8

Detection and Pose Estimation of Piled Objects Using Ensemble of Tree Classifiers

Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitarai and Hiroto Yoshii
Canon Inc.
Japan

1. Introduction

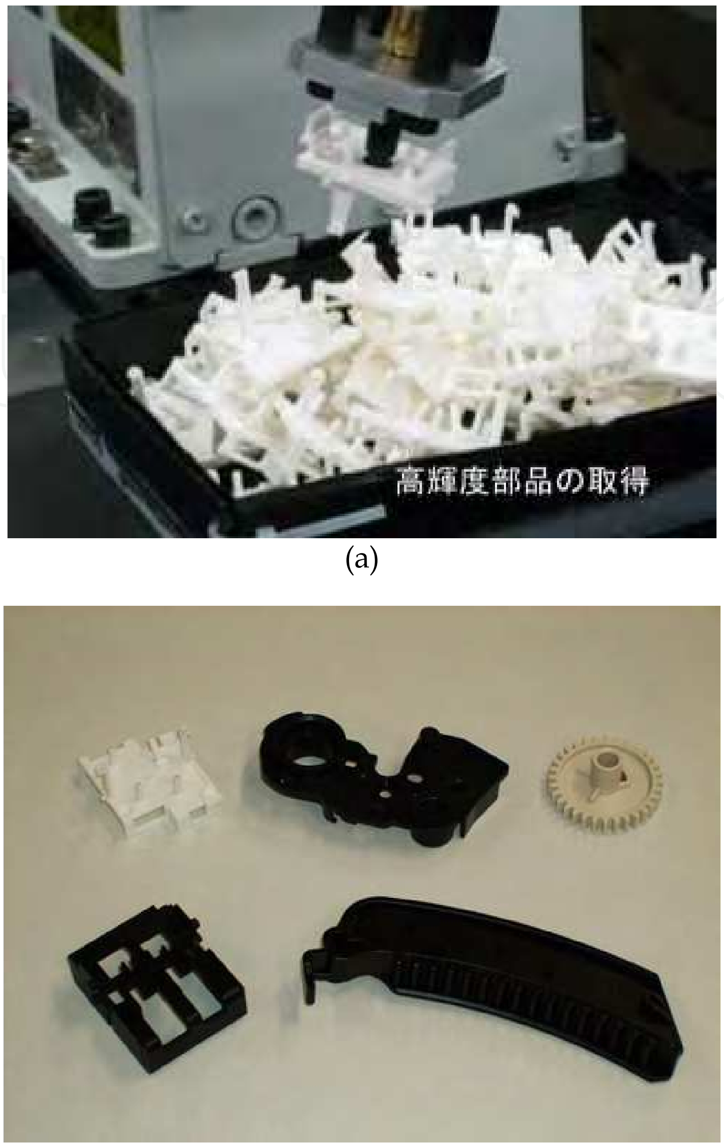
Detection and pose estimation of 3D objects is a fundamental machine vision task. Machine vision for bin-picking system (Figure 1 (a)), especially for piles of objects, is a classical robot vision task.

To date, however, there has been only limited success in this longstanding problem (e.g., picking piled objects), and, to the best of our knowledge, existing algorithms (e.g., Drost, et al., 2010; Ulrich et al., 2009; Hinterstoisser et al., 2007) fall short of practical use in automatic assembly of electronic products composed of various parts with differing optical and surface properties as well as with differing shapes and sizes. We found that even the state-of-the-art, commercially available machine vision software cannot be practically used for picking such piles of parts with unknown pose and occlusion. Specifically, as exemplified in Figure 1 (b), for black (or white)-colored and untextured parts with some degree of complexity in shape, conventional methods turned out to be of little use.

In this chapter, we present a potential solution to this classical, unsolved problem (i.e., Detection and pose estimation of each of piled objects) with an efficient and robust algorithm for object detection together with 3D pose estimation for practical use in robot vision systems. We consider the detection and 3D pose estimation as a classification problem (Lepetit & Fua, 2006) which constitutes a preprocessing stage of subsequent model fitting for further precise estimation (Tateno et al., 2010), and explore the use of ensemble of classifiers in a form of Random Forests (Lepetit & Fua, 2006; Gall & Lempitsky, 2009; Shotton et al., 2011) or Ferns (Bosch et al., 2007; Oshin et al., 2009; Özuysal et al., 2010) that can handle multi-categories.

Based upon sliding window approach, we formulate the problem as classifying a set of patches of input image (local regions) into a sufficient number of conjunct pose and location categories which are supported by distributed representation of leaf nodes in trees.

Main contributions of this paper are 1) spatially restricted and masked sampling (SRMS) scheme, 2) voting through local region-based evidence accumulation for pose categories, 3) cancellation mechanism for fictitious votes (CMFV) suggesting ill-conditioned and degenerated sampling queries for pose estimation, altogether leading to robust detection and 3D pose estimation of piles of objects.



(a)

(b)

Fig. 1. (a) Picking system for piles of parts. (b) Parts with various shape and surface properties.

2. Basic formulation

Detection task of piled objects composed of the same category parts as shown in Figure 1 (b), inherently requires following three properties. 1) Robustness to occlusion, 2) Robustness to high background clutters (i.e., noisy clutters are by themselves some other objects of the same class in the neighborhood of a specific object to be detected), 3) Robustness to drastic variability of object appearance due to varying pose and illumination change, especially for objects with higher specularity.

In this section, we show details about basic formulation of the proposed algorithm. We show here a new patch based method for object localization as well as pose estimation, which is a class of generalized Hough transforms and similar in spirit with Hough Forests (Gall & Lempitsky, 2009), and the basic strategy for the improvement is given in the next section.

