http://dx.doi.org/10.5755/j01.eee.19.10.5885 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 19, NO. 10, 2013

Orientation Invariant Surface Classification Using Uncertainty Level Estimation

S. Sinkevicius[1], A. Lipnickas[1], V. Sinkevicius[2], K. Rimkus[1]
[1]*Department of Control Technology, Kaunas University of Technology, Studentu St. 48–320, LT-51367 Kaunas, Lithuania*
[2]*Department of Electrical Engineering, Kaunas University of Technology, Panevezys Institute, S. Daukanto St. 12-103, LT- 35212 Panevezys, Lithuania*
saulius.sinkevicius@stud.ktu.lt

***Abstract*—This paper proposes and describes orientation invariant surface classification system used for connectors labelling. The presented system was tested by classifying sides of 8 different electrical wire connectors for automated ink-jet print labelling. Connectors were randomly placed on the conveyor, and the system identified their visible sides regardless of the orientation and camera's viewpoint variation. All connectors in one batch were of the same type. Identified wire connector information was further fed to the industrial robot arm, which then could take free oriented connector and face it to the ink-jet printer on the required side. For classification task, four different classifiers were experimentally compared: Artificial Neural Network, Decision Tree, K Nearest Neighbours, and Quadratic Discriminant Analysis. The first order statistic and Scale-invariant feature transform were used for feature extraction from the images. The approach proposed allowed identification with 100º% accuracy depending on the selected level of uncertainty. Experimental results have shown that quantity of unrecognized samples for most connectors varied only in the range of few percents.**

***Index Terms*—Image classification, image matching, object recognition, robot vision systems.**

I. INTRODUCTION

Automatic industrial sorting systems are widely spread and used for many classification tasks. Such systems are now common in food processing [1], [2], surface defect detection [3], [4], garbage recycling [5], letter sorting [6], etc. Many of them are based on the object's surface optical properties but use a different kind of sensors such as CCD cameras, spectroscopy [7], stereo vision, infrared light, and others. Optical sensors can capture colour, shape, texture, and other optical features; in many cases, it is a multiclass [8], [9] identification problem. Optical properties depend on lighting conditions; therefore, isolating objects from environment and implementing artificial lighting sources may be one of the most important key points of the system to work properly. Such systems usually have strict requirements for their performance. They should be able to process requested number of objects fast enough and as accurate as possible.

In wiring industry, before assembling any type of electrical wire connectors, they should be first marked physically. For this task, a conveyor and ink-jet printer are used to label small batches of connectors. Each batch is composed only of the same type of connectors. This process has not been automated yet and performed manually.

It is impossible to use any orienting devices with vibratory-bowl feeders, because the number of connectors in each batch is small and each type of connector varies in size, geometry, etc. The system with robot arm and computer vision may be the solution for automation; connectors then could be reoriented fast without human interaction. Such a system should have the ability to identify the side and orientation angle of the connector viewed by a video camera. Then, a robot arm could undertake the pick-and-place task of correct positioning of a connector for ink-jet printing. The presented approach focuses on connector side identification.

To the best of our knowledge, there is no way to estimate the level of uncertainty in classification precisely. The performance of the classifier depends on data; if samples from different classes correlate, misclassification errors may occur. Although there are some attempts [10], [11] to estimate uncertainly level but those solutions still do not give 100 percent accuracy. Our approach proposed is based on idea that some sides of connectors are easily classified correctly by visual properties such as surface area or mean of greyscale intensity values. For the rest sides, we used one of the popular orientation and viewpoint invariant method - Scale-invariant feature transform (SIFT) [12].

The paper contains five main sections. In the second Section, theoretical background is overviewed. The third Section gives more information about used data and the experiments. Experimental results are presented in Section IV. Finally, conclusions are given in Section V.

II. THEORETICAL BACKGROUND

The approach suggested (Fig. 1) is based on assumption that some sides of connectors can be classified with zero misclassification error (MCE) using just a few first order statistical (FOS) features such as mean or standard deviation calculated from greyscale intensity values. FOS features are orientation invariant and fast to compute; however, they are estimated on individual pixel values, ignoring the spatial interaction between image pixels. Decision Tree (DT) was chosen as a classifier, because experiments show that this method was the fastest and still accurate enough.

Manuscript received January 9, 2013; accepted June 8, 2013.

