Object Recognition and Tracking for Indoor Robots using an RGB-D Sensor

Lixing Jiang, Artur Koch and Andreas Zell

Computer Science Department, University of Tuebingen, Sand 1, D-72076 Tuebingen, Germany {lixing.jiang,artur.koch,andreas.zell}@uni-tuebingen.de

Abstract. In this paper we extend and generalize our previously published approach on RGB-D based fruit recognition to be able to recognize different kinds of objects in front of our mobile system. We therefore firstly extend our segmentation to use depth filtering and clustering with a watershed algorithm on the depth data to detect the target to be recognized. We forward the processed data to extract RGB-D descriptors that are used to recoup complementary object information for the classification and recognition task. After having detected the object once, we apply a simple tracking method to reduce the object search space and the computational load through frequent detection queries. The proposed method is evaluated using the *random forest* (RF) classifier. Experimental results highlight the effectiveness as well as real time suitability of the proposed extensions for our mobile system based on real RGB-D data.

Keywords: RGB-D, Mobile Systems, Segmentation, Classification

1 Introduction

Object recognition deals with the task of identifying visible objects in an image from a previously defined set of known object candidates. Typical object recognition systems usually consist of two basic stages, the offline training and the (online) evaluation. In the training stage, various object data samples are collected and processed to build an object representation using descriptors that characterize the objects. Using the processed object features, a classifier can be trained to capture the differences of the features for single objects and thus to be able to distinguish between the different classes or instances of objects. The final recognition, which may be generative or discriminative, then is carried out by matching the data samples of the objects to be recognized with the stored object classifier models in the previously generated database.

The major focus of this work is a robust real time recognition system for fruits and small textured objects tailored to meet the requirements of our mobile robot equipped with a tray and the Microsoft Kinect RGB-D sensor. Based on our previously published system for fruit recognition [1], we introduce our recent extensions and the resulting pipeline depicted in Fig. 1. The contribution of this work is threefold. First, the segmentation process is adapted to be



Fig. 1. Block diagram of recognition framework.

able to deal with small fruits or textured objects by using depth filtering and clustering. This is due to the more general demand for recognition of textured or inhomogeneously colored objects, as opposed to our previous work, where we were assuming mostly uniformly colored fruits and thus were successfully able to use the homogeneity in the segmentation as well as recognition process. Since a suitable choice of the object representation has a major impact on the recognition accuracy, we secondly propose an adapted and compact representation of essential RGB-D characteristics based on the *color layout descriptor* (CLD) [2,3]. We evaluate the proposed descriptor combination with different machine learning methods and compare the extensive results to our previously published approach. Finally, we integrate Continuously Adaptive Mean Shift (CamShift) [4] into our system to be able to continuously track the position of the target, circumventing the need for frequent detection and recognition queries that would introduce a much higher computational load to the system. The tracking may be used on our mobile system to follow a previously recognized and moving object.

The remainder of the paper is structured as follows. Section 2 briefly reviews the related work on object recognition. Afterwards, we present the different stages of our pipeline in Sect. 3. In Section 4 we investigate the performance of different machine learning algorithms based on the proposed extensions. Finally, we draw conclusions in Sect. 5.

2 Related work

Region-based object recognition usually includes several steps: (1) segmentation, (2) feature extraction, (3) training and (4) testing. Segmentation is the initial step in most vision based applications. Existing segmentation approaches can be generally categorized into grid-, clustering-, contour-based and statistical algorithms. Khan et al. [5] utilized a grid-based segmentation approach for visual terrain classification with quadrotors. Although suited for real-time application, grid-based approaches usually brake up the object's semantic representation regarding shape and size. Lai et al. propose a combination of *Random Sample Consensus* (RANSAC) plane fitting with the adaptive Gaussian mixture model



Fig. 2. Target object detection and tracking.

for segmentation in [6]. This statistical algorithm is calculated by expectation maximization (EM) but features high runtime complexity. Gu et al. [7] use a hierarchical segmentation to construct a region tree for recognition, which is based on an oriented watershed transform and achieves good segmentation results.