Through construction of trees in the training, pose classes of an object are defined in a form of tree-like structure and described by a set of local cues inside patches. For each tree, a set of patches P_i, local regions in an edge-enhanced feature image (given in subsection 3.3) can be defined as $\{P_i = (F_i, C_i)\}$, where F_i is the appearance of the local feature image, C_i is the label of the patch, $C_i = (l_i, Pose_i)$ where l_i denotes its location inside the object and $Pose_i$ is associated pose class. Whole sets of local cues $\{t(F_i)\}$, in the entire trees associated with particular pose of an object, constitute codebooks or dictionary of particular poses. Specifically, the local cues $t(F_i)$ are binary data given by comparison of two feature values at given paired locations (p_1,q_1) and (p_2,q_2) and defined as:

$$t_{p_1,q_1,p_2,q_2}(F_i) = \begin{cases} 0, & if\ F_i(p_1,q_1) < F_i(p_2,q_2) \\ 1, & otherwise \end{cases} \tag{1}$$

2.1 Building ensemble of trees

In the training phase, construction of trees goes recursively by setting patches as well as paired locations (sampling points) inside each patch. For a given number L of trees, we perform L sessions of training and prepare a set of feature images for training.

The feature image is given by preprocessing (explained in subsection 3.3) the input image to obtain edge-enhancement, while suppressing noise. Since our pose classification is succeeded by model fitting for further precise estimation, total number of pose categories is determined by the resolution and accuracy requirement on the initial pose estimation imposed by the subsequent model fitting process (Tateno, et al., 2010), and the number could be huge (see Section 4). At the beginning of each training session, set of patches are first randomly generated subject to the condition that their locations are inside the object, and sampling points are probabilistically set according to the new scenario given in subsection 3.1. For example, in Figure 2 we have four patches for respective five pose categories, which amount to 20 training images.

A *leaf* node is the one which contains less than a fixed number of patches or its depth is the maximum value a-priori set, and if it contains no patch, we call it *null* or *terminal* node. In the training, we set a maximum depth of node constant among trees, and starting from the *root* node, the node expansion continues until it reaches the maximum depth or terminal node. At each node of a tree, if it is not the *terminal* nor *leaf* node, binary tests (1) are performed for a set of patches inside the node, and they are partitioned into two groups which are respectively fed to two child nodes (Figure 3).

We do not have a strict criterion on the training performance, a criterion on good codebooks being generated. One of reasonable criteria is that many of leaf nodes should have only one patch (i.e., single pose category) so that uniqueness of distributed representations is ensured, and another criterion is the diversity of sampling points so that spatial distribution of query points are not biased to some limited local area of the object. For the second criterion, because of geometrical triangulation principle, it is reasonable to consider that estimated pose category at wide spread positions, many of which supports the same category, is more credible than those from narrowly spread positions.

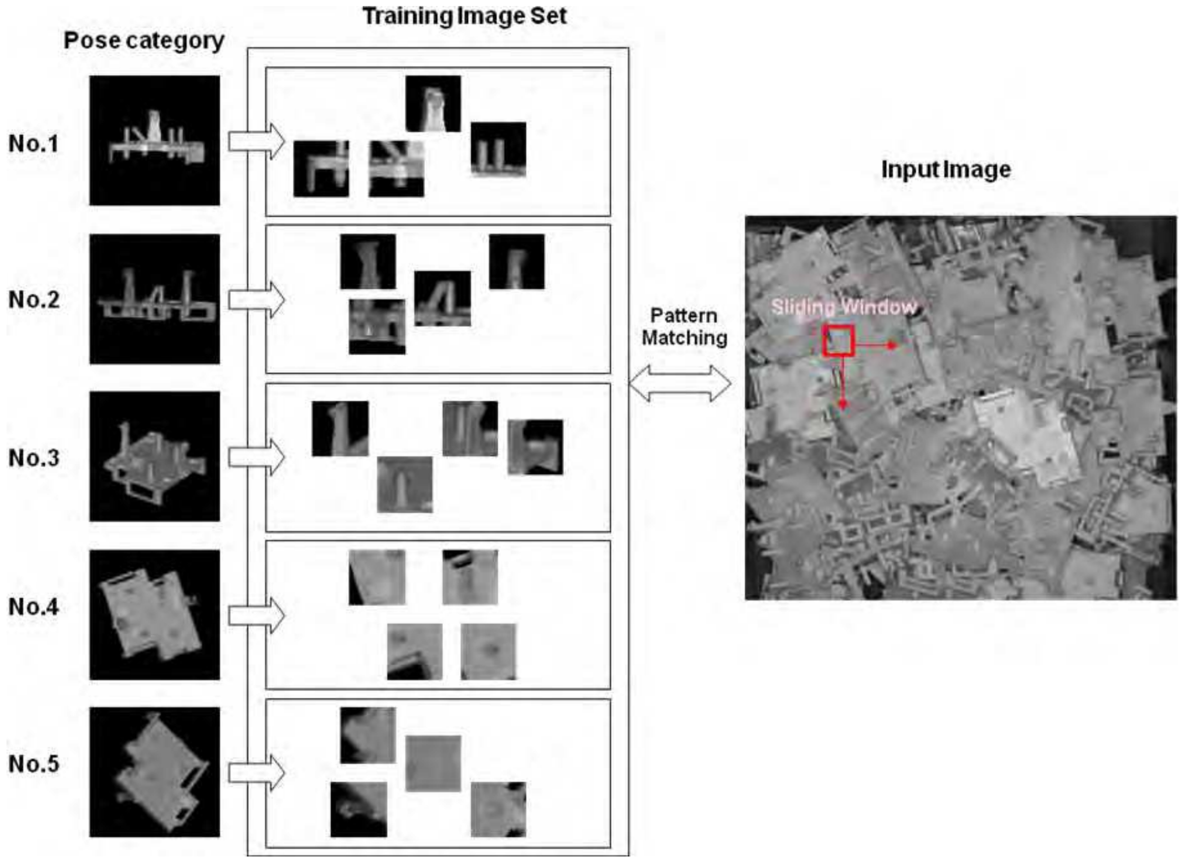


Fig. 2. (left) Schematic patch images for training; (right) Sliding window for matching.

2.2 Detection and pose estimation by ensemble of trees

In sliding window approach, we raster-scan the entire image using a local window of appropriate size, and at each position, parts detection together with pose estimation is done using the ensemble of classifiers .

