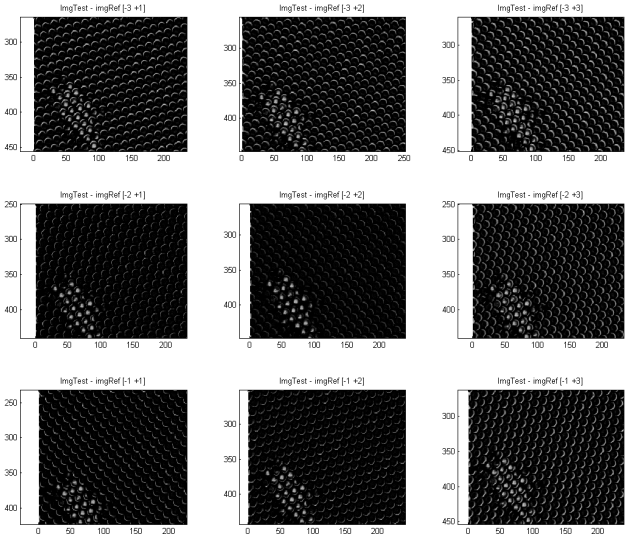


Figure 3. Coarse sampling issue in reference subtraction: difference image $I_t - I_r$ for 9 different translations of the reference image I_r. Sampling artifacts are mixed with genuine error signals (lower left corner).

1. Segmenting and labelling the image to detect all lenses-like shapes.
2. Verifying that the labelled shapes fall within the tolerances set for a correct lens.

The main advantage of this approach is that no alignment of the test image is required. Moreover, since a blob is defined for each lens, it is possible to extend the inspection process by checking various geometrical properties (diameter, circularity, etc). Similarly, this approach can be easily adapted for lens arrays with different lens shapes (e.g. polygonal rather than circular). For circular lenses, this approach is relatively robust with respect to the noise induced by the coarse sampling: the shape of the segmented lens stays roughly circular, and the uncertainty in the measured diameter stays low. The main drawback of this approach is related to the binarization required before blob labelling. Any error introduced in this step is difficult to compensate in later processing steps. This issue could be critical in case of large intensity gradients within the image, or if the image presented a bad contrast (for example if no metal coating is present between the lens). For the problem presented here, simple binarization with a global threshold is sufficient. Moreover, this issue could be taken into account by using more complex binarization techniques, using for example adaptive thresholding (to attenuate the role of illumination gradients), or edge finder based on the first derivatives (if the contrast in absence of metal is too low).

2.3 Methods comparison

The methods listed above can be compared regarding to their ability to cope with the challenges listed in table 1. The results of this comparison are presented in table 2. As noted above, the main disadvantage of the reference subtraction method is that it requires accurate alignment of test and reference images, whereas blob analysis has no such constraint. Therefore, we choose to use the blob analysis approach in our development of an automated defect detection process.

3. PROTOTYPE BLOB ANALYSIS SYSTEM

3.1 Prototype architecture

The structure of an automated defect detection process based on blob analysis is presented. In order to reduce development time and to allow further evolution, the process was implemented in Matlab®.[4]

Table 2. Applicability of image processing methods

Challenge	Reference sub.	Blob analysis
⊖ Illumination may vary (gradients + vignetting)	−	0
⊖ No alignment between array lattice and image axes	−−	++
⊖ Defects may vary greatly in size and intensity characteristics	++	++
⊖ Short processing time ($< 1\ s$)	++	+

The basic idea followed here is to build blobs corresponding to lenses, and then to check if those blobs comply with all requirements for a defect free lens. Practical considerations (see sec. 3.1.1) required to also process blobs corresponding to highly reflective regions. For both type of blobs, the processing is as follows :

- Segment the image to obtain blobs
- Remove noise with basic morphology operations
- Detect and label blobs
- Compute blob features
- Compare blob features to criteria set for defects

The defect detection process marks all images containing defects for review by a human operator in the semi-automated inspection system.

