View planning for efficient contour-based 3D object recognition

C. Urdiales, C. de Trazegnies, J. Pacheco and F. Sandoval

***Abstract*—This paper presents a method for capture planning in view based 3D recognition. Views are represented by their contours, encoded into curvature functions, which are reduced into compact feature vectors by Principal Component Analysis. These vectors are very resistant against transformations, so they can be assumed to be distributed over the surface of a sphere with the object in its center. After clustering these vectors, 3D objects are represented via Hidden Markov Models where classes are states. To recognize an object in a minimum number of steps, we propose to align candidate cluster representations and then subtracting their cluster maps to decide in which locations they differ the most. Then, a TSP is used to decide in which order these distinctive locations are visited. The proposed approach has been successfully tested with several artificial 3D object databases, even though it still presents some errors in objects with strong symmetries.**

I. INTRODUCTION

3D object recognition is an area of key importance in computer vision applications. Yet, most methods are complex and computationally expensive, due to required object segmentation and representation, feature extraction and matching of an usually large amount of data. Roughly, these methods can be divided into two categories. Geometrical methods [28] rely on approximating the input object by a combination of a variety of representations, like generalized cylinders, superquadrics or parametric bicubic patches. If this combination is similar to a stored model, the object is recognized. Else, it can be learnt as a new model. Their main drawback is that full 3D models of objects are typically required for recognition. Also, they are very sensitive to noise, distortions and occlusions and, typically, too slow for on-line processing.

Alternatively, 3D objects can be recognized by studying sets of planar views from the object [7], that are analyzed to extract significant features. Some methods rely on extracting the relevant points of a subset of canonical views [11] [16] [23] [24], later matched to stored models. However, detection of relevant points is sensitive to noise, transformations and illumination changes, so some authors propose multiscale analysis [17], even though it makes processing much slower. Other methods prefer to work with the whole view image instead, even though a represented object yields a very large data volume. A view data is usually reduced by methods like Principal Components Analysis (PCA) [25], so that complete objects become trajectories in their view eigenspace [9] [19]. Objects are recognized by projecting one of its views in the eigen-space and, then, calculating which curve it belongs to. Thus, the nature and the pose of the object are simultaneously obtained. This approach is quite sensitive to shadows and illumination changes [26], but its main drawback is assumming that different objects may not present similar views: if two curves in the eigen-space are intersected, there is no solution to the recognition problem [9].

In order to deal with this problem, the authors proposed a new 3D object recognition to approach the object by a set of planar views that were represented by the curvature of their shapes [27] to achieve robustness against transformations, noise and illumination changes [26]. Unfortunately, shapes could only be obtained via segmentation, so resulting shapes were likely to be distorted and noisy. Noise can be removed by working on the Fourier domain and using PCA to reduce the FFT curvature function dimension, but distortions still affect the resulting vector and, definitely, different objects may present similar views. Hence, a Hidden Markov Model (HMM) method was applied to disambiguate possible candidates during the recognition process. Thus, confidence was increased by analyzing views in a sequential way. This approach could require analyzing too many views if the input object was rotated over a symmetry axis. However, its main problem was that the probability of finding a distorted view grows with the number of views analyzed and these errors reduce the probability of recognition in

the HMM. Yet, given a number of views large enough, the process usually converges to an unique candidate. However, this solution is not efficient in the real world, where either the object or the video camera has to be repositioned to acquire a given view. Also, in objects with many symmetries, most of the views only add redundant information to the recognition process. As it would obviously be desirable to recognize an object in the minimum number of steps, so it is necessary to choose carefully which views need to be analyzed.

This work presents a new method to choose the best sequence of views to recognize a 3D object in an efficient way. First, the object representation and recognition method is briefed in section 2.

II. 3D OBJECT REPRESENTATION AND RECOGNITION

Since real image segmentation is a complex field on its own and not within the scope of this work, in this section we assume that we already have segmented the object views, even though they may well be noisy and distorted.