Decision about the classification is done based on voting the outputs of leaf nodes among trees, followed with thresholding. The voting stage accumulates the supporting, local evidences by collecting outputs of leaf nodes among trees which signify the same pose category. Figure 4 schematically shows concentration of specific pose category as the result of correct voting, and no concentration for other pose classes. For the total number of *L* trees, voting for class *j* is performed for each pose category, yielding score *S(j)* as:

$$S(j)=\sum_{t}^{L}\int_{r}\delta\left(C_{j,t}(r),1\right),$$

where *r* denotes relative position vector of a patch directing to the center of object, $C_{j,t}(r)$ is 1, if, in the *t*th tree, class label *j* is detected by the patch assigned with position vector *r*, and $C_{j,t}(r)$ is 0, if otherwise. $\delta(a,b)$ is 1 for $a = b$, and 0 for otherwise. In practice, we use the following weighted voting given as:

$$S_r(j)=\sum_{t}^{L}\sum_{k}F(r_k,r)\delta\left(C_{j,t}(r_k),1\right),$$

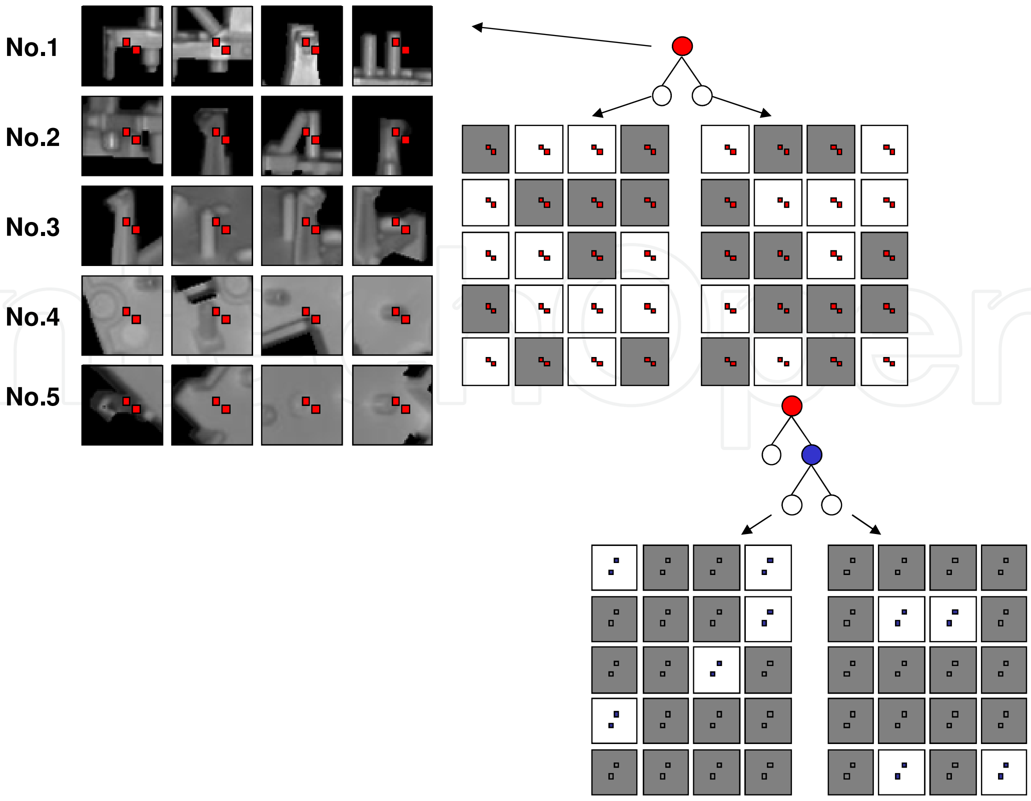


Fig. 3. Patch data partitioning. Based on a comparison of two pixel values, divide a set of patches into two groups with the same content (i.e., left or upper pixel value is larger or not than the other).

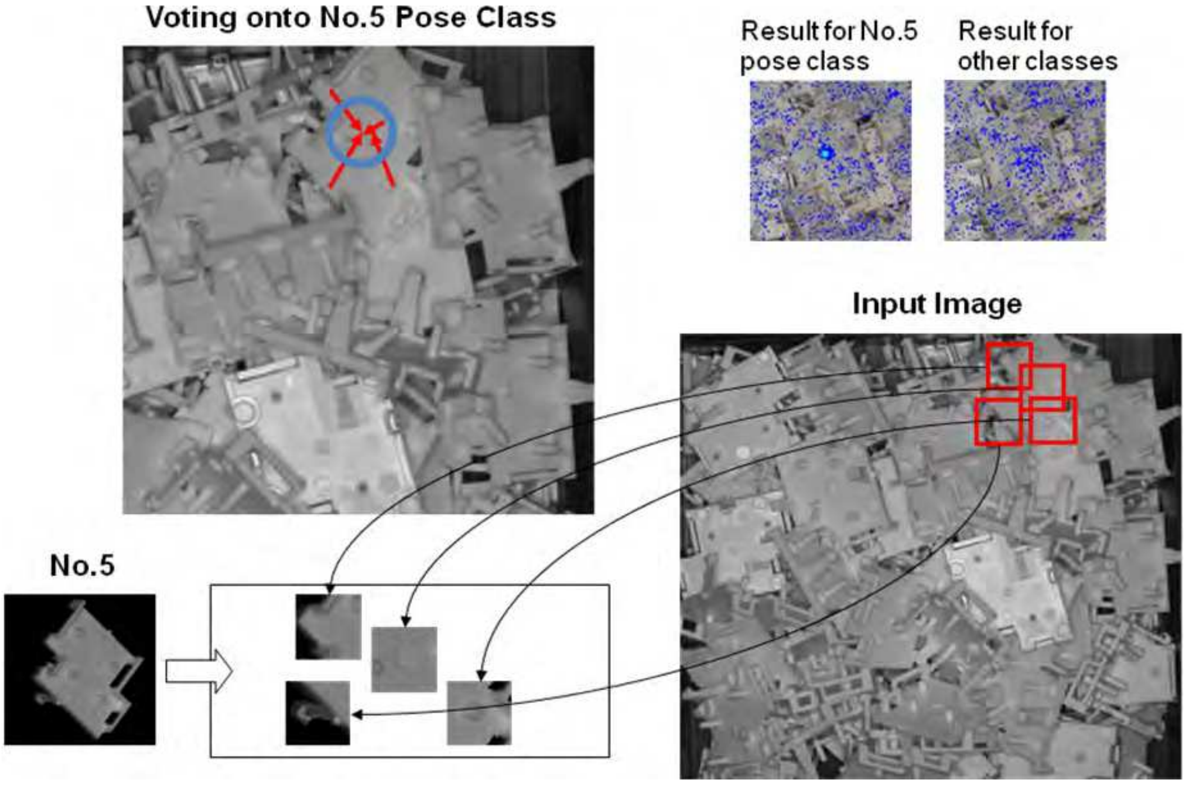


Fig. 4. Observed concentration of voting for correct class.