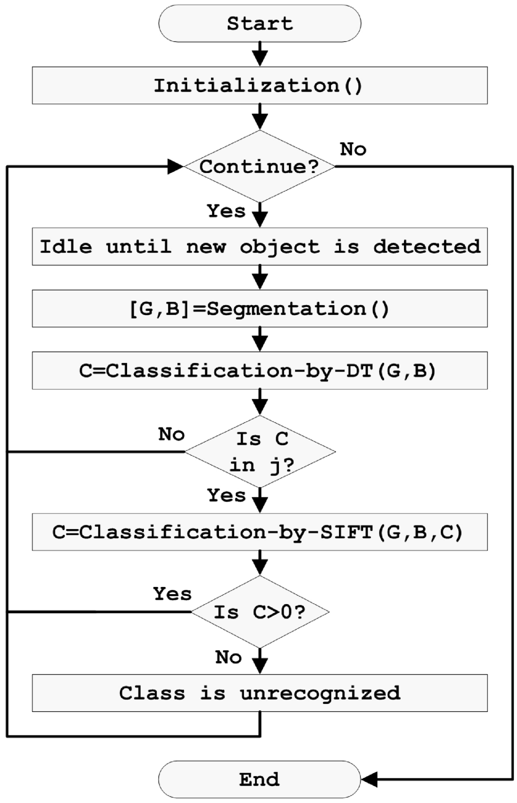


Fig. 1. The abstraction of proposed algorithm using pseudo code fragments.

Visually some sides of the same connector are like mirrored version of the same image or differ by tiny details only; therefore, FOS features are not sufficient in distinguishing these sides. In such cases, the advantage of global spatial information obtained by SIFT method is additionally applied for feature extraction.

For each connector, a subset of different features was selected by SFS (Sequential Feature Selection) [13] dimensionality reduction method. Fewer features still give the same accuracy results, but the model is smaller and classification is faster. The most useful FOS features in experiments were surface area, mean, standard deviation, harmonic mean, median, central moments, kurtosis, skewness, mode, and percentiles extracted from the greyscale images.

For the FOS features classification, four different classifiers were compared: ANN (Artificial Neural Network), DT (Decision Tree), KNN (K Nearest Neighbours), and QDA (Quadratic Discriminant Analysis). Accuracy of all the classifiers is around the same, but DT shows the best performance. MCE is evaluated using holdout method.

To make it easier to reproduce our result, the pseudo code is given. The flow of the algorithm is shown in Fig. 1 and functions are given bellow. `vl_sift()` and `vl_ubcmatch()` are functions provided by VLFeat open source library for Matlab. The first one extracts SIFT features and descriptors; the second matches SIFT descriptors of two images. The actual code was written in Matlab; therefore, pseudo code is similar to Matlab syntax and has the same or similar function names.

The initialization procedure starts all preparation needed for calculations such as: defining necessary constants; reducing data feature space; defining which classes should be joined for the best DT performance; training or loading DT; calculating orientations of each sample in the training set; extracting SIFT features and descriptors for all training set samples.

```
INITIALIZATION()
  s = CONST  // image closing size
  t = CONST  // edge detection threshold
  p = CONST  // SIFT peak threshold
  e = CONST  // SIFT edge threshold
  th = CONST // Features matching threshold
  vt = CONST // Voting threshold
  vm = CONST // Minimal matches required
  trainSet = loadTrainSetForSIFT()
  featureTypes = {area, mean, mode, …}
  reducedTypes = SFS(featureTypes)
  j = {a b; …} // joined classes
  DT = loadDecisionTree()
  F = D = F1 = D1 = angles = {Ø}
  for sample in trainSet
    angles = angles U {orientation(sample)}
    [F1,D1] = vl_sift(sample,p,e)
    F = F U {F1} // SIFT features
    D = D U {D1} // SIFT descriptors
```

In segmentation function, the image is converted to greyscale and binary images. The image is segmented firstly by detecting edges; secondly, closing edges morphologically; and finally filing any holes in that segment. Any other pixels outside the detected object are labelled as a background and are replaced with white colour.

```
SEGMENTATION()
  I = getFrame() // get image from video camera
  G = 0.299I(R) + 0.587I(G) + 0.114I(B) // to greyscale
  B = fillHoles(close(edges(G,t),s)) // find object
  G(find(B == 0)) = 255 // background is white
  return G, B // greyscale and binary images
```

