A template matching approach based on the discrepancy norm for defect detection on regularly textured surfaces

Jean-Luc Bouchot[a], Gernot Stübl[b] and Bernhard Moser[b]

[a]Department of Knowledge-based Mathematical Systems, Altenbergerstr. 69, A-4040 Linz, Austria;

[b]Software Competence Center Hagenberg, Softwarepark 21, A-4232 Hagenberg, Austria

ABSTRACT

In this paper we introduce a novel algorithm for automatic fault detection in textures. We study the problem of finding a defect in regularly textured images with an approach based on a template matching principle.

We aim at registering patches of an input image in a defect-free reference sample according to some admissible transformations. This approach becomes feasible by introducing the so-called discrepancy norm as fitness function which shows particular behavior like a monotonicity and a Lipschitz property. The proposed approach relies only on few parameters which makes it an easily adaptable algorithm for industrial applications and, above all, it avoids complex tuning of configuration parameters.

Experiments demonstrate the feasibility and the reliability of the proposed algorithms with textures from real-world applications in the context of quality inspection of woven textiles.

Keywords: Texture Analysis, Automatic Defect Detection, Discrepancy Norm, Automatic Visual Inspection, Template Matching, Registration, Optimization

1. INTRODUCTION

Texture Analysis is a broad area in image vision and understanding. Many applications have been recently developed which are used in different domains. Some authors are interested in finding errors in seismic data[1] in order to help geologist finding breaks in geological structures. It is for them of prime importance to detect any discontinuities in the patterns they get in order to avoid expensive further analysis. Texture analysis is also needed in mechanical engineering where one wants to find out how a material reacts to a given force. This is known as strain analysis[2].

In this paper we are more interested in automatic quality control. These applications have been an intense research area in the past decades. Indeed one can find as many good algorithms as applications. We refer the reader to[3–6] for some examples. However, it seems that all of the proposed approaches suffer of painful drawbacks such as learning a particular texture or type of defect, decreasing the adaptability of the methods. Therefore we introduce a novel method for automatic inspection of the quality of texture which is based on template-matching and optimization. We claim our method to be at least as reliable as other ones for regularly textured surfaces, with the advantage that it does not need any particular optimized features to be calculated, allowing it to easily adapt itself to new unknown textures of a certain class. Constraining our method to work only on regularly textured surfaces does not strongly affect its relevance as many industrial applications (industrial textile, airbag hose, paper, ...) create such regular textures.

However all template matching methods need a similarity measure to be minimized. We propose to use the discrepancy norm as it has been proven to be monotonic and Lipschitz-continuous with respect to shifting[7], at least for positive signals. Finding translational parameters is done using standard optimization methods. Finally we added a coarse-to-fine approach to accelerate the efficiency of the algorithm.

Further author information: (Send correspondence to J.-L.B.)
J.-L.B.: E-mail: jean-luc.bouchot@jku.at, Telephone: +43 7236 3343 432
G.S., B.M.: E-mail: {gernot.stuebl, bernhard.moser}@scch.at, Telephone: +43 7236 3343 833

This paper is articulated as follows. In Section 2 we first recall the problem we are aiming to solve and describe some state-of-the-art approaches with their advantages and drawbacks in a systematic way. Then we motivate the main contribution of this paper in Section 4 by introducing a naive approach which calculates local dissimilarity in order to assess the quality of a textile. This first guess gave us the idea to locally optimize the registration to avoid artifacts. Finally after giving some experimental results, conclusion and future works are given in Section 5.

2. PROBLEM STATEMENT AND RELATED WORK

Before we introduce our methods we want to first introduce the problem we are aiming to solve and then explain how the known methods will fail at this task.

2.1 Problem statement

In this paper we are aiming to solve the following (general) problem:

Given an image flow of a fabric, can we tell if there are any defects in the finished product? If yes the location of the defect should be given.

This statement describes two problems. The first one is to find whether an image contains a defect or not. The second one is to tell where this defect is actually located in the input image. The problem stated is independent of the types of defect that can occur. Indeed some researchers try to find a defect by looking for a particular pattern. Song et al.[5] for instance, try to detect cracks, and only cracks, in textures. Even if these systems are reliable for a given kind of defect, they need quite a lot of effort to adapt themselves to new or unexpected defects.