where $S_r(j)$ is the score for class j at the position r, and r_k is the location of patch k, $F(r_k)$ denotes weighting function with belle shaped envelope. Location of the object detected can be found by taking summation of position vectors r for patches identified as having the same pose category $C_j(r)$ under the conditions that the score $S(j)$ is maximum or ranked K (K is ordinarily set as 1 or 2) or above among other categories and that S is also above a certain threshold S_0. If K is set as 2, we select the most plausible estimate in the subsequent model fitting process (Tateno, et al. 2010).

3. Patch-based approach in ensemble of tree classifiers

First, to deal with three issues given in Section 2 faced in the detection task of piled objects, we incorporate the SRMS scheme (subsection 3.1) in ensemble of classifiers, a class of Hough Forests (Gall & Lempitsky, 2009). This sampling scheme together with CMFV (subsection 3.2) turned out to be very effective for enhancing robustness to occlusion as well as background clutters. The proposed method uses training images composed of only positive data because of SRMS as well as patch data generation inside the object. This is in contrast to Hough Forests which handle both positive and negative data in a supervised learning. In the second, we perform a series of feature extraction (subsection 3.3) for edge enhancement while suppressing noise, namely bilateral filter and gamma correction for smoothing, Laplacian filter for edge extraction, and Gaussian filters for blurring to form a channel of feature images fed to ensemble of trees.

3.1 Spatially restricted and masked sampling (SRMS)

Hough Forests (Gall & Lempitsky, 2009), a class of both random forests and generalized Hough transform, introduced spatial restriction in a form of patch (e.g., local region in a image) so that sampling queries are generated inside respective patches. In addition to this patch-based framework, we introduce here another spatial restriction in a form of mask which is the silhouette of object with particular pose. Mask defined at each node is used to impose probabilistic restriction onto queries so as to be inside the object with a range of poses. In contrast to the proposed approach, Hough Forest does not restrict location of patches in image. Thus, in the training phase to construct trees, each patch is set at random so that its centre position shall be inside the silhouette of objects to be detected, while respective sampling pairs of points are probabilistically set based on the conjoint mask data given as follows.

This combination of locality restrictions in generating sampling queries helps to enhance robustness against occlusion. We define a *conjoint mask* as a conjunction of silhouettes of objects with different poses in the corresponding node. A node in the tree generally contains multiple classes of pose, and at each node, pose categories are partitioned into two sets of data. Those partitioned data are fed to subsequent nodes, where partitioning follow. Here we show several ways of generating the mask data resulting from conjoint of *composite mask* data at each node of a tree. Here, we use the term, *composite mask*, for one of masks in the node. One way of conjoining silhouettes is taking AND operation on them for a given node in the tree. The resulting mask data M is thus given by

$$M = \left(\prod_{k=1}^{N} M_1^k, \prod_{k=1}^{N} M_2^k, \ldots, \prod_{k=1}^{N} M_n^k \right)$$

where N is the number of mask images (i.e., number of pose classes in the current node), $M^k=(M_k^1, M_k^2, \ldots, M_k^N)$,and M_k^j is binary data at the *k*th location for the *j*th class of pose categories in the current node, *n* is the dimension of the mask image (i.e., number of pixels).

Another way of constructing mask data for a given node is to take OR operation on them:

$$M = \left(\sum_{k=1}^{N} M_1^k, \sum_{k=1}^{N} M_2^k, \ldots, \sum_{k=1}^{N} M_n^k \right)$$

Yet another way of constructing mask data in the node is to take ORs of patch data m_k^j within a composite mask image:

$$M = \left(\sum_{k=1}^{N} m_1^k, \sum_{k=1}^{N} m_2^k, \ldots, \sum_{k=1}^{N} m_n^k \right) \tag{2}$$

Here, *M* in eqn. (2) does not give the conjoint data of mask silhouettes, but it is sufficient for us since sampling pair data are to be generated inside those patches. Next, we show our scheme of probabilistic generation of sampling queries. For conjoint mask data *M* at each node, we define a probability density function *P* by simple normalization, given as follows.

