Hybrid SURF-Golay Marker Detection Method for Augmented Reality Applications

Article *in* Journal of WSCG · January 2012

5 authors, including:

David Procházka
Mendel University in Brno

Ondřej Popelka
Mendel University in Brno

Tomas Koubek
Mendel University in Brno

Jaromir Landa
Mendel University in Brno

Hybrid SURF-Golay Marker Detection Method for Augmented Reality Applications

David Prochazka, Ondrej Popelka, Tomas Koubek, Jaromir Landa, Jan Kolomaznik
Mendel University in Brno
Zemedelska 1
61300, Brno, Czech Republic
david.prochazka@mendelu.cz

ABSTRACT

Augmented reality is a visualization technique widely used in many applications including different design tools. These tools are frequently based on tracking artificial objects such as square markers. The markers allow users to add a 3D model into the scene and adjust its position and orientation. Nevertheless, there are significant problems with marker occlusions caused by users or objects within the scene. The occlusions usually cause a disappearance of the 3D model. Such behavior has substantial negative impact on the application usability. In this article we present a hybrid marker detection approach. With this approach, markers are detected using the well-known SURF method. This method is able to recognize complex natural objects and deal with partial occlusions. Further, we overcome the problem of distinguishing similar markers by using the Golay error correction code patterns. The described approach represents a robust method that is able to identify even significantly occluded markers, differentiate similar markers, and it works in a constant time regardless of the amount of used markers.

Keywords

Augmented reality, augmented prototyping, SURF, Golay error correction code.

1. INTRODUCTION

The augmented reality (AR) research has been running for almost two decades. Nevertheless, it is possible to find just a few applications for common users. There are several principal reasons. One of the key problems is the inability to deal with occlusions of markers that are used for scene augmentation. During the work with an AR application, a marker is frequently obstructed by different solid objects, e.g. **users´** hands. Inability to identify such **a** partially occluded marker leads to frequent disappearances of a visualized 3D model. Despite the obvious importance, this problem is unsolved even in many well-known AR toolkits (e.g. *ARToolKitPlus*).

The presented approach is implemented in the AR application *AuRel* that is focused on an augmented prototyping process**.** The application is developed in cooperation with an automotive company. It allows a car designer to extend a physical car model by selected virtual objects (3D models of car spare parts (see Fig. 1)). The usage of AR for industrial design is mentioned in many papers, e.g. in [FAM*02], [BKF*00] and [VSP*03].

Figure 1: 3D model of a spoiler inserted onto a rear car hood

Although there is a substantial amount of existing augmented reality frameworks (*ARToolkit*, *ARTag*, *Studierstube*, etc.), *OpenCV* library has been used for the implementation [Lag11]. Principal reasons being active *OpenCV* development, cross-platform deployment, 64bit systems support, a wide range of implemented computer vision algorithms and the amount of documentation (books, tutorials, etc.) [PK11].

There are briefly summarized current methods used for recognition of possible markers in the section 2. Two approaches focused on identification of geometric features are compared with the advanced technique usually used for natural object detection. Further, the section 3 outlines our method that is composed of SURF marker detection and Golay error correction code identification. Finally, the section 4 presents our results and concentrates on the ability to deal with occlusions.

2. MARKER RECOGNITION METHODS

The process of marker recognition is usually divided in two parts: *marker detection* and *marker identification*. The former involves recognition of video frame regions that may represent markers. The latter concentrates on verifying the identity of the markers. The marker identity defines which 3D model will be displayed to the user.

2.1 Marker Detection Approaches

The registration process of all further described methods is influenced by many negative factors, e.g. low image resolution, camera distortion (caused by lens), various light conditions or marker occlusions. The methods endeavor to compensate most of these factors. For the purpose of the article, the methods are distinguished into three general groups according to their basic principles**.** In detail the description of object recognition methods can be found e.g. in [Sze11].