A. Single view representation

Shape based object recognition methods rely on capturing the salient shape descriptors which should be: i) invariant to geometric transformations; ii) robust to noise; iii) meaningful to matching algorithms; iv) robust to occlusions; and v) computationally feasible. Contours, reported to be good shape descriptors, are often represented by their curvature functions (CF). However, most CF calculation techniques implicitly filter contours at a fixed cut frequency [7], so relevant curvature information appearing at different scales might be lost. The authors proposed in [6] a new adaptively estimated curvature function (AECF) to filter noise in an adaptive way depending on the natural scale of the curve to avoid both noise and distortions. The proposed method consists of the following stages:

- Contour encoding by means of an incremental chain code. The incremental chain code associated to a given pixel n is a vector $(\Delta x(n), \Delta y(n))$ which presents the difference in x and y between points n and $n + 1$ of the contour. Further steps will represent the function by means of an adaptive code to includes adaptation to the natural scale of the curve
- For every point n, calculation of the maximum contour length $k(n)$ free of discontinuities around n. The value of k for a given pixel n $(k(n))$ is calculated by comparing the Euclidean distance from pixel $n - k(n)$ to pixel $n + k(n)$ of the contour $(d(n - k(n), n + k(n)))$ to the real length of contour between both pixels$(l_{max}(k(n)))$. Both distances tend to be equal in absence of corners, even for noisy contours. Otherwise, $d(n - k(n), n + k(n))$ is significantly shorter than $l_{max}(k(n))$. Thus, $k(n)$is the largest value that satisfies:

$$d(n - k(n), n + k(n)) \geq lmax(k(n)) - Uk \quad (1)$$

  being U_k a constant value that depends on the noise level tolerated by the detector. If U_k is large, $k(n)$ tends to be large and some contour spikes might be softened, but if it is low, the resulting curvature function is very noisy. Fortunately, it is very easy to choose a suitable U_k and it has been empirically proven that $U_k = 0.4$ works correctly in most cases [6].
- Calculation of the incremental adaptive chain code $(\Delta x(n)_k, \Delta y(n)_k)$, associated to n. This new vector shows the variation in x and y between contour pixels $n - k(n)$ and $n + k(n)$ and it is equal to:

$$\Delta x(n)_k = \sum_{j=n-k(n)}^{n+k(n)} \Delta x(j)$$
$$\Delta y(n)_k = \sum_{j=n-k(n)}^{n+k(n)} \Delta y(j) \quad (2)$$

- Calculation of the slope of the curve at every point n. We consider that the slope at point n can be approximated by the angle between the segment $(n - k(n), n + k(n))$ and the vertical axis. This angle is equal to:

$$Ang(n) = arctan\left(\frac{\Delta x(n)_k}{\Delta y(n)_k}\right) \quad (3)$$

- Calculation of the curvature at every point n. The curvature at every point n can be defined as the slope variation respect to n, $\Delta(Ang(n))/\Delta n$ This value can be approximated by the incremental $\Delta(Ang(n))/\Delta n$, or locally by $Ang(n + 1) - Ang(n)$.

The proposed AECF behaves well enough with respect to scale that a filtered version of the original shape can be recovered from our AECF (Fig. 1). The length of an AECF is equal to the view contour length, plus rotations provoke shifting in the curve. However, shiftings do not appear in the module of the its FFT (CFFFT). Furthermore, curvature information is quite redundant, so we use PCA to extract the most relevant information of each CFFFT. As proven in [8], CFFFTs of closed curves can be roughly represented by a reduced number of components that can be extracted from a relatively low number of different samples. In our tests we use a 10-dimensional basis calculated from a set of 27 traffic signals [8]. From this point on, any view can be represented by projecting its CFFFT onto the aforementioned basis, i.e. by a 10 component feature vector (FV).

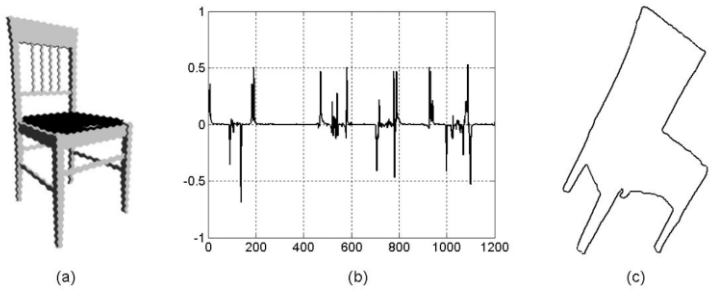


Fig. 1. a) 3D object view; b) AECF; c) recovered view.

