Machine Vision and Applications (2005) 16(5): 298–305
DOI 10.1007/s00138-005-0001-3

ORIGINAL PAPER

Franz Pernkopf

3D surface analysis using coupled HMMs

Received: 31 December 2004 / Accepted: 11 July 2005 / Published online: 25 November 2005

Abstract We propose a coupled hidden Markov model (CHMM) for analysis of steel surfaces containing three-dimensional flaws. The aim is to model surface errors, which are stretched across one or more surface segments because of their strongly varying size. Due to scale on the surface, the reflection property across the intact surface changes and intensity imaging fails. Light sectioning is used to acquire the surface range data. The steel block is vibrating on the conveyor during data acquisition, which complicates robust feature extraction. After depth map recovery and feature extraction, segments of the surface are classified using CHMMs. The CHMM achieves a recognition rate of 98.57%. We compare the CHMM approach to the naïve Bayes classifier, the Hidden Markov Model, the k-nearest neighbor classifier, and to the Support Vector Machine.

Keywords Surface analysis · Surface inspection · Range data analysis · Coupled hidden Markov models · EM training

1 Introduction

In industry there is an increasing demand for automatic inspection systems to control the quality of the products. The ever more stringent customer demands are well founded on their high costs incurred for poor quality, due to the resulting costs for correction. In a survey of different inspection approaches, Newman and Jain [1] defined the task of inspection as follows: *Inspection is the process of determining if a product deviates from a given set of specifications.*

The literature considers two different approaches for acquiring the surface image [1, 2]: Intensity imaging techniques and range imaging methods. There are many surface inspection approaches for steel products available which are based on intensity imaging. Dupont et al. [3] propose a system for flat steel products, which is optimized using a cost matrix approach. They achieve a recognition rate of 84.5% with a multilayer neural network. In [4] an inspection system for machined metallic high-precision surfaces such as bearing rolls is presented. An optimal illumination setup in terms of maximizing the contrast between surface defects and flawless surface is derived by considering the machined surface as composition of micro-facets using the Torrance-Sparrow reflection model [5, 6]. They achieve a classification rate of 85.4% using a k-NN classifier based on a selected subset of features. All these approaches have in common that the reflection property and accordingly the optical appearance of the flawless surface is homogeneous. Many more systems for inspection of metallic surfaces have been suggested in different *Proceedings of SPIE* [7]. Inspection systems which use range images have mostly been applied for complex objects, solder joints, and printed circuit boards. A comprehensive overview is provided in [1, 7].

In this paper, we deal with the surface inspection of milled steel blocks, which are partially covered with scale. The cross-section of the block is approximately quadratic with a size varying from 130 mm × 130 mm to 160 mm × 160 mm, whereby the edges are round. The length is about 10 m. Defects with a minimal length of 3 mm, a width of 0.5 mm, and a depth of 1 mm have to be identified on moving blocks at a speed of 1.85 m/s.

The reflection property across the flawless surface changes strongly due to scale on the surface. As intensity imaging results in a poor performance, we use range imaging based on light sectioning techniques [2, 8] to acquire the surface data of the steel block with its embedded flaws. Of course, the variations of the surface reflectance affects the range accuracy [9]. However, we are interested in a qualitative measure of the surface and the exact dimensions of the product are irrelevant. The range image is less sensitive to the inhomogeneous reflectance of the flawless surface. However, when we use light sectioning, the data are affected by vibrations caused by the movement of the steel block on the conveyor. These vibrations result in a varying position of the

F. Pernkopf (✉)
Laboratory of Signal Processing and Speech Communication,
Graz University of Technology,
Inffeldgasse 12, 8010 Graz, Austria
E-mail: pernkopf@tugraz.at

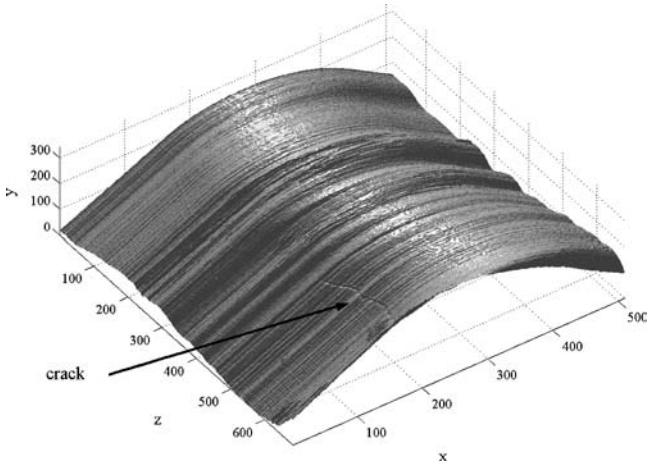

Fig. 1 Acquired surface data with an embedded crack

acquired sections. Figure 1 shows a surface segment with an embedded crack and the vibration artifacts caused by the movement.

The segments of the surface should be classified into two classes:

1. *Surface flaws:* Basically, there are different types of flaws, which might be classified according to their genesis during production. We treat them as non-specific surface error, as they have the degradation of the surface quality in common. Defects occur always at the round edge of the steel block. This means that the inspection is focused on the edge of the steel block. The lateral resolution is ~ 0.1 mm/pixel. The employed camera [10] delivers 2,600 sections/s, which gives a resolution of $\sim$0.7 mm/section in the moving direction of the steel block. Figure 2 shows different flaws and their dimensions. Additionally, the inhomogeneity of the surface caused by scale is observable.
2. *Pseudo errors:* Pseudo errors and intact surface segments do not degrade the quality of the steel blocks. Pseudo errors are caused by an extremely inhomogeneous surface reflection property, which results in a spurious specular reflection [5] of the light in the direction of the imaging sensor. This entails an overmodulation of the sensor and a distorted range data. In particular, we are confronted with these errors when the scale of the surface is removed mechanically through the carrier apparatus during the production. Thus, small surface areas appear shiny. These pseudo errors cannot be avoided by changing the setup of the light sectioning technique.

