Abstract

Due to the fast increasing amount of shape data in the Internet, one wants to search objects with shape retrieval. Whereas in the 2D-case there already exists a default operator, the 3D-case is more difficult. One good candidate is the heat kernel signature but sadly it is not scale-invariant. However this is quite an important property as a lot of different sizes of the same object exist in the Internet. In this report an approach based on logarithmically sampling and use of the Fourier transform is shown. By doing so, shape scaling first corresponds to a translation which is undone using the magnitude of the Fourier transform. The obtained dense scale-invariant local descriptors work under wide classes of transformations such as isometric deformations, global and local scaling, missing parts and noises. We will show that one gets dramatic improvements of performance over the usual heat kernel signature.

1 Introduction

1.1 Motivation

As only a small fraction of visual and geometric data in the Internet is tagged and therefore available through text search, shape retrieval is strongly needed. That means objects of the same kind should be found by an object query as input. In this paper 3D-shapes are meant if we talk about an ‘object’, otherwise it will be mentioned separately. For example if you take a 3D-shape of a man as an input, you want to get all other figures representing a man in whatever positions. Like in the 2D-case of image comparison, there are different problems which have to be dealt with. The main problems are to compare the query shape with shapes of

- different scale,
- different orientation,
- deformations,
- missing data and
- noise.

Our goal is to find a so called descriptor or operator, which remains invariant under transformations or other perturbations mentioned above. In the 2D-case there already exists a default choice for a local descriptor/operator, the so called scale-invariant feature transform (SIFT). The 3D-case however is a lot more complicated, as more degrees of freedom have to be considered. Therefore our general wish is to find a descriptor for 3D shapes which maintains invariance under a wide class of transformations.

In this paper we will have a look on one good candidate of such a descriptor, called Scale-invariant heat kernel signature (SI-HKS). The most informations stem from the paper of Bronstein and won’t be explicitly cited.

1.2 Point Signatures

Based on our observations and wishes before, we sum up some properties an ideal feature descriptor should fulfill:

- be intrinsic (deformation invariant)
• cope with missing parts
• should work across different shape representations and formats
• be insensitive to sampling
• be scale-invariant

The last points are quite important in terms of shape retrieval through the Internet, as a lot of different formats exists. Also we can’t compare every point of a shape or some of the shapes are already sampled. Therefore being insensitive to sampling has great importance in this field. The importance of scale-invariance should be self explaining. Everyone who tried a reverse image search over Google knows the wide variety of image sizes. The same goes for 3D-shapes.

To give a short overview, we want to provide two examples of other descriptors.

1.2.1 Global Point Signature (GPS)

The Global Point Signature uses the most basic idea how to get an intrinsic descriptor - a property every descriptor should fulfill. Like many other descriptors the GPS makes use of the eigenvalues and eigenfunctions of the Laplacian.

Therefore we have a look at the equation

$$\Delta \phi = \lambda \phi$$  \hspace{1cm} (1)

where $\Delta$ is the Laplace-Beltrami Operator. Basically it is the same as the Laplacian but only on manifolds. It has some nice properties which will be used during the paper, but not explained in detail. If you want to know more about the Laplace-Beltrami Operator, please have a look at [6].

Using the eigenvalues $\lambda_i$ and eigenfunctions $\phi_i$ of the Laplace-Beltrami Operator you can define the GPS under the assumption that the eigenvalues $\lambda_i$ can be ordered ascending:

$$0 = \lambda_0 < \lambda_1 < \cdots < \lambda_i < \cdots$$  \hspace{1cm} (2)

$$GPS(p) = \left( \frac{1}{\sqrt{\lambda_1}} \phi_1(p), \frac{1}{\sqrt{\lambda_2}} \phi_2(p), \frac{1}{\sqrt{\lambda_3}} \phi_3(p), \ldots \right)$$  \hspace{1cm} (3)

This signature is isometry invariant (intrinsic), which is just what we wanted. One can also show that the GPS is uniquely determined and quite stable under local deformations [5].