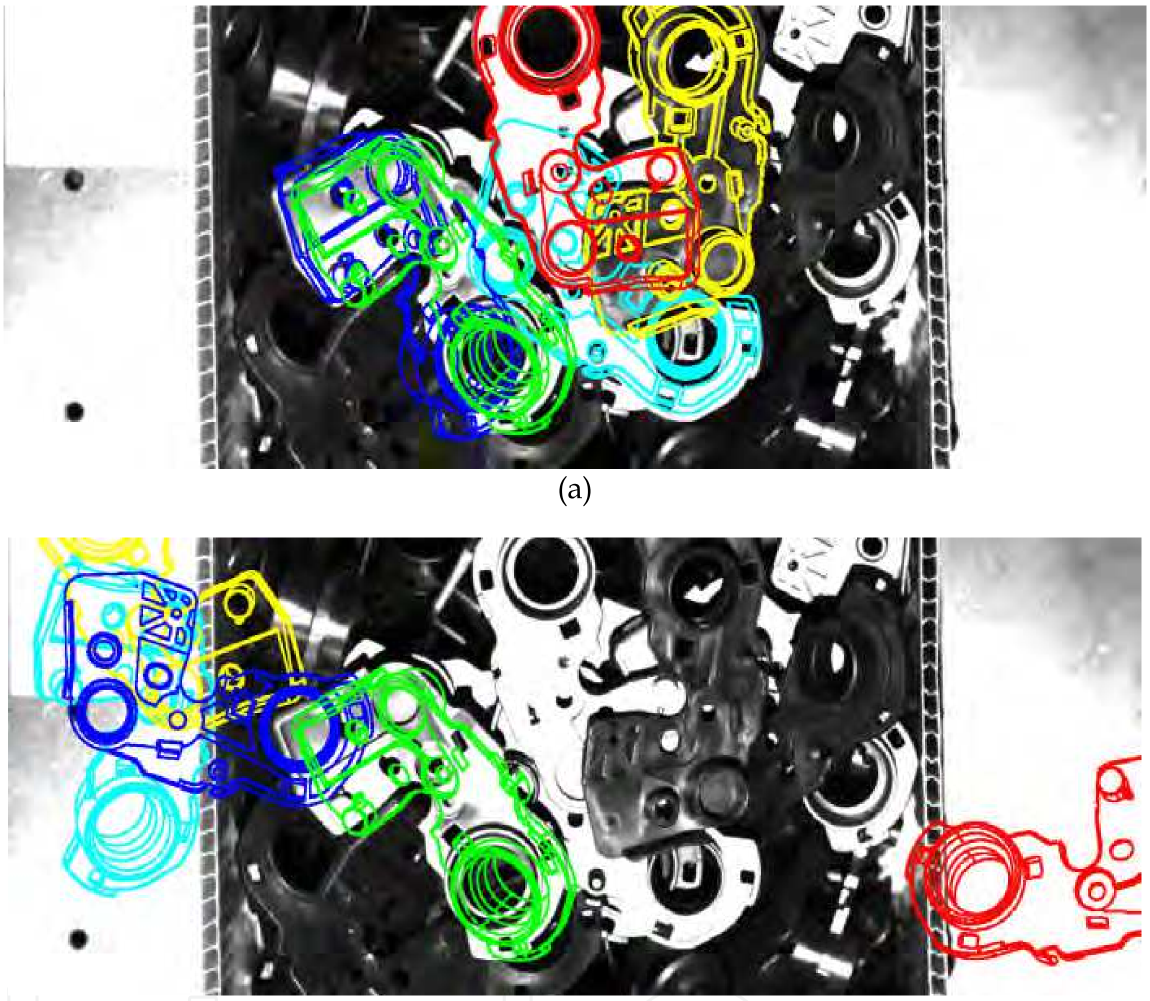
$$P = \frac{\left(\sum_{k=1}^{N} m_1^k, \sum_{k=1}^{N} m_2^k, \ldots, \sum_{k=1}^{N} m_n^k \right)}{\sum_{j=1}^{n} \sum_{k=1}^{N} m_j^k} \tag{3}$$

Then we generate pairs of sampling points based on the probability density function (3) which are guaranteed to be inside either of patches. This sampling scheme together with CMFV in the next subsection turns out to be very effective for enhancing robustness to occlusion as well as background clutters.

3.2 Cancellation mechanism for fictitious votes (CMFV)

In automatic assembly line of manufacturing, containers are used for supplying parts. Those containers are usually kinds of trays or boxes made of rectangular planes. In practice, linear portions of a tray cause detection errors if objects to be detected are made up of many or longer linear portions. In such cases, it is not surprising to confuse linear portion of a tray as a part of object to be detected, since our method as well as randomized tree based approaches are based on accumulation of local evidence (e.g., comparison of feature values at two sampling points). This confusion resulting from such degeneration is reminiscent of so-called aperture problem in computer vision.

Proposed cancellation mechanism is intended to alleviate such confusions. As in generalized Hough transform, we perform voting as local evidence accumulation in which each of evidence is obtained through sliding window. In practice, this local evidence accumulation can cause degenerated results which cannot be in principle disambiguated. As a result, we may have excessive concentration of classification results with the same localized category (i.e., posture observed at particular location of the object) at particular locations. In such cases, we consider the result as fictitious. The criterion for this singularity is empirically set as a threshold, assuming appropriate probability density function. For example, if the number of patches used for training is much larger than the total sliding numbers, then we can assume Poisson distribution, and for other cases, binomial distribution. Details about the threshold and probability density will be given in Section 4.



(a)

(b)

Fig. 5. (a) Result with CMFV, (b) Result without CMFV.

3.3 Pre-processing: Feature extraction

We perform a series of pre-processing to extract feature images as input to the ensemble classifiers. A typical feature image is an edge image. This processing includes edge extraction by Laplacian filter, blurring with Gaussian filters, and some other non-linear processing. Examples of extracted feature images are shown in Figure 6. Since edge extraction tends to enhance noise, blurring process is necessary for the suppression of noise, however, it could affect the performance, since contrast of edge image is degraded.

2D features thus obtained with appropriate parameters are very important and they significantly influence the performance of randomized tree-based classifiers. These set of operations turn out to be important for robustness and precision of final results (see Section 4).