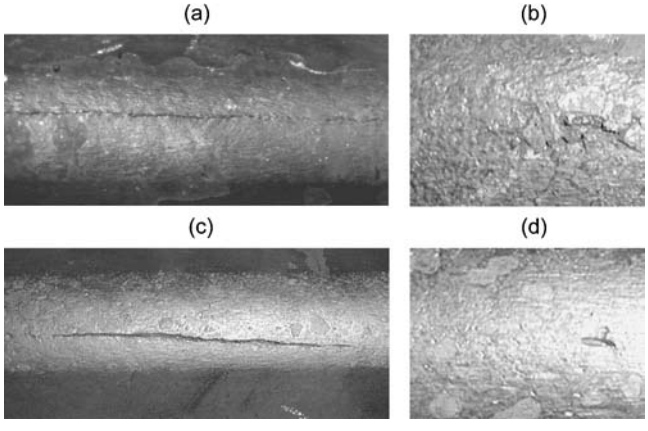

Fig. 2 Different surface flaws: (**a**) 30 mm $\times$ 0.5 mm, (**b**) 10 mm $\times$ 7 mm, (**c**) 60 mm $\times$ 1 mm, (**d**) 3 mm $\times$ 1 mm

For classification of the surface qualities, we use the coupled Hidden Markov Model (CHMM) [11], which is successfully acknowledged for modeling multiple interacting processes in multimedia applications [11], for face recognition [12], and in audio-visual speech recognition [13]. As shown in Fig. 2 we have to deal with flaws of strongly varying sizes. This means that the flaws might be stretched over one or more surface segments. The idea is to improve the classification performance by modeling the interaction between neighboring surface segments. Additionally, in our application, the nature of the production (milling) is inherently sequential over the length of the steel block (time) as well as the surface data acquisition. The data of each time slot can be divided into segments. Each sequence of segments over time can be viewed as an individual process of the CHMM, which is interacting with its neighboring processes. We assume that the dependency of the segment at time t is restricted to its immediate preceding surface segments at time $t-1$. Moreover, the interaction of the coupled processes is limited to the immediate neighbor processes. The advantage of using CHMM for surface analysis is twofold: (i) Surface data of varying size can be classified with one model. (ii) The interactions between the surface segments are modeled. This is useful, as an error might be spread over more adjacent surface segments.

The remainder of the paper is organized as follows: Sect. 2 treats an approach for depth map extraction, whereby the vibration artifacts that exist due to the movement of the steel blocks on the conveyor have to be removed. Additionally, the extracted features used for classification are derived. The coupled HMM and an approach for training and recognition is presented in Sect. 3. We present classification results of the CHMM and compare them to the naïve Bayes classifier, the Hidden Markov Model, the k-nearest neighbor classifier, and to the Support Vector Machine in Sect. 4. Conclusions and future work are presented in Sect. 5.

2 Depth map recovery and feature extraction

Before the extraction of the features, we have to recover the depth map of the surface. Thereby, the vibration artifacts that exist due to the movement of the steel blocks on the conveyor have to be removed. The measuring head for light sectioning is focused on the round edge of the steel block. This means that the edge is in the center of the acquired sections. A surface segment of the range data is shown in Fig. 3a. In

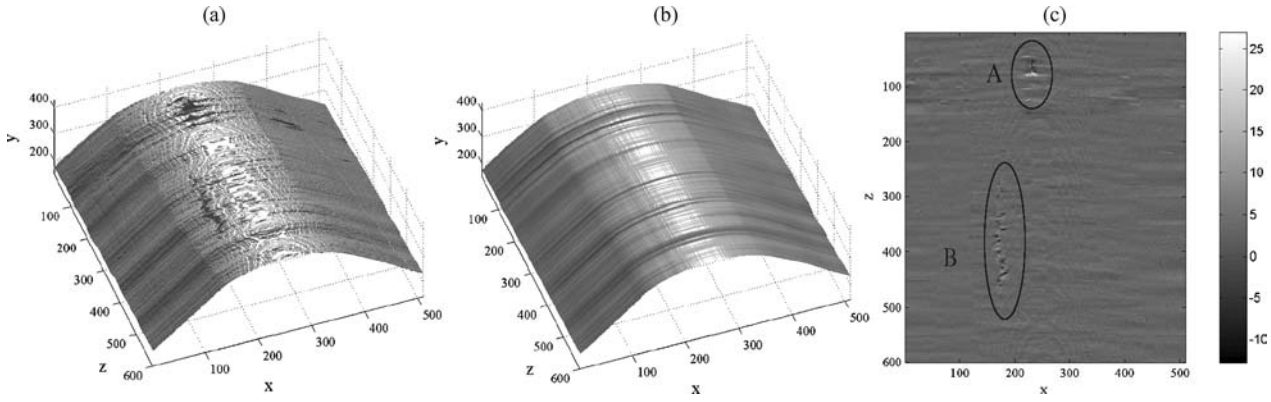

Fig. 3 Surface approximation: (**a**) Acquired surface segment (with vibrations), (**b**) Smoothed surface approximant, (**c**) Depth map (A, pseudo error, B, surface error)