B. Multiple view representation

When a camera moves around a 3D object (or the object is rotated), the variation of FVs of capture views represent the input object. FVs are quite resistant against scale, so we assume that points of view differing only in their radial coordinate are represented by the same FV. Hence, any object can be represented by a 2-dimensional map of FVs which corresponds to the surface of a sphere surrounding the input object. Close views tend to present a certain continuity, so they usually present very similar FVs as well. Hence, our plane can be compacted by clustering FVs into connected classes. We use the Mode Analysis clustering algorithm with the Tanimoto distance to transform our 3D model into a class layout. The resulting number of

classes depends on the object nature and on the cluster radius. If the number of classes is too low, their prototypes will not be representative. A larger number of classes is not strictly negative for recognition, as their prototypes will be very similar to each other, but increases the size of the model. However, it is fairly easy to set a conservative radius that might not return the most efficient number of classes but works fine for most usual 3D objects.

In Fig. 2, we capture 144 views for a cube. View points are equally spaced but they do not need to be. Furthermore, it is not necessary to acquire a whole set views. In objects presenting strong symmetries, some views can be interpolated, specially if they fall within a defined cluster. Also, if part of a new object can not be observed, those views can be labeled as unknown and recognition will operate basically in known areas. The cartesian coordinates in the 2D class layout correspond to the azimuthal and polar angles of the view point with respect to the observed object. Each view is printed in a different color by assigning the value of the first three components of the vector to RGB coordinates. Fig. 2.b show the resulting clusters for a radius equal to 0,075. Symmetries in the cube are clearly appreciated, as it can be represented with a very reduced class set. It can be observed that views corresponding to a polar angle of 90 or -90 belong to the same class because they correspond to rotated versions of the same vertical view. Also, views corresponding to diametrally opposed view points are classified into the same classes.

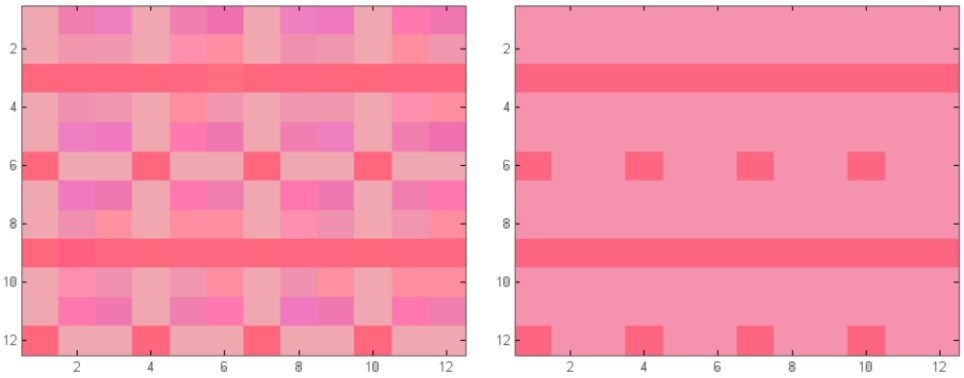


Fig. 2. a) cube vector layout (views equally spaced each 30 degrees); b)

C. Previous HMM based query method

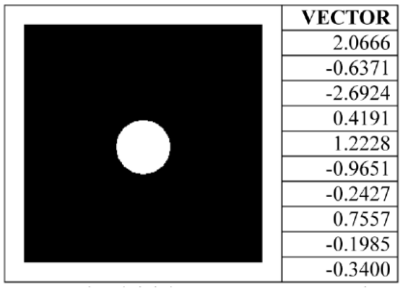
When an object x is detected, its first view V_x^1 is captured and an observation probability distribution, $B(V_x^1)$ relating the observation to every view prototype is assigned to V_x^1. Then, the probability $P(V_x^1|p)$ of the object of belonging to any of the available stored templates is evaluated. In order to choose an unique template, it is usually necessary to acquire a second view of the observed object. Since cameras move in a continuous way, consecutively acquired view are adjacent when projected onto spherical representation. Thus, we can consider that views are sequential for consecutive observations.

If the object is not still recognized, new views are captured and evaluated until the object, represented by a sequence of observations $\{V_x^1, V_x^2, \ldots, V_x^q\}$ is univocally recognized as one of the stored templates. When the sequence length increases, so does the complexity of the probability calculation. Hence, this calculation is usually performed in an iterative way to keep a bounded computational load. In our case, the calculation is performed by using the Forward-Backward Procedure [22].