3.1.1 Binarization

In the test images, the metal background is bright, since it reflects the incoming light. In comparison, lenses appear brighter, except in their top area, where specular reflection occurs. The first processing step is therefore a binarization of the grayscale test image I_t. A global threshold θ is used, and two complementary binary images are formed : B_l showing dark areas (lens bodies) and B_m showing bright areas (metal + lens tops), according to the rule :

$$B_l(i,j) = \begin{cases} 1 & \text{if} \quad I_t(i,j) < \theta \\ 0 & \text{if} \quad I_t(i,j) \geq \theta \end{cases} \qquad B_m(i,j) = \bar{B}_l(i,j) \tag{2}$$

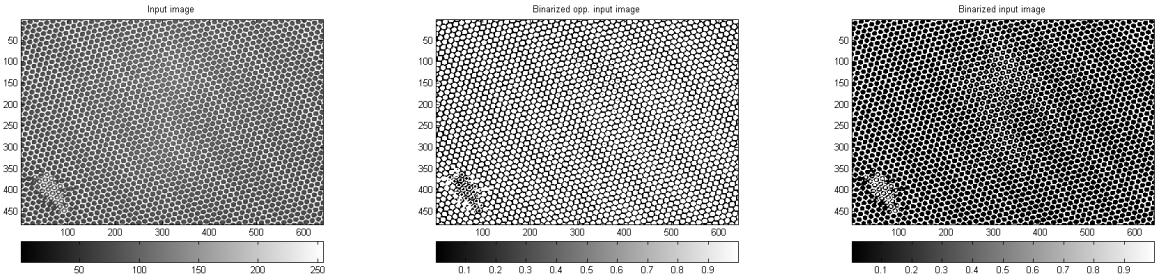
Typical binarization results are presented in fig. 4. The optimal threshold is determined once by a human operator after observation of the binarization results for different values of θ, as illustrated in fig. 2. This figure illustrates the typical illumination gradients observed in the practical inspection system. The low severity of those illumination gradient does not, however, require to use more advanced thresholding techniques (such as adaptive thresholding).[5]

3.1.2 Denoising

In order to avoid artifacts in the subsequent labelling operation, an opening operation with a square 3×3 kernel is applied on the binary lens image B_l. The image for metal and lens tops areas B_m is left unmodified.

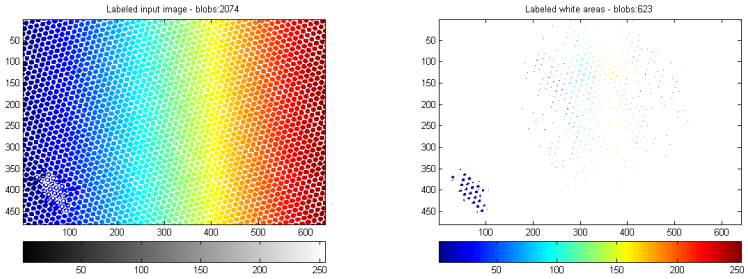
3.1.3 Blob labeling

Blobs, defined as a $V8$ connected region, are detected and labeled in the lens image B_l and in the metal image B_m. We call N_l (resp. N_m) the number of blobs found in the binary image B_l (resp. B_m). Typically in this inspection system, we have $N_l > 2000$ and $N_m > 250$ (fig. 5). In the image B_m, the largest blob corresponds always to the metal region separating the lenses. This blob is deliberately removed from further processing. Other smaller blobs, however, can indicate presence of metal covering lenses. Therefore, all small blobs are labeled on the metal image (fig. 5**b**).