order to classify the data, the depth of the flaws has to be recovered. Therefore, a model of the three-dimensional shape of the surface is approximated. This approximation of the surface segment [14] is determined using singular value decomposition (SVD) [15]. The surface segment of range data $\mathbf{A}$ with the size of $m \times n$ is decomposed by means of SVD as $\mathbf{A} = \mathbf{USV}^{\mathrm{T}}$, where $\mathbf{U}$ is an $m \times m$ orthogonal matrix, $\mathbf{V}$ is an $n \times n$ orthogonal matrix, and $\mathbf{S}$ is an $m \times n$ diagonal matrix composed of the singular values S_{ii} ordered from the largest to the smallest. It has been observed that the shapes of the surface segments are either planar or curved in one direction. Therefore, the first few singular values capture most of the variance of such simple surface shapes. Hence, we establish a surface approximant (or model) $\mathbf{A}^a$ according to $\mathbf{A}^a = \mathbf{US}'\mathbf{V}^{\mathrm{T}}$, where $\mathbf{S}'$ is a diagonal matrix containing the two largest singular values. The result of the surface approximant is shown in Fig. 3b. After establishing the model of the surface segment, the orthogonal distance can be computed between the original surface segment $\mathbf{A}$ and the surface approximant $\mathbf{A}^a$ as proposed in [16]. However, a simple subtraction of the model from the range data $(\mathbf{A} - \mathbf{A}^a)$ giving the depth map is sufficient (see Fig. 3c), as the differences contributed by the orthogonal distance are negligible due to noise in the data. Our approach for surface approximation compensates the vibrations in the data if the size of the surface segments is moderate. An approach for removing the vibrations in the data by means of a geometric transformation is given in [16].

In order to evaluate the quality of the surface, the depth map is divided into a set of overlapping segments (block size 30×30), whereby each segment is represented by five features (see Table 1) [17]. According to these features, a set of segments is assigned to one of the two classes, which are specified in Sect. 1. These features involve parameters such as variance, skewness, and kurtosis [18].

Table 1 List of features

No.	Feature name
1	Standard deviation
2	Skewness
3	Kurtosis
4	Maximum depth
5	Minimum depth

3 Coupled hidden Markov model

The CHMM is used for modeling interacting random processes, which evolve over time. Figure 4 depicts a CHMM for three interacting processes, where the squares represent the hidden discrete nodes and the shaded circles denote the observation nodes. In our application, the surface segments are acquired sequentially. The different surface segments in each time slot can be viewed as individual processes that are interacting with their immediate spatial adjacent neighbor processes. Hence, the CHMM models neighboring relationships. We assume that a hidden node at time t interacts with the three preceding adjacent hidden nodes at time $t-1$. The interaction of the processes is limited to its immediate neighbor processes. This model has the advantage that surfaces of varying size can be classified according to the trained parameters.

A CHMM is characterized by the following set of parameters λ_{CHMM}:

1. The number of states N is assumed to be equal for each process c.
2. The initial state probability

$$\pi^c(i^c) = P\left(q^c_{t=1} = i^c\right), \tag{1}$$

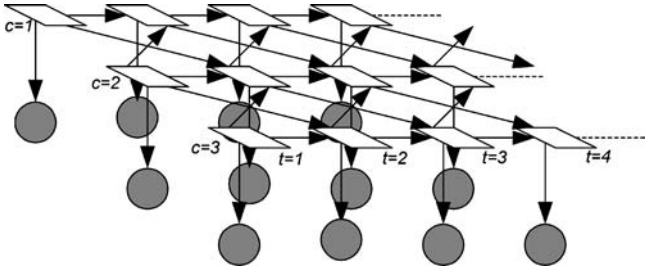

Fig. 4 Coupled hidden Markov model

where q_t^c is the state at time t of the hidden node in the cth process ($c \in \{1, \ldots, C\}$). The state of process c is denoted as i^c. Symbol C denotes the number of coupled processes.

3. The observation probability for state i^c of process c at time t is

$$b_t^c(i^c) = P\left(\mathbf{x}_t^c | q_t^c = i^c\right), \tag{2}$$

where $\mathbf{x}_t^c = [x_{t,1}^c, \ldots, x_{t,d}^c] \in \mathbb{R}^d$ is the observed d-dimensional feature vector of process c at time t. The probability of the observed nodes is modeled as Gaussian mixture model [19] according to

$$b_t^c(i^c) = \sum_{m=1}^{M_{i^c}} \omega_{i^c,m}^c \, p\left(\mathbf{x}_t^c | \theta_{i^c,m}^c\right), \tag{3}$$

where $\omega_{i^c,m}^c$ corresponds to the weight of each component $m = 1, \ldots, M_{i^c}$ of state i^c and process c. These weights are constrained to be positive and $\Sigma_{m=1}^{M_{i^c}} \omega_{i^c,m}^c = 1$. Each component $p(\mathbf{x}_t^c | \theta_{i^c,m}^c)$ is represented as normal distribution with the parameters $\theta_{i^c,m}^c = \{\mu_{i^c,m}^c, \Sigma_{i^c,m}^c\}$, the mean vector and the covariance matrix for the state i^c, component m, and process c. The Gaussian mixture is specified by the set of parameters $\Theta = \{\omega_{i^c,1}^c, \ldots, \omega_{i^c,M_{i^c}}^c, \theta_{i^c,1}^c, \ldots, \theta_{i^c,M_{i^c}}^c\}$.

4. The state transition probability

$$a_{i^c|j^{c-1},j^c,j^{c+1}}^c = P\left(q_t^c = i^c \middle| q_{t-1}^{c-1} = j^{c-1}, q_{t-1}^c = j^c, q_{t-1}^{c+1} = j^{c+1}\right), \tag{4}$$

models the interaction between neighboring processes, whereby for the processes at the edge of the CHMM this probability reduces to $a_{i^1|j^1,j^2}^1$ and $a_{i^C|j^{C-1},j^C}^C$, respectively.