To get an appropriate object representation, local feature extraction based on histograms of gradient directions such as SIFT and SURF [8,9] are widely used for object recognition. For advanced 3D data understanding, 3D shape contexts [10] and spin images [11] are often utilized for 3D feature extraction. Karpathy et al. [12] recently introduced 3D shape measures for discovering object models from 3D meshes of scenes in indoor environments. They are able to analyze a whole RGB-D scene in less as one minute and achieve remarkable results in object discovery.

Vision-based region descriptions have proven to be able to efficiently capture and represent characteristics of real world objects and therefore are comprehensively utilized in recognition and classification. More advanced hybrid approaches using 2D/3D descriptors have for example been presented in [13–15]. Bo et al. [13] have proposed a kernel descriptor, which combines a group of significant RGB-D features capturing the objects' 3D model information such as depth edges, 3D size and shape. Fischer et al. [14] present a novel layout of the point feature descriptor from RGB-D data. Socher et al. [15] propose a combination of convolutional and recursive neural networks for deep learning. We follow the paradigm of hybrid descriptors and utilize RGB based descriptors from the MPEG-7 standard [2,3] as well as the depth-based shape measures introduced by Karpathy et al. [12].

3 Overview of proposed method

The pipeline of our recognition framework consists of two consecutive stages as illustrated in Fig. 1. In the detection and segmentation stage, each image is divided into foreground, unknown and background regions using depth data. These regions are forwarded to a marker-controlled watershed algorithm using depth data for object segmentation.



Fig. 3. Depth mapping to foreground, background and unknown regions: closest point (CP) and its local neighborhood (ε) mark the foreground (green), T marks the distance to the tray.

In the second stage, the previously trained classifier is used to recognize the target object based on a set of feature descriptors (color, texture and shape) extracted from the segmented object region. After a successful initial object detection and recognition, the object may optionally be tracked using CamShift in the RGB-D domain, as shown in Fig. 2. In the following we describe the proposed methods to solve the single tasks of the system at hand.

3.1 Target region detection and segmentation

The goal of the detection and segmentation process is to select possible object region candidates from an intricate environment represented by the RGB-D data stream. It first significantly reduces the underlying search space, and second, filters out unrelated information.

We use the Kinect for Windows mounted above a tray on the robot to capture raw RGB and depth data streams containing the objects to be classified. In our previous approach, the application scenario was limited to fruits placed on top of the tray. We therefore derived and applied simple constraints based on depth data to remove the tray as well as the area behind the tray from the image samples. Afterwards we used a marker-controlled watershed algorithm on the masked RGB data to identify the regions of interest. For the work at hand we loosen the previously made assumptions and constraints to be able to identify small textured objects instead of fruits only, which may be lying on top of the tray or moving in front of the robot. In the following we describe the changes made to the single stages of the segmentation to account for the new application scenario.

Depth clustering and filtering: For every frame, we first detect the closest point inside the depth image. To obtain an initial region of the object to be identified, we apply simple depth thresholding around this point. The obtained mask represents the *foreground* marker and is retained for further processing. We consider all objects being further away than the tray as invalid candidates, and therefore mark those pixels as the *background* region. All not yet marked pixels are labeled *unknown* (see Fig. 3) and represent the major area of interest that should be processed in the next segmentation step.

Depth-marker-controlled watershed transform: To generate satisfying segmentation results, we pass the depth-masks obtained from depth clustering and filtering to the watershed algorithm. Opposed to our previous approach applying watershed transform to RGB data, we now utilize depth data during segmentation. This is due to the fact that the assumption of homogeneously colored objects does not hold for common objects, and thus delivers inadequate segmentation results. Using the supplied depth markers, the watershed transform produces closed object contours through sequential flooding of the marked foreground region. Additionally, it requires low computational times as compared to other more sophisticated vision-based segmentation methods. After successful detection and segmentation, we can build the model for tracking as well as extract the feature descriptors for the classification task.