If even after having evaluated the probability of a view sequence corresponding to a complete turn around the object, it is still unidentified, the object is stored as a new template. Then, a new HMM is trained and included into the template database. This training is performed after the Baum-Welch algorithm [22]. The Baum-Welch algorithm, derived from the expectation-maximization (EM) algorithm, is a local optimization method. Hence, the initialization of iterative parameters is critical for the performance of the system. The choice of the initial system parameters determines i) the number of iterations needed to converge to a stable solution and ii) the tendency to converge to an optimal or to a second order local maximum. We initialize the transition matrix A^p by evaluating the number of transitions between different views at the object cluster map. The initial probability vector Π^p for each known object is initialized with the a priori probabilities of finding the different views of such an object at the beginning of the observation sequence. These probabilities can be also calculated from the prototype class layout.

III. PROPOSED VIEW PLANNING

Our view planning algorithm relies on finding the points of maximum difference between the cluster maps of potential candidates and, then, finding the best path to visit them in order. It works as follows. First, we acquire an initial view V_x^1 and find which objects in the database present a vector whose difference to the input one is under a fixed threshold. For example, Fig. 3 presents a view of the base of a low cylinder and its corresponding vector. We purposefully chose a view which is shared by several objects within our test database, which are presented in the first row of Fig. 4.



VECTOR
2.0666
-0.6371
-2.6924
0.4191
1.2228
-0.9651
-0.2427
0.7557
-0.1985
-0.3400

Fig. 3. First captured view

To improve the efficiency of the recognition process, it would be advisable to capture a second view from the object. Then, all potential matches having no that removes as many of them from the set of candidates. This can be achieved by comparing the cluster maps of the candidates (Fig. 5). Intuitively, green locations in the map -corresponding to circular views- are not a good choice, as they appear in most objects. From a computational point of view, though, we shift all candidate maps so that the location of their first view is aligned at location (1,1). This is feasible because planes are, after all, the surface of an sphere, so rotations do not change their nature.

After that, we calculate the difference between each two maps and accumulate it to check in which points they all differ the most (Fig. 6). It can be observed that the maximum difference corresponds to the whole first line -because of cylindrical symmetry, and then 3 rows later due to symmetry as well. If we move down those three rows, meaning we rotate the camera 90 degrees with respect to the zenith axis, we would obtain a lateral view of each object, which is the best choice to distinguish among them all (second view in Fig. 4). If we apply the proposed HMM method to this second view, all candidates are discarded except our bottle.

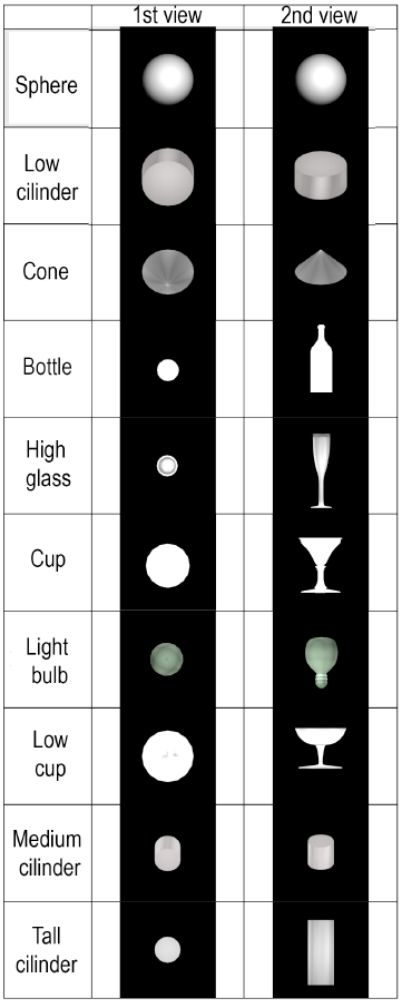


Fig. 4. Potential matches.