In this paper we concentrate our effort on regularly textured surfaces as they are the one we would encounter on many industrial applications such as airbag hose fabrics or paper mills for instance. Moreover, texture inspection should be done differently whether we analyze regular or stochastic textures.

2.2 Related work

According to different recent surveys[8,9] the different methods for automatic inspection of textured surfaces can be divided into several categories. The most common approaches are based on statistics. For instance some authors[10] try to calculate features such as range, means, deviation or median value out of histograms. Then these features are compared to a database to tell whether we have already encountered the input we are currently analyzing.

While histograms are a mean to analyze first order statistics, one might prefer to study second order statistics. The Gray Level Co-occurrence Matrix (GLCM) helps us for this. It was first introduced by Haralick[11] and represents the dependency between two pixel intensities over the whole image. The GLCM are then compared using a set a computed features such as its inertia, mean value or entropy, for instance.

However even if first and second order statistics are easy to understand, they are weak against small transformations. Added to this, one should make a smart guess of which features should be computed and which one should not, eventually putting away some information.

Another idea is to filter an input image with some well chosen filters and to get useful information out of it. Spatial domain[12] , frequency domain[4] or mixed spatial and frequency domain[13] have been studied. Unfortunately, it seems hard to find the perfect filters one should use. Indeed, if one wishes to define a framework as general as possible, many different filters should be used, making this approach unsuitable for general cases.

Other methods have been developed. Structural approaches[14] for instance, seem to have good results for a chosen type of defect. However, as the structural element should be adapted for this given type, it is hard without post processing to get a general framework out of it. Moreover, as the defects are found using morphological operations, the whole process can be time consuming.

The last type of approaches is called model-based and include fractal methods[15] as well as Markov Random Fields[16] or Texem[17]. They basically work as a novelty detection and need first to be taught what they should repeatedly see.

3. TEMPLATE-BASED APPROACH

While the above described methods suffer from different problems, we introduce a novel method for fault detection based on template-matching which turns out to rely only on a few parameters making it simple to apply. After reviewing shortly in Section 3.1 what are the main template-based methods in pattern recognition and their drawbacks for their use in texture analysis, we describe our new approach and give some results of experiments.

3.1 Template Matching Methods

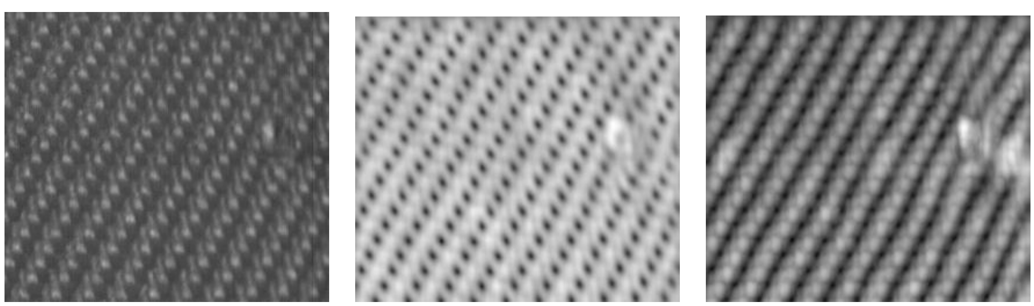
Template matching have been in the last decades used in the context of pattern recognition. They all work in a similar way. Some features are detected (for instance Harris keypoints[18]) and described with a feature vector[19–21]. These descriptors are then matched to a database or to a reference pattern. However in order to be reliable, such methods have some constraints. There should be enough features to describe a given object, and those key points should be distinctive enough which is not guaranteed for the textures we wish to analyze. Here we want to detect different kinds of defects which will have different characteristics, making the matching process harder. Finally we would like to add that some defects will not generate any features which makes them invisible to any feature-based methods.

