Invariant Features for 3D-Data based on Group Integration using Directional Information and Spherical Harmonic Expansion

M. Reisert and H.Burkhardt
Computer Science Department, University of Freiburg, 79110 Freiburg i. Br., Germany

Abstract

Due to the increasing amount of 3D data for various applications there is a growing need for classification and search in such databases. As the representation of 3D objects is not canonical and objects often occur at different spatial position and in different rotational poses, the question arises how to compare and classify the objects. One way is to use invariant features. Group Integration is a constructive approach to generate invariant features. Several variants of Group Integration features are already proposed. In this paper we present two main extensions, we include local directional information and use the Spherical Harmonic Expansion to compute more descriptive features. We apply our methods to 3D-volume data (Pollen grains) and 3D-surface data (Princeton Shape Benchmark)

1. Introduction

In many fields researches deal with a huge amount of three dimensional data. In medical and biological applications one usually has to do with volumetric scans of various types, there is a need for fast and reliable feature extraction methods to handle and classify such huge amounts of data. The development of 3D modeling software has increased the number of free available 3D-surface models, fast retrieval systems are necessary to browse and search for 3D-models in a user-friendly way.

In most cases problem dependent solutions are proposed. The algorithms and features are especially designed for the specific tasks and are not adaptable to other problems. We apply the general Group Integration (GI) framework to volume- and surface data without any data-specific adaption. Additionally we uncover relation to already used ad-hoc features, which lack of theoretical motivation.

Group Integration stands in opposite to Normalization techniques, which obtain invariance by computing features relative to a global reference frame. The determination of the reference frame makes Normalization techniques extremely sensitive to noise. Whereas Group Integration is known to be very robust to many kinds of noise. In [2] a detailed overview over GI-techniques is given. In [5] the GI-framework is joined with Kernel-techniques. Ronneberger et al [8, 9] used GI for the classification of Pollen grains and segmentation of cell nuclei. In [1] GI is applied to discrete graph-like structures.

2. Group Integration Features

For consistent treatment of the volume and surface data we represent them by functions $\mathbf{x} : \mathbb{R}^3 \mapsto \mathbb{R}$ defined on the whole volume, where we interpret the surface data as volume mapping with its 'mass' located only on the surface of the object.

We always consider the Euclidean group $\mathcal{E}$. A group integration feature I_k is obtained by integrating a kernel function k

$$I_k(\mathbf{x}) = \int_{\mathcal{E}} k(g\mathbf{x})\ dg,$$

where $g\mathbf{x}$ denotes the group action of g on $\mathbf{x}$. Typically choices of k are e.g. $k_d(\mathbf{x}) = \mathbf{x}(0) \cdot \mathbf{x}(d)$ or $k_d(\mathbf{x}) = h_1(\mathbf{x}(0)) \cdot h_2(\mathbf{x}(d))$, where h_1, h_2 may be some arbitrary nonlinear functions and $d \in \mathbb{R}^3$ is a parameter.

2.1. Including local directional information

To include directional information instead of gray values only, one should also consider a local quantity that describes the neighborhood of some point, the gradient $\nabla\mathbf{x}$ would be a good idea. We can use kernels as $k_d(\mathbf{x}, \nabla\mathbf{x}) = h_1(\nabla\mathbf{x}(0)) \cdot h_2(\nabla\mathbf{x}(d))$ or further extension, which combine the gradient values with the gray values. We only have to keep in mind that the gradient has the following transformation behavior $(g\nabla\mathbf{x})(r) = R\nabla\mathbf{x}(R^\dagger r - u)$, when $\mathbf{x}$ is shifted by u and

then rotated by R. Of course, one can also use other local neighborhood operators like the Hessian, local jets or wavelets.

For surface data the interpretation of the gradient is not clear. But there is a simple back door. A point on a surface is usually associated with a surface normal, which may be interpreted as the gradient of a solid object surrounded by this surface. The surface normal has exactly the same transformation behavior as the gradient.

