Rehashing for Bayesian Geometric Hashing

Michael Lifshits Ilya Blayvas Roman Goldenberg Ehud Rivlin
Michael Rudzsky
Technion, Computer Science Dept., Haifa 32000, Israel.
{protezhe, blayvas, romang, ehudr, rudzsky}@cs.technion.ac.il

Abstract

Geometric hashing is a model-based recognition technique based on matching of transformation-invariant object representations stored in a hash table. In the last decade a number of enhancements have been suggested to the basic method improving its performance and reliability. One of the important enhancements is rehashing, improving the computational performance by dealing with the problem of non-uniform occupancy of hash bins. However the proposed rehashing schemes aim to redistribute the hash entries uniformly, which is not appropriate for Bayesian approach, another enhancement optimizing the recognition rate in presence of noise. In this paper we derive the rehashing for Bayesian voting scheme, thus improving the computational performance by minimizing the hash table size and the number of bins accessed, while maintaining optimal recognition rate.

1. Introduction

Geometric hashing (GH) introduced by Lamdan and Wolfson [2], based on the indexing approach by Schwartz and Sharir [8], solves the problem of object recognition formulated in the following way. There is a number of predefined geometric models $M_1, ..., M_n$, defined by a set of geometric features (e.g. corner points) and a query image Q, formed from one of the models. The task is to find the corresponding model M_i given the query image Q.

The idea of GH is to shift the computational burden involved in query image comparison to the off-line learning stage. Instead of trying all possible representations of the query image, all possible transformation-invariant model representations are prepared in advance and stored in a hash table. Thus, an arbitrarily chosen representation of the query image has a matching model already stored in the hash table.

Ideally, for every feature point of the query image there is a single model point in the corresponding hash table bin. In practice, features generated by other models can fall into the same bin or even coincide. To deal with this problem one may suggest to reduce the bin size. Unfortunately, the feature points are non-uniformly distributed over the hash table. Therefore, for any bin size there will be either overpopulated or empty bins. Another problem is that the uncertainty in feature point position caused by image noise shifts it away from the corresponding model feature. This can be solved by looking for the matching model in a certain neighborhood of the image feature - voting region.

These problems received a lot of attention over the last 15 years [1, 2, 4, 5, 8]. For a comprehensive review we refer the reader to [9]. Presumably, the optimality criteria for geometric hashing are computational performance and recognition rate. Rehashing improves the computational performance by addressing the problem of nonuniform feature distribution [6]. The Bayesian approach optimizes recognition rate in presence of noise by adapting the size of the voting region [7].

In this paper we reconsider the rehashing approach for the Bayesian framework. We show that in order to optimize both the recognition rate and the computational performance the rehashing is to be redesigned to equalize voting regions rather than feature density, as suggested before [4, 6]. We derive the rehashing scheme for the case of similarity transformations and evaluate its performance in a series of experiments. The similarity transformation was chosen as a simple and yet important case. For example we are implementing it for our on-wafer navigation system.

2. Bayesian approach

The voting region size controls the tradeoff between the number of false votes and the chance to miss the correct model. Rigoutsos and Hummel [7] derive the optimal size for the case of similarity transformations using the maximum likelihood approach, as described below.

Let $\{\mathbf{p}_1, \ldots, \mathbf{p}_{N_k}\}$ be the feature points of model M_k. For similarity transformations the coordinate frame for invariant representation is uniquely defined by two points.

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

Then there are $\binom{N_k}{2}$ such bases for the model M_k. For each one of these bases $B_{\mu\nu} = \{\mathbf{p}_\mu, \mathbf{p}_\nu\}$ and for every remaining model point $\mathbf{p}_i$ the invariant representation $\mathbf{h}_i$ is computed and the entry $\{M_k, B_{\mu\nu}, \mathbf{h}_i\}$ is stored at the hash table bin indexed by $\mathbf{h}_i$. The invariant entries generated by a query Q are denoted by $F_Q = \{\mathbf{q}_1, \ldots, \mathbf{q}_{N_Q}\}$, and their corresponding hash coordinates are $\{(u_i, v_i); i = 1, \ldots, N_Q\}$.

