This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

# An Enhanced Geometric Hashing

Umarani Jayaraman, Amit Kumar Gupta, Surya Prakash and Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, Kanpur-208016, India
Email:{umarani,amitkrg,psurya,pg}@cse.iitk.ac.in

***Abstract*—This paper presents an enhanced geometric hashing technique suitable for object recognition. Unlike the available geometric hashing, the proposed technique needs less amount of time and memory, uniform index distribution in the hash space without using any rehashing function. It performs *indexing* and *searching* in one pass with linear complexity. The proposed technique has been applied in biometric databases. It has been tested for two traits such as ear and iris database. The *hitrate* of 100% has been achieved in top 5 best matches in all cases.**

## I. Introduction

The problem of recognising an image has been extensively studied in the field of computer vision, pattern recognition, etc. The trivial solution for object recognition is to search all images in the database (henceforth called models) against a query image (henceforth called query). It is computationally inefficient to retrieve each model from the database and to compare it against the query for a match. Hence there is a need to index these images to reduce the complexity of recognition significantly. However, following are some issues while indexing images of the large database.

- The number of features points extracted from an image may not be fixed.
- Feature vector representing an object is of high dimension.
- Let $M = \{f_1, f_2, \cdots, f_m\}$ and $Q = \{f'_1, f'_2, \cdots, f'_n\}$ be a model and a query respectively where $f_i s$ and $f'_j s$ are feature points. Assume that $M$ and $Q$ are the two instances of the same object. It can be observed that
  - number of feature points in $M$ and that in $Q$ may not be same.
  - there may not exist any $f'_j$ in $Q$ such that $f_i = f'_j$, for some $f_i$, $f_i \in M$.
- The image of an object may be partially occluded and it may contain some other objects which are not present in the database for recognition.
- The objects may be rotated and translated relative to their database position.

Most of the indexing techniques work a fixed number of feature points. However, in some applications like biometrics, the feature points may vary from image to image and the maximum number of feature points cannot be predicted in advance. Hence, there is a need to design a new indexing technique that supports variable number of features on high-dimensional space.

*Classical geometric hashing* can be used to index the variable feature points in a high-dimensional space. However, classical geometric hashing may not be suitable for its computational time and memory requirement because it uses $nC_2$ bases pair [1][2].

The computational cost of classical geometric hashing can be reduced by the approaches which are based on probabilistic, randomization and modified geometric hashing. In probabilistic approach the computational cost is reduced by thinning out feature points and number of *bases*. In [3] a probabilistic approach of random reduction of feature points has been proposed to reduce the computational cost. But it decreases retrieval accuracy. The theoretical as well as experimental results of the Monte Carlo algorithms (randomization) proposed in [4] suggest that it is possible to avoid the worst case behavior of the deterministic versions of alignment and classical geometric hashing, while ensuring a very small probability of a mismatch. In [5] a modified geometric hashing has been proposed to reduce the computational cost and memory in content-based image retrieval system to retrieve similar shape images from the visual database. To minimize the memory demand, only a subset of image primitives are stored in the hash table.

The non-uniform distribution of invariants over the hash table makes an inefficient storage of the data resulting the slow down of the recognition process. The key solution to the problem is the selection of a "good" geometric hash function which can redistribute the data uniformly over the hash table. All combinations of hash functions have been attempted in [6] to choose the best one. In [7], [8] rehashing functions have been suggested. The basic idea is to find a transformation which maps the distribution of invariants to a uniform distribution. This approach suffers from two drawbacks. First, it is based on the assumption that the probability density function (pdf) of model feature points is known a priori. However, such an assumption can be unreliable, especially in cases where the number of models is small. Second, the derivation of the rehashing transformation involves complex computations and infeasible [9]. In [10] a technique to make uniform distribution of data points has been proposed. It is based on the use of an "elastic hash table" implemented as a Self-Organizing Feature Map Neural Network. Unlike existing approaches which try to redistribute the invariants over the hash bins, it spreads the hash bins over the invariants. The advantage of this approach is that it adapts to the invariants through learning. Hence, it makes no assumption about the statistical characteristics of the invariants. In [11] a technique based on the voting scheme has been proposed. It casts a vote in Hough space. The use of this

Hough transform based vote renders this system robust even against large number of collisions in the hash table.