3.1 Training of the CHMM

The parameter estimation of CHMMs can be formulated as maximum likelihood learning which leads to the EM algorithm [20]. The E-step holds the parameters of the model fixed and estimates the posterior distribution over the hidden states. The M-step holds the distribution fixed and computes the model parameters that maximizes for that distribution. Both steps are repeated alternatively until the total likelihood increase falls below a threshold. Basically, the forward–backward algorithm or Viterbi algorithm can be used in the E-step to estimate the posterior probability distribution over the hidden variables. The computational complexity of the forward–backward procedure for CHMMs is $\mathcal{O}(TN^{2C})$. However, different computationally more efficient training techniques have been suggested in the literature which approximate these estimates. Brand et al. [11] introduce N-head dynamic programming and Pavlovic [21] uses variational approximation to estimate the statistics. We extend the EM algorithm for the HMM to train the CHMM. Initially, robust parameter values are established using the Viterbi algorithm in the E-step. Subsequently, the forward–backward algorithm is used to re-estimate the model parameters. We use three coupled processes ($C = 3$) for the algorithms presented in the following.

3.1.1 Robust initialization using Viterbi algorithm

First, the set of R training sequences $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_r, \ldots, \mathbf{X}_R\}$ are divided according to the number of states into N segments by using the k-means algorithm [22]. This results in an assignment of the observation vectors $\mathbf{x}_{t,r}^c$ to states $q_{t,r}^c$ and an initial state sequence of the coupled nodes is obtained. Each $\mathbf{X}_r$ consists of a set of observation vectors $\{\mathbf{x}_{1:T,r}^{1:C}\}$, where the notation $1:T$ is the set of variables with indices from 1 to T. Once the state segmentation has been established, the observations that belong to a particular state and models the observation probability distribution are further segmented to M_{i^c} clusters by using the k-means algorithm. Hence, the observation vector assigned to a state is additionally associated with a Gaussian mixture component m. We get a state segmentation $q_{t,r}^c$ and an initial segmentation of data to the components of the Gaussian mixture distribution denoted as $z_{t,r}^c$. After initialization, we proceed with the M-Step.

M-step: Once the twofold segmentation is obtained, new parameters of the model are estimated according to the following equations:

$$\omega_{i^c,m}^c = \frac{\sum_{r=1}^R \sum_{t=1}^T \gamma_{t,r}^c(i^c, m)}{\sum_{r=1}^R \sum_{m'=1}^{M_{i^c}} \sum_{t=1}^T \gamma_{t,r}^c(i^c, m')}, \tag{5}$$

$$\mu_{i^c,m}^c = \frac{\sum_{r=1}^R \sum_{t=1}^T \gamma_{t,r}^c(i^c, m) \mathbf{x}_{t,r}^c}{\sum_{r=1}^R \sum_{t=1}^T \gamma_{t,r}^c(i^c, m)}, \tag{6}$$

$$\Sigma_{i^c,m}^c = \frac{\sum_{r=1}^R \sum_{t=1}^T \gamma_{t,r}^c(i^c, m)\left(\mathbf{x}_{t,r}^c - \mu_{i^c,m}^c\right)\left(\mathbf{x}_{t,r}^c - \mu_{i^c,m}^c\right)^T}{\sum_{r=1}^R \sum_{t=1}^T \gamma_{t,r}^c(i^c, m)}, \tag{7}$$

$$\pi^c(i^c) = \frac{\sum_{r=1}^R \epsilon_r^c(i^c)}{\sum_{r=1}^R \sum_{i^c=1}^N \epsilon_r^c(i^c)}, \tag{8}$$

and

$$a_{i^c|j^{c-1},j^c,j^{c+1}}^c = \frac{\sum_{r=1}^R \sum_{t=2}^T \zeta_{t,r}^c(i^c, j^{c-1}, j^c, j^{c+1})}{\sum_{r=1}^R \sum_{t=2}^T \sum_{j^{c-1}=1}^N \sum_{j^c=1}^N \sum_{j^{c+1}=1}^N \zeta_{t,r}^c(i^c, j^{c-1}, j^c, j^{c+1})}, \tag{9}$$

where

$$\gamma_{t,r}^c(i^c, m) = \begin{cases} 1, & \text{if } q_{t,r}^c = i^c \text{ and } z_{t,r}^c = m \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

$$\epsilon_r^c(i^c) = \begin{cases} 1, & \text{if } q_{t=1,r}^c = i^c \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

and

$$\zeta_{t,r}^c(i^c, j^{c-1}, j^c, j^{c+1}) = \begin{cases} 1, & \text{if } q_{t,r}^c = i^c, q_{t-1,r}^{c-1} = j^{c-1}, q_{t-1,r}^c = j^c, \\ & \quad \text{and } q_{t-1,r}^{c+1} = j^{c+1} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

For the transition probabilities $a^1_{i^1|j^1,j^2}$ and $a^C_{i^C|j^{C-1},j^C}$ the modification of Eqs. 9 and 12 is straightforward.