(a) Input image **(b)** B_l : lens bodies **(c)** B_m : metal + lens tops

Figure 4. Image binarization



(a) Lens body blobs **(b)** Lens top blobs

Figure 5. Blobs labelling

3.1.4 Blob features

Each blob in B_l and B_m must be automatically analyzed and classified. Quantitative features describing the blobs must be computed, in order to distinguish between valid lenses and defects, respectively between legitimate metal regions and defects. Features such as diameter, elongation, etc. could be computed for each blob. In the current implementation, only the area feature A is considered. This choice keeps the processing time low. As we will see in the next section, the area feature is sufficient for a successful defect detection.

3.1.5 Defect classification and characterization

Blobs in the binary lens image B_l should all be caused by valid lenses. A defect is reported when a blob is too large or too small to be a valid lens. Additionally, specular reflections from lens tops cause only small blobs in the metal image B_m. Large blobs in this image are also caused by defects. The classifier therefore requires three parameters :

- maximum blob area for a valid lens $A_{max,l}$
- minimum blob area for a valid lens $A_{min,l}$
- maximum blob area for a blob in the metal image $A_{max,m}$

In order to spot defects where chrome is missing between two lenses, $A_{max,l}$ is strictly lower than twice the minimum area for a valid lens ($A_{max,l} < 2 \cdot A_{min,l}$). Moreover, the minimum area $A_{min,l}$ criterion is not enforced for blobs touching the image edge. Since test images I_t are taken with a sufficient amount of overlap, this escape clause does not allow invalid lenses to avoid detection. All blobs b_l from the lens image, respectively b_m for the metal image are processed according to algorithm 1. Finally, for each reported defect, more features can be

Algorithm 1 Defect reporting according to blob area

for all blobs $b_{l,n}$ in the binary lens image B_l ($n = 1$ to N_l) **do**
  if blob area $A_l(n) >$ maximum lens area $A_{max,l}$ **then**
    reportDefect($b_{l,n}$, maxLensArea)
  else if blob area $A_l(n) <$ minimum lens area $A_{min,l}$ **and** $b_{l,n}$ does *not* intersect an image edge **then**
    reportDefect($b_{l,n}$, minLensArea)
  end if
end for
for all blobs $b_{m,n}$ in the binary metal image B_m ($n = 1$ to N_m) **do**
  if blob area $A_m(n) >$ maximum center area $A_{max,m}$ **then**
    reportDefect($b_m(n)$, maxMetalArea)
  end if
end for