Assuming the independence of $\{\mathbf{q}_i\}$, the probability of model M_k given the query Q is

$$P(M_k|Q) \sim P(M_k) \prod_{\mathbf{q}_i \in Q} \frac{P(M_k|\mathbf{q}_i)}{P(M_k)}. \tag{1}$$

Applying the Bayes theorem we get

$$P(M_k|Q) \sim P(M_k) \prod_{\mathbf{q}_i \in Q} \frac{P(\mathbf{q}_i|M_k)}{P(\mathbf{q}_i)}, \tag{2}$$

where $P(\mathbf{q}_i|M_k)$ is the probability of "hashing" to the location $\{u_i, v_i\}$, under the assumption that model M_k is present, while $P(\mathbf{q}_i)$ is the probability without this assumption. The ratio $\frac{P(\mathbf{q}_i|M_k)}{P(\mathbf{q}_i)}$, measuring the factor by which the probability $P(M_k|Q)$ changes due to the feature point $\mathbf{q}_i$, can be reformulated in terms of density functions:

$$\frac{P(\mathbf{q}_i|M_k)}{P(\mathbf{q}_i)} = 1 + \frac{N_k}{N_Q} \cdot \frac{f(\mathbf{q}_i) - g(\mathbf{q}_i)}{g(\mathbf{q}_i)}, \tag{3}$$

where $g(\mathbf{q}) \equiv g(u, v)$ is the density of invariant entries in the hash space, $f(u, v)$ is the density of query points corresponding to the model M_k, N_Q is the number of query points and N_k is the number of points in the model M_k. Assuming all model points present in Q, $\frac{N_k}{N_Q}$ is the probability for $\mathbf{q}_i$ to be one of the model points and $\frac{N_Q - N_k}{N_Q}$ to be a clutter point.

Actually, Equation (3) expresses the relationship between the Bayesian approach and the GH voting scheme. Query points satisfying $f(\mathbf{q}) > g(\mathbf{q})$ add to the value of $P(M_k|Q)$ and therefore vote for the model. The rest of the points decrease $P(M_k|Q)$ and are not counted as non relevant to the M_k hypothesis. The votes are to be weighted according to their contribution to $P(M_k|Q)$. For numerical stability considerations it is convenient to take the logarithm of Equation (2). Then the query point $\mathbf{q}_i$ contributes

$$W(\mathbf{q}_i) = \ln \left(1 + \frac{N_k}{N_Q} \cdot \frac{f(\mathbf{q}_i) - g(\mathbf{q}_i)}{g(\mathbf{q}_i)} \right) \tag{4}$$

to the votes of M_k.

It was shown, that for similarity transformations [4]

$$g(u, v) = \frac{12}{\pi} \frac{1}{(4(u^2 + v^2) + 3)^2}, \tag{5}$$

and

$$f(u, v) = \sum_{i=1}^{N_k} \frac{e^{-\frac{1}{2}(u - x_i, v - y_i) C_i^{-1} (u - x_i, v - y_i)^T}}{2\pi \sqrt{|C_i|}}, \tag{6}$$

where the sum runs over the model points (x_i, y_i),

$$C_i = \frac{(4(x_i^2 + y_i^2) + 3) \cdot \sigma^2}{2\|\mathbf{p}_\nu - \mathbf{p}_\mu\|^2} \cdot \mathbf{I}, \tag{7}$$

is the covariance matrix at (x_i, y_i), σ is the variance of the model feature position, and $\mathbf{p}_\nu, \mathbf{p}_\mu$ are the model points forming the basis [5, 6]. By substituting (5-7) into the inequality $f(\mathbf{q}) > g(\mathbf{q})$ and solving for $\rho(u, v) \equiv \|(x, y) - (u, v)\|$ we obtain the optimal radius for the voting region

$$\rho(u, v) = \epsilon \sqrt{(4(u^2 + v^2) + 3) \cdot ln \left(\frac{4(u^2 + v^2) + 3}{12\epsilon^2} \right)}, \tag{8}$$

where $\epsilon = \frac{\sigma}{\|\mathbf{p}_\nu - \mathbf{p}_\mu\|}$. To simplify the derivation we assume that model points are distant enough from each other, so that $f(u, v)$ is defined by the Gaussian centered at the closest model point (x_i, y_i), while the rest is negligibly small. The covariance matrix of this Gaussian distribution can be approximated by $C(u, v)$.

3. Rehashing for Bayesian recognition

The Bayesian approach, presented in the previous section optimizes the recognition rate. The other parameter we wish to optimize as discussed in Section 1 is the computational efficiency.

The hash table bin size should be chosen to minimize the expected access time to the entries within the voting region. This could be achieved by making bins proportional to voting region. However, from equation (8), the voting region size varies significantly over the hash space. Therefore, there is no fixed optimal bin size for the whole hash table.

To overcome this problem we can re-map the hash entries to make the voting region size constant throughout the hash table. Actually the technique of re-mapping, also known as rehashing, is not new in the field and was used to uniformly distribute hash entries over the table [6]. This makes sense in the classical GH approach, where the voting is preformed in a single bin. However, in the Bayesian framework such a re-mapping is not optimal.