Classification by DT is performed on the features extracted by SFS. The result of classification by DT is a class label, i.e., “1-front”, “2-bottom”, “3-left”, and etc. Variable *c* represents a class as a result from the classifier. If `c` belongs to joined classes list `j` defined in initialization function, the result from DT is fed to SIFT method for a validation.

```
CLASSIFICATION-BY-DT(G,B)
  idx = G(find(B < 255)) // values of all object pixels
  X = {Ø}
  for featureType in reducedTypes // create a vector
    X = X U {getFeature(featureType,idx)}
  return classify X vector using DT
```

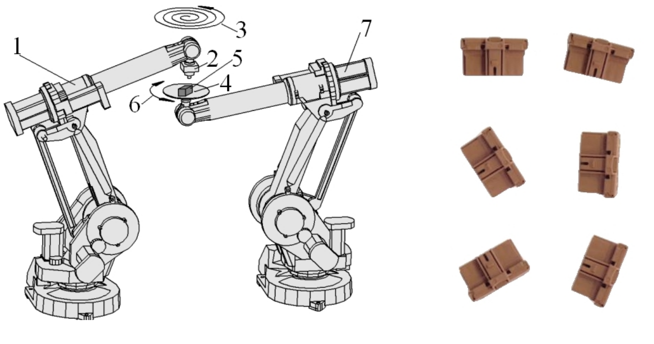
The last piece of pseudo code finds samples from training set with similar angle but only in the case if `c` from DT belongs to the list `j`. Quantity of matches is used for voting. By using coarse Hough space [16], we find similar SIFT features by orientation, scale, and location. Then, only features from the largest bin of Hough space are selected. This allows excluding many incorrect matches and is essential in the solution proposed.

```
CLASSIFICATION-BY-SIFT(G,B,C)
  o = orientation(B)  // calculate orientation of object
  d = f2 = d2 = {Ø}
  for angle in angles // find distances to samples
    d = d U {dist(o, angle)}
  items = find(d < 45); // if item dist < 45
  [f2,d2] = vl_sift(G,p,e)
  classes = get subset from 'j' with C in it
  for cl in classes    // for each class
    for it in items   // for each item
      [m,s] = vl_ubcmatch(D(cl,it),d2,th) // m - matches
      f1 = F(cl,it)
      indices = useCoarseHoughSpace(f1(m),f2(m))
      R(class) = R(class) + count(indices) //sum matches
  [Vmax Imax] = max(R) // value and index
```

```
R(Imax) = 0
[Vmin Imin] = max(R) // second by size
if (Vmax/Vmin <= vt) || (Vmax+Vmin < vm)
  return 0   // unrecognized class
else // uncertainty level is low
  return Imax // index of class
```

III. DATA

The core idea of experimental setup is to train the system and to use industrial robot arms in manufacturing process. Two robot-arms (see Fig. 2) were used; however, it's enough to have one, because the second robot can be easily replaced by a simple drive in training phase. By using robots, we had advantages such as automatic sample labelling, fast database preparation, and the ability to gather a decent number of variations for any connector type. This setup provides the ability to prepare the system for a new connector type extremely fast.



a) b)

Fig. 2. (a) Experimental setup: 1 – first articulated robot, 2 – video camera, 3 – spiral trajectory of camera, 4 – plate, 5 – connector for training; 6 – circular movement of plate, 7 – second robot; (b) A few examples of the back side of one type connector.

The first robot follows a spiral scanning path and the second one rotates a plate with a connector on it. The data acquiring algorithm runs 3 minutes at 10 fps frame rate. After excluding repetitive samples, we got around 1500 unique images per each side of every connector. Connectors have 4 to 6 sides; therefore, a database for each connector has 6000 to 9000 samples.

Eight connector types were chosen for experiments (Fig. 3). They differ by shape, number of sides, size, colour, and texture. During training, arms of both robots move along the pre-defined path until a significant number of variations are acquired.

Fig. 3. Connector types used in experiments; back sides are shown.

The white background of each acquired image had some gradient noise which was removed using edge detection and morphological operations. Then each non-white segment in images was recognized as an area with a connector.

IV. EXPERIMENTS

A. *Classification Using First Order Statistical Features*

Training sets for each connector contain 20 percent of random picked samples and the rest 80 percent is used to calculate MCE. This 20/80 ratio threshold for holdout method was found empirically by comparing accuracies for all connectors. Table I provides the number of sides for each connector and number of features selected up by the SFS method.

TABLE I. NUMBER OF SIDES AND FEATURES FOR EACH CONNECTOR.

