Eurographics Symposium on Geometry Processing (2006)
Konrad Polthier, Alla Sheffer (Editors)

Probabilistic Fingerprints for Shapes

Niloy J. Mitra [†] Leonidas Guibas[†] Joachim Giesen[‡] Mark Pauly[§]

[†] Stanford University [‡] Max-Planck Institut für Informatik [§] ETH Zurich

Abstract

We propose a new probabilistic framework for the efficient estimation of similarity between 3D shapes. Our framework is based on local shape signatures and is designed to allow for quick pruning of dissimilar shapes, while guaranteeing not to miss any shape with significant similarities to the query model in shape database retrieval applications. Since directly evaluating 3D similarity for large collections of signatures on shapes is expensive and impractical, we propose a suitable but compact approximation based on probabilistic fingerprints *which are computed from the shape signatures using Rabin's hashing scheme and a small set of random permutations. We provide a probabilistic analysis that shows that while the preprocessing time depends on the complexity of the model, the fingerprint size and hence the query time depends only on the desired confidence in our estimated similarity. Our method is robust to noise, invariant to rigid transforms, handles articulated deformations, and effectively detects partial matches. In addition, it provides important hints about correspondences across shapes which can then significantly benefit other algorithms that explicitly align the models. We demonstrate the utility of our method on a wide variety of geometry processing applications.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling.

1. Introduction

There has been great progress in recent years in the areas of shape acquisition and modeling, resulting in large collections of digital geometric models. Classification, navigation, and usability of such shape databases largely hinge on the following operation: Given two shapes in arbitrary poses, how can we meaningfully define their similarity and evaluate it efficiently? For database applications, off-line preprocessing of each shape is typically acceptable, if it results in fast query handling. In the same context, it is important to determine, quickly and reliably, when two shapes are *dissimilar*.

At a fine level, global shape similarity is traditionally estimated by comparing optimally aligned models: Translation is factored out by aligning the respective centroids, while rotation is handled using principal component analysis (PCA) for the final alignment. Alternatively, at a coarser level and so as to avoid the alignment step, global shape descriptors can be computed that are invariant under rigid transformations. Shape similarity is then estimated indirectly by comparing these descriptors.

Global approaches fail for partial similarity, which is much more challenging, as factoring out arbitrary rigid transforms without any knowledge of regions of overlap between shapes is difficult. In such cases, we can use more specialized techniques like geometric hashing [GCO06], or explicitly determine correspondence between models and use it for alignment [GMGP05]. However, these methods require high storage or processing time, even if the two shapes are very different.

In this paper we propose an efficient method to define probabilistic fingerprints for 3D shapes and use it to estimate partial similarity. Our approach is complementary to existing work on shape descriptors and signatures, as we can make use of available shape descriptors to define partial similarity across multiple shapes in arbitrary poses. We compress these descriptors using a probabilistic hashing scheme motivated by ideas from the database community. Our fingerprints are such that if they largely disagree, then we

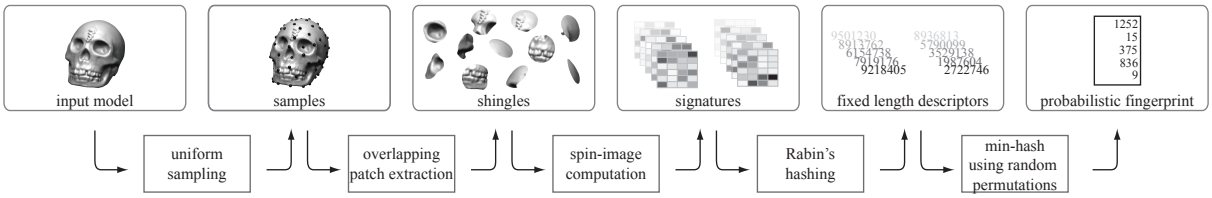

Figure 1: Fingerprint generation. *We first cover an object with ρ-radius balls spaced δ apart with $\rho \gg \delta$. The intersection of each ball with the surface defines a* shingle. *For each shingle, we compute a descriptor, spin-image in this case, which is hashed using Rabin's scheme. Then, in the min-hashing phase, according to m random permutations, we select a small subset of descriptors and store them as the* probabilistic fingerprint.

can claim with high certainty that the corresponding shapes are dissimilar. This yields an efficient way to quickly filter large shape collections when searching for objects matching a particular model.

In our approach to partial shape similarity, we first cover a given 3D shape with a large collection of overlapping patches. Each patch is mapped to a point in a high dimensional space using a compact, local descriptor that is invariant to rigid transformations. We do not preserve any information about the relative spatial ordering of the patches. The shape is thus mapped to an unordered point set in a high dimensional signature space. We select descriptors that are robust to perturbations, so that patches which are very similar are likely to be mapped to the same locale of this signature space. This is important, as similar regions may not be covered by patches in exactly the same way across two shapes. Clearly, if two shapes are similar, then the corresponding point sets will have proximal regions in proportion to the partial similarity between the original objects. However, since we lose relative patch ordering, it is possible that two largely different shapes have a significant overlap in signature space. Statistically this is a rare event and leads to only a few false positives. It is made even more unlikely by ensuring large overlap between neighboring patches. Motivated by this intuition, we define similarity between two shapes in terms of the similarity between the signature sets. Our definition is invariant to rigid transforms, handles partial matching, and is robust to local deformations and articulated motion.

However, these large high-dimensional point sets have high storage requirements and are difficult to compare efficiently. We therefore compress signature information using a technique called min-hashing [Bro97] to generate a short *probabilistic fingerprint* for each signature set. Subsequently, fingerprints of multiple shapes are compared to estimate similarity between the signature sets, and hence between the original objects. We first map the signatures to a finite universe of numbers using Rabin's hashing scheme [Rab81]. Then, during min-hashing, we use a random permutation to assign a complete ordering to all elements of this finite universe of numbers. We can think of this ordering as the ranking of an 'expert', asked to evaluate the patches according to her criteria. According to the expert ranking, we then select the winner among the set of hashed signatures corresponding to an object. For each object, we collect the winners of m randomly chosen permutations and save them as the *probabilistic fingerprint* of the shape. The same random permutations are used for all shapes. This ensures that patches from different shapes are consistently ordered, according to each of the m chosen 'experts'.