3.2 Feature extraction and region description

Feature extraction should be able to efficiently extract and thus characterize the important parts from sample images such that the images may be successfully recognized based on the chosen feature set. We reevaluate the performance of the *scalable color descriptor* (SCD) [3] on the new set of objects and extend our descriptor set with the color layout descriptor [3] to fit the demands of the new application scenario.

The color layout descriptor is designed to capture the spatial distribution of the dominant color inside an object region. Featuring resolution-invariance and compactness, it is particularly useful for video segment identification and image retrieval. In the luminance, blue and red chrominance (YCbCr) color space, this descriptor is obtained by applying the *discrete cosine transform* (DCT) on a 2D array of locally representative colors. The input picture (on RGB color space) is divided into 64 (8x8) blocks to maintain invariance to resolution or scale. By averaging RGB pixel values inside each block, a new set of corresponding RGB values is obtained. The mean color values are transformed from RGB to YCbCr and each channel is filtered by applying the DCT. Using the analogy of the YCbCr color space to the human visual system, chroma subsampling can be applied to the less relevant Cb and Cr channels to reduce the representation size. Additionally, the DC as well as the low-frequency AC coefficients are prioritized since high frequency components are considered to carry less information. The three sets of 64 DCT coefficients are therefore zigzag scanned and non-linearly quantized to group and order low-to-high frequency components. Two cut-off parameters influencing the final representation size may be chosen for the luminance as well as chrominance channels, respectively. The dimension of the descriptor thus may be tuned to the user's preference, whereas a total of 12 coefficients (Y, Cb, Cr) = (6, 3, 3) is recommended in [3].

Additional descriptors capturing relevant characteristics of the images are chosen as follows (cf. [1]). First, *texture* is additionally captured by the edge histogram descriptor [3], which is a useful texture representation if the texture is not homogeneous. And second, *contour and 3D shape* features are extracted by image moments and four intrinsic 3D shape measures [12]: compactness, symmetry, smoothness and local convexity. We refer the interested reader to [1], [12] and [3] for further details on the chosen descriptors.



Fig. 4. Same objects under varying poses and lighting conditions: daylight (top row), evening light (middle row) and artificial light in the night (bottom row).

3.3 Dataset

We extended our fruit RGB-D dataset [1] consisting of 14 fruit instances with 6 objects: bear, cucumber, minion Dave, garlic, pink and black wallet. The new dataset builds a set of approximately 300 RGB-D sample images for each of a total of 11 categories under three different lighting conditions that include day-light, dim light and artificial light in the night. Fig. 4 shows some image samples for the three different lighting conditions. It can be seen that color, brightness or background vary greatly for the same object. We furthermore collected object samples featuring random poses to be able to evaluate the robustness of our approach in regard to pose changes. Due to diverse RGB and depth camera specifications, it is essential to synchronize and align the respective data streams. We therefore use the transformation implemented in hardware within the Kinect for alignment and group consecutive frames in a prefiltering step. Ultimately, we generate a training data set consisting of overall 3,340 aligned RGB-D samples with a resolution of 640×480 pixels.

4 Experimental results

The experimental platform used for our work is a Scitos G5 service robot, equipped with a laser-scanner for Monte-Carlo-based self-localization and an integrated PC for on-board processing. An additional touchscreen is used for IOtasks and human-machine interaction. For our experiments, we use a gray plastic tray on the robot as the major experimental area for sample object placement. It is mounted at a height of approximately 0.6 m and is continuously monitored by a Microsoft Kinect at a vertical distance of approximately 0.5 m above the tray pointing downwards. The RGB-D sensor concurrently reports both color and depth images at a resolution of 640×480 pixels with 30 frames per second. The specific setup makes it necessary to utilize the Kinect for Windows¹, which,

¹ http://www.microsoft.com/en-us/kinectforwindows/

7



Fig. 5. Segmentation results for different object samples, from left to right: small bear, cucumber, Minions Dave and pink wallet. Rows from top to bottom represent raw RGB data (1), raw depth data (2), marker-controlled watershed results with RGB (3) and depth (4). Bottom row (5) represents the point cloud obtained from segmentation.

in contrast to the off-the-shelf Kinect XBox 360 sensor featuring a minimum distance of 0.8 m, supports the so-called near mode. With near mode enabled, the Kinect for Windows provides depth data for objects at a minimum distance of 0.4 m without loss in precision. Our mobile system thus serves as an intuitive platform for object placement and furthermore is able to provide us with the desired image samples (see Fig. 4) for our experiments.