2.2. Spherical Harmonic expansion

Any function $f(s)$ defined on the two-sphere S^2 can be orthogonally expanded in terms of the so called Spherical Harmonics.

$$f(s) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_m^l Y_m^l(s),$$

where s is a unit vector and the a_m^l are the expansion coefficients, that are computed by projections $a_m^l = \int_{S^2} f(s)\overline{Y_m^l(s)}ds$ on basis functions. In practice, the infinite sum is truncated at some finite cutoff parameter l_{max}. The Spherical Harmonic Transformation (SHT) is the analogon to the Fourier Transformation for the rotation group, i.e. the SHT provides a representation, which is invariant to rotations. There are subspaces which preserve their energy while rotating the function. Moreover, the a_m^l show a nice transformation behavior. Suppose $f(s)$ is rotated by some matrix R, then the a_m^l are transformed by the so called D-Wigner matrices $D^l(R)$, i.e. $a_m^l \mapsto \sum_{m=-l}^{l} D_m^l(R) a_m^l$.

Since a integration over the rotation group can always be separated into an integration over a sphere and a circle, we are able to use the SHT to keep more information back. Instead of just integrating the sphere integration out we expand the residue function in terms of spherical harmonics.

3. The Kernel Choice

To choose a kernel function is not a simple question. The choices are typically guided by the application's demands and complexity considerations. The perceptual interesting parts of shape are edges and regions of high variety, hence it is obvious to incorporate the gradient information as the most important part. So we choose

$$k_d(\mathbf{x}) = h_1(\nabla \mathbf{x}(0)) \cdot h_2(\nabla \mathbf{x}(d))$$

as the basis of our kernel function with width parameter d. But how to choose h_1 and h_2. One demand is that both functions should give strong feedback if the gradient is large. The functions should also be direction specific to keep the relative directions of the gradients. The simplest idea fulfilling this demands is $h_n(v) = |v^\dagger n|$, where n is some fixed unit vector. We use the absolute value of the dot-product, because experiments have shown that whether the edge is falling or growing is less important than the actual direction. A disadvantage of the function above is that one is not able to decide whether it has to do with large, disoriented or small, oriented gradients v. A more rational choice is

$$h_n(v) = |v| \cdot \delta_1 \left(\frac{|v^\dagger n|}{|v|} \right), \tag{1}$$

where δ_1 is the Delta-Distribution[1] only giving contribution if its argument is nearby 1. The function $h_n(v)$ is unequal to zero whenever $n \parallel v$, i.e. n and v are parallel or anti-parallel. We now have the kernel function

$$k_{d,n,n'}(\mathbf{x}) = h_n(\nabla \mathbf{x}(0)) \cdot h_{n'}(\nabla \mathbf{x}(d)). \tag{2}$$

For the con-focal microscopy data the gray values should be incorporated. Microscopy data sets are typically not homogeneously illuminated. If we want to circumvent a normalization step, only differences $\mathbf{x}(0) - \mathbf{x}(d)$ of gray values contain comparable information. A direct multiplication of the difference with the kernel in (2) is not a clever idea, because for large distances d the result would be dominated by the fluctuation and interferences of the large differences. So we use a Delta function and have

$$k_{d,n,n',\mu}^{(M)}(\mathbf{x}) = k_{d,n,n'}(\mathbf{x}) \cdot \ \delta_\mu(|\mathbf{x}(0) - \mathbf{x}(d)|), \tag{3}$$

where we again used the absolute difference by the reason mentioned above.