We can efficiently detect if two shapes are similar using our shape fingerprints. However, as mentioned before, we can get a few false positive matches. In practice, the number of such false hits is very small and can be handled by match verification using more expensive partial similarity methods. Further, we can show that if two fingerprints are different, then with high probability the shapes are also different. Thus both false positives and false negatives are bounded.

We provide a probabilistic analysis of our scheme to show the attractive property that the cost of our algorithm depends on the confidence we want from our estimates, and not on the complexity of the shapes themselves. The storage required for preprocessing and query stages depends on the length of the fingerprint m. Finally, though our algorithm only assigns similarity scores between multiple objects without explicitly determining an alignment, it gives important hints about the regions of overlap and correspondences. We apply our fingerprints to address a variety of geometry processing applications, including shape retrieval, automatic scan alignment, adaptive feature point selection, and mesh authentication.

Contributions

We propose a new statistical approach to efficiently estimate partial or total shape similarity. We introduce the concept of probabilistic fingerprints for 3D

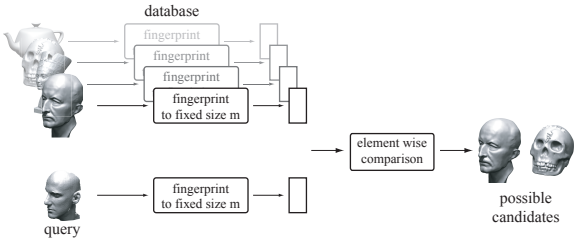

Figure 2: Query Processing. *A query object is first processed to generate its fingerprint using the same parameters used to pre-process the database shapes. Objects with fingerprints similar to fingerprint of the query are returned as possible candidates.*

shapes, provide a statistical analysis on its effectiveness for partial matching, and show its practical use on a number of different applications. The key insight is that the similarity of two shapes can be estimated by comparing signatures derived from very sparse sets of local patches generated on each model. Based on probabilistic arguments, we show how to *preselect* such patches, *without first needing to establish explicit correspondence relations between the models.* This approach produces a fixed-length fingerprint and avoids costly explicit alignment of the models. Our similarity measure is invariant to rigid transforms, robust to perturbations, handles articulations, and most importantly, detects partial matches.

2. Related Work

The problem of shape similarity and retrieval has been extensively studied in computer vision and graphics. It has been addressed in great detail by the extensive work done by the Princeton Shape Retrieval and Analysis Group [FKS*04, FKMS05]. Meaningful similarity between two 3D shapes, partial or whole, has to be invariant to rigid transforms. Global shape descriptors, invariant to rigid transformations, include spherical harmonics [KFR03], shape distributions [OFCD02], reflective symmetry descriptors [KCD*04], and Laplace-Beltrami Spectra [RWP05]. Similarity estimation between two models is then reduced to a comparison of the corresponding global shape descriptors. Alternatively, an object can be canonically oriented using principal component analysis (PCA) and descriptors computed on the rotation normalized shape – examples include extended Gaussian images [Hor84] and shape histograms [AKKS99].

However, all of these global methods are less suitable for detecting partial matches. This problem can be addressed by establishing an explicit correspondence across feature points of the models to compute a good alignment [GMGP05]. Such a solution involves exhaustively considering the various correspondence assignments and is thus computationally expensive. Gal and Cohen-Or [GCO06] proposed a different method for determining partial similarity using geometric hashing techniques. Briefly, in a pre-processing stage, geometric hashing encodes all the possible candidate transforms in a large hash table. While this approach is more efficient, it trades computation time for memory, leading to space requirements of multiple gigabytes even for moderately complex models.

In a different setting, the problem of identifying text or web documents with partial similarity has been extensively studied. Effective solutions to this problem involve clever combinations of hashing and random sampling techniques [Blo70, SGM98, Bro00]. In these schemes, a text document is first converted to a set of overlapping text segments. Similarity between two documents is assigned based on the size of the intersection of the segment-sets which is efficiently estimated using random sampling techniques. Some of these concepts motivated our approach. Our problem, however, is significantly more challenging, as digital 3D shapes have neither the linear ordering nor the canonical decomposition into discrete tokens that is exploited in the text document case.

3. Shape Fingerprinting

Our goal is to reliably and efficiently estimate (partial) similarity between two shapes. The similarity measure should be invariant to rigid transforms and robust to small perturbations. Here we define such a similarity measure for a restricted class of shapes, namely surfaces in $\mathbb{R}^3$ whose normal is defined almost everywhere, e.g., a smooth surface (implicit or explicitly parameterized) or a triangle mesh. The measure is based on surface signatures that allow for effective compression using hashing. We call a hashed signature a fingerprint and start by listing the properties we expect in general from a shape fingerprint.

Fingerprint properties. A *probabilistic fingerprint* is a function f that assigns to each admissible shape a fixed size bit string, i.e. a string in $\{0,1\}^m$. The main purpose of fingerprints is to allow for efficient comparison of shapes. Given a definition of shape similarity (or dissimilarity), any meaningful fingerprint function should have the following properties.

1. Given two shapes S_1 and S_2, we want the following relations to hold with high probability:
   a. If $f(S_1) \neq f(S_2)$, then S_1 and S_2 are dissimilar.
   b. If $f(S_1) = f(S_2)$, then S_1 and S_2 are similar.
2. The number of bits m is small compared to the number of bits needed to encode the actual shapes.

