Production Engineering (2022) 16:719–734
https://doi.org/10.1007/s11740-022-01118-x

QUALITY ASSURANCE

Intelligent predetection of projected reference markers for robot-based inspection systems

Philipp Bauer[1] · Stefan Schmitt[1] · Jonas Dirr[1] · Alejandro Magaña[1] · Gunther Reinhart[1]

Received: 15 October 2021 / Accepted: 8 February 2022 / Published online: 28 February 2022

Abstract

Technical advancements in optical devices like sensors and projectors have led to tremendous innovations in manufacturing metrology, not least due to reductions in cost and the use of sophisticated image processing software. More recently, methods based on machine learning have demonstrated their high potential in meeting challenges that are difficult to overcome using conventional image processing techniques. In this context, we present an approach for the intelligent predetection of projected reference markers in robot-based inspection systems. These markers support the alignment of different sensor views and do not need to be physically attached to any parts. However, their robust detection is challenging under unfavorable lighting conditions. Hence, we introduce trained models of a cascade classifier based on both synthetic and real image data. Subsequently, we present the detection performance for different shapes and designs of markers projected onto real-world sheet metal parts as used in the automotive industry. The results demonstrate that properly trained classifiers can achieve a recall and precision of 90% and higher. The use of intelligent predetection promises more robust results in the subsequent detection of projected markers and, thus, benefits image processing in particular in geometric quality assurance applications.

Keywords Projected reference markers · Machine learning · Robot-based inspection systems · Geometric quality assurance · Manufacturing metrology

1 Introduction

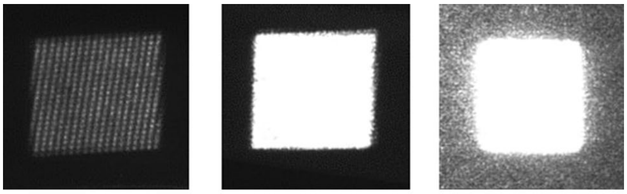
Robot-based inspection systems (RIS) usually comprise an industrial robot and a vision sensor. They are used in tasks involving the inspection of free-form surfaces, such as surface quality assurance or ascertaining the geometric dimensional conformance of sheet metal parts [1]. Due to continuous improvements in vision devices [1], RISs offer advantages over conventional contact measuring methods such as coordinate measuring machines. However, for larger parts, for example from the automotive industry, it is necessary to acquire images from different viewpoints and merge them afterwards [1], because the field of view (FOV) of typically employed vision sensors is too small. The alignment of multiple views is crucial, because it strongly influences the quality of the digitization of a part. Given the high accuracy requirements in the manufacturing industry, physical markers (fiducials) are often used to enhance alignment in close-range photogrammetry applications [1, 2].

Motivated by the idea of reference markers which are no longer attached to a part but are projected onto it using a commercially available projector, a concept was previously proposed for the alignment of point clouds [3]. This concept uses projections which are applied in a region-specific manner onto flat areas on the surface of a part. Consequently, projected markers can be placed exclusively onto dedicated regions of a part, which are selected in advance. Projecting markers requires less manual effort than fiducials [4]. Furthermore, physical contact is avoided, which decreases the chance of damage [4] or accidentally missing markers during detachment.

In an industrial environment, however, the conditions for capturing images with the aforementioned markers are often changing and unfavorable. Such conditions can impact images of markers in the form of deviations in scale, exposure, and contrast, as well as in perspective distortion [5]. In particular, the contrast of markers in relation to their surroundings/background is crucial for their detection. The

✉ Philipp Bauer
philipp.a.bauer@tum.de

[1] Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany



(a) Underexposed

(b) Normally exposed

(c) Overexposed

Fig. 1 Varying lighting conditions for projected markers in different regions of the same measurement object

contrast between areas with and without projection is usually lower than with physical markers due to the limited contrast inherent to conventional projection devices. The detection of projected markers is particularly difficult when the already lower contrast is superimposed by a high contrast variation in the image. Such contrast variations can be caused by unfavorable combinations of pose-dependent illumination and component surface orientation [6]. Examples of challenging lighting conditions in the detection of projected markers are shown in Fig. 1a, c. As opposed to this, Fig. 1b represents a normally exposed marker.

Most conventional marker detection approaches first binarize the image, for instance by thresholding or calculating the edges, and then filter the remaining pixel by querying simple, predefined, and handcrafted features, such as the area of a blob or the length of an edge [7, 8]. While these methods can be effective under favorable conditions or for physical markers with a high contrast, they tend to fail with projected markers. Figure 1, for example, illustrates the varying contrast of such markers. It usually represents markers projected to different locations on the surface of a part for which it is often not possible to find a common parameterization of the aforementioned conventional detection approaches. Therefore, a novel approach is proposed using machine learning for industrial marker detection. It enables an intelligent predetection in order to facilitate the subsequent image processing, see Fig. 2. In this regard, it raises the questions of which machine learning method promises good results to robustly detect projected reference markers in industrial applications and how to train the selected classifier.

From the intelligent predetection comes the advantage of higher robustness of the detection under varying lighting conditions and the ability to successfully apply basic routines, such as thresholding or computing gradients, for feature extraction in significantly more challenging environments. In addition, the predetection is independent of the kinematic chain and internal camera parameters. Thus, under similar environmental conditions the one-time trained classifier for the detection of projected markers can be transferred to various handling and camera systems as well as new applications in production technology without requiring much expert knowledge.

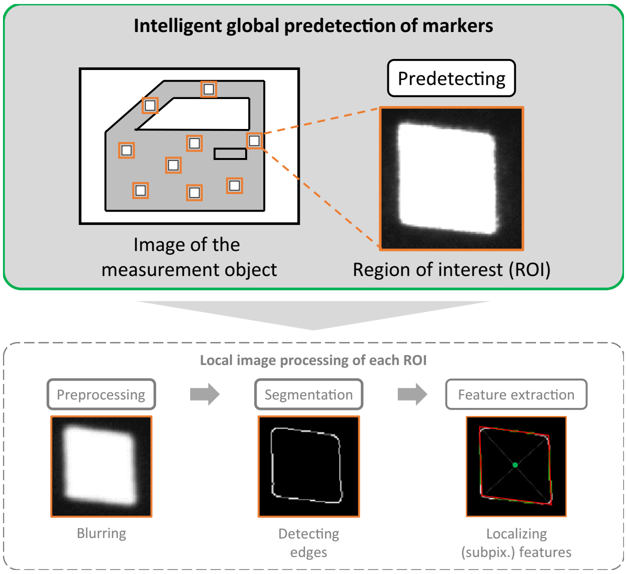

Intelligent global predetection of markers
Predetecting
Image of the measurement object
Region of interest (ROI)
Local image processing of each ROI
Preprocessing
Segmentation
Feature extraction
Blurring
Detecting edges
Localizing (subpix.) features

Fig. 2 Illustration of the image processing method for obtaining features of projected reference markers

1.1 Contributions

The new contributions made by this article are as follows: (1) we apply the Viola–Jones method [9, 10], which comprises a cascade classifier in combination with Haar-like features, for predetecting projected markers in RIS. (2) Our approach is transferable to different camera systems, markers, and sheet metal parts. (3) We train the cascade classifier with synthetic images and find real-world influences, e.g. variation in illumination intensity, blur, noise, which need to be modeled.

1.2 Outline

The following section presents the state of the art of fiducial marker detection. In Sect. 3, we present our novel approach for the intelligent predetection of projected markers followed by the experimental setup in Sect. 4. Section 5 discusses the experimental investigation of the method, while the main conclusions are presented in Sect. 6.

2 State of the art and related work

This section presents methods regarding marker detection in images, divided into established techniques for detecting and verifying possible candidates as markers (Sect. 2.1) and approaches using machine learning (Sect. 2.2).

2.1 Common techniques for localizing artificial markers in images

Several marker systems have been proposed in the literature, including markers for augmented reality (AR) and robot navigation [11–13], photogrammetry [14, 15] and point cloud alignment [2]. Each has its own set of predefined and tuned methods of robustly locating specified targets in images, depending on the target shape, application, and constraints such as hardware or computation time. The localization of artificial markers, as employed in many computer vision tasks, is usually a two-step process, comprising *candidate detection* and *marker verification* [7, 8]. During the detection process, possible marker candidates are found, for example, based on separated blobs or edges. To reduce the number of false positives, an additional verification/identification step is required.

2.1.1 Candidate detection

To detect potential marker candidates, the image is segmented to remove non-significant image regions (image background). Due to their black and white appearance and frequently retroreflective properties [16], artificial markers usually exhibit strong contrast and are captured as bright areas in the image. Therefore, the simplest way of separating any contiguous partial areas from the image background is to conduct a thresholding operation. This process is called *blob detection* [7] and either uses a fixed global threshold, for example determined from an image histogram [15, 17], or a local adaptive threshold, computed in a small neighborhood around each pixel [18–20]. Local adaptive methods are quite common, since they achieve better performance in poor or uneven/inhomogeneous lighting conditions [19].