However there is a lot to improve. The GPS doesn’t work on degenerated meshes which is especially a drawback in Internet applications. Also it is not stable under (certain) perturbations, as the signs of the eigenvectors are not defined.

1.2.2 Wave Kernel Signature (WKS)

The Wave Kernel Signature is based on Schrödinger’s Equation:

$$\frac{\partial \Psi}{\partial t}(x,t) = i \Delta \Psi(x,t)$$  \hspace{1cm} (4)

The idea is to characterize a point $x$ by the average probabilities of frequencies of different energy, so information from all frequencies is captured.

Taking an approximate measurement one obtains an energy probability distribution with expectation value $E$, which leads eventually to the descriptor

$$WKS(E, x) = \lim_{T \to \infty} \frac{1}{T} \int_0^T |\Psi_E(x, t)|^2$$  \hspace{1cm} (5)

This signature is a way better descriptor than the GPS. Of course it is still intrinsic. But it also improves/removes every negative aspect of the GPS, meaning mainly the stability under perturbations such as noise. Due to its natural notion of scale it is also scale-invariant. Another benefit is the fact that the Eigenvalues don’t have to be ordered strictly ascending. So repeated eigenvalues are allowed. [1]
2 Approach

2.1 Heat Kernel Signature

At first we have a look at the heat diffusion equation on non-Euclidean domains. Basically it is the same equation as on Euclidean domains (spaces) but with the Laplace-Beltrami Operator instead of the Laplacian.

\[ \left( \Delta_X + \frac{\partial}{\partial t} \right) u = 0, \]  

\[ u_0(x) = \delta(x - z) \]  

where \( X \) describes the manifold and \( \Delta_X \) the Laplace-Beltrami Operator on this manifold.

The solution \( u(x, t) \) of this equation with the initial condition \( u(x, 0) = u_0(x) \) describes the amount of heat at point \( x \) in time \( t \).

If we take only one point as initial condition, so putting all the heat in just one point \( z \) at the beginning, the solution will be the so called Heat Kernel.

Initial condition: \[ u_0(x) = \delta(x - z) \]  

Where \( \delta \) describes the Delta-Distribution.

The solution is then given through

\[ K_{X,t}(x, z) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_i(z) \]  

with eigenvalues \( \lambda_i \geq 0 \) and eigenfunctions \( \phi_i \) of \( \Delta_X \phi_i = \lambda_i \phi_i \).

If we choose \( x = z \), then we get our Heat Kernel Signature

\[ h(x, t) = K_{X,t}(x, x) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i^2(x), \]

describing the amount of remaining heat at the point \( x \) at time \( t \).

Taking this signature, we can provide isometry invariance (so the descriptor is again intrinsic).

With some extra assumptions one can show that the HKS is also informative [7]. This means if we have the same heat kernel for every point of two shapes, we can say that these were deformed by an isometry. Another good property is the multi-scale which allows us to capture local and global features.
However there is one big drawback which is its sensitivity to scale, especially global scale. Let's consider the scaled shape $X' = \beta X$. The eigenvalues and eigenfunctions of the Laplace-Beltrami operator applied to $X'$ will now change to $\lambda' = \beta^2 \lambda$ and $\phi' = \beta \phi$, leading to

$$h'(x, t) = \sum_{i=0}^{\infty} e^{-\lambda_i \beta^2 t} \phi_i^2(x) \beta^2 = \beta^2 h(x, \beta^2 t)$$  \hspace{1cm} (10)

As we mentioned in the introduction, scale-invariance is one of the most important properties one needs for shape retrieval in Internet applications. In the following we try to save all the good properties of the HKS and make it scale-invariant.

### 2.2 Scale-invariant Heat Kernel Signature

Ideas like the global pre-normalization of the shape to undo the scaling effect only work on rigid shapes. Typical normalizations of e.g. the intrinsic diameter (geodesic distance) or the Laplace-Beltrami eigenvalues would work, but fail if the shape has missing parts in it.

A better approach is to look at local normalization. This can be done in three steps with the goal to remove the dependence of $h$ from the scale factor $\beta$.