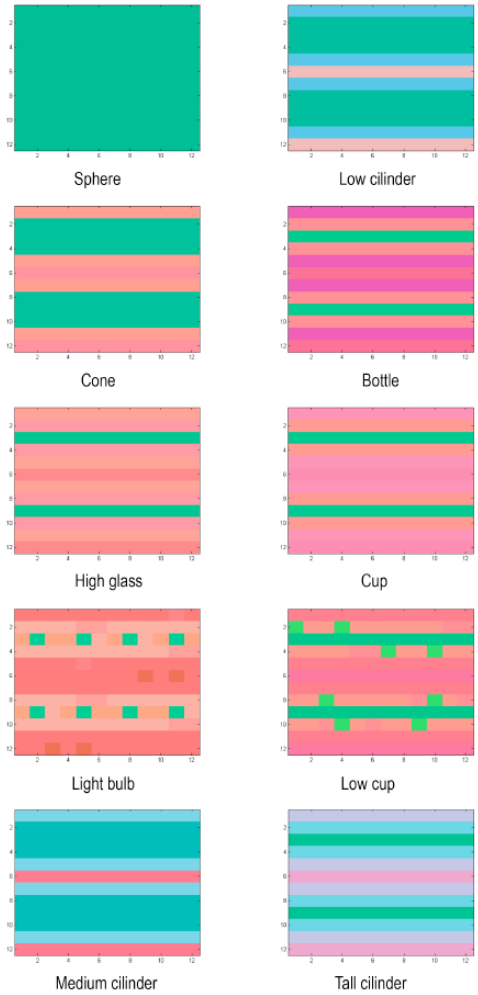


Fig. 5. Potential matches.

In this case, all objects started over their base, but the first alignment could be wrong if some of the objects were learnt in other positions, meaning that their cluster maps are not just shifted, but also rotated. In order to avoid this problem, when the second view is acquired, we realign all candidates, this time using a vector from view 1 to view 2 rather than a single view (realignment). At this point, maps are correctly aligned and HMM based recognition can be applied to views of maximum difference. To choose the most efficient sequence, we use a Traveling Salesman Problem (TSP) solution by force, because the number of views to capture is always lower than 10. Views to be captured are cities and distance are calculated over the view sphere.

IV. EXPERIMENTS AND RESULTS

In this section, we present some results of the proposed system. Basically, it works fine with complex objects presenting clearly different views, so we will stick to problematic ones.

First, we present an object that would be problematic if realignment were not performed. We work with a cube, whose first view is presented in the first row in Fig. 7. The best candidates are presented in the same figure, and winning probabilities at this point correspond to objects 2, 3 and 12 (0.977, 0.94 and 0.926 respectively). After maps are aligned, the second proposed view quickly removes object 12, a table, from potential candidates, as its legs are clearly perceived and its probability in the

HMM goes very low. However, it is not as easy to decide whether we choose object 2 or 3 if we assume that transformations and distortions are allowed and, hence, can not rely on a too strict similarity threshold. Fig. 8 shows the captured view map -this one would not be available in an usual case, as only a few views would be captured, and it is only presented here for comparison and candidate cluster maps 2 and 3. Despite their high symmetries and likeness, we are indeed checking the views we should, as proven by the different colors of the frame where views are captured in both objects. Still, our similarity threshold is not low enough to tell them apart as fast as we would like to. In the end, after 4 new captures, the system can not decide between objects 2 and 3. If realignment is allowed, though, the system recognizes the cube in just 3 captures.

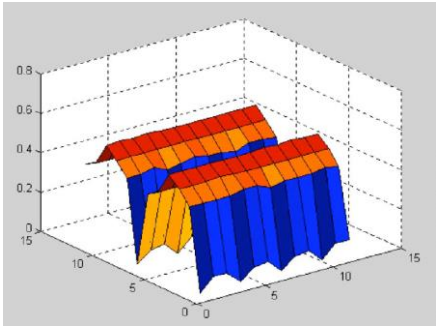


Fig. 6. Potential Matches

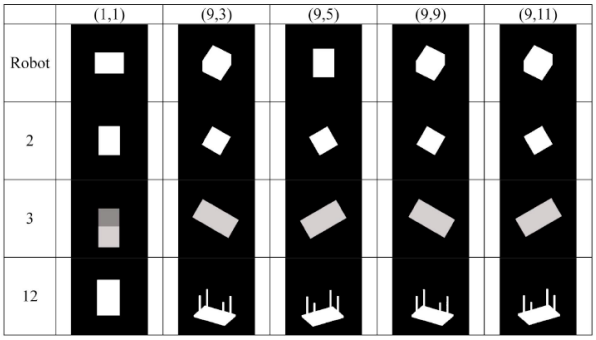


Fig. 7. Potential Matches

Even when we use realignment, some objects may take a while to disambiguate. For example, the low cylinder in Fig. 3. After the second capture, most objects in the candidate set were discarded, but the low cylinder and the tall one where still strong candidates. This happens because the shape of the view of the low cylinder over its base is similar to a rotated version of the tall one and, due to their symmetry, new captures do not help much. This is clearly observed in the captured and model maps (Fig. 9). Yet, the similarity of the input object to its equivalent is enough to slowly produce a success after increasing the probability via several captures. In this case, the final match probability is just 0.6, but the second candidate is merely 0.013.