Another common approach is to use edges for the detection of possible marker candidates [12, 21, 22]. Since the addressed markers exhibit strong and sharp contrast at their boundaries, edges can easily be computed by detecting the maximum gray value gradients in the image. Furthermore, the definition of an absolute threshold, as it is the case with global thresholding operations, can be circumvented, which may be problematic under difficult lighting conditions [7]. In addition, partially occluded markers can be detected using heuristic methods [7, 19].

The majority of markers have a black and white design to achieve maximum contrast in the image to facilitate detection and identification processes. Some authors, however, use colored markers to improve the initial positive detection rate. By using a camera capable of recording color information, these markers can be more easily segmented by switching to the appropriate color channel in the image and performing a thresholding operation [14, 23] or by extracting the edges [24]. DeGol et al. [23] used opposing color patterns, especially red and green, because such color gradients are rare in natural scenes and exhibit a strong contrast. Fraser and Cronk [25] used color as an additional detection cue to distinguish their red-colored markers from bright white reflections, which might otherwise be set as valid markers.

Barone et al. [2] implemented a segmentation method based on texture, in which they employ a statistical measurement approach to classify the direct local texture around each pixel and applied morphological operations to segment the image into homogeneous and inhomogeneous intensity regions.

Dosil et al. [26] developed a method based on the measurement of symmetry. Radial symmetry is identified by a combination of multiple symmetry detectors with different orientations. Local maxima in the responses of the proposed detectors form regions of circular marker candidates.

2.1.2 Marker verification

Once all the marker candidates have been separated from the image background, one or more verification steps are usually applied to enable any false positives to be rejected. Common methods of removing incorrectly detected markers involve checking basic geometric properties such as the size, diameter, aspect ratio, or circularity of the detected blobs or edges [2, 7, 14, 15, 25]. More advanced methods use the response of the Hough transformation [24], the fitting of an ellipse and consideration of the error of the fit [26], template matching and break at a certain error [27], the approximation of a polygon and testing of its topology [18–20] or form factors [28].

Fiala [7, 12] filtered square AR markers called ARTag by segmenting extracted edges into straight line sections and grouping them into quadrilaterals.

With more complex markers, e.g. encoded fiducials, the decoding process is used to verify the integrity of detected markers. This is also called the identification step according to [7]. Examples are ARTag [7, 12], ARToolKit [13], ARToolKitPlus [11], or in the circular case, FourierTag [29]. Furthermore, structural integrity tests can be applied to markers consisting of multiple sub-geometries. These patterns are added to the shape of the markers and are used for verification. A common example are the finder patterns located at the corners of QR codes [30].

Often, the main reason for using artificial markers is to precisely localize artificial feature points, which require a proper candidate detection and verification of potential markers in advance. The aforementioned detection and verification methods are often associated with a succeeding feature point calculation, since calculated feature points, e.g. centroids or corners, often rely on the geometric shape of the artificial markers and thus can be used as an additional verification cue.

2.2 Use of machine learning for marker detection

Most of the methods presented in Sects. 2.1.1 and 2.1.2 rely on handcrafted features. The challenge with such features is to define explicit and simple detection rules that provide a sufficient description of the markers' representations [31]. Hence, the development of these types of features requires a lot of time and a comprehensive knowledge of the application area and its conditions [5, 32]. Inaccuracies in this process can also lead to a declining detection rate, especially when background clutter, deviations in scale, perspective distortion, or inconsistent illumination occur [5, 32].

Some authors address these problems and switch to machine learning methods for detecting and/or identifying markers in images. Among the first authors to use machine learning in the context of marker detection were Claus and Fitzgibbon [5, 33]. They implemented a two-stage cascading classifier based on an ideal Bayes decision rule and using nearest-neighbor classification to detect fiducial targets in real-world scenes. An intensity pair consisting of the center and one edge pixel of each detected candidate window was used for classification.

Belussi and Hirata [30] used a cascading classifier based on the Viola–Jones rapid object detection framework in combination with Haar-like features [9, 10] to detect the finder patterns of QR codes in natural scenes. Yuan et al. [34] applied a QR code detection procedure based on the extraction of BING (binarized normed gradients) features attached to an AdaBoost-SVM (support vector machine). However, the drawback of BING features is their poor proposal window (ROI) localization accuracy [35]. Chou et al. [32] took a deep learning approach by applying a modified CNN (convolutional neural network) to the localization and segmentation of QR codes. CNN-based models are known for their good detection rates, but their performance decreases with varying and inconsistent image data [36]. Jiang et al. [37] introduced an automatic detection algorithm for fiducial markers in medical X-ray images based on the calculation of HOG (histogram of gradients) features in combination with an SVM.

Besides marker detection, machine learning techniques can also be employed to verify and decode previously detected marker candidates. Instead of using a binary classification approach, Mondéjar-Guerra et al. [20] modeled the marker identification step as a multi-class classification problem by including the different marker encodings in the training and labeling process. The authors compared three different types of classifiers, a CNN, SVM, and MLP (multilayer perceptron) for identifying and decoding fiducial markers (ArUco and AprilTags) in difficult image conditions. The employed machine learning approaches showed nearly similar results and their performance was significantly better than conventional identification methods of the aforementioned marker systems. Another interesting aspect of the work presented by Mondéjar-Guerra et al. [20] is the use of synthetic data, which comprises the generation of synthetic images for training purposes. This approach can be advantageous because it avoids exhaustive image data acquisition of real scenes. In addition, it provides the possibility to control the composition of a training dataset more easily, since parameters like quantity, resolution, or influences from image acquisition, e.g. illumination, blur, noise, or perspective distortion, can be adjusted individually. These aspects are often not considered in literature. Mondéjar-Guerra et al. [20] employed synthetic training images of markers. However, the authors did not consider influences, which can occur when using a projection device such as a visible pixel grid or frayed marker contours. Moreover, they solely processed ArUco markers and AprilTags.

Most research into the use of machine learning for the detection of artificial markers concerns AR, QR code detection, or medical imaging, as such applications deal with various and complex real-world scenarios. To our knowledge, comparable approaches have not yet been published in the field of industrial close-range photogrammetry. This could be due to the sole use of physical, retroreflective markers, as established in industrial close-range photogrammetry applications [1]. Retroreflective properties allow better control of marker exposure in the image, so the non-marker background is more likely to be underexposed with the markers shown as very bright areas in the image [16]. The resulting high contrast significantly simplifies marker extraction using conventional candidate detection and verification methods. This circumstance changes when projecting markers onto sheet metal parts. As a consequence, imaged markers are significantly affected by influences such as varying contrast or overexposure, as it can be seen in Fig. 1. This makes subsequent marker detection much more difficult. Machine learning methods have the potential of adapting to those influences since the learning process is designed to find similarities and differences within provided training datasets. This promises marker detection to be more robust under a wider range of environmental conditions, as it was also suggested by authors of other works presented in this section. We therefore propose using an intelligent predetection approach based on machine learning to cope with the challenging lighting conditions inherent to projected markers.

2.3 Application of reference projections

Although the use of physical (retroreflective) markers is quite common, it is not always feasible. Using such markers for measurement applications is usually associated with a lot of manual efforts and a high amount of time [4, 38]. Hence, projecting reference markers provides an alternative to overcome these challenges. Markers are then projected onto the surface of an object by means of a projection device. As a result, it is not necessary to invest time for attaching and removing markers [4, 38]. Furthermore, it facilitates modifying the reference projections, e.g. the marker density and/or size [4]. It also prevents damaging the underlying surface, because there is no physical contact [4]. However, the intensity of projected markers changes with respect to the distance and angle from the projection device, which impacts subsequent image processing [4]. In the case of a digital projector, the effect of pixelation can also occur due to its limited resolution [4].

In the field of close-range photogrammetry and surface reconstruction, some approaches have been presented. Pappa et al. [4] projected a repetitive pattern of white circular points (dots) on large, flexible, thin-film components from the aerospace industry for static and dynamic surface measurements. Feng et al. [22] used a projected grid of white dots to reconstruct surface points of a flexible antenna. Chen et al. [39] applied a pattern consisting of circular points in conjunction with encoded markers to measure the deformation of a crane girder. To detect and verify the projected markers, the authors of the presented applications employed common detection methods, as elaborated in Sect. 2.1, albeit individual thresholds were selected by hand [4], ambient light was controlled [22], or complex routines were needed to eliminate false marker candidates [39].