**Step 1: logarithmical sampling in time**

At first we sample the heat kernel signature logarithmically to the base $\alpha$, so $t = \alpha^s$, with $\tau$ as the new (discrete) time steps. We get:

$$h'_\tau = h'(x, \alpha^s) = \beta^2 h \left( x, \beta^2 \alpha^s \right) = \beta^2 h \left( x, \alpha^s \alpha^\tau \right) = \beta^2 h \left( x, \alpha^{\tau+s} \right) = \beta^2 h_{\tau+s}$$ \hspace{1cm} (11)

with $\alpha^s = \beta^2 \Leftrightarrow s = \log_\alpha \beta^2 = 2 \log_\alpha \beta$.

In this form we can easily see the translation in time stemming from the factor $\beta^2$ in the time slot of \ref{eq:10}, if we choose a logarithmic scale. 2

**Step 2: taking the logarithm of $h$ and then $\frac{\partial}{\partial \tau}$**

Now we take the logarithm to turn the multiplicative constant $\beta^2$ in an additive constant $2 \log \beta$, which vanishes in the later differentiation.

$$\log h'_\tau = \log \left( \beta^2 h_{\tau+s} \right) = \log \beta^2 + \log h_{\tau+s} = 2 \log \beta + \log h_{\tau+s}$$ \hspace{1cm} (12)

$$\frac{\partial}{\partial \tau} (\log h'_\tau) = \frac{\partial}{\partial \tau} (2 \log \beta + \log h_{\tau+s}) = \frac{\frac{\partial}{\partial \tau} (h_{\tau+s})}{h_{\tau+s}}$$ \hspace{1cm} (13)

$$\frac{\partial}{\partial \tau} h_{\tau+s} = \frac{\partial}{\partial \tau} h(x, \alpha^s \alpha^\tau) = \frac{\partial}{\partial \tau} \sum_{i=0}^{\infty} e^{-\lambda_i \alpha^s \alpha^\tau} \phi_i^2(x) = \sum_{i=0}^{\infty} \frac{\partial}{\partial \tau} \left( e^{-\lambda_i \alpha^s \alpha^\tau} \phi_i^2(x) \right)$$ \hspace{1cm} (14)

$$\frac{\partial}{\partial \tau} \left( e^{-\lambda_i \alpha^s \alpha^\tau} \phi_i^2(x) \right) = -\lambda_i \alpha^s \ln(\alpha) \alpha^\tau e^{-\lambda_i \alpha^s \alpha^\tau} \phi_i^2(x)$$ \hspace{1cm} (15)

$$\Rightarrow \frac{\partial}{\partial \tau} h_{\tau+s} = \sum_{i=0}^{\infty} -\lambda_i \alpha^s \ln(\alpha) \alpha^\tau e^{-\lambda_i \alpha^s \alpha^\tau} \phi_i^2(x)$$ \hspace{1cm} (16)

Define:

$$\dot{h}_\tau := \sum_{i=0}^{\infty} -\lambda_i \ln(\alpha) \alpha^\tau e^{-\lambda_i \alpha^s \alpha^\tau} \phi_i^2(x) \Rightarrow \dot{h}_{\tau+s} = \frac{\frac{\partial}{\partial \tau} h_{\tau+s}}{h_{\tau+s}}$$ \hspace{1cm} (17)

$$\dot{h}'_\tau := \dot{h}_{\tau+s}$$ \hspace{1cm} (18)
Which can also be rewritten as
\[ \dot{h}_\tau' = \dot{h}_{\tau+s}, \quad \text{with} \quad \dot{h}_\tau = \log h_{\tau+1} - \log h_\tau \]  

since
\[ \dot{h}_\tau' = \dot{h}_{\tau+s} \]  

\[ \log h_{\tau+1} - \log h_\tau = \log h_{\tau+s+1} - \log h_{\tau+s} \]  

\[ \log \beta^2 h_{\tau+s+1} - \log \beta^2 h_{\tau+s} = \log h_{\tau+s+1} - \log h_{\tau+s} \]  

\[ \log \beta^2 + \log h_{\tau+s+1} - \log \beta - \log h_{\tau+s} = \log h_{\tau+s+1} - \log h_{\tau+s} \]  

\[ 0 = 0 \]  

which is true.