There are some improved techniques of classical geometric hashing which employ other primitive features. Instead of feature point, techniques in [12] and [13] have used a line segment and a chain of connected line segments respectively to index into the hash table. In [14] correlated complex local shape parameters (i.e., four-dimensional local shape descriptors) have been used to produce viewpoint invariant. The use of correlated high-dimensional indices is crucial to overcome the drawbacks of classical geometric hashing [15] such as poor index selectivity, excessive accumulation of votes in each bucket, limited number of useful buckets in the table and extreme sensitivity to noise and quantization parameters [16]. A new hashing technique [17] which is based on non-probabilistic selection of feature points has linear computational cost. It uses only $t$ neighbors of the feature point having computational cost of $O(t^b mN)$ in the *preprocessing* stage and $O(t^b n)$ in the *recognition* stage where $t$ is the nearest neighbors.

For indexing, classical geometric hashing maps feature points of an object to hash table with respect to a bases pair points which are selected from the object itself. In order to handle all possible rotations in an object, the technique considers all possible bases pair (effectively $nC_2$, where $n$ is the total number of feature points in the object) and maps object feature points in the hash table with respect to all the bases pairs. As a result, it increases both memory and searching cost.

In the proposed technique, in order to handle all possible rotations in an object, each image object is aligned using principal components of the feature points obtained using the PCA. So the technique effectively removes the use of bases pairs thus reduces the time complexity by a factor of $nC_2$. Overhead in the proposed technique is the use of PCA which is negligible. In the proposed indexing technique, since each feature point is inserted exactly once, both memory and searching cost has been reduced significantly.

This paper is organized as follows. Section II introduces the proposed enhanced geometric hashing. Experimental results are analyzed in the next section. Conclusions are given in the last section.

## II. Proposed Enhanced Geometric Hashing

Enhanced geometric hashing consists of two stages known as *indexing* and *searching*. In the *indexing* stage, the models are inserted into the hash table. For any new model, it can be added without affecting the performance of the recognition algorithm and without modifying the existing index structures. Indexing consists of three steps known as feature extraction, preprocessing and hash table generation. In the *searching* stage, it extracts the features from a query and preprocessed similar to *indexing* stage and accesses the previously constructed hash table for recognition. It consists of two steps, filtering and refinement.

**Algorithm 1** : Indexing

For each model $M$ in the database do the following

1. Extract the feature points from an image. Assume there are $m$ such feature points are found.
2. For given $m$ feature points, each having their coordinates $(x, y)$, compute the mean centering $i.e$ find the mean point and shift the mean point towards the center of the coordinate system.
3. Determine the first and second principal components of the mean centered coordinate system by using PCA technique.
4. Rotate these principal components such that it becomes the primary axis of the coordinate system.
5. For every feature point, the new coordinate $(x', y')$ is then calculated in the coordinate system defined by the principal components.
6. Use the new coordinate $(x', y')$ as an index to insert the feature points into a $2-D$ hash table, with the information model identity, $M_{id}$ and descriptor vector, $(\vec{D})$ of a feature point obtained through SURF.