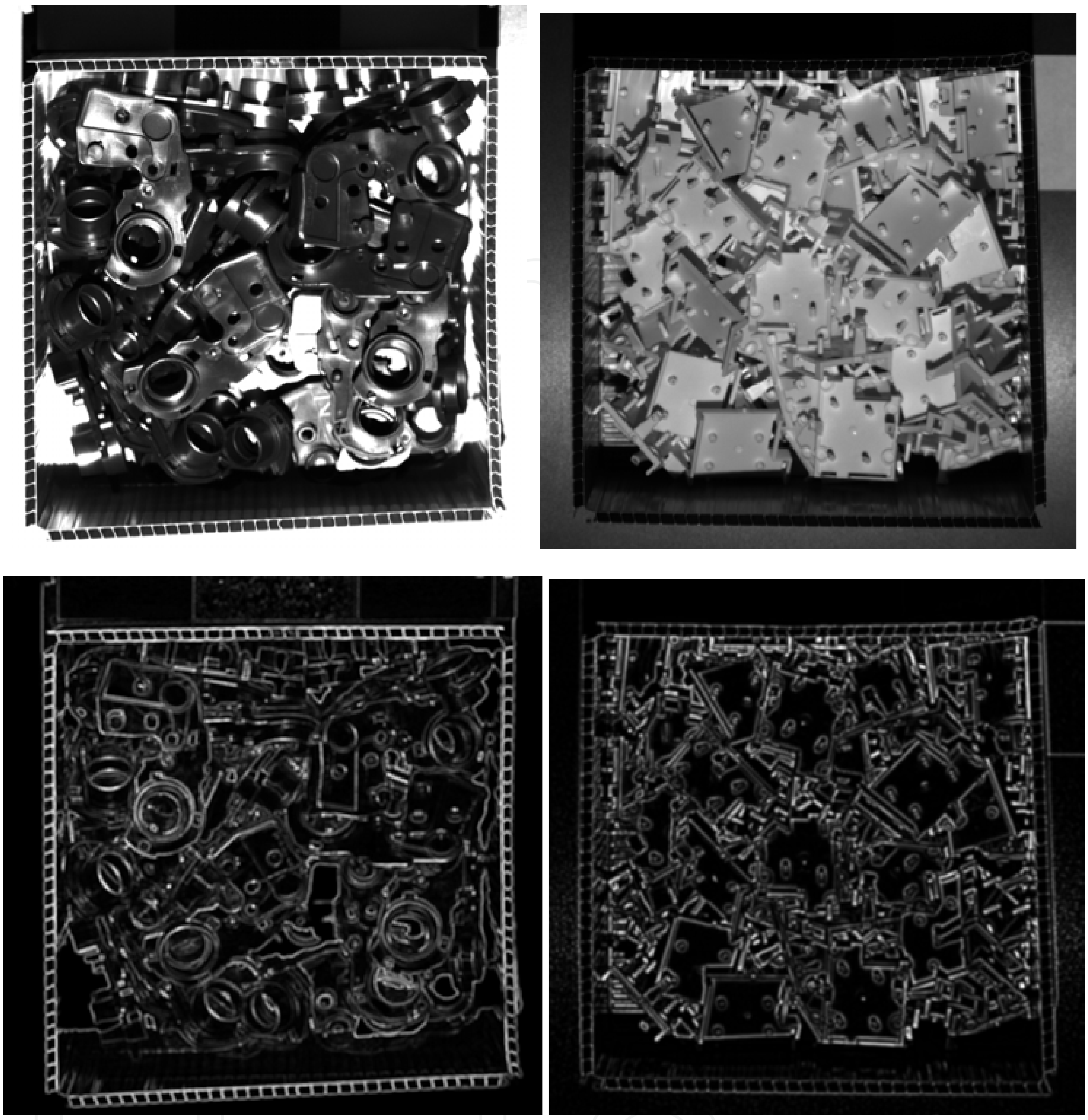


Fig. 6. Feature image obtained by edge extraction and blurring. Upper pictures are input images and lower ones are corresponding feature images as input to the classifier ensemble.

Another possible feature used for the detection task is 2.5 D map (depth map data) obtained by any three dimensional measurement (e.g., stereo vision, triangulation by structured pattern projection, TOF, etc.) method. Various features can be used, in principle, as channels of input to the ensemble classifiers. In this paper we confine to 2D edge-enhanced image as input.

4. Experiments

We used five classes of parts in printers (i.e., inkjet and laser beam printers) for training and testing. Those parts are of plastic mold and many of them are either black or white. Images of parts are taken by Canon EOS Kiss X2. Training image is taken for a single object with

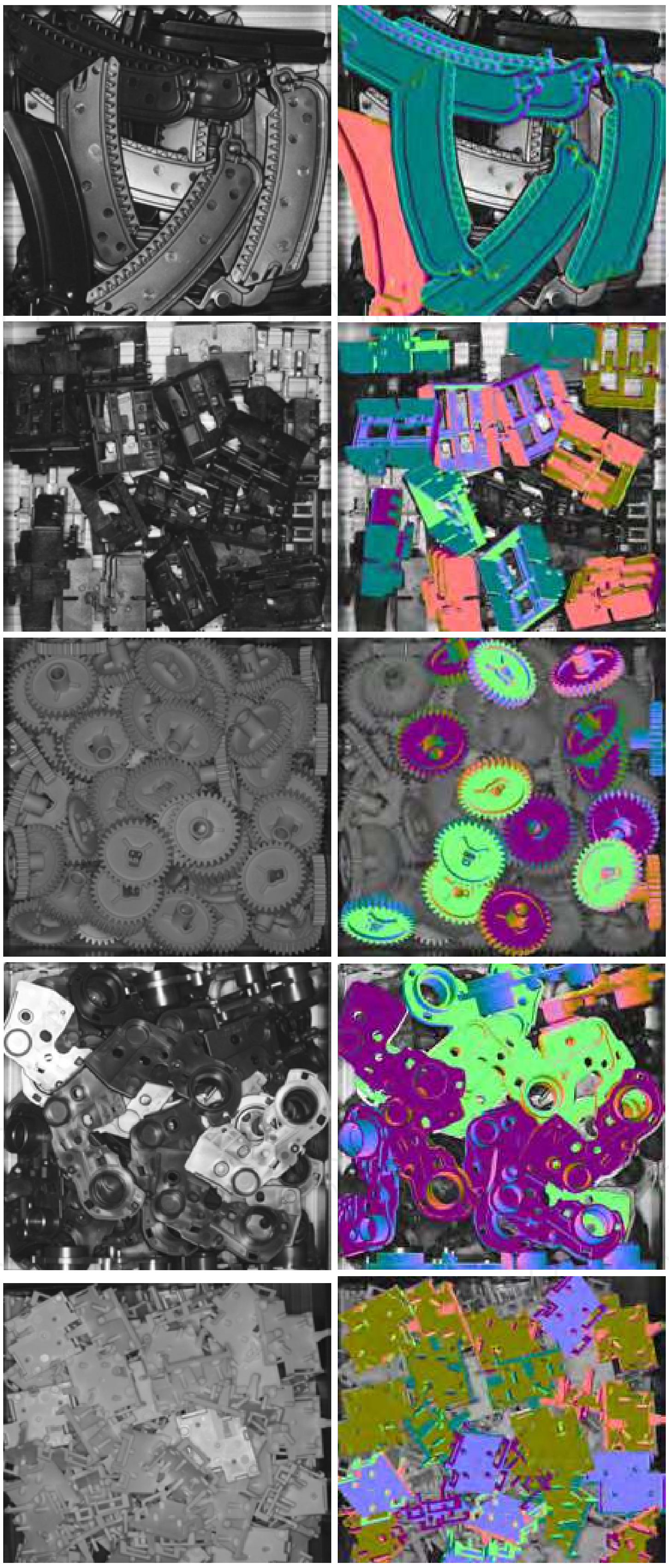


Fig. 7. Detection results for various objects. Correctly detected parts are shown by superimposed CAD image.