3.1. Implementation

A direct implementation of the integration is of high computational complexity. In [8] a convolution with a rotation symmetric kernel is used for a fast evaluation of the integral. In our case this is not possible, because our kernel depends on local directional quantities. We want to point out another way. We write out the integral for the basis kernel

$$I_\Pi = \int_{\mathbb{R}^3} \int_{O_3} h_n(R\nabla \mathbf{x}(u)) \cdot h_{n'}(R\nabla \mathbf{x}(u + R^\dagger d)) \ dR \ du,$$

where Π denotes the parameter set n, n', d. We use the relation $f(u) = \int_{\mathbb{R}^3} f(u')\delta_{u'}(u) \ du'$ for the second kernel factor and get

[1] we write $\delta_y(x)$ for the usual $\delta(x-y)$ of the Delta-Distribution due to space considerations

$$\int_{\mathbb{R}^3,\mathbb{R}^3} \int_{O_3} h_n(R\nabla\mathbf{x}(u)) \cdot h_{n'}(R\nabla\mathbf{x}(u')) \cdot \delta(u + R^\dagger d - u'),$$

where we omit $du\ du'\ dR$. Due to the form of our kernel function we find out that the integral over the rotation group gives contribution only if there is a R such that the four conditions $R^\dagger n \parallel \nabla\mathbf{x}(u)$, $R^\dagger n' \parallel \nabla\mathbf{x}(u')$, $R^\dagger d \parallel (u - u')$ and $|d| = |u - u'|$ are simultaneously met. The first 3 condition are fulfilled if n, n', d have the same configuration up to rotation as $\nabla\mathbf{x}(u)$, $\nabla\mathbf{x}(u')$ and $u - u'$. This is the case when the absolute values of the pair-wise dot-products of the two sets are the same. So we get the following expression for the integral

$$I_\Pi = \int_{\mathbb{R}^3,\mathbb{R}^3} \theta_{\alpha,\beta,\gamma} \cdot \delta_\Delta(|u - u'|) \cdot |\nabla\mathbf{x}(u)| \cdot |\nabla\mathbf{x}(u')|, \quad (4)$$

where

$$\theta_{\alpha,\beta,\gamma} = \delta_\alpha(|\nabla\mathbf{x}(u)^T \nabla\mathbf{x}(u')|) \ \cdot \\ \delta_\beta(|\nabla\mathbf{x}(u)^T (u - u')|) \cdot \delta_\gamma(|\nabla\mathbf{x}(u')^T (u - u')|)$$

is the orientation specific part. We introduced the abbreviations $\alpha = n \cdot n', \beta = n \cdot d, \gamma = n' \cdot d$ and $\Delta = |d|$. From now on we refer to $\Pi = \{\alpha, \beta, \gamma, \Delta\}$ as the parameter set. The question arises how to evaluate integral (4). Due to the high dimensionality of the integral a Monte-Carlo approximation is appropriate. It is well known if we have to do with sparse, non-homogeneous data, that importance sampling is the right way to do it. A tuple (u, u') is chosen according to the probability $p(u, u') = |\nabla\mathbf{x}(u)| \cdot |\nabla\mathbf{x}(u')|/Z$, where Z is a normalizing factor making $p(u, u')$ to a probability distribution. Then we compute for which parameter-configuration $\Pi = \{\alpha, \beta, \gamma, \Delta\}$ the Delta-distributions are unequal to zero and accumulate the contributions in a result-array with appropriate parameter-indices.

In fact, we compute some kind of histogram. This issue is very interesting because it shows a very close connection of GI-features with invariant histograms used in [6, 7]. For example, the D2 Shape-distributions [7] may interpreted as a Group Integration feature with the kernel $k_d(\mathbf{x}) = \mathbf{x}(0) \cdot \mathbf{x}(d)$.