### A. Indexing

At the time of Indexing, it index the features of all models in the database. Features are represented by co-ordinate position. It consists of co-ordinate position of feature points. Generally, based on the number of matched co-ordinate positions, the correspondence between two object is found. It gives a measure of global similarity between the objects, but it does not establish a one-to-one correspondence between them. Although considerable amount of time has been reduced through indexing in classical geometric hashing, the lack of quantitative correspondence makes this technique unsuited for recognizing multiple or partially occluded objects in a noisy image. Much attention has been made to extract geometric primitives (e.g., lines, circles, curvature, edges, etc.) that are invariant to viewpoint change [18]. Nevertheless, it has been shown that such primitives can only be reliably extracted under limited conditions (controlled variation in lighting and viewpoint with certain occlusion). Hence the idea is to use simple and efficient quantitative information along with the feature point based co-ordinate positions. The descriptor vector around each feature point is considered as quantitative information in order to improve the performance of the recognition system. The algorithm for indexing the model image in the enhanced geometric hashing is given above.

*1) Feature Extraction:* For each model in the database, features are extracted by SURF feature extraction algorithm which has more discriminative power than a local feature descriptor such as SIFT [19]. Assume that for a model, $M_J$, SURF has extracted $m$ key points, $f_1, f_2, ..., f_m$. It can be noted that the number of key points may vary from one model to another. Each $f_i$ is a $3-$tuple $(x, y, \vec{D})$ where $x, y$ is used as indices to insert into the hash table and the descriptor vector

$\vec{D}$ is used to filter out the results during searching.

*2) Preprocessing:* During indexing, the features $f_1, f_2, ..., f_m$ are extracted for all models using SURF algorithm [19]. There is a possibility that the model images may appear translated and rotated relative to their original positions. Also, models may not have the same scale. Hence in order to make the proposed indexing technique invariant to translation, rotation and scaling, every model in the database is preprocessed. It consists of three steps, mean centering, rotation with respect to principal components and normalization.

If the model image is translated, the feature points are also translated from their original positions. In that case, mean centering can be used to translate each feature point $f_i$ to $f'_i$ such that mean of all $f'_i$ becomes zero. This can be done by subtracting $\bar{f} = \frac{1}{m}\sum_{i=1}^{m} f_i$, from $f_i$.

The set of feature points $\{f'_1, f'_2, ..., f'_m\}$ is used to determine the largest $(PC_1)$ and the second largest $(PC_2)$ principal components. The geometric properties of PCA are used to make the model image invariant to rotation. $PC_1$ and $PC_2$, are rotated in such a way that they become the primary axes of the co-ordinate system. Let the point set $\{f'_1, f'_2, ..., f'_m\}$ become $\{f''_1, f''_2, ..., f''_m\}$ after the rotation step.

In order to make the feature points invariant to scale, the normalization step is carried out. For the point set $\{f''_1, f''_2, ..., f''_m\}$, each feature point is divided by the standard deviation $(\sigma)$ of the point set. It can be observed that all these normalized feature points become much closer to each other and as a result, it may occupy a single bin into the hash table. In order to avoid such a situation, it is multiplied by some scaling factor $\alpha$ such that every feature point, gets different bin into the hash table. Let the point set $\{f''_1, f''_2, ..., f''_m\}$ become $\{f'''_1, f'''_2, ..., f'''_m\}$ after the normalization step. These preprocessing steps has little effect on well distribution of feature points in the hash table.

*3) Hash Table Generation:* Each preprocessed feature point in a model is mapped into the location $p_l$ of hash table by placing the midpoint of the co-ordinate system at the center of the hash table as follows

$$p_l = f'''_l + \frac{size(H)}{2} \tag{1}$$

where $size(H)$ is the number of bins in the hash table. After mapping, all feature points are inserted into hash table as

$$H(p_l) = H(p_l) \cup (M_{id}, \vec{D}) \tag{2}$$

where $M_{id}$ and $\vec{D}$ are the model identity and the descriptor vector of the feature point respectively. The same process is repeated for each model in the database.

### B. Searching

In contrast to classical geometric hashing, the response time in the enhanced geometric hashing is much faster because it needs to compare with a smaller number of feature points. For a query $Q$, Algorithm 2 gets the top $k$ best matches from the hash table in a two step process, filtering and refinement. In the filtering step, the feature points which are dissimilar to the query's feature points are filtered out while at refinement step, the top $k$ matches are found based on the voting scheme.