	C1	C2	C3	C4	C5	C6	C7	C8
Sides	6	5	4	6	6	6	5	4
Features	3	7	3	5	2	6	5	5

Four classifiers were compared by their MCE and performance (Table II). Results show that all the classifiers give around the same accuracies, but the fastest classifier (DT) was chosen for further experiments. On average the one image computation times are: QDA – 0.0069 s, KNN – 0.0215 s, ANN – 0.0221 s, and DT – 0.0046 s. The overall performance for a connector mostly depends on connector's size and number of features extracted; it's enough to use 2 to 7 features to get high enough accuracy. All experiments were performed on an Intel Pentium 4 3 GHz processor with 2 GB RAM.

TABLE II. CLASSIFIER COMPARISON BY MCE.

Connector	QDA, %	KNN, %	ANN, %	DT, %
C1	0.01	0.15	**0.00**	0.07
C2	2.82	**1.90**	3.08	4.87
C3	0.00	0.00	0.00	0.00
C4	1.38	**0.13**	0.26	1.11
C5	0.08	0.00	0.00	0.29
C6	6.46	**1.42**	2.69	2.39
C7	2.55	1.32	**1.03**	1.40
C8	13.67	12.04	**10.63**	12.44

Whatever classifier was used, some sides had zero MCE. The remaining of the sides had some misclassified cases but with a strict pattern (Table III). For example, front and back sides of C1 connector are very similar as well as their left and right sides; therefore, FOS features are not enough to distinguish them. Other sides for this connector are well identified using only a few FOS features by any classifier.

TABLE III. EASY AND HARD SIDES FOR RECOGNITION.

Connector	Classes without errors	Misclassified classes
C1	top; bottom	front/back; left/right
C2	left; top; bottom	front/back
C3	front; back; top; bottom	-
C4	front; back; top; bottom	left/right
C5	front; back; left; right	top/bottom
C6	top; bottom	front/back/left/right
C7	right; top; bottom	front/back
C8	-	front/back; top/bottom

B. *Classification by SIFT*

Figure 4 shows a relation used for voting. Orientation is expressed as an angle between 0 and 90 degrees. It is measured between the *x*-axis and the major axis of the ellipse that has the same second-moments as the region. All training and testing data have been used for the charts.

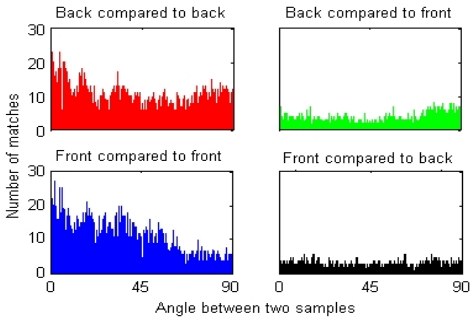


Fig. 4. Two samples of the first connector type were picked, one as a front side example and another as a back side example. Each sample was matched by SIFT features to all samples of the same side (front to front; back to back) and to all samples of very similar another side (front o back; back to front). The quantity of matches for each pair was computed.

As it is seen from Fig. 4, if the number of matches is computed on pair of images from the same class, the number of matching points is much bigger comparing to number of matches obtained for the images from different classes. As well as it is obvious to note that the smaller angle of displacement is present between pair of images, the bigger number of matches could be found within the same class data pairs. This relation is valid for all connectors used in experiments; we use it for voting in classification. SIFT may be more sensitive to viewpoint changes than to orientation changes, and number of matches may be lower if the viewpoint is significantly different.

Sometimes the proposed system gets into a confusing situation by counting equal numbers of matched features for different classes of connectors. In order to reduce MCE the minimal threshold of matched features is used.

So, it is possible to use the matches' ratio threshold and if the ratio is low, the image is left unrecognized. A high ratio means high reliability, while a low ratio means uncertainty or even misclassification. But it is better to leave the image unrecognized than to identify it incorrectly. Table IV shows connector recognition results obtained with the applied thresholding technique. For all connectors the MCE is equal to zero but because of the applied threshold some acquired images are left unrecognized.

TABLE IV. FINAL ACCURACY RESULTS.