3.2 Our approach

While feature-based template-matching methods have been widely used in computer vision[22, 23], textured surface analysis is still left aside from the community. Here we investigate the use of a naive sliding window approach to assess the defectiveness of a given pixel. We propose an algorithm which is somehow similar to Baudrier et al.[24] . While the authors calculate a so-called *Local Dissimilarity Map (LDMap)* for the purpose of binary image classification, we propose an approach based on an LDMap to compute a local *defectiveness* map for defect analysis. That means that for each pixel, we consider a neighboring patch of the size of a given reference pattern, and calculate the (dis-)similarity (i.e. norm, distance, any metric or (dis-)similarity measure) between the reference and the current sliding window patch. Formally speaking, if I denotes an input frame, represented as an $M \times N$ matrix, and g is a single reference pattern of size $m \times n$, then we have

$$\forall 1 \leq i \leq M,\ \forall 1 \leq j \leq N, \mathrm{LDMap}(i,j) = d(A_{ij}, g) \qquad (1)$$

where we have d representing any (dis-)similarity metric (e.g. L_2 norm, discrepancy norm, Bhattacharyya measure[25] or Kullback-Leibler divergence[26]) and A_{ij} is the local patch of size $m \times n$ centered at (i, j).

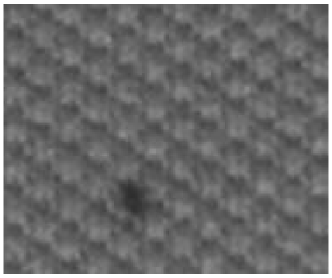


(a) Surface of an airbag hose (b) LDMap of 1(a) with L2 norm (c) LDMap of 1(a) with discrepancy norm

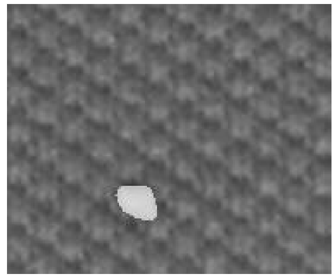
Figure 1. Local Defectiveness Map (LDMap) computed on an image of an airbag hose using either the Euclidean norm 1(b) or the discrepancy norm 1(c) which improves the contrast around the defect.

Figure 1 shows the result of the computation of the LDMap on an image of an airbag hose containing a defect on the right side of it. The lighter pixels denote the most dissimilarity between the reference pattern[*] and the

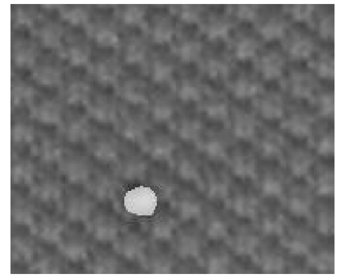
[*]The reference pattern was chosen manually to cover between one and two defect-free pattern of the image. This could be done automatically for instance by studying the Fourier domain of the repetitive pattern like in Xie and Guan[27] .



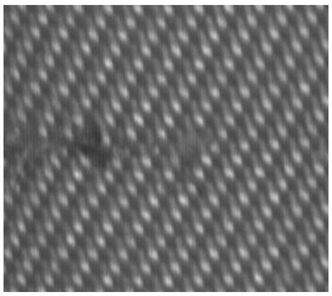
(a) Surface of a woven textile



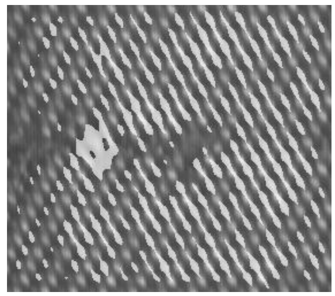
(b) Detected failures of 2(a) based on the L2 norm



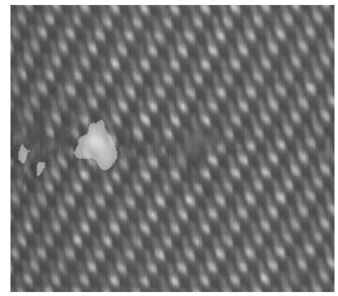
(c) Detected failures of 2(a) based on the discrepancy norm



(d) Surface of an airbag hose



(e) Detected failures of 2(d) based on the L2 norm



(f) Detected failures of 2(d) based on the discrepancy norm

Figure 2. Defect detection in textile and airbag hose production after a threshold on the computed LDMap with either the L_2 or the discrepancy norm. Threshold of DN images is 0.8 and of L_2 0.7, with normalized values between 0 and 1.

current local window. On the L_2 based LDMap 1(b) one sees that the brighter parts cover almost the whole resulting image whereas the discrepancy norm computation on Figure 1(c) yields results with more contrast. On both images the defect appears much brighter, and therefore more distinctive than the rest.