**Algorithm 2** : Searching

For query $Q$ do the following

1: Extract the feature points from an image. Assume there are $n$ such feature points are found.
2: For the set of $n$ feature points each having their coordinate $(x, y)$, preprocessing has been done as like in indexing.
3: For each feature point, compute the new coordinates $(x', y')$ defined by the principal components.
4: The new coordinate $(x', y')$ of a feature point are mapped to the hash table to access the appropriate bin.
5: Compute the Euclidean distance between the feature descriptor $\vec{D}$ of a query feature point and all the points that found in the bin and select only those points as candidate set whose distance is less than threshold $t$.
6: Repeat the same process for remaining query's feature point and accumulate the results into the candidate set $C$.
7: Cast a vote based on their occurrence of their model identity $(M_{id})$ from the candidate set $C$ to declare the top $k$ best matches.

*1) Filtering and Refinement:* In the filtering step, the feature points of the model which are dissimilar to the query's feature points are filtered out based on their descriptor vector $\vec{D}$. It can be noted that the feature points of the different images of the same model may be missed due to the noise present in the images. In order to improve the recognition performance, it considers the feature points not only from its mapped bin but also from its nearest bins of size $K \times K$.

Let $\{f_1, f_2, \cdots, f_n\}$ be the $n$ feature points in the query. For a feature point, $f_i$, let q be the mapped index in the hash table $H$. As discussed above, it is considered $K \times K$ neighbors of $q$ in $H$. Let $z$ be such a neighboring bin, i.e., $z \in \left[q - \left\lfloor \frac{K}{2} \right\rfloor, q + \left\lfloor \frac{K}{2} \right\rfloor\right]$. Note that there may be some feature points of different models from the database lying in the bin $z$ of $H$. Let $c$ be a feature point of a model lying in $z$. Euclidean distance between $q$ and $c$, $\forall c \in H(z)$ and $z \in \left[q - \left\lfloor \frac{K}{2} \right\rfloor, q + \left\lfloor \frac{K}{2} \right\rfloor\right]$, can be obtained by

$$d(c) = ||\vec{D}(q) - \vec{D}(c)|| \tag{3}$$

A candidate set $C_i$ for the corresponding feature point $f_i$ of the query $Q$ contains all the model identity $M_{id}(c)$ such that $d(c) \leq t$, $\forall c \in H(z)$ and $z \in \left[q - \left\lfloor \frac{K}{2} \right\rfloor, q + \left\lfloor \frac{K}{2} \right\rfloor\right]$, where $t$ is the assigned threshold. The same procedure is followed for all query points $f_1, f_2, \cdots, f_n$. Thus, there are $n$ candidate sets $C_1, C_2, \cdots, C_n$ for given $n$ feature points in a query $Q$.

In refinement step, these candidate sets $C_1, C_2, \cdots, C_n$ are concatenated and the number of occurrences of each model identity $M_{id}$ is determined. Let $C$ be the set of the form as $C = \{M_{id}, l\}$ where $l$ is the number of occurrence of each model identity $M_{id}$. The elements of $C$ are arranged in decreasing order with respect to the number of occurrences. First $k$ model identities of the set $C$ are considered as top $k$ matches against the query $Q$.

### C. Time Complexity

Enhanced geometric hashing allows to solve the *indexing* and *searching* problem in one pass with linear time. It reduces the size of the hash table by inserting each feature points exactly once into the hash table. Hence the computational cost reduces to $O(mN)$ in the *indexing* stage and $O(nk)$ in the *searching* stage where $k$ is the window size considered at the time of searching. The number of stored entries in the hash table of enhanced geometric hashing is $mN$. Therefore, the number of entries in a sub region is reduced to $mN/k$ in average. The preprocessing step has little effect on uniformity distribution in enhanced geometric hashing. Hence there is no need of any rehashing function. Further, enhanced geometric hashing has $mN$ bins in the voting table. Therefore, the computational cost is $O(mN)$.