Connector	Threshold	Unrecognized	Correct	MCE, %
C1	1.2	116	8884	0
C2	1.3	88	7412	0
C3	-	-	6000	0
C4	1.7	1248	7752	0
C5	1.1	0	9000	0
C6	1.3	394	8606	0
C7	1.4	428	7072	0
C8	1.3	1790	4210	0

V. CONCLUSIONS

A novel approach for electrical wire connector sides' classification with uncertainty level estimation is introduced. The uncertainty level estimation proposed allows identifying connector sides with 100 % accuracy, even if they have very similar visual appearance. The proposed system is designed for robot-arm pick and place autonomous operation. The first order statistical features were extracted by SFS method from gray scale images captured by the video camera. Four different classifiers were experimentally compared for connectors' data analysis: ANN, DT, KNN, and QDA. The decision tree classifier has shown the best performance.

In order to reduce the MCE, the thresholding technique were introduces on matched points found by SIFT method. Experimental investigation have shown that the number of unrecognized samples slightly increased but overall system performance has lead to zero misclassification error.

In the presented work we have also shown that the samples of connectors' narrow sides have a high unrecognising rate, but this result is not critical in real application, because the connector most likely will land on the bigger flat side in front of the camera as well as the acquired instance will be rejected if the quantity of matches will be lower than the predefined threshold.

REFERENCES

[1] T. Pearson, D. Brabec, S. Haley, "Color image based sorter for separating red and white wheat", *Sensing and Instrumentation for Food Quality and Safety*, vol. 2, pp. 280–288, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1007/s11694-008-9062-0

[2] M. Omid, A. Mahmoudi, and M. H. Omida, "Study on Fresh Fish Sorting Techniques", *Expert Systems with Applications*, vol. 37, pp. 7205–7212, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.04.008

[3] W. Guifang, K. Hoonsung, J. Seyoung, X. Ke, X. Jinwu, "Design of online surface inspection system of hot rolled strips", in *Proc. IEEE Int. Con. Automation and Logistics*, Sept. 2008, pp. 2291–2295.

[4] J. Guzaitis, A. Verikas, "An efficient technique to detect visual defects in particleboards", *Informatica*, vol. 19, pp. 363–376, 2008.

[5] S. Shahrani, H. Aini, A. W. Dzuraidah, M. M. Mohd, R. Suzaimah "Support Vector Machines for Automated Classification of Plastic Bottles", in *Proc. 6th Int. Colloquium. Signal Processing and Its Applications CSPA*, 2010, pp. 1–5, 21–23.

[6] E. Normanyo, D. Ayim, A. Isaac, "Designing of a Letter Sorting Machine for the Regional Post Offices in Ghana", *Engineering and Applied Sciences*, vol. 4, pp. 1–13, Aug. 2009.

[7] M. O'Farrell, E. Lewis, C. Flanagan, W. B. Lyons, N. Jackman, "Design of a system that uses optical-fiber sensors and neural networks to control a large-scale industrial oven by monitoring the food quality online", *IEEE. Sensors Journal*, vol. 5, pp. 1407–1420, Dec. 2005. [Online]. Available: http://dx.doi.org/10.1109/JSEN.2005.858963

[8] M. Fuerst, C. Woegerer, G. Kronreif, I. Hollaender, H. Penz, "Intelligent high-speed, high-variant automation of universal coin sorting for charity organizations", in *Proc. IEEE Int. Con. Robotics and Automation*, 2006, pp. 303–308.

[9] A. Ramanan, P. Ranganathan, M. Niranjan, "Speeding up multi-class texture classification by one-pass vocabulary design and decision tree", in *Proc. IEEE Int. Con. Industrial and Information Systems (ICIIS)*, 2011, pp. 255–260.

[10] A. Buschermohle, N. Rosemann, W. Brockmann, "Stable Classification in Environments with Varying Degrees of Uncertainty", in *Proc. Int. Con. Computational Intelligence for Modelling Control & Automation*, 2008, pp. 441–446.

[11] K. E. Graves, R. Nagarajah, "Uncertainty Estimation Using Fuzzy Measures for Multiclass Classification", *IEEE Trans. Neural Networks*, vol. 18, pp. 128–140, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1109/TNN.2006.883012

[12] D. G. Lowe, "Object recognition from local scale-invariant features", in *Proc. IEEE Int. Con. Computer Vision*, 1999, vol. 2, pp. 1150-1157.

[13] R. Kohavi, and G. H. John, "Wrappers for Feature Subset Selection", *Artificial Intelligence*, vol. 97, pp. 273–324, 1997. [Online]. Available: http://dx.doi.org/10.1016/S0004-3702(97)00043-X

[14] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, 1986, pp. 679–698. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.1986.4767851