Figure 2 goes one step further and shows the results of overlapping both input images with (global) thresholds of the LDMap. The discrepancy norm based defect detection (Figures 2(c) and 2(f)) yields better results than the ones obtained with a classic L_2 norm (Figures 2(b) and 2(e)). Indeed, the defects are better localized and one sees less artifacts in the defect free regions. However with this approach, some defects are discarded (particularly in Figure 2(f)) and we do not make use of interesting properties of the discrepancy norm like monotonicity and Lipschitz behavior for autocorrelation[7]. We will give in the next section an optimized variant making full use of these properties.

4. TEMPLATE MATCHING BASED DEFECT DETECTION

This section holds the main contribution of our paper, which refines the previous naive approach by using standard optimization techniques.

4.1 General Description of our Approach

In the last section we described a way of calculating an LDMap. Its pseudo code is shown in Algorithm 1. The additional Statement 13 improves the contrast in the LDMap by taking the minimum of defectiveness values on a neighborhood of δ pixels. In this case Statements 3 to 13 are equivalent to the following optimization problem:

$$\alpha(i,j) \quad = \quad \min_{f \in V(i,j,\delta)} \|f - g\| \tag{2}$$

where g represents the chosen defect-free reference and α designates the defectiveness of a pixel at location (i,j). $V(i,j,\delta)$ represents a set of patches of the size of g, whose centers are located within δ pixels from (i,j). Then,

Algorithm 1 Local Defectiveness Map

1: **function** LDMap(referenceImage, defectImage(size $M \times N$))
2: extract $m \times n$ patch from referenceImage, where $m \times n$ covers 1-2 periods of pattern
3: **for** $x = 0, n, 2*n, \cdots \div (N-n)$ **do**
4: **for** $y = 0, m, 2*m, \cdots \div (M-m)$ **do**
5: set position of $m \times n$ sliding window to (x, y) of defectImage
6: **for** all pixels $p = (i, j)$ of sliding window **do**
7: center reference patch on pixel p
8: calculate discrepancy between sliding window and reference patch
9: enter dissimilarity value in (x, y) of resultImage
10: **end for**
11: **end for**
12: **end for**
13: *aggregate temporary images with min-function in an area of δ pixels*
14: binarize resultImage
15: **return** resultImage
16: **end function**.

in Statement 14 if α is higher then a threshold, the pixel at position (i, j) is labeled as defect, and defect-free otherwise. Literally speaking, Equation (2) reads

If I can register the reference pattern in a small neighborhood of a given point, then there should not be any defect at this considered point

The idea we introduce next tackles the converse problem. Instead of finding a perfect match within the input image for the reference pattern, we try to register a small input sample into a larger reference defect-free pattern by some transformation. Such idea is not completely new as it has already been used for surfaces of wafers[27] or ceramic tiles[28, 29] which show special structures of orthogonal line segments. However, these algorithms can not be adapted to cases where the textures are more complex. Because of its monotonicity property the discrepancy norm (thereafter denoted as $\|\cdot\|_D$) is an appropriate candidate for modeling the fitness function for template matching. This allows to reduce a full exhaustive search of quadratic complexity with respect to the diameter of the sliding window by applying optimization techniques.

Now if we denote by f an input image, by $A_{i,j}$ the patch of size $m \times n$ centered at pixel (i, j) and by g a larger defect-free reference pattern then we determine the defectiveness at a pixel (i, j) by solving the following optimization problem

$$\alpha(i, j) \quad = \quad \min_{\tilde{g} \in \mathcal{G}} \|A_{i,j} - \tilde{g}\|_D, \tag{3}$$

where $\mathcal{G}$ represents all possible $m \times n$ sub-windows of the reference pattern g.

Algorithm 2 is the pseudo code of Equation (3), which we will from now on refer to as OptLDMap algorithm. By applying a gradient descent like algorithm we avoid exhaustive search for solving Equation (3), by which super linear convergence can be achieved. In our experiments, see Figures 3, we apply a method outlined in Subsection 4.3 and 4.4 for which the number of image evaluations remains always below them of Algorithm 1. Note that gradient descent optimization techniques require subpixel evaluation which is done by linear interpolation.