## III. Experimental Results

The proposed hashing technique has been used in the biometric databases for indexing. To determine the performance of the proposed indexing scheme, four measures, namely, *hitrate*, *bin-miss rate*, *penetration rate*, and *Cumulative Match Characteristic curve* are used.

- The *hitrate* $(H_r)$ is defined by $H_r = \left(\frac{X}{L}\right) \times 100\%$ where $X$ is the number of queries whose corresponding identity is occurred at top $t$ best match (here $t$ is 5) and $L$ is the total number of queries.
- The *bin-miss rate* $(B_r)$ is defined as $B_r = 100 - H_r$
- The *penetration rate* $(P_r)$ is defined $P_r = \left(\frac{1}{X}\sum_{i=1}^{X} \frac{d_i}{N}\right) \times 100\%$ where $X$ is the total number of query images correctly identified, $d_i$ is the number of images retrieved per query and $N$ is the database size.
- *Cumulative Match Characteristic (CMC) curve* represents the relationship between identification accuracy against rank (top position).

The proposed technique has been tested on two databases, namely ear and iris databases. The ear database is obtained from Indian Institute of Technology Kanpur (IITK). It consists a total of 1200 ear images collected from 150 subjects. From each subject 8 ear images are collected. Out of these 8 images, 7 images are used as model and the remaining image is used as query.

It also uses the publicly available UPOL iris database [20]. It consists a total of 384 iris images collected from 64 subjects. From each subject, 3 images from left and 3 images from right eye are collected. For each subject among 3 left/right iris images 2 images are considered as model images and the remaining one image is used as query.

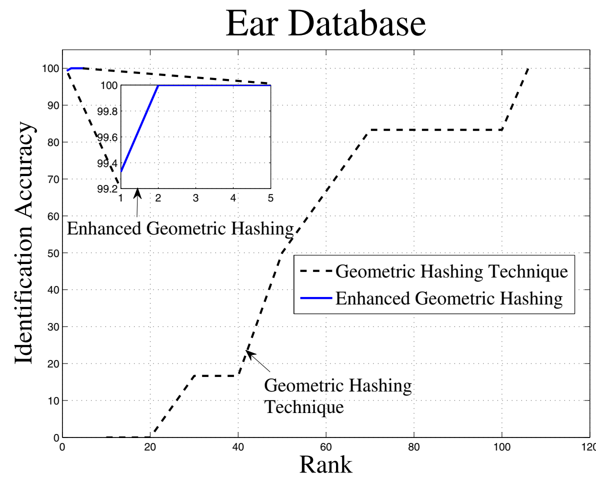


Fig. 1: Ear Database: *CMC* Curve for Optimum Window Size

To improve the performance in terms of time to achieve the same accuracy as exhaustive search gives, the biometric databases are indexed using enhanced geometric hashing. The bin-miss rate is obtained for various window size $K \times K$. It has been observed that penetration rate increases with the increase of window size while bin-miss rate decreases. High penetration rate results in more search time while low bin-miss rate results in a less secure identification system. Thus there exists a trade off between these two measures. We have to choose the window size (optimum) in such a way that both are at acceptable level. The value of window size is considered as optimum where two curves are intersected $i.e, B_r = P_r$. In our experiment, the window size is chosen as $3 \times 3$ and $5 \times 5$ for ear and iris databases such that both bin-miss rate and penetration rate are equal.

### A. Comparison with Classical Geometric Hashing

In order to show the effectiveness of the enhanced geometric hashing with classical geometric hashing two databases such as IITK ear and UPOL public iris images are considered. The number of query images considered for comparison is small because the classical geometric hashing takes huge amount of time while searching because of $nC_2$ possible bases pairs. In our experiment, each image consists an average of 80 feature points and there are $80C_2$ possible match results approximately 2.34 Hrs. All the experiment has been executed in matlab. For ear database 1200 images for training and 6 images for testing are considered. Similarly, for iris database 256 images and 10 images are considered for training and testing respectively.