Overall, based on the works presented in this section we selected a machine learning method. The Viola–Jones method [9, 10] in combination with Haar-like features suggests an easy-to-use and effective approach for our application in industrial marker detection. The binary composition of the Haar-like features promises a good detection of markers under challenging lighting conditions since it is not directly related to absolute pixel values. In addition, these features are suitable to detect edges, lines, and other simple image structures [9], which is beneficial regarding the marker shapes which we intend to project. Furthermore, Belussi and Hirata [30] demonstrated that the Viola–Jones method achieves good detection results for printed square markers in the context of finder patterns of QR codes. Therefore, we propose the application of this method for the predetection of projected reference markers. In addition, we aim at modeling various influencing factors to examine the required complexity of generated synthetic training datasets.

3 Methodology

In general, we follow the common procedure in machine learning, see Fig. 3. We acquired different datasets (Sect. 3.1) in order to train parametrized models according to the Viola–Jones method (Sects. 3.2, 3.3). The models were applied to real-world data to analyze their detection performance based on the performance indicators presented (Sect. 3.4).

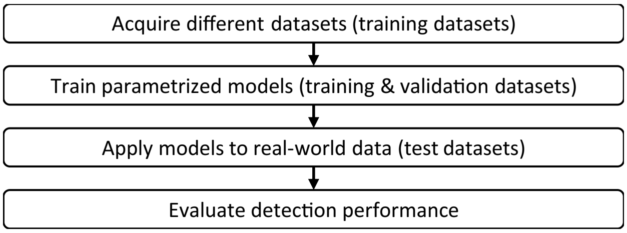

Acquire different datasets (training datasets)
Train parametrized models (training & validation datasets)
Apply models to real-world data (test datasets)
Evaluate detection performance

Fig. 3 General procedure

3.1 Dataset collection

To train a model according to the Viola–Jones method (Sect. 3.2), one positive (samples of single markers), one negative (samples without markers), and one validation dataset (images of a scene containing several markers) are required. A positive image sample depicts only a single marker, as it is shown, for example, in Fig. 4 for square markers. A negative image sample shows no markers but only background. An image of the validation dataset contains a potential scene with several markers present, for example illustrated in Fig. 5. The validation dataset is employed to assess the success during the training procedure. Furthermore, test datasets are finally used to evaluate the detection performance of the trained models. Since we examined different shapes of markers, it was necessary to obtain datasets for each marker shape. The negative dataset was independent of markers. Therefore, it was acquired once and used for all marker shapes. In the following, we describe the procedure of the dataset collection in more detail. The data acquisition parameters are explained in Sect. 3.1.2 and the labeling process of the validation and test datasets is described in Sect. 3.1.3.

3.1.1 Procedure

For the collection of positive training samples, we pursued two approaches:

First, we implemented a software pipeline that can generate large amounts of synthetic images of the projected markers based on a template which depicts the desired reference marker, see Fig. 4, since the acquisition of real-world datasets can be a tedious task. This is advantageous since the detection performance often depends on the number of available samples, cf. Sect. 3.3. It is important that environmental influencing factors, which are usually present during imaging, for example variation in illumination intensity, blur, or

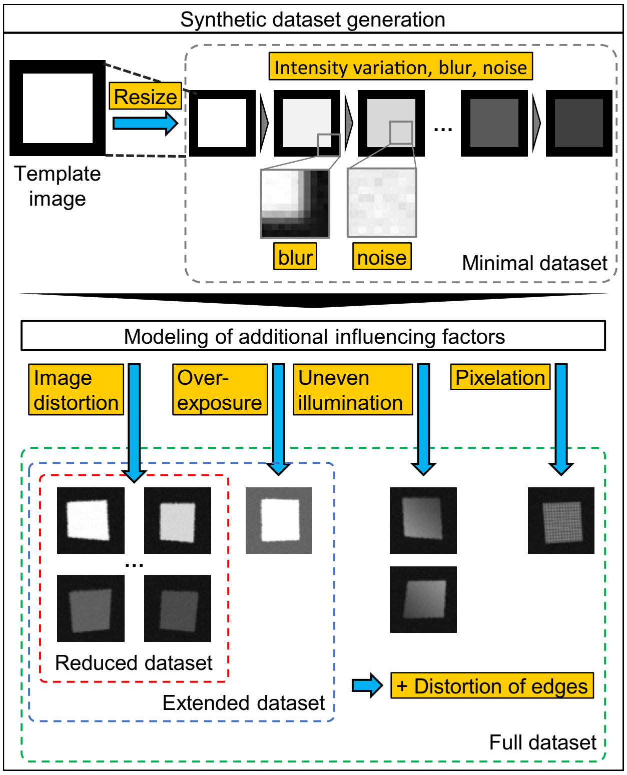


Fig. 4 Overview of the generation of different synthetic positive samples (illustrated by a square marker)

noise, are modeled to sufficiently train the classifier. In order to examine the impact of dataset complexity, we generated multiple datasets, see Fig. 4, which incorporate apparent and observable influencing factors. The "Minimal" dataset includes such conditions as (illumination) intensity, blur, and noise. The "Reduced" dataset additionally considers the effect of the perspective distortion of markers, while the "Extended" dataset also includes overexposure (increase in background intensity and blur), and the "Full" dataset incorporates all the modeled effects, including uneven illumination, pixelation, and edge distortion. Pixelation occurred in imaged markers at low exposure levels in conjunction with limited digital projection device resolution. Note that each aforementioned dataset of positive samples considers a different subset of modeled influencing factors. With regard to subsequent investigations, this does not necessarily imply that the total number of samples in a dataset automatically increases with a larger number of modeled effects. All positive training samples were generated at a pixel resolution of 100 × 100. This size approximates the dimensions of the imaged projected markers.

Second, we also produced a dataset of real-world positive training samples by taking images of projected markers to examine the impact on the detection performance compared to the synthetic ones. To obtain a large variety of training samples, we projected the desired markers onto a flat sheet metal plate in a repetitive pattern. Routines were subsequently applied to cut out individual markers from the imaged data. For a quick overview of the real-world datasets employed, please see Table 4 in the Appendix.

To collect representative negative data, we took several images of an automotive side door, see use cases Sect. 4.1, with no markers visible. Then the images were divided into multiple subimages such that a large set of potential backgrounds were available for training purposes. Black subimages were omitted.

For the validation of the models during training, we used images showing the inside of the door with projected markers (validation datasets). The image data was then manually labeled by using bounding boxes to obtain a ground truth about the present markers, see Sect. 3.1.3.

To investigate the detection performance of the trained models, test datasets are required. For this purpose, we employ several imaging devices to obtain images with region-specifically projected markers applied to the two reference sheet metal parts (see Sect. 4.1). Subsequently, labeling was conducted analogous to the validation datasets, see Sect. 3.1.3.

3.1.2 Data acquisition parameters

This section provides further details about the implementation and data acquisition parameters of the dataset collection, in particular regarding the acquisition of the real-world image data. In the Appendix, a summary is provided, see Table 4. These parameters are elaborated in the following.

As described earlier, the datasets were depending on the marker shape used, except for the negative dataset. We examined four different markers: filled square, circular, and cruciform shapes, which are referred to in this paper as square, circle, cross, as well as circular encoded markers. The last one is referred to in the following as "encoded markers". It incorporated an encoding, similar to the circular, physical fiducials commonly found in measurement applications (cf. Sect. 2).

The parameters and ranges for generating positive synthetic datasets were chosen in a way that the obtained synthetic images exhibited similarities to the real-world samples. The real-world positive samples were acquired with the ZEISS sensor (vision sensor of the employed RIS, see Sect. 4.1) and an exposure time of 250 ms at a working distance of approximately 570 mm. Further specifications about the sensor are provided in Sect. 4.1. This exposure usually provided imaged markers with an adequate illumination intensity. Additionally, the sensor was tilted relatively to the plate (0°–30°) in order to introduce perspective distortion to the markers.

The images of the negative dataset were captured using the ZEISS sensor and different exposure times (250 ms, 350 ms, 550 ms). Consequently, we obtained a large variety of potential backgrounds ranging from almost dark to overexposed regions with bright reflections (long exposure times). We applied a black background to the projection device since it also introduces a small amount of illumination to the scene. Different measurement poses were employed as compared with the image data acquired for the validation and test datasets.

The validation dataset comprised images with region-specifically projected markers which were taken with five different poses, covering most of the side of the door, with and without a tilt angle of 15° (along the short side of the FOV of the ZEISS sensor). The exposure time was set to 250 ms. As for the encoded markers, we additionally used exposure times of 100 ms and 400 ms but with no tilt angle of the sensor.

Compared with the validation datasets, a different parameterization was used for acquiring the test datasets. For instance, they differed in terms of their measurement poses, tilt angle of the sensor, and incorporated views (inside and outside of the door). Furthermore, we employed various sensors, see Sect. 4.1 for the specifications, and different sheet metal parts. More details on the test datasets are provided in Sect. 5 along with the results.