Formally, we need to define a mapping $T : (u, v) \to (u', v')$, such that $\rho'(u', v') = 1$. Then

$$\rho'(u', v') = \rho(u, v) J, \tag{9}$$

where

$$J = \det \begin{pmatrix} \frac{\partial u'}{\partial u} & \frac{\partial v'}{\partial u} \\ \frac{\partial u'}{\partial v} & \frac{\partial v'}{\partial v} \end{pmatrix} \tag{10}$$

is the Jacobian of T. In other words, we are looking for T solving the following differential equation

$$det \begin{pmatrix} \frac{\partial u'}{\partial u} & \frac{\partial v'}{\partial u} \\ \frac{\partial u'}{\partial v} & \frac{\partial v'}{\partial v} \end{pmatrix} = \frac{1}{\rho(u, v)}. \tag{11}$$

We find that the transformation T can be well approximated by

$$\begin{cases} u' = \frac{\pi^2}{\sqrt{r+e^2}} ln(1+r) \\ v' = \arctan(\frac{v}{u}) \end{cases}, \quad (12)$$

where $r = \sqrt{u^2 + v^2}$.

Figure 1 presents (a) the initial voting regions distribution described by Equation (8) and voting regions transformed by (b) uniform density rehashing and (c) by our rehashing scheme. The varying voting region size in cases (a) and (b) lowers the recognition rate for the classical single-bin voting and the computational performance for the Bayesian schemes.

4. Experimental Results

To verify the theoretical results we performed a number of simulations and compared several variations of GH algorithms.

We experimented with two voting schemes: (a) classical single bin voting - query point votes for all the entries found in the hash bin it falls in, and (b) Bayesian approach - query point votes for all the points within its voting region. For each one of the voting schemes three rehashing options were evaluated: (i) no rehashing, (ii) uniform density rehashing and (iii) the uniform voting region rehashing suggested here. For each one of the six voting/rehashing combinations we tested nine different hash bin sizes.

We used five models, each consisting of 20 points randomly scattered over a unit square. 500 queries were formed by adding a Gaussian noise with $\sigma = 0.025$ to the models and randomly choosing a two-point basis. Small bases ($\|\mathbf{p}_\nu - \mathbf{p}_\mu\| < 0.2$) where discarded as unreliable. Recognition rate was taken as the relative number of queries for which the correct model/basis combination received the maximal voting score. The computational performance was estimated by counting the number of the accessed hash entries.

Figure 2 presents the recognition rate as a function of the bin size. Three graphs for the single bin voting are shown (no rehashing, uniform density and uniform voting region rehashings). The fourth graph corresponds to the Bayesian voting, for which the recognition rate does not depend on the rehashing scheme or the bin size.

As expected, the Bayesian voting yields the best recognition rate. Both rehashing schemes improve the recognition rate for the single bin voting. As one can see, the proposed uniform voting region rehashing outperforms the uniform density rehashing for any bin size.

Obviously, for unreasonably small bin sizes rehashing schemes become ineffective, since only a small part of the voting region is covered by the bin, and therefore most of the bins miss the relevant vote. On the other hand, without rehashing, bins in the densely populated areas still contain the relevant vote.

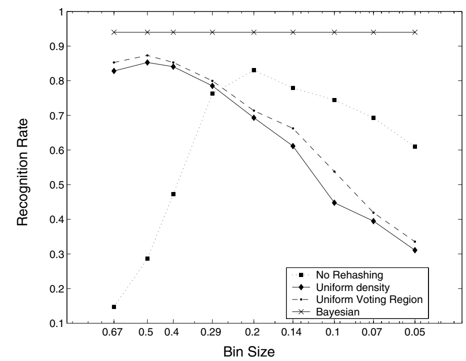


Figure 2. Recognition rates in classical scheme for three different rehashing schemes and Bayesian scheme.

Figure 3 presents the number of the accessed entries for the single bin voting scheme. This can be seen as an implementation independent indicator of the time complexity. As expected, the uniform density rehashing has the best performance, since any non-uniformity in distribution results in a higher probability to hash into a more populated bin. The proposed uniform voting region rehashing, as a side effect, also distributes hash entries more uniformly. Therefore, the number of the accessed entries in this case decreases as well.