In our experiment, It has been observed that the value of bin-miss rate and penetration rate is less when compared with classical geometric hashing in both ear and iris databases. In addition, the search time in the enhanced geometric hashing technique has reduced significantly when compared with classical geometric hashing which is 0.23 sec. Fig. 1 and Fig. 2 shows *Cumulative Match Characteristic* curve for

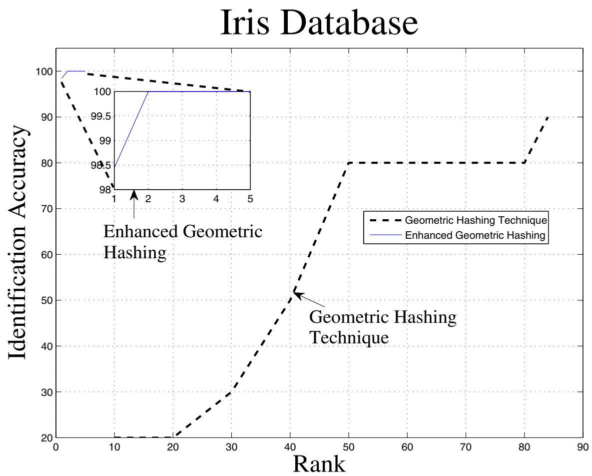


Fig. 2: Iris Database: *CMC* Curve for Optimum Window Size

both enhanced and classical geometric hashing. This graph is obtained for an optimum window size of $3 \times 3$ and $5 \times 5$ for ear and iris databases respectively. It could be noted that $100\%$ identification accuracy is achieved in top $5$ rank while the same is in top $106$ rank for enhanced and classical geometric hashing in ear database. Similarly, in iris database $100\%$ identification accuracy is achieved in top $5$ rank while only $90\%$ accuracy is achieved in top $84$ rank for enhanced and classical geometric hashing.

## IV. Conclusions

This paper has proposed an enhanced geometric hashing to index the features of models. It reduces significantly the computational cost and memory. Features are extracted in such a way that they are invariant to translation, rotation and scaling. These features of all models along with their respective model identity $(M_{id})$ and the descriptor vector $\vec{D}$ are inserted in the hash table $H$. During searching, feature points of query $Q$ which are invariant to rotation, translation and scaling are mapped on hash table to access the stored points in the bin and its $K \times K$ neighbors. Then distance has been computed between the descriptor vector of mapped query feature point and that of each feature point found at mapped location. And the model identity $M_{id}$ whose distance is less than a threshold is retained in the filtering step. The $M_{id}$'s of all mapped query feature points are concatenated and occurrence of each $M_{id}$ is found out. Then $M_{id}$ having large number of occurrences are selected for the top $k$ matches.

In enhanced geometric hashing the size of the hash table is reduced by inserting each feature point exactly once into the hash table. Hence the computational cost reduces to $O(mN)$ in the *indexing* stage and $O(n)$ in the *searching* stage. The number of stored entries in the hash table is $mN$. Therefore, the number of entries in a sub region is reduced to $mN/k$ in average. The preprocessing step makes the distribution of feature points uniformly into hash table. Hence, there is no need of any rehashing. Further, enhanced geometric hashing has $mN$ bins in the voting table. Therefore, the computational cost is $O(mN)$. The enhanced geometric hashing has been tested on IIT Kanpur ear database of $1200$ images and UPOL iris database of $384$ images. For top $5$ best matches, the identification accuracy of $100\%$ is achieved in both cases.

## Acknowledgment

The authors acknowledge support provided by the Department of Information Technology, Govt. of. India to carry out this work.