2.1.1 Morphology-based marker detection

These methods are based on recognition of shapes in preprocessed images. An approach described in [HNL96] uses a system of *concentric contrast circles* (CCC). The marker is composed of a black circle around a white middle or vice versa. The detection process starts with image thresholding and noise removal. Further, connected components are found, and their centers are determined. The results are two sets of centers: centers of white connected components and centers of black connected components. CCC marker position is given by the cross section of black and white centers**.**

An example of another approach is implemented in the frequently used *ARToolKit* [KB99]. In this case, square markers with black borders and black-and-white inner pictures are detected. A camera image is thresholded and connected components contours are found. Further, quadrangles are selected from the contours set. These quadrangles represent potential markers [KTB*03].

The obvious limitation of these methods is their inability to deal with occlusions. Such occlusion causes a change in the image morphology. Therefore, the required shape cannot be detected.

2.1.2 Edge-based marker detection

These methods are more flexible with regard to the marker occlusions than the image morphology-based methods. One solution that is based on this principle is the *ARTag* system [Fia05]. Although the *ARTag* markers are similar to the *ARToolKit* markers (significant is a thick black border), the implemented detection method is completely different. The *ARTag* method is based on detection of *edgels* (edge pixels) of an object. Further, a set of lines is constructed from the found *edgels*. It is not necessary to detect all line *edgels*; therefore, the edge could be partially occluded. Finally, four corresponding lines represent edges of a potential marker.

The same detection principle is used also in *StudierStube* project [Hir08] and many others.

2.1.3 Feature-based marker detection

These methods are based on key points detection. The key points are various regions of interest: edges, corners, blobs. To identify whether a given key point really represents a part of a marker, it is necessary to match it with a key point in a marker template. This matching process requires that both key points to be matched are described by gradient changes in their neighborhood. The process of key point

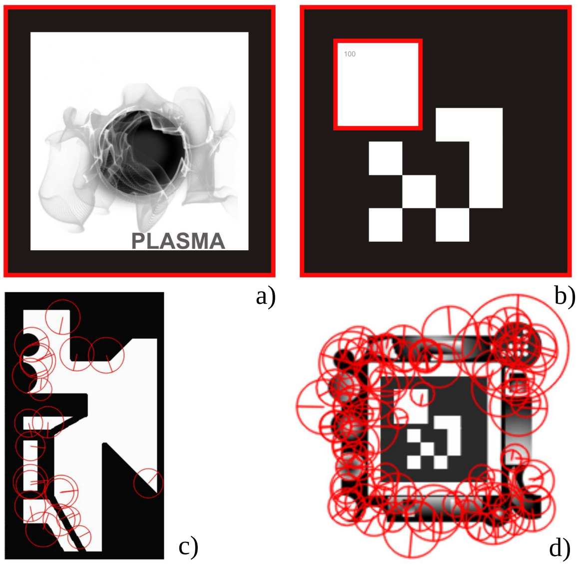

Figure 2: Markers used with different detection methods. Detected features are highlighted with a red color. From left: a) Match template, b) Golay error correction code, c) SURF and d) S-G detection

neighborhood description is usually called feature extraction. The output of this process is a set of feature descriptors vectors. The feature descriptors are later compared and their distance is computed [Low04]. This enables to match points of interest between a template and a camera image.

There are several approaches for feature-based detection. Widely used are e.g. SIFT [CHT*09] and SURF [BTG06]. A thorough comparison of selected methods is described in [TM08]. The SURF (Speeded Up Robust Features) algorithm has a good ratio between detection capabilities and performance. The SURF algorithm application is composed of three steps: detection of key points (points of interest), feature extraction and key point matching.

The detection of image key points that are used for the image description is based on gradient changes in the grayscale version of the image. Each key point is identified by position and radius that specifies the size of the key point neighborhood. Then the process of feature extraction is performed.

During this process, each key point neighborhood is described using 64-dimensional or 128-dimensional vector that describes the gradient changes of each key point neighborhood. The descriptors are produced both for a template and a camera image, so that the corresponding key points are identified.

The SURF main advantage is the scale and rotation invariance [BTG06]. This allows the SURF to work even with low resolution images or small objects. Another advantage is that the algorithm compares only the points; therefore, the object can be partially occluded. Although the SURF method is usually used for natural object identification (see e.g. [BCP*08]), it can be used also for marker detection as described in our method outlined in section 3.