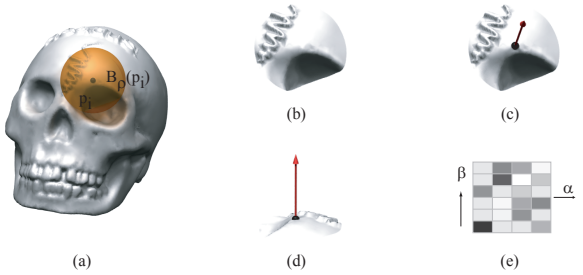

Figure 3: Shingle generation. *(a) $B_\rho(p_i)$ is the neighborhood ball for a point p_i. (b) The selected surface patch (shingle) T_i around p_i. (c) The patch along with the surface normal at p_i. (d) The normal oriented along the z-axis. (e) Computed spin image for patch T_i. The signature is invariant to rigid transforms, and robust to sampling and small surface perturbations.*

3. The function f is efficiently computable. Also $f(S_1) = f(S_2)$ can be quickly checked.

In the following we describe in detail a fingerprint and the corresponding notion of similarity/dissimilarity it is based on. The pipeline for computing the fingerprint is depicted in Figure 1.

Our fingerprint relies on the concepts of *sample*, *shingle*, *signature*, *resemblance*, and *hashing*. Next we describe these concepts and how we use them for defining and computing a probabilistic fingerprint.

Sample. In the first step, we generate a set of (approximately) uniformly sampled points on the input shape S. Let $P = \{p_1, \ldots, p_n\}$ be the set of sample points with sampling spacing δ. For our fingerprint to work well we assume that for any $p \in S$, $\exists\, p_i \in P$ such that $\|p_i - p\| \leq \delta$. Further, the number of such neighboring points is bounded by a small constant, preventing the sampling from being arbitrarily dense.

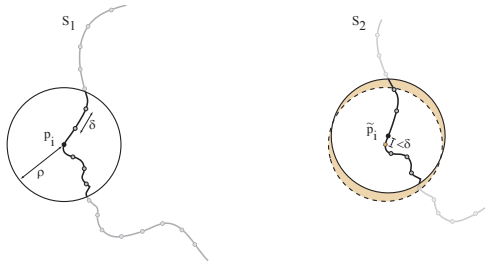

Figure 4: Overlapping Shingles. *Shingles for two shapes S_1 and S_2 are computed using ρ-radius balls spaced roughly δ apart. If $\tilde{p}_i$ lies in a matching region between S_1 and S_2, then with high probability the shingle at $\tilde{p}_i$ will have a corresponding shingle from S_1 with a significant overlap as $\rho \gg \delta$.*

There is a simple and efficient process for generating such a sample set: Let A be the surface area of the shape. On the surface of the object, we randomly place $n = \lceil A/\pi\delta^2 \rceil$ samples and uniformly spread them out using particle repulsion [Tur92].

Shingles. For each sample point $p_i \in P$, we define a neighborhood of radius ρ where $\rho \gg \delta$. A surface patch $T_i \subset S$ corresponding to point p_i is obtained as $T_i = S \cap B_\rho(p_i)$ where $B_\rho(p_i)$ denotes the ball of radius ρ around p_i (Figure 3). If multiple components are present in T_i, we retain only the component containing p_i (the surface is assumed to be a manifold). We refer to these patches as *shingles* and denote the multi-set of all shingles by $\mathcal{P}$. Keep in mind that $\mathcal{P}$ depends on the sample P. Given two shapes S_1 and S_2, with high probability, any shingle from the matching region has a corresponding shingle on the other shape with significant overlap (Figure 4).

Signatures. We compute a signature σ_i for each shingle $T_i \in \mathcal{P}$ and denote the multi-set of all signatures by $\mathcal{S}$. A signature σ_i is essentially a string that represents a shingle T_i. Any signature that is invariant to rigid transforms and robust to sampling and local perturbations can be used to this end. Here we use spin images [Joh97] which are defined as follows: Let the surface normal at any sample point $p_i \in P$ be n_i. For any point x in the corresponding shingle T_i, its spin-map is defined as:

$$(\alpha, \beta) = \left(\sqrt{\|\bar{y}\|^2 - \langle n_i, \bar{y} \rangle^2}, \langle n_i, \bar{y}/\|\bar{y}\| \rangle \right)$$

where $\bar{y} = x - p_i$. The spin-image s_i of T_i is simply the quantized version of the (α, β) space recording the spin-map of the points of T_i falling into a set of discrete bins (Figure 3). Since spin images are robust to perturbations, if two shingles have significant overlap then they are likely to have the same signatures.

Resemblance. Now we introduce our similarity/dissimilarity measure. Given two surfaces S_1 and S_2 we define their resemblance r with respect to their corresponding signatures $\mathcal{S}_1$ and $\mathcal{S}_2$. Remember that $\mathcal{S}_1$ and $\mathcal{S}_2$ are multi-sets. For each $\sigma \in \mathcal{S}_i$ let $m_i(\sigma)$ denote its multiplicity in $\mathcal{S}_i$. The resemblance of S_1 and S_2 is defined as:

$$r(S_1, S_2) = \frac{|\mathcal{S}_1 \cap \mathcal{S}_2|}{|\mathcal{S}_1 \cup \mathcal{S}_2|},$$

where $|\mathcal{S}_1 \cap \mathcal{S}_2|$ denotes

$$\sum_{\sigma \in \mathcal{S}_1 \cap \mathcal{S}_2} \min(m_1(\sigma), m_2(\sigma))$$

and $|\mathcal{S}_1 \cup \mathcal{S}_2|$ denotes

$$\sum_{\sigma \in \mathcal{S}_1 \cup \mathcal{S}_2} \max(m_1(\sigma), m_2(\sigma)).$$

Since resemblance $0 \leq r(S_1, S_2) \leq 1$ is higher when two shapes are similar, we define distance between shapes as $1 - r(S_1, S_2)$. Observe that this definition is based on the signature sets, and hence depends on the scale parameter ρ used to define the shingles.

Finally, we want to estimate the resemblance using a much sparser representation for the shapes than their signatures, namely their fingerprints. To this end each signature is first hashed into a finite set $\mathcal{U}$ using Rabin's hashing scheme.