E-step: The optimal state sequence of the coupled nodes is determined using the Viterbi algorithm. The Viterbi algorithm is presented in the following:

Initialization: $(1 \le i^c \le N, \forall c = \{1, \ldots, 3\})$

$$\delta_{t=1,r}(i^1, i^2, i^3) = \prod_{c=1}^{C} \pi^c(i^c) b_{t=1,r}^c(i^c), \tag{13}$$

$$\psi_{t=1}(i^1, i^2, i^3) = 0. \tag{14}$$

Recursion: $(1 \le i^c, j^c \le N, \forall c = \{1, \ldots, 3\}, 2 \le t \le T)$

$$\delta_{t,r}(i^1, i^2, i^3) = \max_{1 \le j^1, j^2, j^3 \le N} \Big\{ \delta_{t-1,r}(j^1, j^2, j^3) \times a^1_{i^1|j^1,j^2} a^2_{i^2|j^1,j^2,j^3} a^3_{i^3|j^2,j^3} \Big\} \prod_{c=1}^{C} b_{t,r}^c(i^c), \tag{15}$$

$$\psi_{t,r}(i^1, i^2, i^3) = \arg\max_{1 \le j^1, j^2, j^3 \le N} \Big\{ \delta_{t-1,r}(j^1, j^2, j^3) a^1_{i^1|j^1,j^2} a^2_{i^2|j^1,j^2,j^3} a^3_{i^3|j^2,j^3} \Big\}. \tag{16}$$

Termination: $(t = T)$

$$P_r^* = \max_{1 \le i^1, i^2, i^3 \le N} [\delta_{T,r}(i^1, i^2, i^3)], \tag{17}$$

$$\big[q^1_{T,r}, q^2_{T,r}, q^3_{T,r}\big] = \arg\max_{1 \le i^1, i^2, i^3 \le N} [\delta_{T,r}(i^1, i^2, i^3)] \tag{18}$$

Backtracking of the path: $(t = T-1, T-2, \ldots, 1)$

$$\big[q^1_{t,r}, q^2_{t,r}, q^3_{t,r}\big] = \psi_{t+1,r}\big(q^1_{t+1,r}, q^2_{t+1,r}, q^3_{t+1,r}\big). \tag{19}$$

The Viterbi algorithm achieves a new state segmentation of the data. The assignment of the data to a particular component of the Gaussian mixture distribution is obtained according to

$$z_{t,r}^c = \arg\max_{m=1,\ldots,M_{i^c}} P\big(m|\mathbf{x}_{t,r}^c, q_{t,r}^c = i^c, \Theta\big) = \frac{\omega_{i^c,m}^c p\big(\mathbf{x}_{t,r}^c|\theta_{i^c,m}^c\big)}{\sum_{m'=1}^{M_{i^c}} \omega_{i^c,m'}^c p\big(\mathbf{x}_{t,r}^c|\theta_{i^c,m'}^c\big)}. \tag{20}$$

The E-step and the M-step are repeated until the sum $\Sigma_{r=1}^R P_r^*$ of the training set converges to an optimum. The classification of sequences of surface segments with the Viterbi algorithm is based on P_r^*.

3.1.2 Forward–backward algorithm for training the CHMM

Once the CHMM has been initialized by means of the Viterbi training, the parameters are re-estimated using the forward–backward algorithm in the E-step of the EM algorithm.

E-step: First the forward probabilities $\alpha_{t,r}(i^1, i^2, i^3) = P(\mathbf{x}_{1:t,r}^{1:C}, q_t^1 = i^1, q_t^2 = i^2, q_t^3 = i^3)$ are computed. The forward probabilities are initialized $(t = 1)$ according to

$$\alpha_{1,r}(i^1, i^2, i^3) = \prod_{c=1}^{C} \pi^c(i^c) b_{1,r}^c(i^c) \tag{21}$$

for all states $i^c = 1, \ldots, N$ and processes $c = 1, \ldots, 3$. Afterwards, the forward probabilities are determined recursively for $t = 2, \ldots, T$ as

$$\alpha_{t,r}(i^1, i^2, i^3) = \prod_{c=1}^{C} b_{t,r}^c(i^c) \sum_{j^1=1}^{N} \sum_{j^2=1}^{N} \sum_{j^3=1}^{N} \alpha_{t-1,r}(j^1, j^2, j^3) \times a^1_{i^1|j^1,j^2} a^2_{i^2|j^1,j^2,j^3} a^3_{i^3|j^2,j^3}. \tag{22}$$

In a similar manner, the backward probabilities defined as $\beta_{t,r}(i^1, i^2, i^3) = P(\mathbf{x}_{t+1:T,r}^{1:C}|q_{t,r}^1 = i^1, q_{t,r}^2 = i^2, q_{t,r}^3 = i^3)$ are computed. The backward probabilities are initialized for $t = T$ as

$$\beta_{T,r}(i^1, i^2, i^3) = 1. \tag{23}$$

Afterwards, the backward probabilities are determined recursively according to

$$\beta_{t,r}(i^1, i^2, i^3) = \sum_{j^1=1}^{N} \sum_{j^2=1}^{N} \sum_{j^3=1}^{N} \prod_{c=1}^{C} b_{t+1,r}^c(j^c) \times a^1_{j^1|i^1,i^2} a^2_{j^2|i^1,i^2,i^3} a^3_{j^3|i^2,i^3} \beta_{t+1,r}(j^1, j^2, j^3), \tag{24}$$

where $t = T-1, T-2, \ldots, 1$. The computational complexity for the forward–backward algorithm is $\mathcal{O}(TN^{2C})$. Hence, it is only practical for small values of C and N.