## References

[1] H. J. Wolfson, "Model Based Object Recognition by Geometric Hashing," in *Proceedings of the 1st European Conference on Computer vision*, 1990, pp. 526–536.
[2] Y. Lamdan and H. J. Wolfson, "Geometric Hashing: A General And Efficient Model Based Recognition Scheme," in *Proceedings of 2nd International Conference on Computer Vision*, 1988, pp. 238–249.
[3] M. Hoffman and M. Lindenbaum, "Some Tradeoffs and a New Algorithm for Geometric Hashing," in *Proceedings of the 14th International Conference on Pattern Recognition*, vol. 2, 1998, pp. 1700–1704.
[4] S. Irani and P. Raghavan, "Combinatorial and Experimental Results for Randomized Point Matching Algorithms," in *Proceedings of the 12th Annual Symposium on Computational Geometry*, 1996, pp. 68–77.
[5] J. Wang, W. Chang, and R. Acharya, "Efficient and Effective Similar Shape Retrieval," in *Proceedings of International Conference on Multimedia Computing and Systems*, vol. 1, 1999, pp. 875–879.
[6] R. H. M. Costa and L. Shaprio, "Optimal Affine Matching," in *Proceedings of the 6th Israeli Conference on Artificial Intelligence and Computer Vision*, 1989.
[7] I. Rigoutsos and R. Hummel, "Robust Similarity Invariant Matching in the Presence of Noise," in *Proceedings of the 8th Israeli Conference on Artificial Intelligence and Computer Vision*, 1991.
[8] R. Hummel and I. Rigoutsos, "Implementation of Geometric Hashing on the Connection Machine," in *Workshop on Directions in Automated Cad-based Vision.*, 1991, pp. 76–84.
[9] I. Rigoutsos, "Massively Parallel Bayesian Object Recognition," Ph.D. dissertation, New York, NY, USA, 1992.
[10] G. Bebis, M. Georgiopoulos, and N. da Vitoria Lobo, "Learning Geometric Hashing Functions for Model Based Object Recognition," in *Proceedings of the Fifth International Conference on Computer Vision*, 1995, p. 543.
[11] B. Lamiroy and P. Gros, "Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing," in *Proceedings of the 4th European Conference on Computer Vision - Volume I*, 1996, pp. 59–70.
[12] F. C. Tsai, "Robust Affine Invariant Matching with Application to Line Features," in *Proceedings of Computer Vision and Pattern Recognition*, 1993, pp. 393–399.
[13] Rothwell, C.A. and Zisserman, A. and Forsyth, D.A. and Mundy, J.L., "Using Projective Invariants for Constant Time Library Indexing in Model Based Vision," in *Proceedings of the British Machine Vision Conference, Glasgow Scotland*, 1991.
[14] A. Califano and R. Mohan, "Multidimensional Indexing for Recognizing Visual Shapes," *IEEE Tran. on Patrn. Ana. and Mach. Int.*, vol. 16, no. 4, pp. 373–392, 1994.
[15] A. Kalvin, E. Schonberg, J. T. Schwartz, and M. Sharir, "Two Dimensional Model Based Boundary Matching Using Footprints," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 38–55, 1987.
[16] D. H. W Grimson, "On the Sensitivity of Geometric Hashing," in *Proceedings of 3rd International Conference on Computer Vision*, 1990, pp. 334 –338.
[17] T. Nakai, K. Kise, and M. Iwamura, "Use of Affine Invariants in Locally Likely Arrangement Hashing for Camera Based Document Image Retrieval," in *Proceedings of 7th International Workshop on Document Analysis Systems*, ser. Lecture Notes in Computer Science 3872. Springer, 2006, pp. 541–552.
[18] J. L. Mundy and A. Zisserman, Eds., *Geometric Invariance in Computer Vision*. Cambridge, MA, USA: MIT Press, 1992.
[19] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *Proceeding of 9th European Conference on Computer Vision*, 2006, pp. 404–417.
[20] M. Dobe and L. Machala, "Iris Database," http://www.inf.upol.cz/iris/.