Experiments show further that the number of image evaluations are independent of the window size. This seams reasonable since the test pattern is always aligned within one period in the reference image. The window size is chosen to contain two periods of the pattern which can be automatically determined by methods such as proposed by Xie and Guan[27].

Algorithm 2 Basic OptLDMap

```
function OptLDMap(referenceImage, defectImage(size M × N))
    input reference image
    for x = 0, n, 2 ∗ n, · · · ÷ (N − n) do
        for y = 0, m, 2 ∗ m, · · · ÷ (M − m) do
            set position of m × n sliding window to (x, y) of defectImage
            repeat
                transform sliding window content with H to referenceImage
                calculate discrepancy between sliding window and area in referenceImage
                recalculate transformation H to minimize discrepancy
            until optimum with some ϵ is reached
            enter minimum distance in resultImage for the area of the sliding window
        end for
    end for
    binarize resultImage
    return resultImage
end function.
```

4.2 Transformation

The optimization problem described in Equation (3) can be extended to a more general registration by introducing a set H of admissible transformations, leading to

$$\alpha(i,j) \quad = \quad \min_{\tilde{g}\in\mathcal{G},\tau\in H} \|A_{i,j} - \tau(\tilde{g})\|_D. \tag{4}$$

More general transformations allow to model different variations in the appearance of the analyzed textures. For instance blurring is caused by vibration of the conveyor, scale changes appear with variations in the working distance. Even projective transformations can be used if the products surface is not necessarily perpendicular to the camera. For simplicity we restrict the transformation to translation only which is also the most common case on production lines.

4.3 Optimization

We describe here the heart of the the OptLDMap algorithm, which is the optimization part. Motivated by its mathematical properties and by the experiments from Section 3 we suggest to use the discrepancy norm as fitness function for template-matching. Be aware that the discrepancy norm is a discrete mathematical concept which requires the use of numerical differentiation methods and thereby appropriate optimization techniques which are a future research topic. For this paper, as a starting point we apply a quasi Newton algorithm with a so-called BFGS update proposed by Broyden[30,31] , Fletcher[32] , Goldfarb[33] and Shanno[34] , which shows reasonable convergence.

Motivated by the Lipschitz property of the discrepancy norm we apply a pre-optimization step based on the DIRECT[35,36] algorithm which is especially designed for Lipschitz or near-Lipschitz continuous fitness-functions in order to get a robust guess for the initialization of the gradient descent method.

It is worth mentioning that experiments on registering random patches of images show that after a given number of DIRECT iterations the accuracy of the optimization does not change significantly. Therefore the optimization procedure is build up of a constant number of DIRECT iterations followed by a gradient descent search until the change of the step size falls below a certain threshold.

In the experiments on the TILDA database[†] we have found that six DIRECT iterations and a threshold of 0.2 yield reasonable results.

[†]Available from Universität Freiburg, Institut für Informatik, Lehrstuhl für Mustererkennung und Bildverarbeitung (LMB).

4.4 Application

For defect detection applications an overlap of 10% is added to the OptLDMap to detect failures which could appear on the border of the sliding window.

We further introduce a coarse-to-fine approach that adaptively refines the sliding window size as a function of the defectiveness. While this improves the speed of the algorithm, it creates some overhead too. One of them is that discrepancy values between different levels of recursion cannot be compared, since the window size is different. Therefore the maximum discrepancy of each level is estimated on the reference image in advance and all values of a level are normalized to these maxima.

In our experiments a recursion is started if the discrepancy value of a window exceeds the maximum discrepancy of the level. The window size of each level decreases by a factor of two until the original window size $m \times n$ is reached, since optimization below one period does not make sense. It is clear that the starting window size and therefore the recursion depth is fixed by the size of the reference image. With this coarse-to-fine approach we exploit the fact that defect-free patches are cheaper to process in terms of computation speed.

Figure 3 demonstrates the applicability of our approach on regularly textured surfaces taken from the TILDA database and real world industrial examples. The images from the TILDA database cover different types of failures on regularly textured images. The first column shows the input image, the second one illustrates the result of OptLDMap as a defectiveness image. Brighter regions are more likely to be defect. The last column depicts an overlay between the input and the thresholded defectiveness image. Figures 3(a) to 3(f) show the result of a real world industrial texture inspection‡. Images from type of Figure 3(d) turn out to be challenging since the defect is visually quite similar to the regular pattern.