3.1.3 Labeling

The validation and test datasets were manually labeled by attaching a bounding box to each recognizable marker. This is important to determine the success of the binary classification according to the Viola–Jones method (ground truth for one marker type), i.e. whether or not a positive region proposal provided by the trained classifier actually contains a single projected marker. In addition, each label was annotated with one of the four attributes, which were introduced for a better analysis of the detection results: “Underexposed”, “Normally exposed”, “Overexposed”, or “Excluded”. If a pixel grid was visible within a projected marker, the category “Underexposed” was chosen. In contrast, “Overexposed” was selected when the adjacent background of a marker was affected by the induced illumination, i.e. a significant increase in background intensity was perceptible. Markers were excluded if they exhibited impairing modifications, for instance if they were cropped due to their position relative to the image borders or extensive superimposed reflections were present which rendered them impractical for further feature extraction. An example of labeled markers with attributes is shown in Fig. 5.

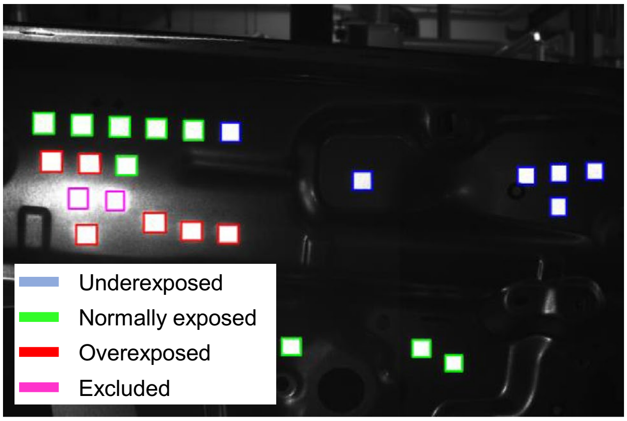


Fig. 5 An example of the labeling procedure used to obtain the ground truth of the acquired validation and test datasets

3.2 Model for the intelligent predetection

To enable intelligent predetection of the projected markers, we adapted the classifier according to [9, 10]. This object detection method was originally developed for high-performance face detection applications and makes use of so-called *Haar-like features* as elements of the classifier, see Fig. 6. These features have a rectangular shape, and their object classification values (thresholds) are obtained by training on the basis of positive and negative samples (cf. Sect. 3.1). A classifier usually contains several stages. Each stage is composed of one or multiple (trained) Haar-like features. Once the classifier has been trained, sub-windows of an image (sliding window) are given to the classifier. The features employed at each stage are applied to the sub-window and their values calculated on the basis of the corresponding intensity values in the image. These values are then compared to the trained thresholds of each feature. Only if a sub-window of an image passes through all stages of the cascade classifier, it is assumed that this sub-window shows the desired object of interest. Since the stages are arranged sequentially during the detection process, the classifier is also referred to as a “cascade classifier”.

The aforementioned features (Fig. 6) consist of rectangular forms and provide horizontal, vertical, and diagonal orientations. They operate on a sub-window of an image, such that the calculation routine applied considers the pixel values covered by the black and white rectangles of each feature. Thus, the classification process is related more closely to those features than the absolute pixel values. This supports the detection of objects under challenging lighting conditions, which was beneficial to our application. Furthermore, Viola and Jones [9] emphasize that the simple features make it possible to incorporate “*ad-hoc domain knowledge*”. This was useful, since we intended to project primitive shapes, such as a squares, circles, or crosses, as markers onto sheet

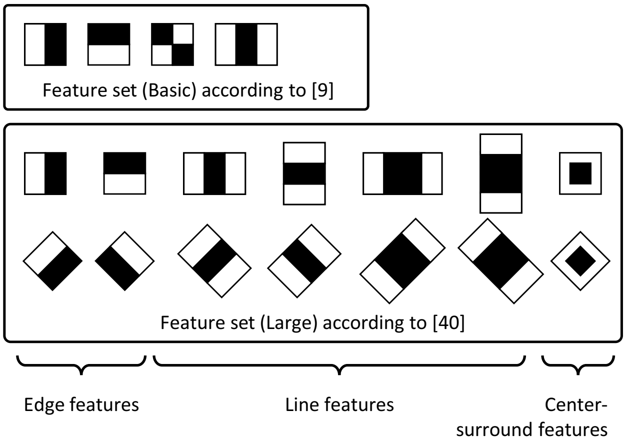

Feature set (Basic) according to [9]
Feature set (Large) according to [40]
Edge features
Line features
Center-surround features

Fig. 6 Illustration of two rectangle feature sets

metal parts. These shapes, referred to in the following as *projection primitives*, consisted of horizontal, vertical, and diagonal edges, and should therefore be well-detectable by these features. Hence, we tested two feature sets: one according to [9], referred to in the following as “Basic”, and the other one based on [40], referred to as “Large”, cf. Fig. 6.

Regarding the training process, we used the learning algorithm based on AdaBoost according to [9] to obtain the values of the threshold classification functions of each feature. This algorithm automatically chooses a suitable subset of those Haar-like features and trains the classifier by means of the provided training datasets (see Sect. 3.1). It is designed to efficiently determine the “learned” thresholds and to achieve a clear differentiation between the positive and negative training samples. For more details on the learning algorithm and the boosting routine in the context of the Viola–Jones method, the reader is referred to [9, 41].

3.3 Classifier training

The parameterization of the training also impacts the success of the detection. Based on our software implementation (OpenCV, cf. Sect. 4.2) and in line with [30], the following parameters were considered relevant to the training process and, thus, investigated more closely (parameter tuning): selected (Haar-like) feature set (cf. Fig. 6), the overall false alarm rate (OFAR), the total number and ratio (positive to negative data) of the samples, and the scaled size of the samples. The OFAR describes an exponential relation between the maximum false alarm rate per stage (stage max. FAR) and the number of stages, cf. Sect. 3.2. The stage max. FAR is a criterion of the permitted misclassification per stage during the training process. For other (not examined) parameters, such as the minimal hitrate or maximal depth, we used the default values suggested by the OpenCV implementation, see Sect. 4.2.

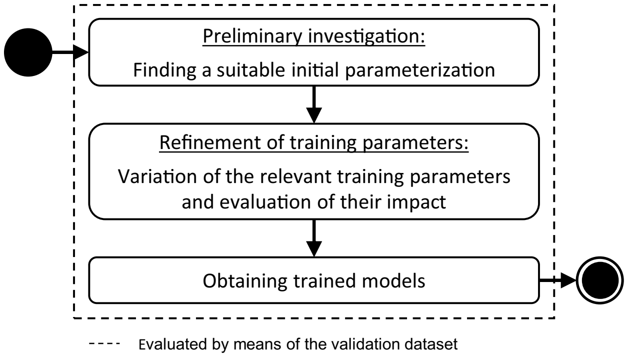

Preliminary investigation:
Finding a suitable initial parameterization
Refinement of training parameters:
Variation of the relevant training parameters and evaluation of their impact
Obtaining trained models
Evaluated by means of the validation dataset

Fig. 7 Training procedure for the classifier

Since no rigid process for classifier training is generally available, our training was based on the procedure shown in Fig. 7. First, we conducted preliminary investigations to determine a suitable initial parameterization. Next, these parameters were successively refined in the subsequent step by varying the training parameters, as discussed in the previous paragraph. Finally, the trained models which showed the best detection performance were selected. The success of the training was determined by means of the performance indicators *recall* and *precision*, cf. Sect. 3.4, based on the validation datasets acquired for training purposes.

As part of the preliminary investigations, it was first necessary to conduct some explorative studies to obtain initial meaningful training results. These mainly focused on the amount of positive and negative training samples required for the different datasets, see Sect. 3.1. The parameter ratio of a primitive relatively to its background was also considered. A recall, see Sect. 3.4, of about 80% indicated a good level of detection performance. For this value, we observed that most of the recognizable markers, i. e. with no significant under- or overexposure, were detected. Eventually, we obtained the following initial parameterization, which was then used in all subsequent trainings: feature set (Large), stage max. FAR (0.5), number of stages (17), primitive-to-background-ratio of the positive sample (2:1), ratio between the number of positive and negative samples (1:2), number of training samples (depending on the dataset employed), and sample size (24 × 24 px).

In the next step, the training process was refined. We varied the parameters on the basis of the training steps introduced by [30]. The order in which the training parameters were varied was as follows: feature set (Basic, Large), OFAR (0.4^{16}, 0.5^{17}, 0.6^{19}), ratio of positive and negative samples (2:1, 2:2, 1:1, 1:2 for the “Reduced”, “Extended”, and “Full” datasets; 1:0.5, 1:1, 1:2, 1:4 for the “Minimal” and “Real samples” datasets), number of training samples, and their size. An overview is provided in Table 1.