In the following, we evaluate the two major pipeline stages inside our object recognition. Additionally, we present extensive experimental results of the proposed single as well as compound descriptors combined with various machine learning algorithms.

4.1 Segmentation

We first wish to investigate the influence of the new object samples in regard to the previous watershed algorithm based on color. Fig. 5 shows some results of the proposed depth-based watershed algorithm with depth filtering (4th row from top) in comparison to our previous segmentation used for fruit recognition (3rd row). As expected, the color-based watershed segmentation is not able to deliver adequate segmentation results for the newly added textured objects while still performing well for new homogeneously colored samples. The depth-based



Fig. 6. Random forest classification accuracy vs. model building time with CLD on different scales.

watershed transform on the other hand delivers robust and satisfying results for all samples. A further advantage is, that the computational complexity decreases slightly while preserving segmentation performance. This is due to the fact, that we replaced the previously necessary RANSAC plane fitting by the proposed simple depth filtering based on thresholding only. Throughout our experiments, object data samples were segmented in 8 ms on average, thus making the proposed detection and segmentation approach highly suitable for our real time scenario.

4.2 Descriptors and Classifiers

We analyse the performance of several classifiers based on the introduced descriptors and compare the results with our previously proposed descriptor *SESH* (cf. [1]). We therefore automatically precompute segmented RGB-D object samples using the proposed detection and segmentation approach. The segmented samples are used for feature extraction and then supplied to the machine learning platform Waikato Environment for Knowledge Analysis (Weka) [16] to train and evaluate various classifiers with different descriptors. We evaluate six machine learning algorithms, namely Native Bayes (NB), Sequential Minimum Optimization (SMO), k-Nearest Neighbor (KNN), Bagging based on *REPTree*, Decision Tree (DT) and Random Forest (RF).

Based on the discrete cosine transform, the CLD descriptor features various lengths of the YCbCr DCT-coefficients. Due to the zigzag scanning pattern, the different scales may be set to 3, 6, 10, 15, 21, 28 to 64, respectively. We investigate the influence on the classification performance using differently scaled CLD descriptors. Fig. 6 shows the results obtained from RF classification on our dataset. As can be seen, on the one hand, the recognition accuracy rises with higher dimension of the CLD descriptor to up to 83.69% at a scale of 58, i.e. (Y, Cb, Cr) = (28, 15, 15). On the other hand, higher values than 58 do not introduce a significant accuracy increase, despite the cost of higher computation

| | CLD scale | | | | |
|------------|-----------|-------|-------|-------|-------|
| Descriptor | 12 | 21 | 58 | 120 | 192 |
| SCD | | | 96.1 | | |
| CLD | 53.32 | 68.82 | 83.69 | 86.33 | 86.78 |
| SCD_CLD | 97.93 | 98.17 | 98.56 | 98.47 | 98.1 |

Table 1. Random forest classification accuracy of color descriptors

times (due to proportionality to the descriptor size). If computation times are of high importance to the application, a value of 58 seems a good choice.

In the next experiment we wish to compare the performance of the two color descriptors CLD and SCD and the combination of both. We choose the SCD with 256 elements, since it delivered significantly higher accuracies with negligible runtime penalty compared to the lower scales. The results in Table 1 show that the SCD solely already achieves a very high accuracy (96.1%) using the RF classifier on our dataset. As also shown in Fig. 6, using higher scales of the CLD descriptor produces better results, which surprisingly is not strictly the case if both descriptors are combined in our experiments. This result might be due to an overlap of single features from SCD and CLD, whereas the CLD high-frequency components seem to introduce redundant or even contradicting object characteristics in regard to the SCD descriptor. The compound color descriptor (SCD_CLD) achieves its peak performance of 98.56% with CLD using 58 elements, and features a slightly improved accuracy (2.46%) when compared to the SCD solely.