2.2 Marker Identification Approaches

Morphological and edge-based detection methods are commonly used with following marker identification approaches: template matching and decoding of various binary codes.

Match template identification is based on computation of a pixel value correlation between a potential marker and a list of templates. In case the correlation fulfills a given threshold, the marker is identified. Obviously, the method has a linear time complexity. It is necessary to compute correlations with all templates until the required one is found or all templates are tested. Moreover, it is difficult to choose a threshold that allows to distinguish a large amount of markers [Bru09]. Therefore, methods based on different binary codes are frequently used to compensate this problem. One of the possible codes is the Golay error correction code.

A marker based on the Golay error correction code (ECC) can be composed of a large white square in the top left corner and e.g. 24 black or white squares that encode a number. The large square provides information about the marker orientation (see Fig. 2-b).

In the first step, a Golay ECC decoder for such a marker detects the position of the large white square. Further, it divides the code area into 24 blocks and calculates an average pixel value in all segments. Finally, the average value is thresholded to zero or one and the binary code is reconstructed. Possible implementation of the code reconstruction is outlined in [MZ06].

A significant advantage of this approach is that the binary code is reconstructed in a constant time. Another important advantage is the ability to correct errors caused by occlusions or an image corruption. Finally, the amount of distinguishable markers is limited just by the binary code length.

A feature-based method, such as the SURF is, is capable of both marker detection and marker identification. Therefore, it is not usually used with an identification method. As mentioned above, the method relies on searching for distinctive key points in a camera image that are then matched against image template descriptors. This process has linear time complexity because all template descriptors must be tested until the required one is found.

2.3 Summary of the Marker Recognition

In general, there are three approaches for marker recognition. The first one is based on image morphology. Detection can be fast and precise. However, it cannot deal with marker occlusions. The edge-based methods can detect partially occluded markers. Nevertheless, this ability is limited. Larger occlusions of the edges are problematic. Both detection methods can be accompanied by a binary code identification method that is able to work in a constant time and reliably distinguish a substantial amount of markers.

Feature-based approaches are able to detect and identify even substantially occluded markers. However, they work in a linear time. This complexity is usually problematic for real-time applications with larger amounts of markers. Even more, feature-based methods have problems with distinguishing of similar markers [SZG*09].

3. S-G HYBRID RECOGNITION METHOD

The proposed identification method combines the positive properties of two previously mentioned methods. We take advantage of robustness of the SURF feature-based object identification and combine it with high reliability and effectiveness of the Golay error correction code detection, hence the name *S-G hybrid detection method*.

3.1 Marker design

As described in section 2.1.3, the SURF algorithm is suitable especially for natural objects identification. However, many applications use this method to identify only a single object in an image. This is marker may appear in a scene.

The most problematic part of the SURF marker identification is the matching of corresponding marker key points in both images. The key points similarity is determined by gradient changes in the key points neighborhoods (these are represented by feature descriptors). If the image contains areas with similar gradient changes, then such areas will be identified as the same or similar key points.

Therefore, it is important to design markers so that the key points identified in them have distinctive gradient changes. Furthermore, these key points must be distinguishable both from the scene image and from other markers.

We use artificial markers very distinctive from the scene objects. Acceptable results are obtained using complex asymmetric markers composed of arbitrary geometric shapes (see Fig. 2–c). These markers are easily detected because they contain a substantial amount of features which can be tracked. The development of such marker, however, requires a lot of manual work. Even with a thorough testing it seems that only a very low number (approx. 3) of these markers could be reliably distinguished in an image.

Therefore, to ensure the correct marker identification we propose a hybrid detection method – *S-G Detection* – in which we combine the SURF algorithm with the Golay error correction code. In this case, the marker template is divided into two parts: the marker border and marker content. These two parts of a template may be combined independently.

Marker content is composed solely of a Golay code image. Only the marker content is used for marker identification. This has the advantage of very high identification reliability and allows to distinguish large number of markers – see section 2.2.