Table 1 Overview of the relevant training parameters in the refinement step

	Training parameters				
Dataset	Feature set	OFAR	Ratio # pos : # neg	# pos, # neg	Size in px
"Minimal","Real samples"	Basic, Large	0.4^{16}, 0.5^{17}, 0.6^{19}	1:0.5, 1:1, 1:2, 1:4	Individually	12×12, 24×24, 32×32
"Reduced", "Extended", "Full"	Basic, Large	0.4^{16}, 0.5^{17}, 0.6^{19}	2:1, 2:2, 1:1, 1:2	Individually	12×12, 24×24, 32×32

Based on the determined initial parameterization, we tested the first three steps thoroughly because they seemed to be the ones with the largest impact on the detection performance. The number of training samples depended on the employed datasets. The sample size was examined for the "Full" dataset, but showed little effect. Hence, it was kept at 24 × 24 px for all datasets. For the interested reader, we provide a flowchart of the parameter variation of the "Full" dataset (square primitive) in the Appendix (see Fig. 12). We usually considered the recall as the criterion of improvement. When it increased, with no significant drop in the precision, did we move on to the next variation step. In the case that the recall decreased only slightly but the precision improved significantly, we adapted this parameterization.

Finally, we obtained suitable training parameters for the introduced datasets, as presented in Table 2. For real samples the Basic feature set showed the best performance, whereas synthetic datasets usually exhibited good training results with the Large feature set, except for the "Minimal" dataset with the primitive cross. The resulting trained models (classifiers) were employed for the experiments in Sect. 5.

3.4 Performance indicators

The Viola–Jones method provides positive region proposals (suggesting the presence of markers), which can be handled as a binary classification problem. With regard to the (labeled) ground truth data, a proposal window was classified as a true positive (TP) only if it contained a complete single projected marker. If the proposed region comprised no marker, more than one marker, or just a fraction of a marker, it was counted as a false positive (FP). False negatives (FN) consisted of markers that were labeled but not detected. To enable an intuitive analysis of the results, we employed simple metrics such as *recall*, *precision*, and F_1 *score* for evaluation purposes, similarly as presented in [30].

The recall metric quantifies the percentage of all true positive proposals in relation to all positively labeled markers and can be defined as:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

The precision metric indicates the ratio of correct proposals to the total proposals made:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

The F_1 score represents the weighted average of recall and precision and can be obtained by

$$\text{F}_1\text{score} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}. \tag{3}$$

Table 2 Parameterization of the trained models

	Feature set	Stage max. FAR	# pos	# neg	Size in px
Dataset square					
"Minimal"	Large	0.5	40	80	24 × 24
"Reduced"	Large	0.4	160	320	24 × 24
"Extended"	Large	0.4	120	240	24 × 24
"Full"	Large	0.5	1000	1000	24 × 24
"Real samples"	Basic	0.4	380	760	24 × 24
Dataset circle					
"Minimal"	Large	0.5	40	160	24 × 24
"Reduced"	Large	0.4	320	320	24 × 24
"Extended"	Large	0.5	240	480	24 × 24
"Full"	Large	0.4	500	500	24 × 24
"Real samples"	Basic	0.4	370	370	24 × 24
Dataset cross					
"Minimal"	Basic	0.5	40	160	24 × 24
"Reduced"	Large	0.5	500	1000	24 × 24
"Extended"	Large	0.5	750	1500	24 × 24
"Full"	Large	0.5	500	1000	24 × 24
"Real samples"	Basic	0.4	370	370	24 × 24
Dataset encoded markers					
"Reduced"	Large	0.4	500	1000	24 × 24
"Full"	Large	0.4	1000	1000	24 × 24

4 Experimental setup

4.1 Hardware

All measurements were taken in the working space of the ZEISS *AIBox*. The measurement setup comprised a Fanuc M-20iA industrial robot, which functioned as a flexible manipulator for the ZEISS COMET Pro AE optical 3D sensor. The sensor provided a resolution of 4896 × 3264 pixels with an FOV of approximately 600 × 450 mm^2 at a working distance of 570 mm. Furthermore, a Canon EOS 760D DSLR camera with a full resolution of 6000 × 4000 pixels and a Xiaomi Mi A2 smartphone camera with 4000 × 3000 pixels were used in the measurement investigations.

For the projection of reference markers, a Sony VPL-PHZ10 LCD projector served as a front projection device for displaying images with a resolution of 1920 × 1200 pixels, with a maximum light output of 5000 lm and a contrast ratio of 500,000:1. Blue markers were applied when using the ZEISS sensor due to a bandpass filter. Otherwise, we employed white projections.

As use cases we employed two sheet metal parts from the automotive industry: a side door and B-pillar, see Fig. 8. The surface of the parts was untreated, i.e. sprayed coating, which are often used to improve reflection properties, was not applied.

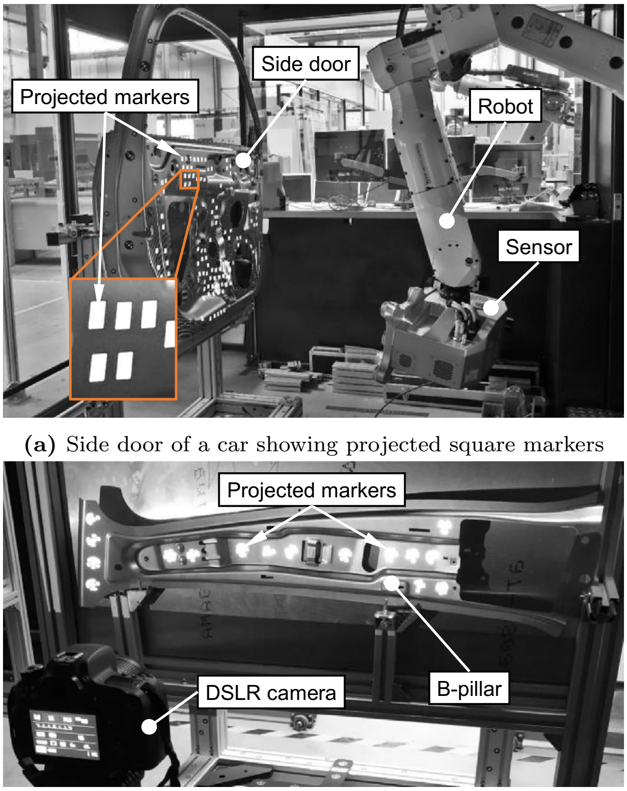
4.2 Software

The software was developed in C++. To enable the implementation of image processing tasks, such as image data handling, generating synthetic samples, or applying the trained classifiers to images, we also used the open source framework OpenCV[1]. All acquired images were processed as gray-scale images. In addition, we employed utilities provided by OpenCV, such as the tool for sample creation and cascade training. To label the acquired image datasets, the available annotation tool was modified to include multiple illumination categories.

5 Results and discussion

5.1 Results

To evaluate the performance of trained classifiers for the predetection of projected reference markers under real-world conditions (working space in the ZEISS AIBox), we used the two reference use cases introduced above. The projection primitives were applied region-specifically meaning that they were displayed on plane homogeneous regions on the surface of the parts. For the side door, we employed the approach proposed in [42] regarding spatial interactive projections. This comprised a calibration procedure and modeling the projection device for the calculation of projection images with the corresponding reference markers. On the B-pillar, the primitives were positioned manually with visual selection of the plane regions.



(a) Side door of a car showing projected square markers

(b) B-pillar showing projected circular encoded markers

Fig. 8 Technical setup with sheet metal parts from the automotive industry (projection device is not shown). The smartphone was mounted on the tripod instead of the DSLR camera by means of a cell phone holder (not shown)

As for the test datasets of the door, images of the projected primitives (square, circle, cross) were acquired separately using the ZEISS sensor. The procedure was kept the same. We used a variety of measurement poses (seven poses covering different views on the inside and outside of the door) and applied several exposure times (100 ms, 250 ms, 400 ms), cf. Table 4 in the Appendix. Additionally, a tilt angle of ten degrees was introduced around the short side of the FOV. Each view contained several primitives for the detection process. The size of the displayed primitives was kept at a constant side length (for square or cross) or diagonal (for circle) of 26 projector pixels. Overall, 2340 labels were drawn for each projected primitive (square, circle, cross) as ground truth and used in its entirety for the

[1] https://opencv.org.