M-step: With the forward and backward probabilities, the parameters Θ of the observation probability distribution, i.e. $\omega_{i^c,m}^c$, $\mu_{i^c,m}^c$, and $\Sigma_{i^c,m}^c$, are re-estimated as in Eqs. 5–7, where

$$\gamma_{t,r}^c(i^c, m) = \frac{\overbrace{\sum_{i^1=1}^{N} \sum_{i^2=1}^{N} \sum_{i^3=1}^{N}}^{\text{so that } q_{t,r}^c = i^c} \alpha_{t,r}(i^1, i^2, i^3) \beta_{t,r}(i^1, i^2, i^3)}{\sum_{i^1=1}^{N} \sum_{i^2=1}^{N} \sum_{i^3=1}^{N} \alpha_{t,r}(i^1, i^2, i^3) \beta_{t,r}(i^1, i^2, i^3)} \cdot \frac{\omega_{i^c,m}^c p\big(\mathbf{x}_{t,r}^c|\theta_{i^c,m}^c\big)}{\sum_{m'=1}^{M_{i^c}} \omega_{i^c,m'}^c p\big(\mathbf{x}_{t,r}^c|\theta_{i^c,m'}^c\big)}. \tag{25}$$

The initial state probability and the transition probability are re-estimated according to

$$\pi^c(i^c) = \frac{\sum_{r=1}^{R}\sum_{m=1}^{M_{ic}} \gamma_{1,r}^c(i^c, m)}{\sum_{r=1}^{R}\sum_{i^c=1}^{N}\sum_{m=1}^{M_{ic}} \gamma_{1,r}^c(i^c, m)} \tag{26}$$

and

$$a^2_{i^2|j^1,j^2,j^3} = \frac{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T-1}\overbrace{\sum_{i^1=1}^{N}\sum_{i^2=1}^{N}\sum_{i^3=1}^{N}}^{\text{so that } q^2_{t,r}=i^2}\alpha_{t,r}(j^1,j^2,j^3)a^1_{i^1|j^1,j^2}a^2_{i^2|j^1,j^2,j^3}a^3_{i^3|j^2,j^3}\prod_{c=1}^{C} b^c_{t+1,r}(i^c)\beta_{t+1,r}(i^1,i^2,i^3)}{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T-1}\alpha_{t,r}(j^1,j^2,j^3)\beta_{t,r}(j^1,j^2,j^3)}. \tag{27}$$

The likelihood P_r of the observation sequence $\mathbf{X}_r$ given the model λ_{CHMM} is obtained using the forward probabilities as follows

$$P_r = P(\mathbf{X}_r|\lambda_{\text{CHMM}}) = \sum_{i^1=1}^{N}\sum_{i^2=1}^{N}\sum_{i^3=1}^{N}\alpha_{T,r}(i^1,i^2,i^3). \tag{28}$$

The transition probabilities $a^1_{i^1|j^1,j^2}$ and $a^C_{i^C|j^{C-1},j^C}$ are derived analogously.

3.2 Classification using CHMM

The classification of the sequences of surface segments is based on the likelihood of an observation sequence $\mathbf{X}_r = \{\mathbf{x}^{1:C}_{1:T,r}\}$ with respect to the CHMM λ_{CHMM}. This means that we train a CHMM for both, surface segments with errors and intact surface segments. The likelihood of the observation sequence given the model is obtained with the probabilities of the forward–backward algorithm according to

$$\begin{aligned} P(\mathbf{X}_r|\lambda_{\text{CHMM}}) &= \sum_{i^1=1}^{N}\sum_{i^2=1}^{N}\sum_{i^3=1}^{N}\alpha_{T,r}(i^1,i^2,i^3) \\ &= \sum_{i^1=1}^{N}\sum_{i^2=1}^{N}\sum_{i^3=1}^{N}\left[\prod_{c=1}^{C}\pi^c(i^c)b^c_{1,r}(i^c)\right]\beta_{1,r}(i^1,i^2,i^3). \end{aligned} \tag{29}$$

The model that achieves the largest likelihood for an observation sequence $\mathbf{X}_r$ gives the class of the data.

4 Experimental results

For the experiments, we have a data set $\mathcal{S}$ consisting of 420 samples uniformly distributed into two classes (see Sect. 1). The data set is divided into five mutually exclusive subsets $\mathcal{S} = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5\}$, which are used to determine the fivefold cross validation classification rate. As shown in Fig. 2, the size of the flaws can vary strongly. The aim is to model surface flaws, which are stretched over one or more neighboring surface segments. Due to the setup of the system (∼0.1 mm/pixel lateral resolution, ∼0.7 mm/section resolution in moving direction, segment size of 30 × 30) we use a CHMM with $C = 5$ coupled processes and a sequence length of $T = 10$. This is a reasonable choice considering the size of the flaws in the data set (some samples are shown in Fig. 2). This means that flaws of a size of ∼15 mm × ∼200 mm can be modeled. Each sample $\mathbf{X}_r = \{\mathbf{x}^{1:C}_{1:T,r}\}$ consists of 50 surface segments and each surface segment is described by five features (see Table 1). The observation probabilities for the CHMM and HMM classifiers are modeled with a Gaussian mixture model using diagonal covariance matrix and three Gaussian components ($M_{ic} = 3$). The transition probabilities $a^c_{i^c|j^{c-1},j^c,j^{c+1}}$ for $c = 2,\ldots,4$ are tied to keep the number of parameters of the model low.

Table 2 compares the naïve Bayes classifier, the Hidden Markov Model, the k-nearest neighbor classifier, and the Support Vector Machine with the proposed CHMM. We use the following abbreviations for the different classification approaches:

- CHMM-2: Coupled Hidden Markov Model with $N = 2$ states. The CHMM has been trained with the EM algorithm using first the Viterbi algorithm and subsequently the forward–backward algorithm in the E-step (see Sect. 3.1).
- CHMM-3: Coupled Hidden Markov Model with $N = 3$ states.
- HMM-2: Hidden Markov Model with $N = 2$ states. The feature vectors are used row by row to form the data sequence $\{\mathbf{x}^1_{1,r}, \mathbf{x}^2_{1,r}, \ldots, \mathbf{x}^5_{1,r}, \mathbf{x}^1_{2,r}, \ldots, \mathbf{x}^5_{T,r}\}$ for the HMM. This means that the length of the sequence is $TC = 50$.
- HMM-3: Hidden Markov Model with $N = 3$ states. The usage of the data is the same as for HMM-2.