particular pose in a flat background (Figure 2 left). The number of pose categories is dependent on the resolution and accuracy requirement as initial estimate of pose which is given as input to the model fitting. Based on experiments, we empirically set the number of basic pose as follows. Pose categories necessary for the initial estimate in the model fitting are found to be defined by every 162 viewpoints evenly distributed on a geodesic dome and it was found necessary to discriminate poses for every 8 degree in-plane rotation, which results in total number of poses, 162 x 45 = 7620. In the training, we set 100 patches for each pose category. The size of input image is 660 x 640 pixels, and the size of patches is 50 x 50 pixels which is set constant for the five parts. Maximum depth of trees is empirically set from 15 to 30, which is dependent on the shape of parts.

Feature extraction was obtained by Laplacian filtering followed with blurring using Gaussian filter as shown in Figure 6. After the feature extraction, ground truth data given by (1) are taken for the respective patch images using queries generated by the probabilistic sampling in subsection 3.1.

4.1 Detection results

We show some detection results for piles of parts in Figure 7 obtained using the proposed SRMS as well as CMFV (Section 3) with the number of trees 32. We indicate here correct detections by superimposing CAD data of corresponding pose category. We do not use depth map data at all in the detection as well as training process.

4.2 Benchmarking

We compared the proposed method with the state-of-the-art, commercially available machine vision software (*HALCON* 9.0 produced by company *MVTec*). Technology related with the reference software can be found in Ulrich et al. (2009), which relies on edge-based fast matching scheme. Here we show some results in Figure 8. As is evident from this figure, it is very difficult for the reference software to detect and estimate 3D pose in the case of parts with white surface properties. Moreover, for black parts in Figure 1 (b) it was entirely unable to detect.

Since our method as well as the reference software *HALCON* in this comparative experiment is 2D-based, it is essentially hard to estimate rotation angle in depth. Our criterion for correct detection is based on the requirement set by model fitting process, which is given by allowable error in position and pose. Maximum allowable error for 'correct' result was given by approximately 10% of size in terms of positional error and approximately 10 deg. in terms of in-plane angle error measured in the image.

Shown in Figure 9 is a kind of 'RPC' curves for varying number of trees in the case of the piled parts shown in the upper picture in Figure 8, the reference software *HALCON* could detect up to only three parts, whereas proposed method could detect increasing number of parts on the order of 20 with growing number of trees.

For the five classes of parts shown in Figure 1 (b), the number of correctly detected parts and average detection time are as follows. It is clear that the proposed method outperforms the reference software *HALCON* in terms of precision and detection time. For fair comparison, we set the same criterion on correct detection.

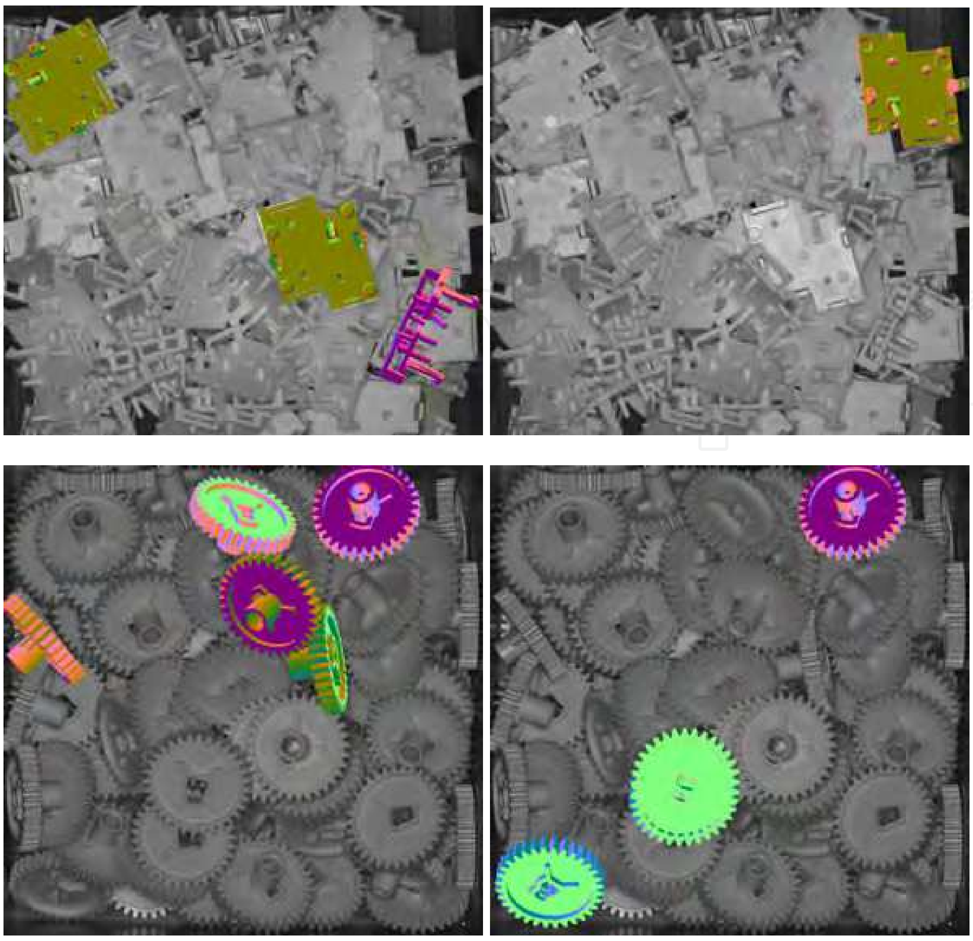


Fig. 8. Results obtained for the reference software *HALCON (9.0)*.