Table 3 Detection results of markers projected onto the side door of a car

	Recall	Precision	F_1 score
Model of dataset square			
"Minimal"	0.9500	0.8338	0.8881
"Reduced"	0.9496	0.9586	0.9541
"Extended"	0.9748	0.9698	0.9723
"Full"	0.9949	0.9058	0.9483
"Real samples"	0.9739	0.9167	0.9444
Model of dataset circle			
"Minimal"	0.9697	0.8905	0.9284
"Reduced"	0.9692	0.9570	0.9631
"Extended"	0.9923	0.9595	0.9756
"Full"	0.9842	0.9685	0.9763
"Real samples"	0.9868	0.9440	0.9649
Model of dataset cross			
"Minimal"	0.9333	0.5796	0.7151
"Reduced"	0.9932	0.9839	0.9885
"Extended"	0.9885	0.9889	0.9887
"Full"	0.9962	0.9881	0.9921
"Real samples"	0.9868	0.9360	0.9607

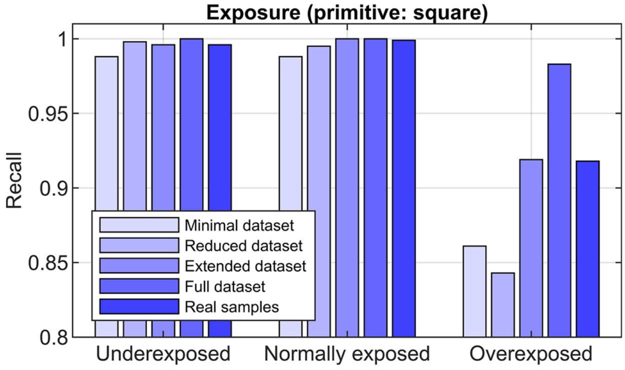


Fig. 9 Detection results of projected square markers for different exposure levels

evaluation of the detection performance. The trained models from the previous section were applied to detect these projected markers in the acquired and labeled images.

Table 3 shows the detection performance results, i.e. the performance indicators (introduced in Sect. 3.4), of the primitives investigated for the different trained models. For all primitives and datasets, a recall of over 93% was achieved. We therefore conclude that the chosen predetection approach is well suited to the detection of projected reference markers in the form of simple shapes. It also demonstrates that real-world positive samples were not needed to achieve a high detection performance. With the exception of the synthetic samples in the "Minimal" dataset, a precision above 90% and an F_1 score over 94% were achieved. In comparison with the "Reduced" dataset, it is apparent that including the effect of marker distortion in the generation of positive training samples contributed to the improved precision rates of 90% and more, and resulted in an effective detection of the primitives.

To analyze the impact of different lighting conditions on the detection process, we introduced three attributes of displayed markers, see Sect. 3.1.3. Figure 9 shows the recall rates for the exposure levels of "Underexposed", "Normally exposed", and "Overexposed" for square primitives of the test dataset. It seems to be more challenging to deal with overexposed markers than with underexposed ones, especially with the synthetic datasets, for which the effect of overexposure was not considered. A similar trend was also observed for the primitive circle, see Appendix Fig. 13. The primitive cross also missed markers in the "Overexposed" class. However, it should be noted that the "Minimal" trained model used the basic feature set (Fig. 14). Regarding the intelligent predetection of markers, we assume that avoiding overexposed markers during image data acquisition could improve the detection results in general.

Similar to the procedure described for the test dataset of the side door, we also employed the ZEISS sensor to acquired validation images of another sheet metal part, a B-pillar of a car. The detection performance results for the primitives square, circle, and cross are given in the Appendix, see Table 5. Comparable values of the recall and precision were achieved as for the door. This demonstrates that the trained models also work on other parts.

Since we observed that modeling the effect of marker distortion affected precision (cf. Table 3 and 5), we also determined the admissible tilt angle for our imaging device. For this purpose, the sensor was tilted by up to 30° towards a sheet metal plate. Higher angles were not considered feasible in practical applications. Zero degrees corresponded to the pose in which the optical axis of the ZEISS sensor was perpendicular to the plate. The angle was then increased in increments of 10°. For square projection primitives, the results show (see Appendix Fig. 15) that despite image distortion and distorted squares, respectively, the markers were well detected by the trained models. However, it is worth mentioning that distortions and illumination conditions occurring with real parts might be more complex as the underlying surface is usually not planar in a mathematical sense as it is for a plate.

In addition to the aforementioned projection primitives (square, circle, cross), we also tested the detection performance of projected markers that incorporated an encoding. Training was as described in Sect. 3. The test dataset of the encoded markers was acquired in a similar manner to that described above for projection primitives. Figure 8b shows an exemplaric illustration of projected encoded markers.

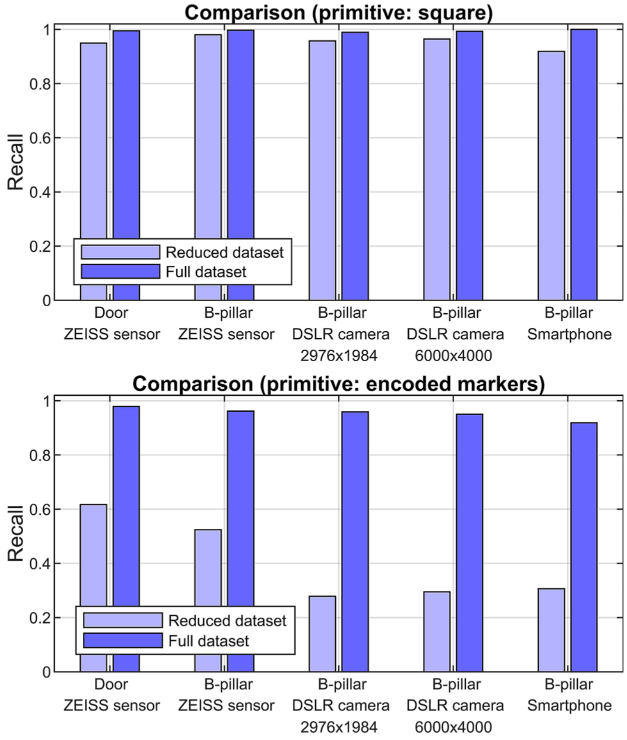


Fig. 10 Detection results of projected square and encoded markers obtained with different imaging devices

We also employed additional imaging devices as well as a different part to demonstrate the transferability. To enable an intuitive comparability between the detection results of the parts, simple and more complex marker shapes, as well as the various cameras, we used the performance indicator recall. The results are presented in Fig. 10. The “Full” dataset achieved recall rates over 91% for both marker types (square and encoded markers). This shows that the proposed approach is capable of detecting more complex shapes based on synthetic training datasets and that it is transferable to other setups. For encoded markers, it appears that the “Reduced” dataset is not sufficient and reveals the need for additional modeled effects in the training samples. The classifier based on the “Full” dataset on the other hand demonstrates high detection rates. This might be useful for other applications, for instance the detection of conventional, physically attached fiducials under challenging lighting conditions.

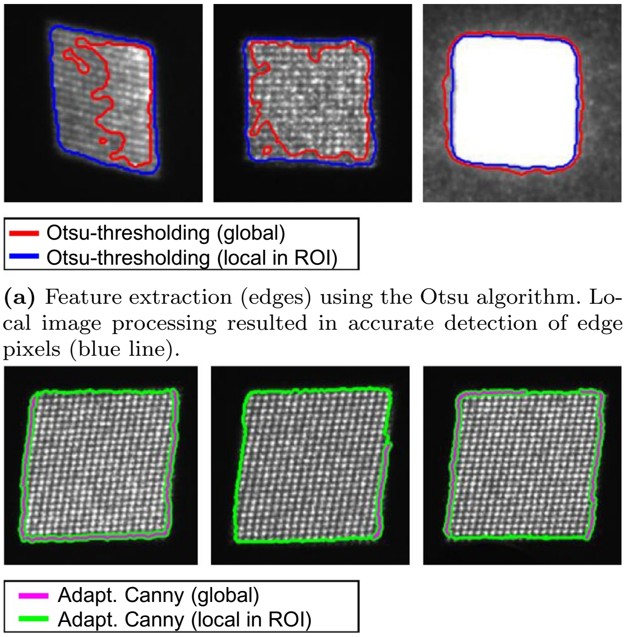
5.2 Discussion

The results presented in this section demonstrate that cascade classifiers trained with real-world or synthetic datasets were able to cope with both varying lighting conditions and marker distortion in the course of the detection process. Note that although they displayed almost invariant behavior to the variation in illumination intensity in our investigations, the classifier is not invariant to rotations. Circular markers like the circle primitives or encoded markers should be less affected by this issue. For squares on the other hand, this circumstance needs to be considered, for example, by incorporating the pose information of the sensor into the display of the projected markers or by adding rotated samples to the training pipeline. The latter was not considered for this paper and, thus, requires further examination. Overall, the proposed approach showed a very high detection performance for all tested markers, although it did not reach a precision value of 1. This means that other regions of an image were occasionally misclassified as valid markers as they exhibited local properties indistinguishable from true markers for the trained classifier. Consequently, post-processing routines of the obtained ROIs might still be necessary to eliminate false positives. Nevertheless, the intelligent predetection step introduced here promises a more robust marker detection under challenging lighting conditions and enables subsequent image processing techniques to be tailored more easily to the local ROIs.