Marker border is composed of different geometric shapes selected so that they are distinctive from real scene objects. However, the border is no longer used for identification of the markers. This is possible because each marker border may be combined with any number of Golay codes to identify the marker. This combination solves the problem of distinguishing between marker templates while maintaining great robustness against template occlusion (see Fig. 4). Different marker borders may be used in the application. However, it is not necessary. We use the same border for all markers.

3.2 Marker Detection

As been described in the previous section, we use the SURF method to identify key points only in the marker border (see Fig. 2–d). This border is the same for all markers. A strong advantage of this approach is that the time complexity of the whole algorithm is not a function of a number of templates (see section 2.1). Therefore, we can use a high number of markers without a performance hit. This is an important usability feature.

A common approach [Lag11] in matching the template and video frame points of interest is: find the best matches of key points using a defined metric, filter out false positives (invalid matches), repeat filtering until a sufficient number of adequately reliable points are obtained.

Errors in matched points may occur when a template key point is matched to an unrelated video frame key point because it happens to have similar neighborhood. Another source of errors occurs when a video frame contains two or more markers and template points are matched to correct points but on different marker borders (two or more physical markers).

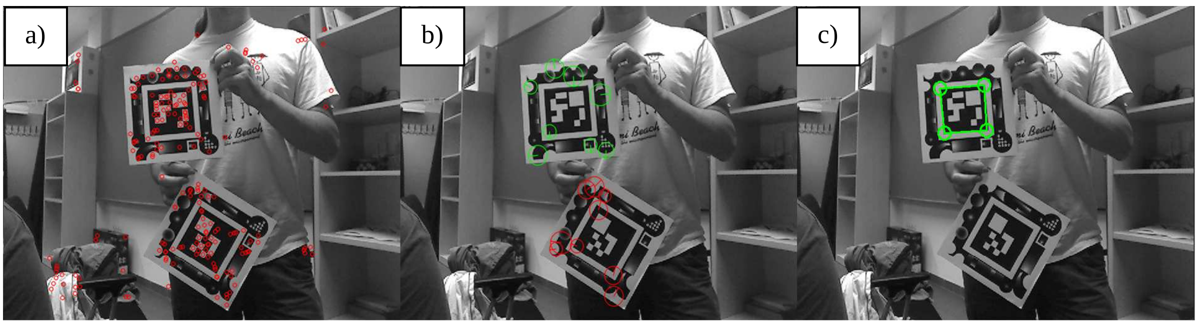

Figure 3: S-G hybrid method. From left: a) key points are detected and filtered b) angle filter is applied so that the key points on both markers are distinguished c) marker specified in application configuration is detected.

For many applications, it is enough to identify if the template is present in the image, other applications require approximate template positions. Our application requires the exact position (translation and rotation) of the marker so that the virtual object may be inserted to the real scene.

The first step of marker matching feature extractor is to discover key points in the processed image. Then a descriptor vector for each key point is found using a feature extractor. These vectors are matched by computation of Euclidean distance between each pair of points. Moreover, we use symmetric matching filter for the key points.

First, template key points are matched against video frame image and the best matches are selected. Then the frame key points are matched against template key points, and best matches are selected. The intersection of these two sets is a set of matched points [Lag11].

Further, we filter the set of key points by application of an angle filter. The idea behind the angle filter is to take advantage of the information stored in a SURF key point itself. Each SURF key point contains an angle value, which defines the direction of the most significant gradient descent in the neighborhood of the key point. In our application, we use artificial markers; therefore we search for a set of predefined objects. This means that relative differences in rotation of the matched key point must be similar for all matched key points. That is – if the template has two key points and their rotation is 45° and 70°, then the two key points matched in the frame must have the rotation difference approximately 25°. Due to perspective deformations, the differences can be computed only approximately. An example of this filtering is shown in Fig. 3 – each set of differently colored points maintains the same relative rotation differences between points (in other words the same rotation difference between a template and a video frame).