As already mentioned we want to use the SHT to keep more information. Rewriting (4) by evaluating $\delta_\Delta(|u - u'|)$ leads to the sphere integral

$$I_\Pi = \int_{\mathbb{R}^3,S^2} \theta_{\alpha,\beta,\gamma} \cdot |\nabla\mathbf{x}(u)| \cdot |\nabla\mathbf{x}(u + \Delta s)|\ du\ ds,$$

where s ranges over the unit-sphere S^2. Instead of simply integrating the above expression we now compute the projection of it on $Y_m^l(s)$, i.e.

$$I_\Pi^{lm} = \int_{\mathbb{R}^3,S^2} \theta_{\alpha,\beta,\gamma} \cdot |\nabla\mathbf{x}(u)| \cdot |\nabla\mathbf{x}(u + \Delta s)| \cdot \overline{Y_m^l(s)}\ du\ ds.$$

For $l = 0$ the integral is exactly the same as the one in (4). For $l > 0$ the implementation of the above integral is very similar to the computation of (4). Instead of a simple accumulation, the contributions are weighted by the complex factor $Y_m^l(\frac{u-u'}{|u-u'|})$. For every l and m a new random tuple (u, u') is chosen to keep the results independent. After computation, the results are made invariant by computing the band-wise energy $\sum_{m=-l}^{l} |I_\Pi^{lm}|^2$.

4. Experiments

4.1. Volume Data

The considered database consists of the 26 most important German pollen taxa (385 samples). The data was recorded by con-focal laser scanning microscopy, for details see [8]. Each sample is of dimension 128^3 with 8-bit resolution. As preprocessing we blurred the volume-image by a Gaussian of width $\sigma = 4$ in pixel units. The gradients were computed by finite differences within pixel distance 3. Both parameter choices were driven by the signal-to-noise ratio of the data. For a fast determination of the random points according to the gradient-magnitude we compute the cumulated distribution and use bisectional search. Having determined two random points (u, u'), we compute the arguments of the Delta-Distributions in $\theta_{\alpha,\beta,\gamma}$ and additionally $|\mathbf{x}(u) - \mathbf{x}(u')|$ to accumulate afterwards the result array at appropriate indices $\Pi = \{\alpha, \beta, \gamma, \Delta, \mu\}$. In the experiments the parameters are discretized in $4 \cdot 2 \cdot 2 \cdot 16 \cdot 10 = 2560$ bins, where the scaling of Δ and μ is chosen such that the range of all values, which can give contribution, is covered. We do experiments with and without the SH-transform. The cutoff-parameter $l_{max} = 2$ is chosen. The final features are computed by taking the energy $\sum_{m=-l}^{l} |I_\Pi^{lm}|^2$, resulting in an overall feature size of $3 \cdot 2560 = 7680$.

For classification we use a simple one-Nearest-Neighbor Classifier (NN) with L_1-norm and a Support Vector Machine (SVM) with Histogram-Intersection Kernel. Due to the small dataset size it makes no sense to divide the corpus into test- and training set, so we use leave-one-out cross-validation for performance evaluation. Since pollen forecasts are only interested in the allergologically relevant species, we also make experiments where the irrelevant classes merged in one class, resulting in a corpus divided in only 7 classes. In Table 1 the results are presented. Comparing this results to [8], where for all 26 pollen taxa a recognition accuracy of 92% were reached and 97.4% for the 7 class problem, we have nice performance improvements.

Feature	Method	Acc. (26C)	Acc. (7C)
noSH	1NN	87.0%	93.8%
noSH	SVM	93.0%	98.2%
SH	1NN	94.5%	97.9%
SH	SVM	96.9%	99.7%

Table 1. Results for the Pollen datasets. LOO cross-validation accuracy for the different features and classifiers on the fine (26C) and coarse (7C) class distributions.

4.2. Surface Data

The PSB consists of 1814 3D-surface models retrieved from the web. The PSB provides a hierarchical classification, we used the finest granularity with about 90 classes. The features are computed as follows: the models are rendered in a 256^3 voxel-grid by using a binary version of Incremental Triangle Voxelization [4]. Additionally, each voxel gets a reference to the triangle it is stemming from to enable us to incorporate the original surface normals in our calculations. Models collected from the web sometimes contain constructional artefacts inside the closed surface. Since we are only interested in the visual appearance of the model we exclude the artefacts from the computation by doing the following: after voxelization we perform a flood-fill operation starting from the border of the cube. If the surface is closed, only the outer surface is 'wet' and the artefacts inside stay 'dry'. To exclude the artefacts from the inside of the object we only include the 'wet' voxels in our computations. The rest of the feature computation is nearly the same as for the volume data except that each voxel is chosen with equal probability and the surface normals attached to the voxels play the role of the gradients. Again we compare two type of feature: without SH (noSH) and with SH. We use the following quantization $4 \cdot 4 \cdot 4 \cdot 16 = 1024$ of the parameters.