We provide selected examples in Fig. 11 to demonstrate the impact and benefit of intelligent predetection for subsequent local image processing and, thus, for feature extraction under challenging lighting conditions. We used two established methods (the Otsu algorithm and an adaptive Canny algorithm) to detect the contours (edges) of projected square markers. While the Otsu algorithm applied to the ROI (local image processing) is able to detect clear contours for the depicted squares, its global parameterization fails to detect the four edges or exhibits a visible offset. As for the adaptive Canny algorithm, the locally applied routine appears to detect all four edges of the squares, while the global parameterization does not. To complement this qualitative comparison, we plan to provide quantitative measures on this matter in future work, for example using intersection over union (IoU).

6 Conclusion

In the context of geometric quality assurance of sheet metal parts, robot-based (optical) inspection systems have shown high potential for coping with current challenges in manufacturing metrology. To further improve RIS, we previously proposed a concept based on region-specific projections for the alignment of different views of a sensor. This eliminates the need for physically attached fiducials, but requires an image processing method for robustly detecting projected markers in challenging lighting conditions. Hence, in this paper, we introduced trained cascade classifiers with Haar-like features based on the Viola–Jones method as a


Otsu-thresholding (global)
Otsu-thresholding (local in ROI)
Adapt. Canny (global)
Adapt. Canny (local in ROI)

(a) Feature extraction (edges) using the Otsu algorithm. Local image processing resulted in accurate detection of edge pixels (blue line).

(b) Feature extraction (edges) using the Canny algorithm. Local image processing resulted in detection of edge pixels on each side of the projected square (green line) unlike the open contours obtained by the global Canny.

Fig. 11 Impact of predetection and local image processing on feature extraction (edges), as compared to global processing (illustrated are selected examples)

predetection step for an improved image processing in measurement applications.

The results showed that properly trained models based on *both* positive synthetic *and* positive real image data achieved a high recall of above 90% for shapes like squares, circles, or crosses. Besides intensity variation, blur, and noise, modeling the effect of marker distortion on synthetic datasets improved the precision to above 90%. With projected encoded markers (similar to circular fiducials), we observed that incorporating influencing factors, such as overexposure and uneven illumination, into the generation of synthetic training data significantly benefited detection performance. We also demonstrated that the approach presented in this paper was transferable to other sheet metal parts and imaging devices.

Since our results indicate that overexposure of markers tends to affect the detection performance, we suggest analyzing the impact of overexposed samples in future studies more closely. We are also interested in investigating the detection of more complex shapes on the basis of synthetic positive datasets, not only to further explore the limitations of the proposed approach but also to foster ideas for novel marker designs that are adapted to the use of digital projection devices. In general, we intend to identify new use cases for the proposed approach outside the field of manufacturing metrology since we believe that it could benefit many applications in production engineering across all industries.

Appendix

See Tables 4, 5 and Figs. 12, 13, 14, 15.

Table 4 Overview of the acquisition parameters of real-world datasets employed

Dataset	Projection primitive	Vision sensor	Projection surface	# measurement poses	Exposure time in ms	Sensor tilt angle in deg
Real positive samples	Square, circle, cross	ZEISS sensor	Sheet metal plate	4	250	0, 10, 20, 30
Negative samples	none	ZEISS sensor	Side door	5	250, 350, 550	0
Validation	Square, circle, cross	ZEISS sensor	Side door	10	250	0, 15
Validation	Encoded markers	ZEISS sensor	Side door	5	100, 250, 400	0
Test	Square, circle, cross,encoded markers	ZEISS sensor	Side door	14	100, 250, 400	0, 10
Test	Square, circle, cross,encoded markers	ZEISS sensor	B-pillar	4	100, 250, 400	0, 10
Test	Square, circle, cross,encoded markers	DSLR camera	B-pillar	4	4, 20, 100	0, 20
Test	Square, circle, cross,encoded markers	Smartphone camera	B-pillar	4	4, 20, 100	0, 20

Table 5 Detection results of markers projected onto the B-pillar

	Recall	Precision	F_1 score
Model of dataset square			
"Minimal"	0.9243	0.8959	0.9099
"Reduced"	0.9806	0.9786	0.9796
"Extended"	0.9949	0.9868	0.9908
"Full"	0.9969	0.9503	0.9730
"Real samples"	0.9806	0.9697	0.9751
Model of dataset circle			
"Minimal"	0.9918	0.9390	0.9647
"Reduced"	0.9888	0.9827	0.9857
"Extended"	0.9918	0.9908	0.9913
"Full"	0.9918	0.9939	0.9928
"Real samples"	0.9908	0.9690	0.9798
Model of dataset cross			
"Minimal"	0.9601	0.7039	0.8123
"Reduced"	0.9918	0.9969	0.9943
"Extended"	0.9898	0.9979	0.9938
"Full"	0.9928	0.9959	0.9943
"Real samples"	0.9796	0.9383	0.9585

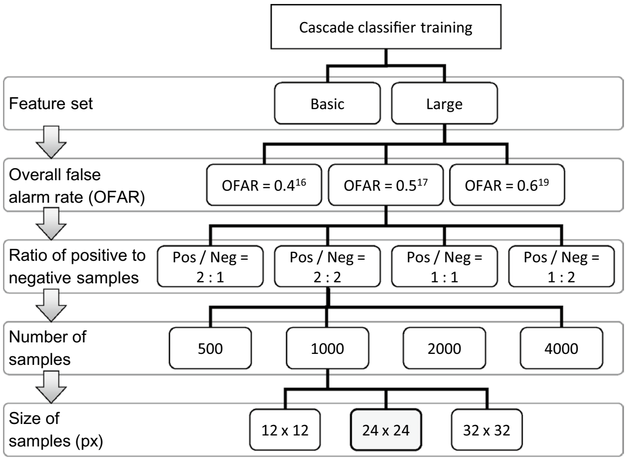


Fig. 12 Example flowchart of parameter variation during training ("Full" dataset, square primitive)

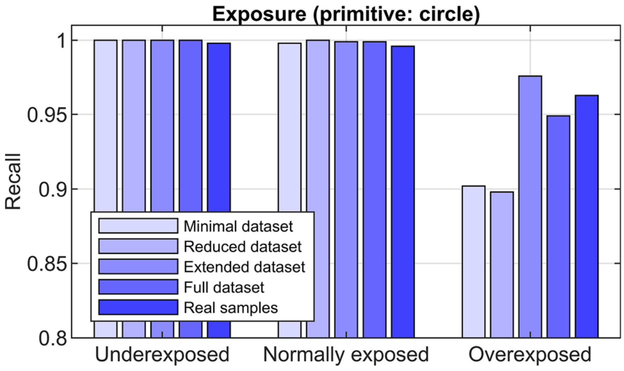


Fig. 13 Detection results of projected circular markers for different exposure levels (automotive side door)

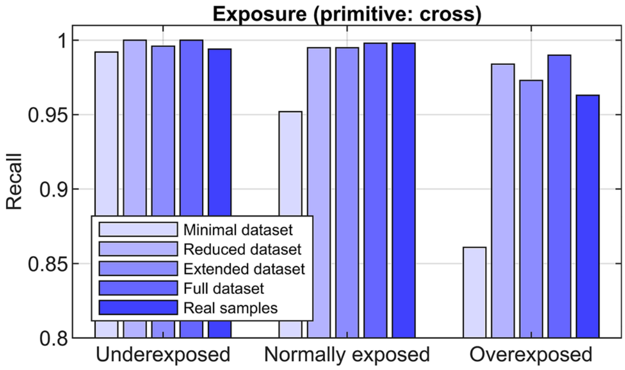


Fig. 14 Detection results of projected markers with a cross as the projection primitive for different exposure levels (automotive side door)

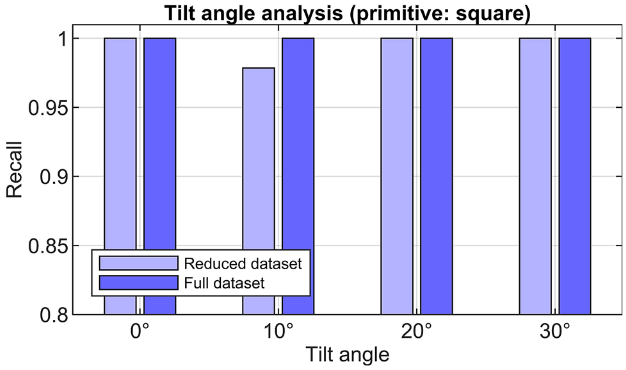


Fig. 15 Detection results of projected square markers on a sheet metal plate with different tilt angles of the sensor