A difficult part of the angle filtering algorithm is defining initial conditions. This is caused by the fact that until the marker is identified, its key points, their rotations and order are all unknown. To overcome this problem, the angle filter algorithm is implemented by marshaling all possible rotations into overlapping intervals of a defined width (rotation difference tolerance – RT). Each interval overlaps half of neighboring intervals so that there are no artificial boundaries. Key points in each interval are then processed individually as if it was a standalone key point set. This introduces a performance hit as another loop iterating over sets of key point has to be processed. Fortunately this is upper bounded – maximum number of iterations is 360 / (RT · 2). This upper bound is hardly reached because only sets containing at least four points need to be processed. A minimum of four points is required for a correct positioning of a 3D model which will be added to the image later in the process. The angle filter algorithm may be described by the following pseudo-code:

```
FOR each matched_point
  difference =
    matched_point_template->angle -
    matched_point_frame->angle;
  div = difference / RT
  angles[div * RT]->add(matched_point)
  angles[(div + 1) * RT]
      ->add(matched_point)
END FOR
FOR each angle
  find homography
  identify Golay marker
  IF marker identified THEN
      display 3D object
END FOR
```

3.3 Marker Identification

For each set of points detected by the angle filter, we compute homography matrix so that the Golay code can be identified. By applying the homography transformation to the camera image we compensate the perspective deformation. This image transformed

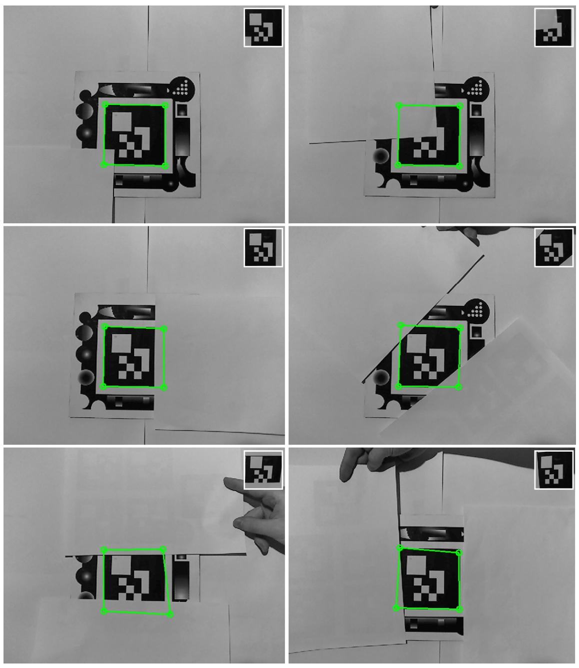

Figure4: Examples of S-G method capability of occluded marker identification from a close distance.

to the camera plane is cropped and processed by the Golay code detector.

If a Golay code is found, it means that the marker is identified. This identification introduces important feedback for the SURF marker detection. Given the reliability of the Golay detector, false positives are almost impossible. In other words, if the code is identified, we can be sure it is one of searched markers. It also means that the homography was computed correctly. This is also important because we can use the points to compute projection matrix. Reliable projection matrix is important for correct 3D models positioning.

In section 2.2 that describes the Golay codes is stated that the Golay code rotation is determined by the position of the large white square in the top left corner. Since the S-G detection method is focused on robustness against marker occlusions, it is undesirable to have parts of the marker with greater importance. In the S-G method, the rotation of the marker is determined solely by the position of key points. This part of the Golay code is therefore unused.

4. COMPARISON

The S-G hybrid method was tested against two other solutions: *ARToolKitPlus* (http://handheldar.icg.tugraz.at/artoolkitplus.php) and *ALVAR Toolkit* (www.vtt.fi/multimedia/alvar.html). All tests were made in a laboratory under artificial light. We used markers with 14 cm long edge for testing. The solutions were tested from three aspects:

- Distance – minimum, maximum and maximum distance without visible jitter.
- Angles – marker was placed at different distances from the camera and rotated around *x* and *y* axis (the *z* axis was not tested because all solutions are capable of 360 degrees rotation).
- Occlusion – occlusion was tested with stationary marker and camera.