Table 2 Classification performance (fivefold cross validation)

Method	CR-Sample (%)	CR-Segment (%)
CHMM-2	98.10 ± 1.35	–
CHMM-3	98.57 ± 1.55	–
HMM-2	93.33 ± 3.22	–
HMM-3	94.76 ± 3.43	–
HMM25-2	93.75 ± 3.65	–
HMM25-3	93.32 ± 3.81	–
3NN	91.19 ± 4.26	99.67 ± 0.14
5NN	79.28 ± 3.53	99.02 ± 0.26
7NN	75.00 ± 5.58	98.64 ± 0.31
SVM	96.19 ± 4.48	99.80 ± 0.29
NB	49.05 ± 1.30	85.83 ± 1.00

CR-Sample: classification rate based on the data set samples. CR-Segment: classification rate of the surface segments.

- HMM25-2: Hidden Markov Model with $N = 2$ states. The features of the $C = 5$ processes are concatenated into a single vector $\mathbf{x}_{t,r} = [\mathbf{x}_{t,r}^{c=1}, \mathbf{x}_{t,r}^{c=2}, \mathbf{x}_{t,r}^{c=3}, \mathbf{x}_{t,r}^{c=4}, \mathbf{x}_{t,r}^{c=5}]$ with a dimension of $d = 25$. Hence, we form a HMM with the observation sequence $\mathbf{x}_{1:T,r}$.
- HMM25-3: Hidden Markov Model with $N = 3$ states. The usage of the data is the same as for HMM25-2.
- k-NN: k-nearest neighbor classifier, $k \in \{3, 5, 7\}$.
- SVM: The experiment with the Support Vector Machine (SVM) uses a radial basis function (RBF) kernel [23, 24]. The SVM uses two parameters C^* and σ, where C^* is the penalty parameter for the errors of the non-separable case and σ is a parameter for the RBF kernel. The values for these parameters $C^* = 2$ and $\sigma = 0.0625$ have been determined by a search using cross validation.
- NB: Naïve Bayes classifier.

For the k-NN, the NB, and the SVM we provide the five-fold cross validation classification performance for the data set samples (CR-Sample) and for the individual surface segments (CR-Segment). For these classifiers we use the class information of all surface segments during training. Then we classify each individual surface segment $\mathbf{x}_{1:T,r}^{1:C}$ in the sample. If all segments in the sample are classified to the manually assigned labels then the sample is correctly classified (CR-Sample). For the CHMM and HMM we use the class information of the whole sample (CR-Sample) containing 50 surface segments. The sample belongs to the erroneous class if at least one of its segments contain a flaw. Hence, we are using much more class information for training the k-NN, the NB, and the SVM compared to the HMM and CHMM classifiers.

The CHMM trained with the Viterbi and the backward–forward algorithm performs better than the HMM approach. The best sample-based recognition rate of 98.57% is achieved with the CHMM using three states. However, the performance of 96.19% of the SVM on the samples (CR-Sample) is comparable.

As mentioned earlier, the NB, SVM, and k-NN classifiers are trained with the class information of each surface segment. Hence, we present the classification performance of the surface segments (CR-Segment) and of the samples (CR-Sample). The SVM classifier achieves a classification accuracy of 99.80% on the surface segments. This is the best compared to the NB and the k-NN classifier.

The computational complexity for classification and the number of parameters (#Parameters) for the different classification approaches are listed in Table 3.

The number of support vectors of the SVM and the number of reference samples of the k-NN are denoted as N_{SV} and N_{ref}, respectively. The N_{ref} and the N_{SV} for the k-NN and SVM classifiers are large, which leads to relatively high computational costs for classification. For the classification of one sample (CR-Sample) we have to classify TC surface segments. The CHMM and HMM approaches outperform the SVM and k-NN classifiers in terms of memory requirements (see #Parameters in Table 3). However, the computational complexity for these models is high for large N, especially for the CHMM it is exponential with C. For large values of C the CHMM is clearly intractable. The second term $TNCM_{i^c}d$ in the table gives the complexity for computing the observation probabilities $b_t^c(i^c)$ of the Gaussian mixture model. It can be neglected in contrast to the first term TN^{2C}, which models the complexity of the forward–backward procedure of the CHMM. However, we keep this term for the purpose of comparison. The intention of the CHMM is to improve the classification performance by modeling the interaction between neighboring surface segments. This is useful, as we have to deal with flaws of strongly varying sizes. Compared to the SVM we achieve a slightly better recognition rate. The CHMM uses a lower number of parameters. However, the computational complexity for classification is exponential in C.