Acknowledgements The presented results were part of the research project "CyMePro" (Cyber-physical measurement technology for 3D digitization in the networked production), which was funded by the Bavarian Ministry of Economic Affairs, Energy and Technology (funding code ESB036/001). We also thank our research partners AUDI AG and Carl Zeiss Optotechnik GmbH for their support.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Luhmann T (2010) Close range photogrammetry for industrial applications. ISPRS J Photogramm Remote Sens 65:558–569. https://doi.org/10.1016/j.isprsjprs.2010.06.003
2. Barone S, Paoli A, Razionale AV (2012) Three-dimensional point cloud alignment detecting fiducial markers by structured light stereo imaging. Mach Vis Appl 23:217–229. https://doi.org/10.1007/s00138-011-0340-1
3. Bauer P, Magaña Flores A, Reinhart G (2019) Free-form surface analysis and linking strategies for high registration accuracy in quality assurance applications. Proced CIRP 81:968–973. https://doi.org/10.1016/j.procir.2019.03.236
4. Pappa RS, Black JT, Blandino JR, Jones TW, Danehy PM, Dorrington AA (2003) Dot-projection photogrammetry and videogrammetry of gossamer space structures. J Spacecr Rockets 40(6):858–867. https://doi.org/10.2514/2.7047
5. Claus D, Fitzgibbon AW (2004) Reliable fiducial detection in natural scenes. Eur Conf Comput Vis 3024:469–480. https://doi.org/10.1007/978-3-540-24673-2_38
6. Jones TW, Pappa RS (2002) Dot projection photogrammetric technique for shape measurements of aerospace test articles, p 532. https://doi.org/10.2514/6.2002-532
7. Fiala M (2010) Designing highly reliable fiducial markers. IEEE Trans Pattern Anal Mach Intell 32:1317–1324. https://doi.org/10.1109/TPAMI.2009.146
8. Köhler J, Pagani A, Stricker D (2011) Detection and identification techniques for markers used in computer vision. Schloss Dagstuhl-Leibniz-Zentrum für Informatik GmbH, Wadern/Saarbrücken, Germany. https://doi.org/10.4230/OASIcs.VLUDS.2010.36
9. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features, pp 511–518. https://doi.org/10.1109/CVPR.2001.990517
10. Viola P, Jones M (2004) Robust real-time face detection. Int J Comput Vis 57:137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb
11. Wagner D, Schmalstieg D (2007) Artoolkitplus for pose tracking on mobile devices
12. Fiala M (2005) Artag, a fiducial marker system using digital techniques. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp 590–596
13. Kato H, Billinghurst M (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings 2nd IEEE and ACM international workshop on augmented reality (IWAR'99), pp 85–94. https://doi.org/10.1109/IWAR.1999.803809
14. Cronk S, Fraser CS (2006) Hybrid measurement scenarios in automated close-range photogrammetry. Int Arch Photogramm Remote Sens Spatial Inf Sci Pt:XXXVII B3b
15. Shortis MR, Seager JW (2014) A practical target recognition system for close range photogrammetry. Photogram Rec 29:337–355. https://doi.org/10.1111/phor.12070
16. Burgess G, Shortis MR, Scott P (2011) Photographic assessment of retroreflective film properties. ISPRS J Photogramm Remote Sens 66:743–750. https://doi.org/10.1016/j.isprsjprs.2011.07.002
17. Andziulis A, Drungilas D, Glazko V, Kiseliovas E (2015) Resource saving approach of visual tracking fiducial marker recognition for unmanned aerial vehicle. Adv Electr Electron Eng. https://doi.org/10.15598/aeee.v13i4.1492
18. Wijenayake U, Choi SI, Park SY (2014) Automatic detection and decoding of photogrammetric coded targets
19. Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, Marín-Jiménez MJ (2014) Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn 47:2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005
20. Mondéjar-Guerra V, Garrido-Jurado S, Muñoz-Salinas R, Marín-Jiménez MJ, Medina-Carnicer R (2018) Robust identification of fiducial markers in challenging conditions. Expert Syst Appl 93:336–345. https://doi.org/10.1016/j.eswa.2017.10.032
21. Naimark L, Foxlin E (2002) Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. IEEE Int Symp Mixed Augmented Reality. https://doi.org/10.1109/ISMAR.2002.1115065
22. Feng T, Bingguo L, Fengdong C, Guodong L (2012) Dots-array photogrammetry of flexible antenna surfaces. Int Conf Image Anal Signal Process 2012:1–5. https://doi.org/10.1109/IASP.2012.6425067
23. DeGol J, Bretl T, Hoiem D (2017) Chromatag: a colored marker and fast detection algorithm. IEEE Int Conf Comput Vis 2017:1481–1490. https://doi.org/10.1109/ICCV.2017.164
24. Valença J, Dias-da Costa D, Júlio E, Araújo H, Costa H (2013) Automatic crack monitoring using photogrammetry and image processing. Measurement 46:433–441. https://doi.org/10.1016/j.measurement.2012.07.019
25. Fraser CS, Cronk S (2009) A hybrid measurement approach for close-range photogrammetry. ISPRS J Photogramm Remote Sens 64:328–333. https://doi.org/10.1016/j.isprsjprs.2008.09.009
26. Dosil R, Pardo XM, Fdez-Vidal XR, García-Díaz A, Leborán V (2013) A new radial symmetry measure applied to photogrammetry. Pattern Anal Appl 16:637–646. https://doi.org/10.1007/s10044-012-0281-y
27. Niederöst M, Maas HG (1997). Entwurf und Erkennung von codierten Zielmarken. 16. Oldenburg, Germany. https://doi.org/10.3929/ethz-a-004332841
28. Ahn S J (1997) Kreisförmige Zielmarke (circular target). Proc. 4. ABW Workshop Optische 3D-Formerfassung
29. Sattar J, Bourque E, Giguere P, Dudek G (2007) Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction. In: Fourth Canadian conference on computer and robot vision (CRV '07), 165–174. https://doi.org/10.1109/CRV.2007.34
30. Belussi L, Hirata N (2011) Fast QR code detection in arbitrarily acquired images. https://doi.org/10.1109/SIBGRAPI.2011.16
31. Cheng G, Han J (2016) A survey on object detection in optical remote sensing images. ISPRS J Photogramm Remote Sens 117:11–28. https://doi.org/10.1016/j.isprsjprs.2016.03.014
32. Chou TH, Ho CS, Kuo YF (2015) QR code detection using convolutional neural networks. Int Conf Adv Robot Intell Syst. https://doi.org/10.1109/ARIS.2015.7158354
33. Claus D, Fitzgibbon AW (2005) Reliable automatic calibration of a marker-based position tracking system. In: Seventh IEEE workshops on applications of computer vision 1, pp 300–305. https://doi.org/10.1109/ACVMOT.2005.101
34. Yuan B, Li Y, Jiang F, Xu X, Zhao J, Zhang D, Guo J, Wang Y, Zhang S (2019) Fast QR code detection based on bing and adaboost-svm. https://doi.org/10.1109/HPSR.2019.8808000
35. Zitnick CL, Dollár P (2014) Edge boxes: locating object proposals from edges. Eur Conf Comput Vis 8693:391–405
36. Ahmadvand P, Ebrahimpour R, Ahmadvand P (2016) How popular CNNs perform in real applications of face recognition. https://doi.org/10.1109/TELFOR.2016.7818876
37. Jiang Y, Wang H, Liu H (2017) A robust fiducial mark extraction method in X-ray image based on HOG operator. https://doi.org/10.1109/IAEAC.2017.8054084
38. Ürün M, Wiggenhagen M, Nitschke H, Heipke C (2017) Stabilitätsprüfung projizierter Referenzpunkte für die Erfassung großvolumiger Messobjekte

39. Chen ZP, Lin XX, Ling X, Wang Y, Huang CL, Li ZW, Sun ZJ (2020) Design and detection algorithm of white-light markers in close-range potogrammetry. In: Duan B, Umeda K, Hwang W (eds) Proceedings of the Seventh Asia international symposium on mechatronics lecture notes in electrical engineering, vol 589. Springer, Singapore, pp 654–662. https://doi.org/10.1007/978-981-32-9441-7_68
40. Lienhart R, Maydt J (2002) An extended set of Haar-like features for rapid object detection. https://doi.org/10.1109/ICIP.2002.1038171
41. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139. https://doi.org/10.1006/jcss.1997.1504
42. Bauer P, Fink F, Magaña A, Reinhart G (2020) Spatial interactive projections in robot-based inspection systems. Int J Adv Manuf Technol 107(5):2889–2900. https://doi.org/10.1007/s00170-020-05220-1

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.