Hashing. Rabin's hashing scheme [Rab81] gives a low collision probability for a fixed bit budget by working with irreducible degree k polynomials over $\mathbb{Z}_2$. Let the number of bits required to represent a signature be t. For instance, if a spin image is computed using b bins and each bin is of length l bits, then t is upper bounded by the maximum length of the spin images which is $\log(b2^l)$. If Rabin's hashing scheme h maps each of the n signatures σ_i corresponding to a shingle T_i down to k bits, then the probability of collision is bounded by

$$\Pr[h(\sigma_i) = h(\sigma_j) | \sigma_i \neq \sigma_j] \leq n^2 t / 2^k. \qquad (1)$$

For example, if $n = 10^8$ and $t = 128$ then for $k = 80$ the probability of collision is less than 10^{-6}. Thus even with 10 bytes for each signature, we get low collision probability. Further, Rabin's hashing scheme can be very efficiently computed using simple bit arithmetic [Bro93, CL01]. For each signature we store only k bits corresponding to the coefficients of the degree k polynomial in $\mathbb{Z}_2$. We denote the universe of all k-bit numbers by $\mathcal{U}$ and we denote the multi-set of all hashed signatures as $\mathcal{I}$.

We can define the analog of our resemblance function r for multi-sets of hash values as

$$r'(S_1, S_2) = \frac{|\mathcal{I}_1 \cap \mathcal{I}_2|}{|\mathcal{I}_1 \cup \mathcal{I}_2|},$$

where $\mathcal{I}_i$ is the multi-set of hash values corresponding to surface S_i. Evaluating this function instead of $r(S_1, S_2)$ remains impractical, as the involved multi-sets are still too large even though we need less bits to store their elements than we need to store the original signatures. Moreover, set operations between these large unordered multi-sets require $O(n_i \log n_i)$ time where n_i is the number of set elements. As a solution, we further compress each of the multi-sets $\mathcal{I}$ of hash values to generate a small fingerprint. This is done by *min-hashing* using random permutations on the universe $\mathcal{U}$.

Probabilistic Fingerprint. Let $\pi_1, \ldots, \pi_m$ be m random permutations on $\mathcal{U}$, the universe of k-bit numbers. Intuitively, each permutation is like an 'expert' assigning an ordering to $\mathcal{U}$ according to her criteria. Given a multi-set $\mathcal{I}$ of hash values we use the random permutations π_i to compress the set as follows: We replace the multi set $\mathcal{I}$ by the length m sequence of strings obtained as

$$f(S) = \left(\min\{\pi_1(\mathcal{I})\}, \ldots, \min\{\pi_m(\mathcal{I})\}\right),$$

where the corresponding multiplicities are propagated in the obvious way. This sequence is our definition of a *fingerprint* for a surface S. To generate the permutations π_i, we apply 2-universal hashing [MR00] as an approximation for random permutations, using a random pair of numbers as parameters.

Based on the fingerprints we estimate the resemblance $r(S_1, S_2)$ by

$$\hat{r}(S_1, S_2) = \frac{\sum_{j=1}^{m} \min(m_{1j}, m_{2j}) \chi(f(S_1)_j = f(S_2)_j)}{\sum_{j=1}^{m} D_j},$$

with

$$D_j = max(m_{1j}, m_{2j}) \chi(f(S_1)_j = f(S_2)_j) + m_{1j} \chi(f(S_1)_j < f(S_2)_j) + m_{2j} \chi(f(S_1)_j > f(S_2)_j)$$

where $\chi(\cdot)$ is the indicator function taking value 1 if the condition of its argument evaluates to *true*, and 0 otherwise. For the surface S_i, the j-th component of its fingerprint is $f(S_i)_j$ with multiplicity m_{ij}. When the fingerprint consists of strings with all multiplicities equal to one, the resemblance estimate reduces to $\hat{r}(S_1, S_2) = \sum_j \chi(f(S_1)_j = f(S_2)_j)/m$. Notice that to compare two fingerprints, we simply need to compare them element-wise without any need to solve for correspondences.

In the next section we show that choosing a large enough m gives, with high probability, a good estimate of resemblance. In practice $m \approx 1000$ is sufficient and hence the probabilistic fingerprints, in the order of 10KBytes, are very compact.

4. Analysis

In this section we analyze the performance of our fingerprints — as mentioned before our goal is to approximate the resemblance of two surfaces effectively and efficiently.

Rabin's hashing scheme maps any signature σ to a number with bit length k. This mapping obviously results in some collisions that can be quantified as:

$$\sigma_i = \sigma_j \;\Rightarrow\; h(\sigma_i) = h(\sigma_i)$$
$$\sigma_i \neq \sigma_j \;\Rightarrow\; \Pr[h(\sigma_i) = h(\sigma_j)] \leq p, \qquad (2)$$

where p is $n^2t/2^k$, see Equation 1. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be multi-sets of signatures for the surfaces S_1 and S_2, respectively. Let

$$\mathcal{A} = \mathcal{S}_1 \cap \mathcal{S}_2,\ \mathcal{B} = \mathcal{S}_1 \setminus \mathcal{S}_2, \text{ and } \mathcal{C} = \mathcal{S}_2 \setminus \mathcal{S}_1,$$

where the set operations again are defined in the multi-set setting, i.e. the $\setminus$ operation respects the multiplicities. We can relate $a = |\mathcal{A}|$ (size of a multi-set is the sum of the multiplicities of its elements), $b = |\mathcal{B}|$, and $c = |\mathcal{C}|$ to the resemblance of S_1 and S_2 as follows:

$$r(S_1, S_2) = a/(a+b+c). \tag{3}$$

Let $\mathcal{I}_1$ and $\mathcal{I}_2$ denote the multi-sets of hashed signatures for S_1 and S_2, respectively. Having set the terminology we now quantify the errors incurred by Rabin's hashing scheme.