We now see that only a shift in \( \tau \) appears, but no scaling anymore.

**Step 3: taking the discrete-time Fourier transform of \( \dot{h}_\tau \)**

To undo the shift, we take the Fourier transform so that the shift becomes a complex phase:
\[ \mathcal{F}[\dot{h}_\tau'](\omega) = H'(\omega) = H(\omega)e^{-i2\pi \omega s} = \mathcal{F}[\dot{h}_{\tau+s}](\omega) \]  

with \( \omega \in [0, 2\pi] \).

Finally we eliminate the phase by taking the modulus:
\[ |H'(\omega)| = |H(\omega)e^{-i2\pi \omega s}| = |H(\omega)| \cdot |e^{-i2\pi \omega s}| = 1 \]  

\[ = |H(\omega)| \]  

Figure 2: Visualization of the construction of the SI-HKS. From left to right you can see the three steps at a corresponding point. At first the scale (by the factor of 11) and translation, where \( h \) is red and \( h' \) is blue. Secondly the multiplicative constant is undone and the change in scale corresponds to a shift in \( \tau \) only. Here \( \dot{h}_\tau \) is red and \( \dot{h}'_\tau \) is blue. Finally the first 10 frequencies of \( |H(\omega)| \) and \( |H'(\omega)| \), computed at the two different scales, are virtually identical.

With the above steps we created a scale-invariant quantity \( |H(\omega)| \) at each point \( x \), without performing scale selection. This quantity is from now on called SI-HKS.

An other advantage besides the scale invariance is that our descriptor is computable at any point, even where scale selection could be impossible.

Additionally most of the signal information is contained in low-frequency components. Therefore we are able to build a compact descriptor by sampling \( |H(\omega)| \) at a small number of low frequencies and reduce computation time without getting much worse results.

One possible objection could be that as we made use of the Fourier transform, resampling \( \dot{h}_\tau \) makes the samples at the boundaries of the range of \( \tau \) change. This could have dramatic effects if most of the signal information lies there. But the HKS is typically smooth at low and high scales and its derivative is equal to zero for a broad range of \( \tau s \) at the beginning and end of \( h \). This comes from some properties of Fourier transformations and so-called band-limited functions which we won’t analyse here now.
3 Experiments

Now that we have successfully found a way to get our scale-invariant heat kernel signature, we want to do some numerical experiments and also compare the SI-HKS with the HKS to see how much improvement we obtained by making the HKS scale-invariant with the above method.

3.1 Numerics

At first some words about the numerics behind it.
As we stated above, it is sufficient to consider only the first $k$ low-frequency components:

$$K_{X,t}(x, z) \approx \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x) \phi_i(z)$$  \hfill (27)

Of course we also need to choose a discrete Laplace-Beltrami operator. The discretization depends on the shape, e.g. if point clouds or meshes are used. Here we will take the so called cotangent weight scheme, which is one of the most common discretizations on triangular meshes.

This discretization is defined as a weighted sum over the edges of triangles:

$$(\Delta \hat{X} f)_i = \frac{1}{a_i} \sum_j w_{ij} (f_i - f_j)$$  \hfill (28)

with $w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$, where $\alpha_{ij}$ and $\beta_{ij}$ are the two angles opposite to the edge between $i$ and $j$ inside the triangles sharing the edge. $a_i$ are normalization coefficients which are proportional to the area of triangles sharing $x_i$.

The cotangent weight scheme preserves many important properties of the continuous Laplace-Beltrami operator like positive semi-definiteness, symmetry, locality and is also numerically consistent.

The equation (28) can also be written as

$$\Delta \hat{X} f = A^{-1} L f$$  \hfill (29)

where $A = \text{diag}(a_i)$ and $L = \text{diag} \left( \sum_{l \neq i} w_{il} \right) - w_{ij}$.