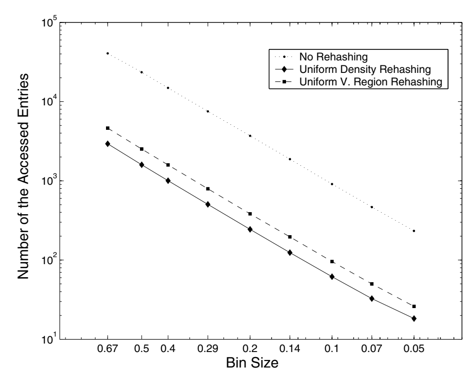
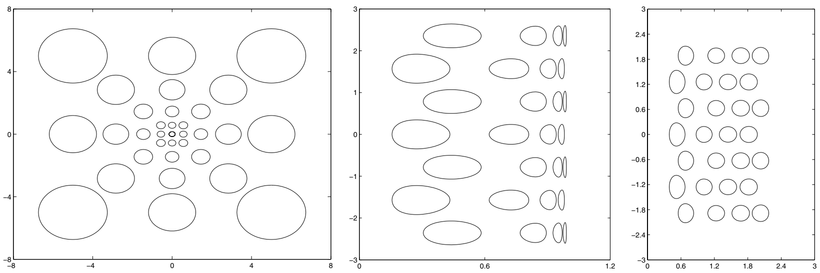


Figure 3. Number of accessed entries for the single bin voting scheme.

We apply the proposed rehashing scheme in the system built for navigation on wafers, which is defined as following: given an "eye point" (e.g. partial image of the wafer)

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE



(a) without rehashing (b) after uniform density rehashing (c) after uniform voting region rehashing

Figure 1. Voting regions with and without rehashing.

determine its exact position on the wafer map. This system uses GH to establish a correspondence between the observed eye-point and the known in advance wafer map. Wafer map can be divided into many adjacent parts to be identified during navigation. These parts correspond to models in pattern recognition framework, whereas the eye-point plays a role of a query pattern. Matching the eye-point to one of the previously prepared parts of the wafer map during navigation is essentially the same, as associating a query pattern to one of the predefined models in pattern recognition. An example of the wafer eye-point and the corresponding part of the wafer map is shown in Figure 4.

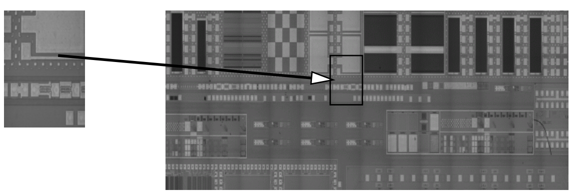


Figure 4. Wafer eye-point within a wafer map.

This approach answers a demand from machine vision tools to become more adaptive to in-process variations and allow navigation on wafers despite changes in visual appearance occurring during the manufacturing process. Such changes may include non-linear contrast variation, rescaling, rotations and partial pattern obliteration [3].

The proposed method proved high reliability and robustness to process variations, while its decreased navigation time (due to the proposed scheme) makes it appropriate for inline use.

5. Summary

In this paper we presented a new rehashing scheme for geometric hashing. The scheme is consistent with the Bayesian voting approach and optimizes its computational performance. In addition the scheme can be used for the traditional single bin geometric hashing yielding a higher recognition rate compared to the known uniform density rehashing. Our theoretical results were confirmed by a comparative experimental analysis.

References

[1] W. Grimson, D. Huttenlocher, and D. Jacobs. A study of affine matching with bounded sensor error. In *Proceedings of the 2nd European Conference on Computer Vision*, pages 291–306, 1992.

[2] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. *Proceedings of the International Coference on Computer Vision*, pages 238–249, 1988. Computer Society Press, Los Alamitos, CA, 1988.

[3] S. Melikian. Geometric searching improves machine vision. *Lasers and Optronics*, 18(7):13, July 1999.

[4] I. Rigoutsos and R. Hummel. Implementation of geometric hashing on the connection machine. In *Proceedings of the IEEE Workshop on Directions in Automated CAD-Based Vision*, Los Alamos, CA, June 1991. Computer Society Press.

[5] I. Rigoutsos and R. Hummel. Robust similarity invariant matching in presence of noise. In *Proceedings of the 8th Israeli Conference on Artificial Intelligence and Computer Vision*, Tel Aviv, Israel, December 1991.

[6] I. Rigoutsos and R. Hummel. Several results on affine invariant geometric hashing. In *Proceedings of the 8th Israeli Conference on Artificial Intelligence and Computer Vision*, Tel Aviv, Israel, December 1991.

[7] I. Rigoutsos and R. Hummel. A bayesian approach to model matching with geometric hashing. *Computer Vision and Image Understanding*, 62(1):11–26, 1995.

[8] J. Schwartz and M. Sharir. Identification of paritially obscured objects in two and three dumensions by matching noisy characteristic curves. *International journal of Robotics Research*, 6(2):20–44, 1986.

[9] H. J. Wolfson and I. Y. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.

IEEE COMPUTER SOCIETY