Using Equation 2 and the definitions of a, b, and c in expectation we get:

$$\begin{array}{ccccc} a & \leq & |\mathcal{I}_1 \cap \mathcal{I}_2| & \leq & a+d \\ a+b+c-d & \leq & |\mathcal{I}_1 \cup \mathcal{I}_2| & \leq & a+b+c, \end{array}$$

where $d = (2bc+ac+ab)p$ is obtained by a simple union bound argument: For any element in $\mathcal{B}$ the probability to participate in a collision that affects the set operations is upper bounded by $(a+c)p$. Thus by linearity of expectation the expected number of such collisions contributed by $\mathcal{B}$ is upper bounded by $(a+c)bp$. Similarly, the expected number of collisions contributed by $\mathcal{C}$ is upper bounded by $(a+b)cp$. Adding these two bounds gives the bound d.

For the approximation quality of the resemblance by $r'(S_1, S_2) = |\mathcal{I}_1 \cap \mathcal{I}_2|/|\mathcal{I}_1 \cup \mathcal{I}_2|$ we get in expectation the following bounds:

$$r(S_1, S_2) = \frac{a}{a+b+c} \leq r'(S_1, S_2) \leq \frac{a+d}{a+b+c-d}.$$

Crucial is only the upper bound which we can also write as

$$\frac{a+d}{a+b+c-d} = \frac{r(S_1, S_2)+\epsilon}{1-\epsilon},$$

if we set $\epsilon = d/(a+b+c)$. By increasing k, i.e. the number of bits each signature gets mapped to, we can make ϵ arbitrarily small. For small enough ϵ and assuming $r(S_1, S_2) > 0$ we get in expectation

$$\begin{aligned} r'(S_1, S_2) &\leq \frac{r(S_1, S_2)+\epsilon}{1-\epsilon} \leq (r(S_1, S_2)+\epsilon)(1+\epsilon) \\ &\leq r(S_1, S_2)(1+\sqrt{\epsilon}). \end{aligned}$$

Using Markov's inequality this yields

$$\begin{aligned} &\Pr[r'(S_1, S_2) \geq \lambda(1+\sqrt{\epsilon})r(S_1, S_2)] \\ &\leq \Pr[r'(S_1, S_2) \geq \lambda \mathrm{E}[r'(S_1, S_2)]] \\ &\leq 1/\lambda. \end{aligned} \tag{4}$$

model	# vts.	uniform samp.	spin image	Rabin hash	min hash
skull	54k	0.8	7.5	0.05	4.5
Caesar	65k	1.4	7.3	0.08	10.3
bunny	121k	1.8	13.8	0.04	2.9
horse	8k	0.7	5.7	0.05	7.3

Table 1: Performance. *Timing in seconds for the different stages of the fingerprint computation ($m = 1000$) on a 3 GHz Pentium 4 with 2GB RAM. Caesar and bunny refer to the complete models. Average query time is roughly 15msec.*

Finally, we have to check how our estimate behaves under the random permutations that we used to compute the fingerprints. When all the strings have multiplicities one, we use the following fact, see [Bro97],

$$\Pr[f(S_1)_j = f(S_2)_j] = r'(S_1, S_2),$$

for all $j = 1, \ldots, m$. Hence estimating $r'(S_1, S_2)$ by using m random permutations is equivalent to performing m coin tosses to evaluate the bias of the coin. Using strong Chernoff bounds we can bound the estimated resemblance $\hat{r}(S_1, S_2)$ as

$$\begin{aligned} &\Pr[(1-\delta)r'(S_1, S_2) \leq \hat{r}(S_1, S_2) \leq (1+\delta)r'(S_1, S_2)] \\ &\quad \geq 1-\eta, \end{aligned} \tag{5}$$

if $m \geq 4\ln(2/\eta)/(\delta^2 r'(S_1, S_2))$.

Combining our probabilistic bounds by taking a union bound for the event we dealt with in Equation 5 and the complement of the event we dealt with in Equation 4 we conclude that

$$(1-\delta)r(S_1, S_2) \leq \hat{r}(S_1, S_2) \leq \lambda(1+\delta)(1+\sqrt{\epsilon})r(S_1, S_2)$$

with probability at least $1-(\eta+1/\lambda)$. That is, with high probability $\hat{r}(S_1, S_2)$ is very close to $r(S_1, S_2)$. The analysis can be easily extended for multi-sets giving us a similar result. Observe that the size m of the fingerprint depends on the desired confidence in our estimated resemblance.

5. Results and Applications

We have implemented the framework shown in Figure 1. Along with each fingerprint, we store some additional header information: a seed for the random number generator, sample spacing δ, shingle radius ρ, parameters for computing spin-images, and k, the degree of the polynomial used in Rabin's hashing scheme. The choice of these parameters is not critical for the success of our scheme provided the conditions given in Section 4 are satisfied. However, we can only compare

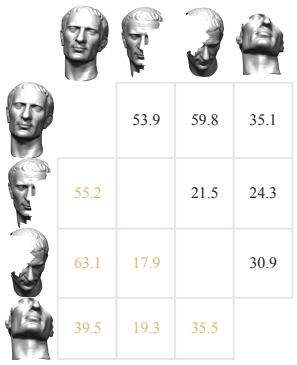

Figure 5: Resemblance between Partial Scans. *In black, resemblance (in %) computed using fingerprints. In yellow, approximate ground truth computed from spin-images. Our resemblance definition being symmetric, difference between diagonally opposite elements quantifies the corresponding estimation error.*

fingerprints computed using consistent sets of parameters. Typical time requirements for the various stages are shown in Table 1.

Partial Matching. Our scheme is tailored to detect partial matches efficiently. In an experiment we take a bust of Caesar along with its three partial scans (Figure 5). The triangulations of the models are very different and thus test the robustness of our scheme. For each model, we independently compute their probabilistic fingerprint. Then, for each model pair, we compute its resemblance using the corresponding fingerprints (shown in black). For comparison, we compute the ground truth resemblance (shown in yellow) via spin image signatures. Since our resemblance measure is symmetric, the difference between diagonally opposite elements in the resemblance matrix quantifies our estimation error.