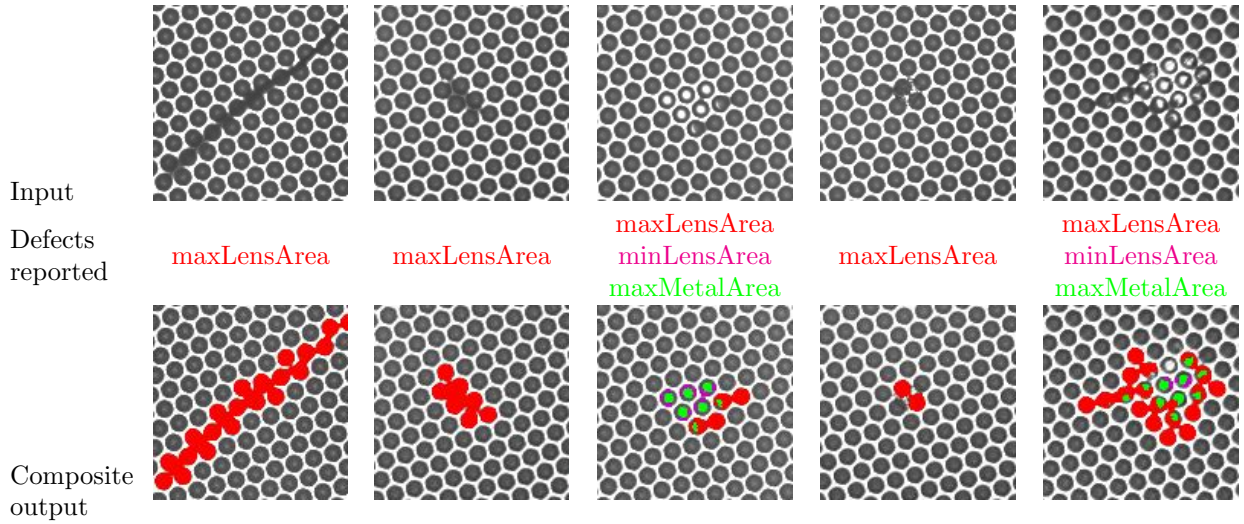


Figure 6. Defects detection results

computed. In the current implementation, the blob center position is computed and written to a log file, along with the blob area and the failure condition triggered. The log file is used to produce statistics on the number and type of defects found in a series of test images. Moreover, a composite image, with defects highlighted by distinctive colors is produced, to facilitate the review by a human operator (fig 6).

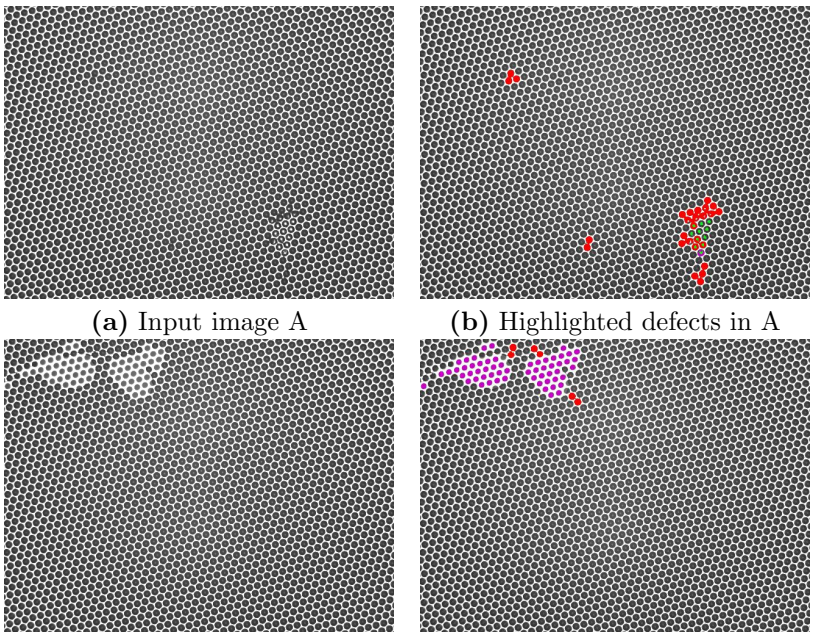
3.2 Possible extensions

In order to improve the inspection performance, more features could be computed for each blob, and used in the blob classification procedure. This could prove useful in situations where the lens *shape* is critical (e.g. for square or rectangular lenses). Lens with a wrong elongation or aspect ratio could efficiently be classified as defective.

4. SEMI-AUTOMATED INSPECTION SYSTEM

4.1 Highlighting of defective samples

The developed software module aims to *assist* a human operator inspecting a microlens array device, mainly by performing a first detection on the input images. Only images containing defects are presented to the human operator, along with a synthetic image where the defects found are automatically highlighted in color. The human operator then estimates the severity of the defect, and its effect on the overall device performance. Figure 7 shows some examples of original input images and defect detection output images.



(a) Input image A **(b)** Highlighted defects in A

(c) Input image B **(d)** Highlighted defects in B

Figure 7. Examples of detected and highlighted defects

Table 3. First test results on manually annotated database

Device	Images	Defects (human)	Defects (autom.)	**False pos. rate** [%]	**False neg. rate** [%]
A	1804	133	446	**17.4**	**0**
B	1804	58	242	**10.2**	**0**

4.2 Automated defect detection performance

In order to evaluate the performance of the developed defect detection process, a test on a manually annotated database was carried out. The test database contains images from two sample microlens devices with a high number of defects. For each device, the number of images was 1804. The database was first inspected by a human operator and then automated defect detection was applied to the same database (tab. 3). The performance of automated defect detection can be qualified as good, since all defects found by human inspection were also found by the automated system (absence of false negatives).