Throughout the whole experiment the threshold is kept constant, only the window size has to be adapted to the analysed texture.

5. CONCLUSION AND FUTURE WORK

We have introduced a novel algorithm used for defect detection in regularly textured surfaces which benefits from the mathematical properties of Hermann Weyl's discrepancy norm. The algorithm works with a reference image and avoids complex tuning of configuration parameters. Indeed, only a window size and a binarization threshold have to be adapted. In the future we aim at developing a completely parameter free approach by automatically determining the optimal window size, for instance using spectral analysis, and the binary threshold value, by statistical analysis.

Although the optimization of the discrepancy norm as fitness function has not yet been studied in depth the results on a standardized texture database are very promising. We expect to improve the results in terms of performance, reliability, distinctiveness and sensitivity by refining the optimization. As soon as this is done a systematic comparison with state-of-the-art algorithms will be performed.

ACKNOWLEDGMENTS

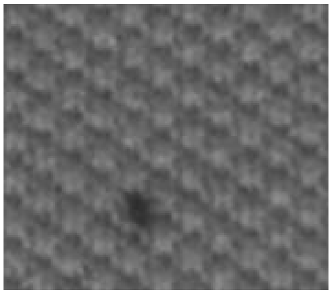
This work was supported in part by the Austrian Science Fund (FWF) under grant no. P21496 N23 and the Austrian COMET program.

REFERENCES

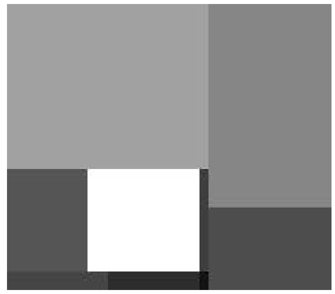
[1] Timm, F., Klement, S., Martinetz, T., and Barth, E., "Welding inspection using novel specularity features and a one-class SVM," in [*Proc. of the Int. Conf. on Imaging Theory and Applications*], **1**, 146–153, INSTICC, Lisboa, Portugal (2009).

[2] Hussain, F., Kehtarnavaz, N., Kallel, F., and Ophir, J., "Texture analysis to characterize strain distribution in elastograms," in [*Proc. of 12th IEEE Symp. on Computer-Based Med. Syst.*], 124 –129 (1999).

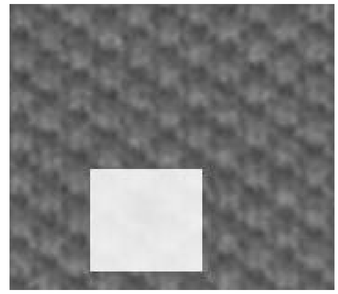
‡The images can be requested by email from the authors.



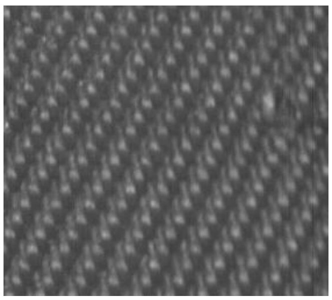
(a) Woven fabrics



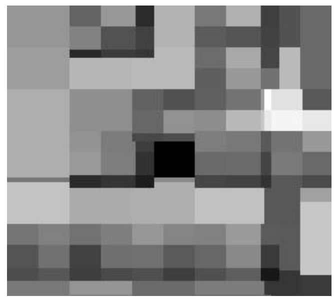
(b) Results of OptLDMap on 3(a)



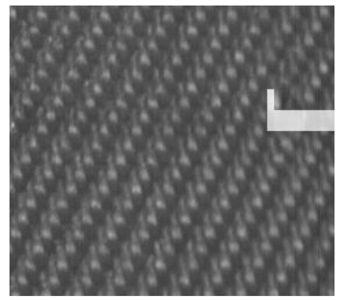
(c) Detected failures of 3(a)