In a last step, the eigenvalues $\lambda_0, \ldots, \lambda_k$ and eigenfunctions $\phi_0, \ldots, \phi_k$ are computed by solving

$$A \Phi = \Lambda L \Phi$$  \hfill (30)

where $A \in \mathbb{R}^{(k+1) \times (k+1)}$ is a diagonal matrix of the $k + 1$ smallest eigenvalues and $\Phi \in \mathbb{R}^{N \times (k+1)}$ is a matrix of the corresponding eigenfunctions such that $\phi_l$ is the value of the $l$th eigenfunction at the point $x_i$.

Approximating the Laplace-Beltrami operator on triangular meshes by finite element methods (FEM) is also an often used alternative.

Whatever method one applied, the discrete heat kernel signature is approximated by

$$h(x_i, t) \approx \sum_{l=0}^{k} e^{-\lambda_l \alpha^T} \phi_l = \Psi e^{-TA},$$  \hfill (31)

where $T = \text{diag} \left( \alpha^T \right)$ and $\Psi = \left( \phi_l^2 \right)$.

After that, one has to do the steps as presented in our approach section.

Theoretically one can compare shades in different representations as the heat kernel depends only on the eigenvalues and eigenfunctions of the Laplace-Beltrami operator, which is a very appealing property when it comes to shape retrieval applications in the Internet, because of the vast variety of formats.
3.2 Set Up

In the experiments 13 shape classes has been selected. For these classes 208 isometry, 208 global scale (with scaling factor $\beta$ ranging from 0.7 to 1.35), 128 local scale and 48 partiality+scale (i.e. missing parts in shapes, global scaling) transformations were simulated. The above introduced cotangent weight approximation has been used with $k = 200$. The parameters for our steps to make the HKS scale-invariant were $\alpha = 2$ and $\tau$ ranging from 1 to 25 with increments of $\frac{1}{16}$. As a local descriptor only the first 6 discrete lowest frequencies have been used. Also 48 bag of features for the shape descriptor [2], where the (SI)-HKS is replaced by the closest geometric word using vector quantization (in general just using areas for the same signature instead of having another signature at every single point of the shape), were used to construct the shape descriptors.

3.3 Comparison of HKS and SI-HKS

Since we took so much effort into improving the HKS by making it scale-invariant, lets compare them and see what we got.

![Figure 3: Comparison of HKS (left) and SI-HKS (right). Upside global scaling was applied and local scaling below. Three components of the HKS and SI-HKS are represented as RGB color at three points on the shape.](image)

In Figure 3 one can see some interesting informations. At the very first glimps one is quite satisfied as it seems that the SI-HKS, which is on the right of the figure, is clearly behaving better than the HKS. The drawn through lines represent the signature of the original shape and the dashed lines the scaled version. To get a match, i.e. the scaled shape is detected as a transformation of the original shape, the signatures have to be equal. In the global scaling case one clearly sees that the HKS behaves badly as non of the same colored lines match. The SI-HKS behaves completely different, all lines are matching almost perfectly. That’s exactly the result we hoped for, the SI-HKS is indeed scale invariant.

In the local setting it looks a bit different. The HKS lines still don’t match perfectly but quite good, because of the local nature of the descriptor. However it has some problems at the point...
at the thumb. This also lies in the nature of the Heat Kernel, as the heat cannot be distributed fast enough in every direction because of the strong curvature, resulting in errors. The matches of the SI-HKS are pretty similar to the ones in the HKS setting here. One sees that the SI-HKS still has problems at points with strong curvature, which is not surprising as it stems from the heat equation as well.

Figure 4: Shape retrieval performance using HKS (left) and SI-HKS (right).

Now that we looked on a specific example, let’s see how the signatures behave in general. For this purpose we at first introduce the

- **precision**, which is a ratio of the number of relevant shapes retrieved (i.e. real/right matches) and the total number of shapes retrieved.

- **recall**, which is a ratio of the number of relevant shapes retrieved and the total number for existing relevant shapes that could be ideally retrieved.