Articulated Motion. Resemblance, as measured by our scheme, is robust to articulated deformations. If

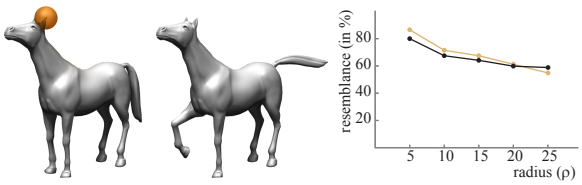

Figure 6: Resemblance between Articulated Shapes. *The yellow ball shows a neighborhood with $\rho = 10$ used for defining shingles. At low values of ρ, we get a high resemblance, since the effect of articulation is felt only by few of the shingles. As ρ increases, resemblance goes down. In yellow, we show the ground truth.*

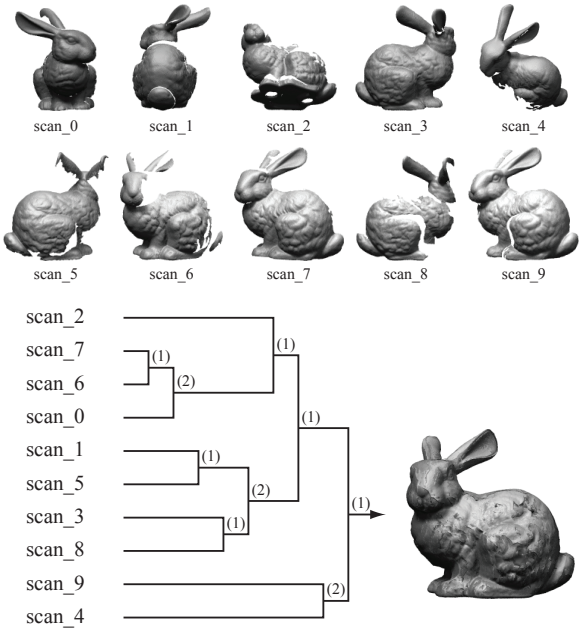

Figure 7: Automatic Scan Alignment. *Given ten initial scans of the Stanford bunny in* arbitrary *poses, for each scan we compute its probabilistic fingerprint with $m = 1000$. Scan pairs with highest resemblance are picked, and then using [GMGP05] their alignment verified. If the scan-pair align, the fingerprint for the merged scan is estimated. The process continues until no more scans can be combined. The number of required global alignment steps is shown in parenthesis.*

large chunks of a model are rigidly deformed across two poses, then the corresponding shingles and their hashed descriptors are also preserved. Results on two articulated poses of a horse model are shown in Figure 6. The size of the shingle, determined by ρ, affects the resemblance score: smaller ρ gives higher resemblance and vice versa. The ground truth (yellow curve), determined using the spin-image signatures, is within $\pm 5\%$ of our estimated values.

Automatic Scan Alignment. The problem of automatic scan alignment has been previously addressed by Huber and Hebert in [HH03]. Their system can be made significantly faster using our scheme. We explain our method with reference to scans (in arbitrary initial orientations) of the Stanford bunny (Figure 7). In the pre-processing stage, for each of the ten scans, we independently compute its fingerprint. Now for each pair of models, we estimate their resemblance using the respective fingerprints and store the edge joining that pair in a heap with the largest element on top. We then extract the top edge, try to explicitly align the corresponding patches using a global aligner [GMGP05],

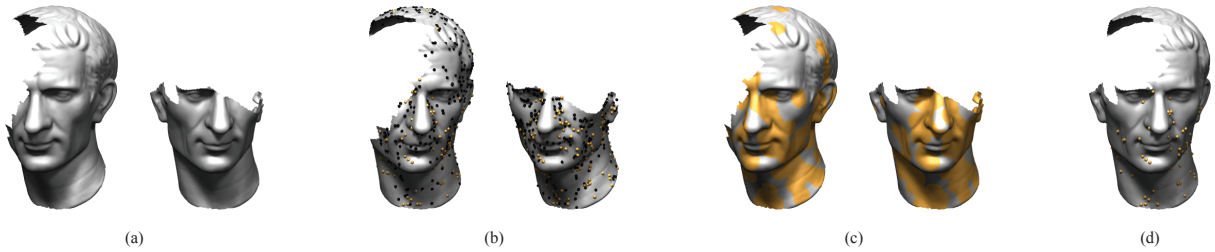

Figure 8: Adaptive Feature Point Selection. *(a) Two shapes in arbitrary poses. (b) For each scan, black and yellow balls denote shingle centers chosen by min-hash. In yellow, shingle centers whose min-hashes agree across the two models. (c) Hints about possible overlap regions obtained by mapping matching min-hashes back onto the objects. These are used for adaptive feature point selection. (d) Final alignment using chosen feature points. The set of features with correct correspondence is shown in yellow.*

and if the alignment is not valid we just pick the next largest edge. If an alignment is valid, we merge the respective patches, and need to compute the fingerprint for the merged patch. However, given two patch fingerprints $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_m)$, we can very efficiently estimate the fingerprint for the merged scan simply as $(\min(a_1, b_1), \ldots, \min(a_m, b_m))$ without explicitly computing the fingerprint for the merged scans. Using this estimated fingerprint, the heap can be efficiently updated by re-evaluating only the affected edges. In the figure, the number of required global registration or verification steps are shown in parenthesis. Since global registration is much more costly compared to fingerprint matching, our method, by quickly pruning away non-matching scans, greatly speeds up the whole process.

Adaptive Feature Selection. As shown previously, we can even identify shapes that match only partially. However, with a bit more effort we also get very good hints about regions of overlap. While comparing two fingerprints, we identify the min-hashes that agree and map them back to the original surface shingles. The union of these shingles give us a very good estimate of the region of overlap (Figure 8).