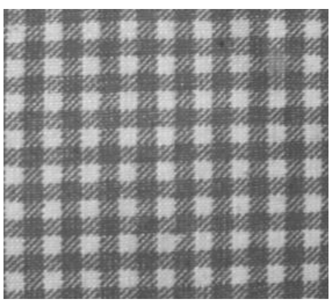
(d) Airbag hose



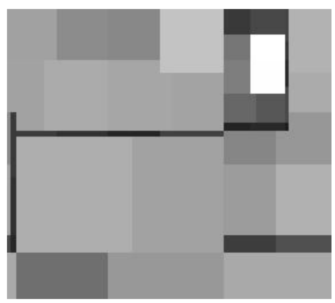
(e) Results of OptLDMap on 3(d)



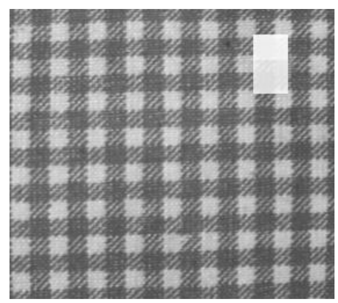
(f) Detected failures of 3(d)



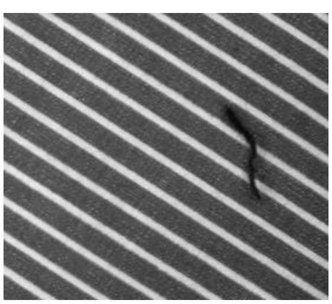
(g) C3R1E2N3 of TILDA



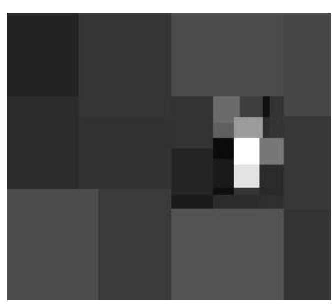
(h) Results of OptLDMap on 3(g)



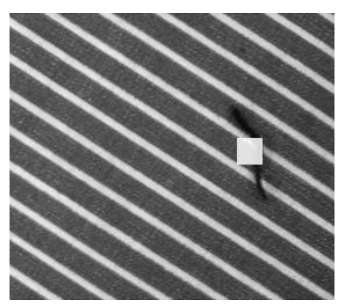
(i) Detected failures of 3(g)



(j) C3R3E4N23 of TILDA



(k) Results of OptLDMap on 3(j)



(l) Detected failures of 3(j)

Figure 3. Examples of OptiMap runs on images of our customers a to f and test images of the TILDA database. The first column is always the input image, the second the defectiveness result of OptLDMap before thresholding and the last column an overlay between input image and the thresholded result. The window sizes are 30×20 for the airbag hose, 25×25 for the woven fabrics and 40×40 for the others. Threshold is fixed to 0.9, but could be adopted.

[3] Gibson, D., Spann, M., and Turner, J., "Automatic fault detection for 3d seismic data," in [*Proc. VIIth Digital Image Computing: Techniques and Applications*], Sun C., Talbot H., O. S. and T., A., eds. (December 2003).

[4] Sari-Sarraf, H. and Goddard, J. S., "On-line optical measurement and monitoring of yarn density in woven fabrics," in [*Autom. Opt. Inspection for Ind.: Theory, Technology, and Application*], Wu, F. Y. and Ye, S., eds., 444–452, SPIE (November 1996).

[5] Song, K. Y., Petrou, M., and Kittler, J., "Texture crack detection," *Machine Vision and Applications* **8**, 63–76 (1995).

[6] Haindl, M., Grim, J., and Mikes, S., "Texture defect detection," in [*Computer Analysis of Images and Patterns*], 987–994 (2007).

[7] Moser, B., "Similarity measure for image and volumetric data based on Hermann Weyl's discrepancy measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **99**, DOI 10.1109/TPAMI.2009.50 (2009).

[8] Xie, X., "A review of recent advances in surface defect detection using texture analysis techniques," *Electr. Letters on Computer Vision and Image Analysis* **3**, 1–22 (2008).

[9] Kumar, A., "Computer-vision-based fabric defect detection: A survey," *IEEE Trans. Indus. Electr.* **55**, 348–363 (2008).

[10] Ng, H.-F., "Automatic thresholding for defect detection," *Pattern Recogn. Lett.* **27**, 1644–1649 (2007).