Figure 4 shows the tradeoff between precision and recall. The more in the upper right of the plot the better the solutions. The SI-HKS performs better in every case but especially in global scale there is a huge improvement. The same goes for partiality+scale, where we used global scale, so even missing parts don’t disrupt the scale invariant heat kernel that much. Of course over all cases partiality is still the hardest case, since the heat distribution changes noticeable. Both signatures don’t have any problem with isometry, the local scaling case is also fine with both signatures, however improves slightly but notable when SI-HKS is used.

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Queries</th>
<th>HKS precision</th>
<th>SI-HKS precision</th>
<th>ShapeDNA precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isometry</td>
<td>208</td>
<td>99.96%</td>
<td>99.97%</td>
<td>99.52%</td>
</tr>
<tr>
<td>Scale</td>
<td>208</td>
<td>61.32%</td>
<td>99.95%</td>
<td>36.72%</td>
</tr>
<tr>
<td>Local scale</td>
<td>128</td>
<td>85.83%</td>
<td>92.60%</td>
<td>72.17%</td>
</tr>
<tr>
<td>Partiality+scale</td>
<td>48</td>
<td>54.67%</td>
<td>89.95%</td>
<td>27.42%</td>
</tr>
<tr>
<td>All</td>
<td>1061</td>
<td>85.30%</td>
<td><strong>97.25%</strong></td>
<td>74.47%</td>
</tr>
</tbody>
</table>

Table 1: Shape retrieval performance (mAP in percents) using HKS and SI-HKS based bags of features and ShapeDNA. Best result is shown in bold.

To quantify the results we define the **average precision** (AP) which is the area below the precision-recall curve and the **mean average precision** (mAP) which is the average of AP over all queries. In table 1 beside HKS we compare ShapeDNA with the SI-HKS. ShapeDNA uses the first 15 eigenvalues of the Laplace-Beltrami operator, calculated with the cotangent weight.
scheme. The presented approach shows a dramatic improvement when it comes to global scaling (99.5% mAP compared to 61.32% with the HKS). It also shows better performance for local scaling but the HKS behaves still pretty well in this case. That’s what we already noticed in figure 3. However the improvement in partiality+scale (89.95% compared to 54.67%) is again very good which may originate in the global scaling part as well. On the class of isometries every descriptor shows nearly perfect performance. The difference of 0.01% between the HKS and SI-HKS is negligibly. ShapeDNA performs very poorly on global and local scale transformations.

For visualization of the results we look at the first four matches of some queries in figure 5. The scaled down centaur is confused with a dog (row a) and the scaled up horse is confused with an elephant (row c) when using the HKS, whereas the SI-HKS has no problems with these queries and delivers only right matches (row b and d). The same goes for the cat, where the HKS only has one right match but four completely other objects. As already mentioned in figure 3 and table 1 the HKS has some good results when it comes to local scale. However it confuses between male and female shapes. The SI-HKS shows better robustness here.

4 Summary

In this report we presented an extension of the HKS, allowing us to deal with global and local scaling. The use of the Fourier transform magnitude is mentionable, as it extracts a scale-invariant quantity out of the heat kernel signature whereas other methods would only work in specific cases. By using the three steps in the approach to make the HKS scale-invariant, we didn’t use any assumptions, so this way can be applied to other signatures as well. As long as the underlying function is smooth enough and fulfills the same properties concerning the change of samples at the time boundaries, the so improved (scale-invariant made) signature should be robust. We got a dense scale-invariant feature descriptor defined at every point and a dramatic improvement as well in global scale and when it comes to missing parts and global scale at once. In local scale the SI-HKS shows better robustness, whereas in the isometry case there is no difference, since the HKS already behaves very good there. Further improvements could be that one might divide the shape in parts and compute a separate descriptor for each part. With this approach, inelastic deformations could be addressed.
Figure 5: Shape retrieval results. On the left are the queries and on the right the first four matches using HKS (a,c,e,g) and SI-HKS (b,d,f,h).
References