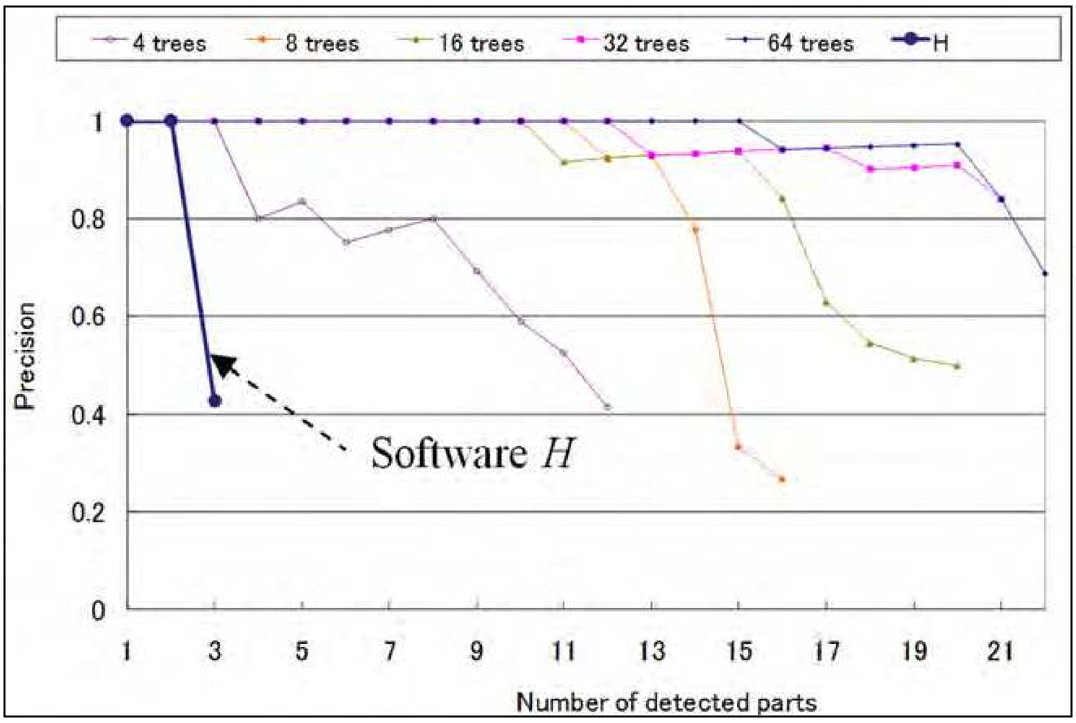


Fig. 9. RPC-like curve obtained for the reference software *HALCON (9.0)*.

	Class 1	Class 2	Class 3	Class 4	Class 5	Detection time (s)
Proposed alg.	20	10	14	10	5	5.8
Ref. Software H	3	0	5	0	0	30.4

In the table above, we show processing times as compared with the reference software (*HALCON*). It clearly shows that the proposed algorithm significantly outperforms the reference software in terms of processing speed.

5. Discussion and conclusions

In this chapter, we proposed a new algorithm based on ensemble of trees for object localization and 3D pose estimation that works for piled parts. We define pose estimation as classification problem which provides initial estimate for the subsequent model fitting processing to obtain further precise estimation.

One important aspect of object detection in the bin-picking task is that it is sufficient to localize and estimate the pose of one 'adequate' object for picking. In fact we used the number of parts detected as measure of 'recall' in the RPC-like curve (Figure 9). The proposed method significantly outperformed the state of the art, commercially available software in terms of precision and processing speed.

6. Acknowledgment

We appreciate T. Saruta, Y. Okuno, and M. Aoba for their efforts in obtaining various results.

7. References

Amit, Y. & Geman, D. (1997) Shape Quantization and Recognition with Randomized Trees, *Neural Computation*, Vol.9 No.7, pp.1545-1588.

Bosch, A., Zisserman, A., & Munoz, X. (2007) Image Classification using Random Forests and Ferns, *Proceedings of ICCV'07*

Drost, B., Ulrich, M., Navab, N., & Ilic, S. (2010) Model Globally, Match Locally and Robust 3D Object Recognition, *Proceedings of CVPR'10*

Gall J. & Lempitsky V. (2009) Class-Specific Hough Forests for Object Detection, *Proceedings of CVPR'09*

Hinterstoisser, S., Benhimane. S., & Navab, N. (2007) N3M: Natural 3D Markers for Real-Time Object Detection and Pose Estimation, *Proceedings of ICCV'07*.

Lepetit, V. & Fua, P. (2006) Keypoint Recognition Using Randomized Trees, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 9, pp. 1465-1479.

Oshin, O., Gilbert, A., Illingworth, J., & Bowden, R. (2009) Action Recognition using Randomized Ferns, *Proceedings of ICCV2009 Workshop on Video-oriented Object and Event Classification*.

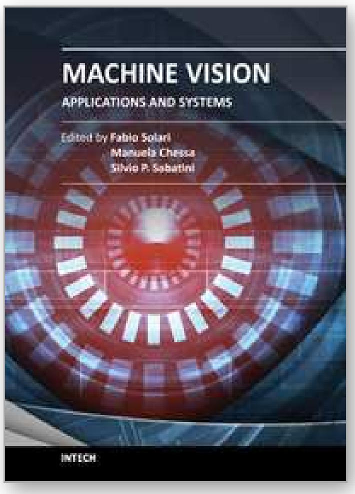
Özuysal, M., Calonder, M., Lepetit, V., & Fua, P. (2010) Fast Keypoint Recognition Using Random Ferns, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 3, pp. 448-461.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011) Real-Time Human Pose Recognition in Parts from Single Depth Images, *Proceedings of CVPR'11*

Tateno, K., Kotake, D., & Uchiyama, S. (2010) A Model Fitting Method using Intensity and Range Images for Bin-Picking Applications (in Japanese)*, Proceedings of Meeting on Image Recognition & Understanding 2010*

Ulrich, M., Wiedemann, C., & Steger, C. (2009) CAD-Based Recognition of 3D Objects in Monocular Images, *Proceedings of ICRA'09*

Yoshii, H., Okuno, Y., Mitarai, Y., Saruta, T., Mori, K., & Matsugu, M. (2010) Parts Detection Algorithm using Ensemble of Tree Classifiers (in Japanese), *Proceedings of Meeting on Image Recognition & Understanding 2010*


MACHINE VISION
APPLICATIONS AND SYSTEMS
Edited by Fabio Solari
Manuela Chessa
Silvio P. Sabatini
INTECH