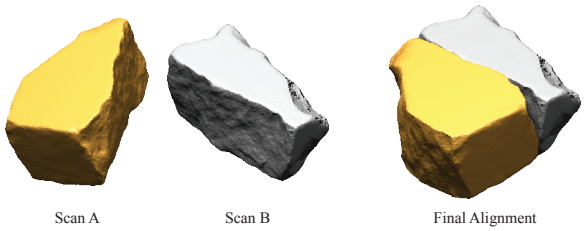

Figure 9: Complementary Shapes. *Given two complementary scans in arbitrary poses, we find their alignment using our adaptive feature selection. To detect complementary shapes, flipped normals are used for computing fingerprint of scan B.*

Subsequent global registration algorithms benefit significantly from this stage, since the adaptive feature points, given by matching shingle patches, very likely lie in areas of overlap and have correct correspondences. In cases when fingerprints are computed independently, we can similarly identify potential overlap regions across multiple shapes, if we additionally store shingle locations for the min-hashed patches along with the fingerprints. Timing complexity and storage requirements still remain $O(m)$.

In order to identify partial complementary matches between two shapes S_i and S_j, we can use a similar method. The fingerprint for S_i is computed as usual. For S_j, when computing spin-images, we flip the point normals to take care of complementary shapes. More generally, for each shape we can compute its fingerprints and its complement fingerprint. A possible application is automatic alignment of broken fragments as shown in Figure 9. In this special scenario, using prior information, the flat surface of the scans can be automatically removed as proposed by Huang et al. [HFG*06] as these regions are known not to be in overlap areas.

Database Classification and Retrieval. We use resemblance between pairs of shapes to efficiently classify a shape database and retrieve models from it (Figure 2). Our database comprises of models, in arbitrary initial positions, from the Princeton shape benchmark [FKMS05]. For each shape, we first compute its fingerprint. A shape distance matrix is then build using $1 - r(S_i, S_j)$ as the distance between any pair of shapes S_i and S_j. We extract a 2D embedding of the fingerprint shape space using multi-dimensional scaling [CC94] on the computed distance matrix. Figure 10 shows a selection of models in the embedded shape space. We get meaningful clustering of shapes even in the presence of articulations and partial matches. A typical query result from the processed database is shown in Figure 11. The resem-

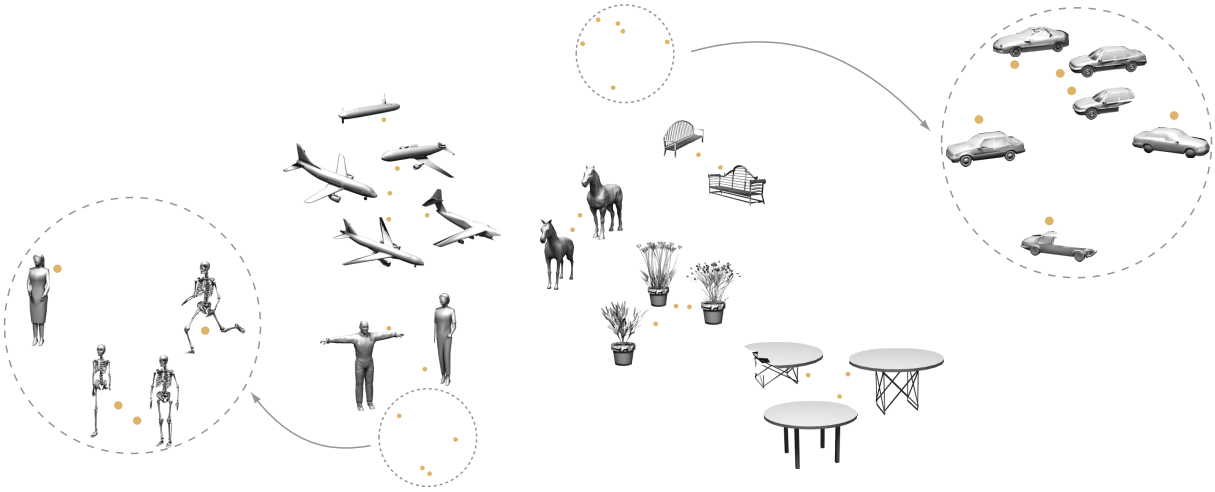

Figure 10: Database Classification. *Shape classification result according to our probabilistic fingerprints. Given any two shapes S_i and S_j, distance between them is defined as $1 - r(S_i, S_j)$. Using this notion of distance, we compute the full distance matrix for a database of shapes. The 2D projection of the fingerprint shape space is computed using classical multi-dimensional scaling (MDS).*

blance scores for this query with any of the tables, planted pots, furniture, or car models is less than 2%. Most of the models in our database being degenerate meshes, we expect a volumetric representation coupled with a suitable signature like spherical harmonics [KFR03] will further improve our performance. Our method is in complement with existing algorithms for shape matching and hence we can use many of the popular shape descriptors in our framework. However, a careful study has to be done to fully evaluate these benefits of our algorithm.

Mesh Authentication. Our scheme can be modified for authenticating geometric models. In the signature computation phase we increase the number of bins making the spin-images sensitive to minor perturbations. Then given a mesh and a partially modified copy, we can use fingerprints to probabilistically identify regions that remain unchanged. For example, if we compute such fragile fingerprints for the original skull model and one corrupted with 1% (of the bounding box) noise, their resemblance is 1.8%. However, if we reorder the vertices, or apply any rigid transform to the original mesh, the resemblance is almost 100%. Local deformations or partial matches can be detected as before. Moreover, for authentication, only a small fingerprint needs to be transmitted and compared with the fingerprint computed from a copy of the mesh. Further investigation needs to be done for quantifying immunity against other types of attacks [WC05].

5.1. Improvements and Limitations

Rabin's method can hash similar signatures to very different values. Such error can be reduced by using locality sensitive hashing (LSH) [IMRV97] where probability of collision between any two signatures is inversely related to their distance. In practice, this may improve the resemblance estimates.

As seen in Figure 6, in some cases the scale ρ has a significant effect on resemblance. To deal with this, we can compute a multi-scale fingerprint over ν different choices of ρ. Storage requirement increases ν fold.