Compared to the other two solutions, S-G has a smaller maximum distance where it is capable to identify a marker. The S-G method is able to detect a marker placed at a distance 2 m from the camera. The *ARToolKitPlus* and *ALVAR* have maximal distance at approx. 5 m.

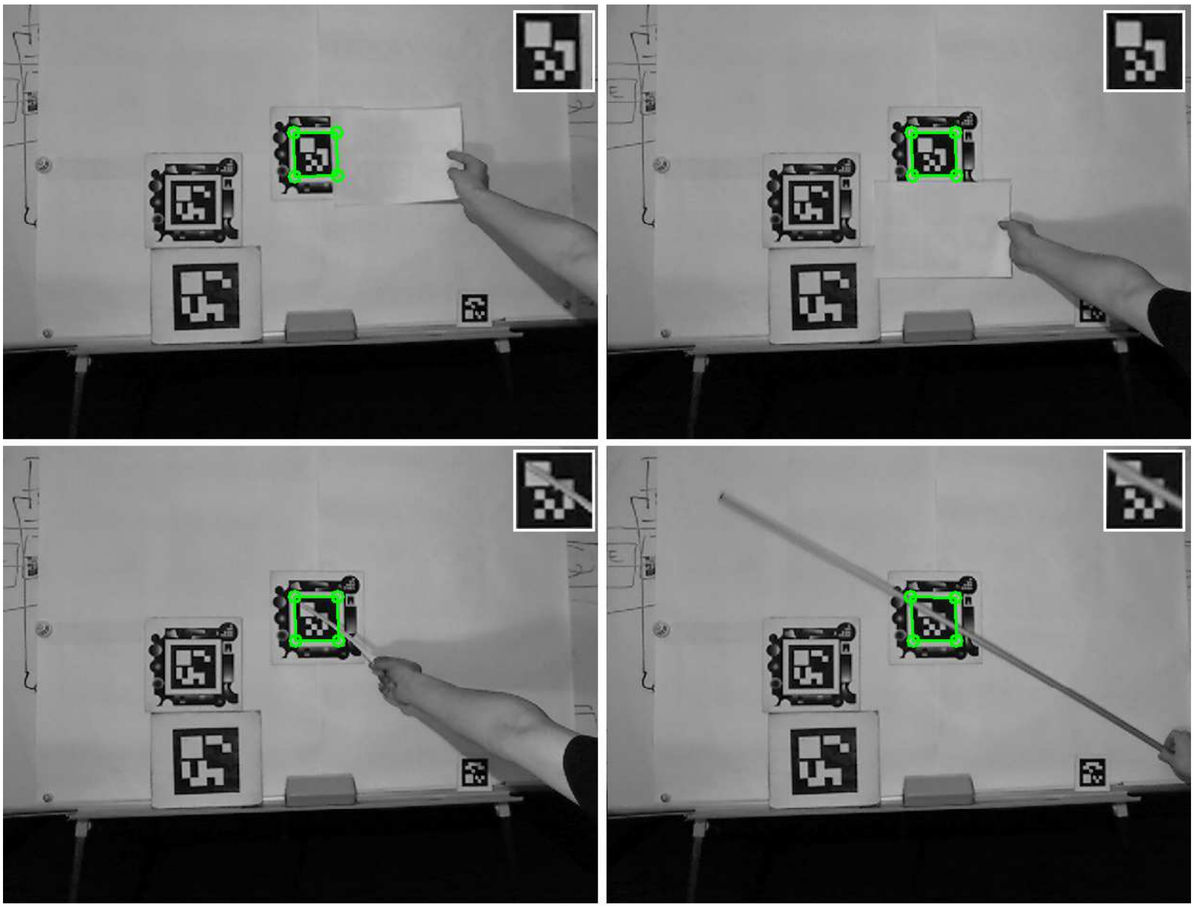

Figure 5: Examples of S-G method capability of occluded marker identification from a large distance. Both the marker boarder and marker content may be occluded.