To further analyze the performance of the texture descriptor as well as the shape descriptors, we run experiments with various classifiers and different descriptor combinations. We refer to the 3D shape measures (**3DSM**) [12] and compound descriptors as follows:

 $\begin{aligned} \mathbf{3DSM} &= \{\text{Compactness, Symmetry, Local Convexity, Smoothness}\} \\ &= \{Co, Sy, LC, Sm\} \\ \mathbf{SH} &= \{Hu7\} \cup \mathbf{3DSM} = \{Hu7, Co, Sy, LC, Sm\} \\ \mathbf{SESH} &= \{SCD, EHD, Hu7, Co, Sy, LC, Sm\} \\ \mathbf{SCESH} &= \{SCD, CLD58, EHD, Hu7, Co, Sy, LC, Sm\} \end{aligned}$

Fig. 7 and Table 2 depict the results obtained from 10-fold cross validation on our dataset. First, the random forest classifier always delivers best results when compared to other machine learning methods, except for the EHD descriptor, where kNN slightly outperforms RF. This finding is similar to our last results obtained on the fruits only dataset. Nevertheless RF shows to be a solid and accurate choice throughout our experiments.

Furthermore, it can be seen, that texture (EHD) and shape information (SH) solely do not deliver good results, which may be due to the low resolution of the Kinect especially for distant objects and the varying lighting and pose conditions inside our dataset, which also seems to be valid for the shape features. Although

10 L. Jiang, A. Koch and A. Zell



Fig. 7. Classification accuracy of several single and compound descriptors.

| Accuracy(%) | NB | SMO | KNN | Bagging | DT | RF |
|-------------|-------|-------|-------|---------|-------|-------|
| SH | 74.58 | 48.84 | 82.19 | 89.98 | 82.42 | 93.24 |
| EHD | 49.44 | 73.48 | 84.80 | 65.21 | 51.97 | 82.58 |
| SCD | 79.09 | 91.08 | 94.71 | 92.43 | 91.38 | 96.10 |
| SESH | 87.92 | 97.66 | 98.00 | 96.49 | 94.35 | 98.01 |
| CLD58 | 61.07 | 69.03 | 80.17 | 78.04 | 77.68 | 83.69 |
| SCESH | 89.52 | 99.01 | 98.65 | 96.64 | 95.35 | 99.58 |

Table 2. Classification accuracy of compound descriptors on our dataset.

our previously proposed descriptor (SESH) [1] suffers a minor loss in precision (98.01%) when compared to previous results obtained (99.36%) on the fruit dataset, it still performs better as most other candidates. By combining the prior with the CLD descriptor with 58 elements, SCESH achieves the best performance with a 99.58% recognition rate inside our dataset. Again, the best performance is achieved using a Random Forest classifier.

Finally, we want to examine the runtime performance of the discussed descriptors. We therefore evaluate the times taken for feature extraction of each single descriptor on our dataset. Table 3 shows, that the RGB based descriptors are highly suitable for real time applications regarding runtime performance. Shape measures, on the other hand, introduce a high computational load to the recognition system, especially if symmetry and smoothness are taken into account. This is primarily due to the more costly operations on RGB-D point cloud data, which are necessary for feature extraction.