Nevertheless, the automated defect detection system reported many defects not observed by human inspection, resulting in relatively large false positives rate. This high discrepancy between human and automated results motivated a second human inspection. In the spirit of semi-automated inspection, this second human examination was performed only on images marked as defective by the automatic classifier, and was helped by color annotated defects map from automated defect detection. Results of this experiment are reported in table 4. In almost all cases, the result of the automatic classifier was confirmed by the second inspection. It was observed that most of the new defects found by the automated system consist mostly of small defects, such as tiny amounts of metal missing between two lenses. Those defects may have been deliberately left out by the human expert in the first

Table 4. Test results, revised manual annotation

Device	Images	Defects (rev.)	Defects (autom.)	**False pos. rate** [%]	**False neg. rate** [%]
A	1804	433	446	**0.72**	**0**
B	1804	242	242	**0**	**0**

inspection of the entire database, since they are not expected to significantly affect microlens device performance. The last two columns in tab. 4 show that the performance of the developed classifier is good :

- on the test database, no **false negative** was found, which indicates a high security of the inspection system.
- the amount of **false positive** stays low. Note that this may be subject to interpretation, as different human observers may have different interpretations concerning the missing metal defects.

4.3 System extension

The prototype software presented in this work allows automation of a tedious and time-consuming task. Note that the current system was designed and implemented for lens arrays with metal coating and that the defect detection procedure relies on good contrast for a simplified binarization procedure. In devices without metal coating, segmentation methods more advanced than the global thresholding used here (3.1.1) should be employed.

5. CONCLUSION

In this paper, we studied the possibility to automate a time-consuming task in an inspection pipeline for microlens array devices. Two image processing methods were considered. In the context of inspection of metal coated microlens devices, it was possible to develop an automated vision process which greatly reduces the strain on the human operator during the inspection task :

- the number of images to inspect is reduced ,
- in the images shown, the defect locations are clearly highlighted.

A test on real-world images allowed to ensure the software performance : no false negatives were found, meaning that all defects identified by independent human inspection were also detected by the automated procedure. The estimation of the false alarm rate for automated detection proves more difficult, mainly since the criteria considered for an alarm may be pondered differently by different human operators. In the worst case scenario, the false alarm rate was lower than 20%. Therefore, the number of images presented to the operator is always greatly reduced (by a factor 4 or more: tab. 3). Additionally, the composite image, with defects highlighted, further reduces the time required for the human review. Finally, the implemented software shows close to real-time capability ($\approx 1\,\text{image}/s$), even in an interpreted language (Matlab®[4]).

ACKNOWLEDGMENTS

The authors would like to thank B. Putz and K. Weible at SUSS MicroOptics, for providing the annotated test image databases.

REFERENCES

1. M. Kujawinska, C. Gorecki, H. Ottevaere, P. Szczepanski, and H. Thienpont, "Micro-optic measurement techniques and instrumentation in the european network of excellence for micro-optics (nemo)," in *Proc. SPIE Nano- and Micro-Metrology*, **5858**, p. 585801, aug 2005.
2. R. Voelkel, M. Eisner, and K. J. Weible, "Micro-optics: manufacturing and characterization," in *Proc. SPIE Optical Fabrication, Testing, and Metrology II*, **5965**, p. 596501, oct 2005.
3. P. Nussbaumy, R. Voelkel, H.-P. Herzig, M. Eisner, and S. Haselbeck, "Design, fabrication and testing of microlens arrays for sensors and microsystems," *Pure Appl. Opt.* **6**, pp. 617–636, 1997.
4. The MathWorks, "MATLAB v7.0.1," 2006. http://www.mathworks.com.
5. R. Fisher, S. Perkins, A. Walker and E. Wolfart., "Adaptive Tresholding," 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm (accessed 15.08.2007).