5 Conclusions and future work

We have proposed a novel approach based on coupled HMMs that detects and classifies defects on steel surfaces. We have to deal with two difficulties: (i) Due to scale the reflection property across the surface changes strongly. Hence, intensity imaging fails. We use light sectioning to acquire the surface range data. (ii) The acquired data shows vibrations, which are caused by the movement of the steel block on the conveyor. We show an approach for the recovery of the depth map of the surface. Subsequently, sequences of

Table 3 Number of parameters (#Parameters) and computational complexity for classification

Method	#Parameters	Computational complexity
CHMM-2	$(2d+1)M_{i^c}CN + CN + 2N^3 + N^4 = 372$	$\mathcal{O}(TN^{2C} + TNCM_{i^c}d)$
CHMM-3	$(2d+1)M_{i^c}CN + CN + 2N^3 + N^4 = 645$	
HMM-2	$(2d+1)M_{i^c}N + N + N^2 = 72$	$\mathcal{O}(TN^2 + TNM_{i^c}d)$
HMM-3	$(2d+1)M_{i^c}N + N + N^2 = 111$	
HMM25-2	$(2d+1)M_{i^c}N + N + N^2 = 312$	$\mathcal{O}(TN^2 + TNM_{i^c}d)$
HMM25-3	$(2d+1)M_{i^c}N + N + N^2 = 471$	
k-NN	$d \times N_{\text{ref}}(N_{\text{ref}} = 16{,}800$ segments)	$\mathcal{O}(TCdN_{\text{ref}})$
SVM	$d \times N_{\text{SV}}(N_{\text{SV}} = 5{,}200)$	$\mathcal{O}(TCdN_{\text{SV}})$
NB	$2(2d+1) = 22$	$\mathcal{O}(TCd)$

surface segments are classified by means of CHMMs. We compared the CHMM with the naïve Bayes classifier, the Hidden Markov Model, the k-nearest neighbor classifier, and the Support Vector Machine. The CHMM outperforms the other approaches and achieves a recognition rate of 98.57%. However, the classification performance of 96.19% for the Support Vector Machine is comparable. The computational complexity for the CHMM is exponential with the number of coupled processes C. This means that for a large number of C the CHMM is intractable. In future, we want to focus on different approximate learning algorithms for the CHMM to reduce the computational complexity of this approach.

References

1. Newman, T., Jain, A.: A survey of automated visual inspection. Comput. Vision Image Understand. **61**(2), 231–262 (1995)
2. Pernkopf, F., O`Leary, P.: Image acquisition techniques for automatic visual inspection of metallic surfaces. NDT E Int. **36**(8), 609–617 (2003)
3. Dupont, F., Odet, C., Carton, M.: Optimization of the recognition of defects in flat steel products with the cost matrices theory. NDT N Int. **30**(1), 3–10 (1997)
4. Pernkopf, F., O`Leary, P.: Visual inspection of machined metallic high-precision surfaces, special issue on applied visual inspection. EURASIP J. Appl. Signal Process. (7), 667–678 (2002)
5. Nayar, S., Ikeuchi, K., Kanade, T.: Surface reflection: physical and geometrical perspectives. IEEE Trans. Pattern Anal. Machine Intell. **13**(7), 611–634 (1991)
6. Torrance, K., Sparrow, E.: Theory for off-specular reflection from roughened surfaces. J. Optical Soc. Am. **57**(9), 1105–1114 (1967)
7. Proceedings of SPIE. Machine Vision Applications in Industrial Inspection. URL: www.spie.org
8. Kanade, R.: Three-Dimensional Machine Vision. Kluwer Academic Publishers, Drodrecht (1987)
9. Curless, B., Levoy, M.: Better optical triangulation through spacetime analysis. In: 5th International Conference on Computer Vision, pp. 987–994 (1995)
10. IVP (Integrated Vision Products), Company. IVP Ranger SAH5 Product Information. URL: www.ivp.se.
11. Brand, M., Oliver, N., and Pentland, A.: Coupled hidden markov models for complex action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 994–999 (1997)
12. Nefian, A.: Embedded Bayesian networks for face recognition. In: IEEE International Conference on Multimedia and Expo, vol. 2, pp. 133–136 (2002)
13. Nefian, A., Liang, L., Pi, X., Liu, X., Murphy, K.: Dynamic Bayesian networks for audio-visual speech recognition. EURASIP J. Appl. Signal Process. **11**, 1–15 (2002)
14. Long, A., Long, C.: Surface approximation and interpolation via matrix SVD. College Math. J. **32**(1), 20–25 (2001)
15. Golub, G., Loan, C.V.: Matrix Computations, 3rd ed. The John Hopkins University Press (1996)
16. Pernkopf, F., Pernkopf, F., O`Leary, P.: Detection of surface defects on raw milled steel blocks using range imaging. In: IT&S/SPIE 14th Symposium Electronic Imaging, pp. 170–181 (2002)
17. Pernkopf, F.: Detection of surface defects on raw steel blocks using Bayesian network classifiers. Pattern Anal. Appl. **7**(3), 333–342 (2004)
18. Gonzalez, R., Woods, R.: Digital Image Processing. Addison-Wesley, Reading (1992)
19. McLachlan, G., Peel, D.: Fintite Mixture Models. Wiley, New York (2000)
20. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood estimation from incomplete data via the EM algorithm. J. R. Stat. Soc. **30**(B), 1–38 (1977)
21. Pavlovic, V.: Dynamic Bayesian networks for information fusion with applications to human–computer interfaces. PhD thesis, University of Illinois at Urbana-Champaign (1999)
22. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, New York (2000)
23. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowledge Discov. **2**(2), 121–167 (1998)
24. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2001)

Franz Pernkopf received his MSc (Dipl. Ing.) degree in Electrical Engineering at Graz University of Technology, Austria, in summer 1999. He earned a PhD degree from the University of Leoben, Austria, in 2002. In 2002 he was awarded the Erwin Schrödinger Fellowship. In 2004 he was a research associate at the Department of Electrical Engineering at the University of Washington, Seattle. Currently, he is an university assistant at the Signal Processing and Speech Communication Laboratory at Graz University of Technology, Austria. His research interests include graphical models, generative and discriminative learning of Bayesian network classifiers, feature selection, finite mixture models, image processing and vision, and statistical pattern recognition.