In the angles comparison, measured results are influenced by the SURF algorithm limitations. The S-G method is able to detect a marker that is under 55° angle to the camera axis. (0° represents a marker perpendicular to the camera axis. The maximal theoretical angle is therefore 90°.) The other two solutions have maximal angles ranging from 74° to 85°.

Neither *ARToolKitPlus* nor *ALVAR* can deal with any type of occlusion. This is the most important disadvantage of these solutions. The S-G method can deal with significant marker occlusion. Because S-G works with key points instead of morphological operations or e.g. *edgels*, it is able to withstand a substantial number of different occlusions. We tested several of them (see Fig. 4).

The marker border can be obstructed up to 50 %. It is irrelevant what part of marker border is obstructed (all corners, two whole sides, etc.). The marker content (the Golay error correction code) must be visible at least from 75 % in case the large white square is obstructed. In case the obstruction is in other part of the Golay code, maximum allowed occlusion is approx. 15 %. This occlusion is limited by the Golay code redundancy.

This is the most important contribution of our solution in comparison to other used methods**.**

Because of the nature of the detection, the solutions capable of occlusion (e.g. *ARTag*) need at least three visible marker corners to detect and identify the marker. Our method is capable of the identification of a marker with all corners or sides covered. Our method has capability even of overcoming of the occlusion of marker contents. This is possible because of the Golay error correction code usage.

Figure 6: Marker occlusion. The marker is approx. 2 m distant from the camera.

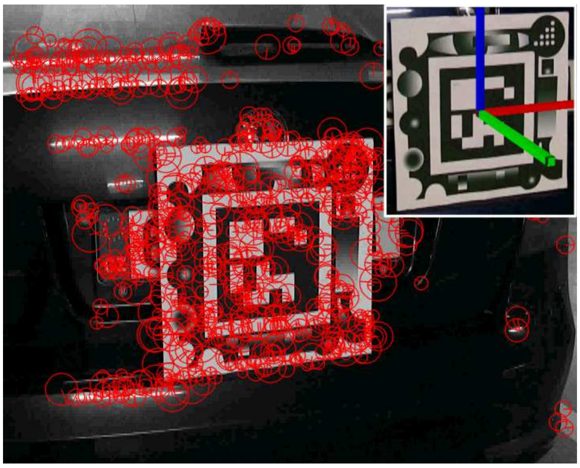

Figure 7: Key points detected by the S-G method and augmented 3D model.

5. CONCLUSION

Our application aims to improve the car design process. Therefore, several criteria must be fulfilled: Our marker detection and identification methods must be able to distinguish several hundred markers (one marker represents one spare part). Further, it must be possible to compute the precise position and rotation of the marker. Finally, the methods must be able to deal with occlusions that are common in real situations.

The SURF detection method as well as the Golay error correction code is able to deal with the occlusions. The proposed S-G registration method is slower than other frequently used approaches (e.g. image morphology approach with the Golay error correction codes). Still, it works in a constant time that is significant for real-time applications.

Nevertheless, in case of very good lighting conditions and absence of occlusions we recommend techniques based on the image morphology. With these methods, the video stream processing speed is substantially improved. Our *AuRel* application supports both approaches; therefore, registration technique is chosen according to the current conditions. By default, the morphology-based method (16 fps) is used. In case a marker detected in previous frame is missing, we switch to the S-G method (4 fps). Following frame is again processed by morphology-based method. Frame rates are measured on 640×480 px camera stream processed by Intel Core i5 2.6 GHz, 4 GB RAM, HDD 7200 rpm.

We consider our approach very promising. Nonetheless, there must be further research focused on several technical aspects. Particularly, the marker detector performance should be optimized (on the reference hardware configuration, *ARToolKitPlus* and *ALVAR* have above 20 fps). This could be done by reducing the number of key points in exchange for lower reliability. Also the maximum detection distance needs to be improved. Possible solution can be to improve marker design so that the marker detector response is increased as outlined in [Sch*09].

SURF method can be easily used to design a marker-less tracking method as outlined in many articles. The absence of markers can substantially improve the application usability. Nevertheless, there could be a problem with selection of a correct 3D model and its manual position adjustment.