To make our application suitable for real time, we examine the effects of using compound descriptors without the costly 3D shape measures on the classification accuracy in our last series of experiments. We therefore define two new candidate

| Descriptor | SCD12 | 8 S | CD256 | 56 CLD | | EHD | | Hu7 |
|----------------|-------|-------------|-------|----------|--|----------------|--|------------|
| Time (ms) | 52.5 | | 52.8 | 14.9 | | 9.8 | | 1.4 |
| | | | | | | | | |
| Shape Measures | Com | Compactness | | Symmetry | | Loc. Convexity | | Smoothness |
| Time (ms) | |).03 | 40 | 1.5 | | 13.3 | | 400.8 |

Table 3. Descriptor extraction runtime

Table 4. Classification accuracy with random forest

| Descriptor | SESH | SCESH | SCEHu | SCEHuCL |
|--------------|-------|-------|-------|---------|
| Accuracy (%) | 98.01 | 99.58 | 99.31 | 99.49 |

descriptors as follows:

$SCEHu = \{SCD, CLD58, EHD, Hu7\}$ $SCEHuCL = \{SCD, CLD58, EHD, Hu7, Co, LC\}$

The results in Table 4 show, that on our dataset very good recognition results of 99.31% can already be achieved using feature descriptors based on RGB only. The additional 3D shape measures only feature a minor improvement, whereas introducing considerably higher computational load on the mobile system. Nevertheless, best results may be obtained by using the combined RGB as well as depth data descriptors if the runtime performance is of minor importance to the application.

5 Conclusion

In this paper we proposed an object recognition system being able to detect, recognize and track common objects on our mobile robot. We therefore extended our own object dataset from 2333 to 3340 image samples including overall 11 object categories and 20 object instances. To solve the more generally defined task of detecting common objects, we improved our previous segmentation method to be able to automatically detect and successfully identify major regions of interest for the objects to be recognized. We evaluated various compound descriptors with different machine learning methods. We furthermore tested the descriptors for real-time suitability and showed that the recognition task based on our object samples can be carried out in real time using only RGB data for the compound SCEHu descriptor. Final extensive results prove that the classification accuracy of our proposed system using the random forest classifier does not significantly suffer from the reduced dimension and is able to detect 99.31% of the evaluated samples. Optionally, we are now able to continuously track objects once they have been identified. By this, the run time overhead of frequent detection queries is reduced. Additionally, the detected object's pose may be forwarded continuously to other modules for further processing.

References

- L. Jiang, A. Koch, S. A. Scherer, and A. Zell, "Multi-class fruit classification using RGB-D data for indoor robots," in *IEEE Int. Conf. Robotics and Biomimetics* (*ROBIO*), (Shenzhen), 2013.
- M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy, "Bilvideo-7: An MPEG-7- compatible video indexing and retrieval system," *IEEE Multimedia*, vol. 17, no. 3, pp. 62–73, 2010.
- B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits and Systems for Video Technology (CSVT)*, vol. 11, pp. 703–715, 2002.
- G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Proc. of the Fourth IEEE Workshop on Applications of Computer* Vision (WACV'98), pp. 214–219, Oct. 1998.
- Y. Khan, A. Masselli, and A. Zell, "Visual terrain classification by flying robots," in IEEE Int. Conf. Robotics and Automation (ICRA), (Saint Paul, MN), pp. 498–503, May 2012.
- K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, (Shanghai, China), pp. 1817–1824, 2011.
- C. Gu, J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *IEEE Int.* Conf. Computer Vision and Pattern Recognition (CVPR), (Miami, FL), pp. 1030– 1037, 2009.
- D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Computer Vision, vol. 60, pp. 91–110, 2004.
- H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346–359, 2008.
- A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *IEEE Pro. European Conf. Computer Vision (ECCV)*, pp. 224–237, May 2004.
- A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence* (*PAMI*), pp. 433–449, 1999.
- A. Karpathy, S. Miller, and L. Fei-Fei, "Object discovery in 3D scenes via shape analysis," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, (Karlsruhe, Germany), pp. 290–294, May 2013.
- L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, (California), pp. 821–826, 2011.
- J. Fischer, R. Bormann, G. Arbeiter, and A. Verl, "A feature descriptor for textureless object representation using 2D and 3D cues from RGB-D data," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, (Karlsruhe, Germany), pp. 2104–2109, May 2013.
- R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutionalrecursive deep learning for 3D object classification," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, 2009.