[11] Haralick, R. M., Shanmugam, K., and Dinstein, I., "Textural features for image classification," *IEEE Trans. Syst., Man, Cybera* **3**, 610–621 (1973).

[12] Baeg, S., Popov, A. T., Kamat, V. G., Batman, S., Sivakumar, K., Kehtarnavaz, N. D., Dougherty, E. R., and Shah, R. B., "Segmentation of mammograms into distinct morphological texture regions," in [*Proc. of 11th IEEE Symp. on Computer-Based Med. Syst.*], 20–25 (June 1998).

[13] Bodnarova, A., Bennamoun, M., and Latham, S., "Optimal gabor filters for textile flaw detection," *Pattern Recognition* **35**, 2973–2991 (2002).

[14] Mirmehdi, M., Marik, R., Petrou, M., and Kittler, J., "Iterative morphology for fault detection in stochastic textures," *Electronic Letters* **32**, 443–444 (1996).

[15] Conci, A. and Proença, C. B., "A fractal image analysis system for fabric inspection based on a box-counting method," *Computer Networks and ISDN Systems* **30**, 1887–1895 (1998).

[16] Cohen, F., Fan, Z., and Attali, S., "Automated inspection of textile fabrics using textural models," *IEEE Trans. on Patt. Anal. Mach. Intell.* **13**, 803–808 (1991).

[17] Xie, X. and Mirmehdi, M., "Texems: Texture exemplars for defect detection on random textured surfaces," *IEEE Trans. on Patt. Anal. Mach. Intell.* **29**, 1454–1464 (2007).

[18] Harris, C. and Stephens, M., "A combined corner and edge detector," in [*Proc. of the 4th Alvey Vision Conf.*], 147 – 151 (1988).

[19] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**, 91 – 110 (November 2004).

[20] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V., "Surf: Speeded up robust features," *Computer Vision and Image Understanding* **110**, 346 – 359 (2008).

[21] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., "Brief: Binary robust independent elementary features," in [*11th European Conference on Computer Vision (ECCV)*], (2010).

[22] Huang, Y. and Huang, T., "Facial tracking with head pose estimation in stereo vision," in [*Proc. of the 2002 Int. Conf. on Image Processing*], **3**, 833 – 836 (June 2002).

[23] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in [*IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*], Schmid, C., Soatto, S., and Tomasi, C., eds., **2**, 886 – 893 (June 2005).

[24] Baudrier, E., Nicolier, F., Millon, G., and Ruan, S., "The adaptative local hausdorff-distance map as a new dissimilarity measure," *Pattern Recognition* **41**, 1461–1478 (2008).

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

[26] Kullback, S. and Leibler, R. A., "On information and sufficiency," *The Annals of Math. Stat.* **22**, 79–86 (Mar. 1951).

[27] Xie, P. and Guan, S.-U., "A golden template self-generating method for patterned wafer inspection," (2000).

[28] Costa, C. and Petrou, M., "Automatic registration of ceramic tiles for the purpose of fault detection," *Machine Vision and Applications* **11**, 225–230 (February 2000).

[29] López-García, F., Acebrón-Linuesa, F., Valiente-González, J.-M., and Pérez, E., "A study of registration methods for ceramic tiles inspection purposes," in [*Proc. of IX Spanish Symp. on Pattern Recognition and Image Analysis*], 145–150 (2001).

[30] Broyden, C. G., "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations," *IMA J Appl Math* **6**, 76–90 (March 1970).

[31] Broyden, C. G., "The Convergence of a Class of Double-rank Minimization Algorithms: 2. The New Algorithm," *IMA J Appl Math* **6**, 222–231 (September 1970).

[32] Fletcher, R., "A new approach to variable metric algorithms," *The Computer Journal* **13**, 317–322 (March 1970).

[33] Goldfarb, D., "A Family of Variable-Metric Methods Derived by Variational Means," *Mathematics of Computation* **24**, 23–26 (January 1970).

[34] Shanno, D. F., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computation* **24**, 647–656 (July 1970).

[35] Jones, D. R., Perttunen, C. D., and Stuckman, B. E., "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications* **79**, 157–181 (October 1993).

[36] Shubert, B. O., "A sequential method seeking the global maximum of a function," *SIAM Journal on Numerical Analysis* **9**(3), 379–388 (1972).