6. ACKNOWLEDGMENTS

This paper is written as a part of a solution of the project IGA FBE MENDELU 7/2012 and FBE MENDELU research plan MSM 6215648904.

7. REFERENCES

[BCP*08] Barandiaran, I., Cottez, Ch., Paloc, C., Grana, M.: Comparative Evaluation of Random 159 Forest and Ferns Classifiers for Real-Time Feature Matching in WSCG 2008 Full Papers Proceedings, Plzen: University of West Bohemia, 2008, pp. 159-166.

[Bru09] Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, 2009, ISBN 978-0-470-51706-2.

[BKF*00] Balcisoy, S., Kallmann, M., Fua, P., Thalmann, D.: A framework for rapid evaluation of prototypes with Augmented Reality. In Proceedings of the ACM symposium on Virtual reality software and technology, pp. 61-66. 2000. ISBN:1-58113-316-2.

[BTG06] Bay, H., Tuytelaars, T., Gool, L. V.: Surf: Speeded up robust features. In ECCV, 2006, pp. 404–417.

[CHT*09] Cui, Y., Hasler, N., Thormahlen, T., Seidel, H.: Scale Invariant Feature Transform with Irregular Orientation Histogram Binning. In Proceedings of Image Analysis and Recognition: 6th International Conference, pp. 258-267, 2009. ISBN: 978-3-642-02610-2.

[FAM*02] Fiorentino, M., De Amicis, R., Monno, G., Stork, A.: Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design. In Proceedings of International Symposium on Mixed and Augmented Reality, p. 86. 2002. ISBN:0-7695-1781-1.

[Fia05] Fiala, M.: ARTag, a fiducial marker system using digital techniques. Computer Vision and Pattern Recognition, 2, June, 2005.

[Hir08] Hirzer, M.: Marker detection for augmented reality applications, 2008. Inst. for Computer Graphics and Vision, Graz University of Technology, Austria.

[HNL96] Hoff, W. A., Nguyen, K., Lyon T.: Computer vision-based registration techniques for augmented reality. In Intelligent Robots and Computer Vision, XV, 1996, pp. 538–548.

[KB99] Kato, H., Billinghurst, M.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality. 1999, s. 85-94.

[KTB*03] Kato, H., Tachibana, K., Billinghurst, M., Grafe, M.: A registration method based on texture tracking using artoolkit. In Augmented Reality Toolkit Workshop, 2003. IEEE International, 2003, IEEE, pp. 77–85.

[Lag11] Laganiere, R.: OpenCV 2 Computer Vision Application Programming Cookbook. Packt Publishing, 2011.

[Low04] Lowe, D. G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60, 2004, pp. 91–110.

[MZ06] Morelos-Zaragoza, R. H.: The Art of Error Correcting Coding (Second Edition). John Wiley & Sons, 2006.

[PK11] Prochazka, D., Koubek, T.: Augmented Reality Implementation Methods in Mainstream Applications. Acta of Mendel University of agriculture and forestry Brno 59, 4,. 2011, p. 257.

[Sch*09] Schweiger, F. et al.: Maximum Detector Response Markers for SIFT and SURF, Proceedings of the Vision, Modeling, and Visualization Workshop 2009, Germany.

[SZG*09] Schweiger, F., Zeisl, B., Georgel, P., Schroth, G., Steinbach, E., Navab, N.: Maximum Detector Response Markers for SIFT and SURF. In Int. Workshop on Vision, Modeling and Visualization (VMV), 2009.

[Sze11] Szeliski, R.: Computer Vision: Algorithms and Applications, Springer, 2011.

[TM08] Tuytelaars T., Mikolajczyk K.: Local invariant feature detectors: a survey, Foundations and Trends® in Computer Graphics and Vision archive, Volume 3 Issue 3, 2008, pp. 177-280.

[VSP*03] Verlinden, J. C., Smit, A. D., Peeters, A. W. J., Gelderen, M. H. V.: Development of a flexible augmented prototyping system. Journal of WSCG, 11, 2003, pp. 496-503