In the current form, we cannot handle scaling. Though pre-scaling of objects may be done using

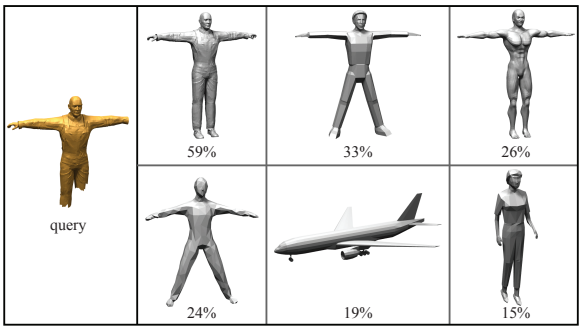

Figure 11: Database Retrieval. *Given a query shape, we show the models retrieved by our algorithm from a database of shapes in arbitrary poses. Our scheme handles partial matches, and is robust to articulations. Corresponding resemblance scores are shown.*

anisotropic scaling [KFR04], such a method fails for partial similarities. We are currently researching other possibilities to handle this scenario.

6. Conclusions

In this paper we have shown how to build compact probabilistic fingerprints for digital geometry models that allow efficient model comparison for partial or total similarity. It is interesting that our scheme relies on randomness for selecting the 'shape features' through the presence or absence of which similarity is estimated. We give provable bounds on the quality of our shape comparisons demonstrating the power of randomization in a geometric context.

The compactness of our fingerprints may also enable distributed geometry processing tasks in sensor network settings, where geometry acquisition, storage, and retrieval may be required over geographically dispersed deployments.

Acknowledgements. This research was supported in parts by DARPA grant 32905, NSF grants FRG-0354543 and ITR 0205671, NIH grant GM-072970, Max-Planck Center for Visual Computing and Communication and a Stanford Graduate fellowship. The authors thank Natasha Gelfand, Marc Levoy, Daniel Russel, and Afra Zomorodian for their helpful comments and suggestions at various stages of the work. We thank the Aim@SHAPE Shape Repository for the Caesar model, the Stanford 3D repository for the bunny scans. Special thanks to the anonymous reviewers for their valuable comments.

References

[AKKS99] Ankerst M., Kastenmüller G., Kriegel H. P., Seidl T.: 3D shape histograms for similarity search and classification in spatial databases. In *SSD* (1999), pp. 207–228.

[Blo70] Bloom B.: Space/time trade-offs in hash coding with allowable errors. In *Communications of the ACM* (1970), vol. 13(7), pp. 422–426.

[Bro93] Broder A. Z.: Some applications of rabins fingerprinting method. In *Sequences II* (1993).

[Bro97] Broder A. Z.: On the resemblance and containment of documents. In *Sequences* (1997).

[Bro00] Broder A. Z.: Identifying and filtering near-duplicate docs. In *CPM* (2000), pp. 1–10.

[CC94] Cox T., Cox M.: *Multidimensional Scaling.* Chapman and Hall, London, 1994.

[CL01] Chan C., Lu H.: *Fingerprinting Using Polynomial (Rabin's Method).* Tech. rep., University of Alberta, 2001.

[FKMS05] Funkhouser T., Kazhdan M., Min P., Shilane P.: Shape-based retrieval and analysis of 3d models. *Comm. of the ACM* (2005), 58–64.

[FKS*04] Funkhouser T., Kazhdan M., Shilane P., Min P., Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: Modeling by example. *ACM Transactions on Graphics 23*, 3 (2004), 652–663.

[GCO06] Gal R., Cohen-Or D.: Salient geometric features for partial shape matching and similarity. In *ACM TOG* (2006), vol. 25(1), pp. 130–150.

[GMGP05] Gelfand N., Mitra N. J., Guibas L., Pottmann H.: Robust global registration. In *Symp. on Geometry Processing* (2005), pp. 197–206.

[HFG*06] Huang Q., Flory S., Gelfand N., Hofer M., Pottmann H.: Reassembling fractured objects by geometric matching. In *Siggraph* (2006), p. to appear.

[HH03] Huber D. F., Hebert M.: Fully automatic registration of multiple 3D data sets. In *Image and Vision Computing* (2003), vol. 21(7), pp. 637–650.

[Hor84] Horn B.: Extended gaussian images. In *Proc. of IEEE* (1984), vol. 72(12), pp. 1671–1686.

[IMRV97] Indyk P., Motwani R., Raghavan P., Vempala S.: Locality-preserving hashing in multidimensional spaces. In *Symposium on Theory of Computing* (1997), pp. 618–625.

[Joh97] Johnson A.: *Spin-Images: A Representation for 3-D Surface Matching.* PhD thesis, Robotics Inst., Carnegie Mellon Univ., August 1997.

[KCD*04] Kazhdan M., Chazelle B., Dobkin D., Funkhouser T., Rusinkiewicz S.: A reflective symmetry descriptor for 3d models. In *Algorithmica* (2004), pp. 201–225.

[KFR03] Kazhdan M., Funkhouser T., Rusinkiewicz S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Symp. on Geometry Proc.* (2003), pp. 167–175.

[KFR04] Kazhdan M., Funkhouser T., Rusinkiewicz S.: Shape matching and anisotropy. *ACM TOG 23*, 3 (2004), 623–629.

[MR00] Motwani R., Raghavan P.: *Randomized Algorithms.* Cambridge University, 2000.

[OFCD02] Osada R., Funkhouser T., Chazelle B., Dobkin D.: Shape distributions. In *ACM Transactions on Graphics* (2002), vol. 21(4).

[Rab81] Rabin M. O.: Fingerprinting by random polynomials. In *Center for Research in Computing Tech., Harvard Univ., Report TR-15-81* (1981).

[RWP05] Reuter M., Wolter F.-E., Peinecke N.: Laplace-spectra as fingerprints for shape matching. In *SPM* (2005), pp. 101–106.

[SGM98] Shivakumar N., Garcia-Molina H.: Finding near-replicas of documents on the web. In *Proceedings of Workshop on Web Databases* (1998).

[Tur92] Turk G.: Re-tiling polygonal surfaces. In *SIGGRAPH* (1992), pp. 55–64.

[WC05] Wu H.-T., Cheung Y.-M.: A fragile watermarking scheme for 3d meshes. In *Multimedia and Security* (2005), pp. 117–124.