We compare our results to Shape Distribution (D2) [7] and extension of it (AD) [6] , both are invariant features which are obtained without normalization, similar to our features. Further we compare to the so called Light Field Descriptors [3], which are based on 2D-images rendered from different viewing angles. Similarity computation is done by searching for the best matching pair of views. In Table 2 the results are presented. For evaluation we use the five performance measures proposed in [10]. The 1NN-rate is the percentage that a object from the same class has the highest similarity to the query object. For description of the other measures see [10]. The results show that our features outperform D2 and AD. In comparison to the pollen experiments the SH-expansion do not show such an advantage. The LFD-descriptors still outperform our features, but they rely on a matching scheme, i.e. the retrieval times are much higher, hence a direct comparison is not really fair.

Feature	1NN	1T	2T	EM	DCG
noSH	58.1	30.1	40.8	24.1	58.3
SH	60.2	31.8	41.5	24.8	58.5
D2	31.1	15.8	23.5	13.9	43.4
AD	42.9	21.6	31.3	18.8	49.9
LFD	65.7	38.0	48.7	28.0	64.3

Table 2. Retrieval Performance for the PSB.

5. Conclusion and Future Work

In this paper we applied the Group-Integration framework to two different types of 3D-data. We showed that without any special adaption of the features to the specific problem we are able to compete or even outperform state-of-art features. The main contributions of the paper are two extensions of the standard GI-features. We included local directional information and kept more information back by a Spherical Harmonic expansion. We also gave a fast algorithm using importance sampling. For future work we want to use the D-Wigner matrices to keep even more information back than with a Spherical Harmonic expansion.

References

[1] H. Burkhardt, M.Reisert, and H.Li. Invariants for discrete structures - an extension of haar integrals over transformation groups of dirac delta functions. In *Proceedings of the 26th DAGM Symposium*, 2004.

[2] H. Burkhardt and S.Siggelkow. *Invariant features in pattern recognition - fundamentals and applications. In C. Kotropoulos and I. Pitas, editors, Nonlinear Model-Based Image/Video Processing and Analysis.* John Wiley and Sons, 2001.

[3] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Computer Graphics Forum*, volume 22/3, 2003.

[4] F. Dachille and A. Kaufman. Incremental triangle voxelization. In *Proceedings of Graphics Interface 2000*, pages 205–212, 2000.

[5] B. Haasdonk, A. Vossen, and H. Burkhardt. Invariance in kernel methods by haar-integration kernels. In *Proceedings of the 14th Scandinavian Conference on Image Analysis*, pages 841–851, 2005.

[6] R. Ohbuchi, T. Minamitani, and T. Takei. Shape-similarity search of 3d models by using enhanced shape functions. *International Journal of Computer Applications in Technology (IJCAT)*, 23:70–85, 2005.

[7] R. Osada, T. Funkhouser, and B. C. und David Dobkin. Matching 3d models with shape distribution. In *Proceedings Shape Modeling International*, 2001.

[8] O. Ronneberger, H. Burkhardt, and E. Schultz. General-purpose Object Recognition in 3D Volume Data Sets using Gray-Scale Invariants. In *Proceedings of the International Conference on Pattern Recognition*, Quebec, Canada, Sept. 2002.

[9] O. Ronneberger, J. Fehr, and H. Burkhardt. Voxel-wise gray scale invariants for simultaneous segmentation and classification. In *In Proceedings of the 27th DAGM Symposium, Vienna, Austria*, 2005.

[10] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling International, Genova, Italy*, 2004.