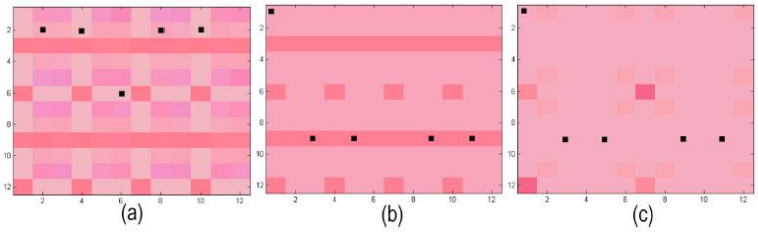


Fig. 8. Models for a) captured object; b) object 2; c) object 3.

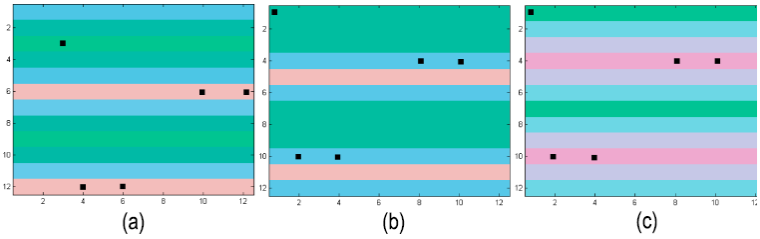


Fig. 9. Models for a) captured low cylinder; b) low cylinder; c) tall cylinder.

V. CONCLUSIONS AND FUTURE WORKS

This paper has presented a new method to plan which views of a 3D input object should be analyzed to recognize it in an efficient way. Views are represented by the curvature function of their contours, captured via a method that preserves significant features at different scales. CFs are encoded into a reduced feature vector by PCA. Views of a given object are distributed over the surface of a sphere with the object in its center, as the proposed FV is very resistant against scale. Then, 3D objects are represented by a set of FV clusters over the sphere surface and modelled with Hidden Markov Models. In objects with strong symmetries or when several objects in the database present different views, HMM may require a large number of steps to converge to a single solution. Furthermore, distorted FV due to segmentation errors and noise also affect the recognition process and increase this number of steps. In order to recognize objects in as few steps as possible, we propose to align candidate representations using two captured views and then subtracting their cluster maps to decide in which locations they differ the most. Then, a TSP [2] is used to decide in which order they are visited in order to minimize camera motion. In most cases, the proposed HMM based recognition method converges before all these locations are visited, usually just after just 4-5 steps depending on the nature of the input object and the database. Detected errors are mostly related to objects with strong symmetries which are very similar to several candidates in the database. If similarity thresholds were reduced, these problems would mostly disappear, but the system would be less resistant to distortions, perspective changes and noise. Future work will consequently focus on finding additional view features to improve recognition rate.

VI. ACKNOWLEDGMENTS

This work has been supported by the spanish Ministerio de Ciencia e Innovacin (MICINN) project TEC-2008-06734 and Junta de Andalucia (JA) project TIC-03106.

REFERENCES

[1] Aas, K., Eikvil, L. and Huseby, R., 1999. Applications of hidden Markov chains in image analysis. *Pattern Recognition*, **32** (4), 703- 713.

[2] Applegate, D. L., Bixby, R. M., Chvtal, V., Cook, W.J., 2007. *The Traveling Salesman Problem: A computational study*.

[3] Agam, G. and Dinstein I., 1997. Geometric separation of partially overlapping nonrigid objects applied to automatic chromosome classification. *IEEE Trans. Pattern Analysis and Machine Intell.*, **19** (11), 1212-1222.

[4] Ansari, N. and Delp, E. On detecting dominant points, 1991*Pattern Recognition*, **24**(5), 441-451.

[5] Bandera, A., Urdiales, C., Arrebola, F. and Sandoval, F.,1999. 2D object recognition based on curvature functions obtained from local histograms of the contour chain code. *Pattern Recognition Letters*, **20** (1), 49-55.

[6] Bandera, A., Urdiales, C., Rodriguez, J.A. and Sandoval, F., 2000. Corner detection techniques for planar images, in *Pandalai, S.G. (Ed.), Pattern Recognition* **I**, 137-150, Transworld Recent Research Network: Kerala.

[7] Xiao, C., Jianxun, L. 2009. Three-dimensional projective invariants of points and lines from multiple images Export. *Optical Engineering*, **48** (5), 057201-057201-5.

[8] Burns, B., Weiss, R. and Riseman, E., 1992, The non-existence of general case view invariants, *In Mundy, J.L. and Zisserman (eds.), Geometric Invariance in Computer Vision. Cambridge, MA: MIT Press*.

[9] Campbell, R. and Flynn, P., 1999. Eigenshapes for 3d object recognition in range data. *Proc. of the Intl. Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, 505-510, Fort Collins-Colorado.

[10] Chang, F.S. and Chen, S.Y., 2000. Deformed Shape Retrieval Based on Markov Model. *Electronic Letters*, **36** (2), 126-127.

[11] Cross, G., Fitzgibbon, A. W. and Zisserman, A., 1999. Parallax geometry of smooth surfaces in multiple views. *Proc. 7th Int. Conf. on Computer Vision, I*, 323-329, Korfu-Greece.

[12] Deichsel G. and Trampisch H.J., 1985. Clusteranalyse und Diskriminanzanalyse, *Gustav Fischer Verlag*, Stuttgart, 24.

[13] He, Y. and Kundu, 1991, A. 2D shape classification using Hidden Markov Models, *PAMI* 13(11), 1172-1184.

[14] Hornegger, J., Niemann, H., Paulus, D. and Schloteke, G., 1991, Object recognition using Hidden Markov Models, in E.S. Gelsema and L.N. Kanal (eds.), *Pattern Recognition in Practice IV*, Elsevier, Amsterdam, 37-44.

[15] Kuo, S.S. and Agazzi, O.E., 1994. Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models. *IEEE Trans. Pattern Analysis and Machine Intell.*, **16** (8), 842-848.

[16] Lo, K.C. and Kwok, S.K.W., 2001. Recognition of 3D planar objects in canonical frames. *Pattern Recognition Letters*, **22** (6-7), 715-723.

[17] Shotton, J., Blake, A., Cipolla, R. 2008. Multiscale Categorical Object Recognition Using Contour Fragments, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30** (7) 1270-1281.

[18] Mokhtarian, F. and Mackworth, A.K., 1986. Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8 (1)**, 34-43.

[19] Mukherjee, S. and Nayar, S., 1993. Object Recognition and Pose Estimation in Eigenspace Using a RBF Network. *Technical Report 40-93*, Department of Computer Science, University of Columbia.

[20] Murase, H. and Nayar, S.K., 1995. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, **14**, 5-24.

[21] Natarajan, P., Lu, Z., Schwartz, R., Bazzi, I. and Makhoul, J., 2001. Multilingual machine printed OCR. *International Journal of Pattern Recognition and Artificial Intelligence*, **15** (1), 43-63.

[22] Rabiner, L.R., 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, **77** (2), 257-286.

[23] Roh, K.S. and Kweon I.S., 2000. 3-D object recognition using a new invariant relationship by single view. *Pattern Recognition*, **33** (5), 741- 754.

[24] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, 2008. Groups of adjacent contour segments for object detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30** (1), 3651.

[25] Sirovich, L. and Everson, R., 1992. Analysis and management of large scientific databases. *Int. J. Supercomputing Applic.*, **6** (1), 50-68.

[26] Startchik, S., Milanse R. and Pun, T., 1998. Projective and Illumination Invariant Representation of Disjoint Shapes. *Fifth European Conference on Computer Vision (ECCV '98)*, **1**, 264, Freiburg-Germany.

[27] Urdiales, C., Bandera, A. and Sandoval, F., 2002. Non parametric planar shape representation based on adaptive curvature functions. *Pattern Recognition*, **35** (1), 43-53.

[28] Wang, P.S., 2000. 3D image understanding and recognition in virtual environment, in M. Cheriet and Y. H. Yang (Eds.), *Vision Interface: Real World Applications of Computer Vision*, World scientific: Nueva York.