CLASSIFIED 3D MODEL RETRIEVAL BASED ON CASCADED FUSION OF LOCAL DESCRIPTORS

Konstantinos Mochament[1], Athanasios Tsakiris[1], Dimosthenis Ioannidis[1], Dimitrios Tzovaras[1]

[1]Information Technologies Institute Centre for Research & Technology Hellas, Thermi-Thessaloniki, Greece

ABSTRACT

One of the core tasks in order to perform fast and accurate retrieval results in a content-based search and retrieval 3D system is to determine an efficient and effective method for matching similarities between the 3D models. In this paper the "cascaded fusion of local descriptors" is proposed for efficient retrieval of classified 3D models, based on a 2D coloured logo retrieval methodological approach, suitably modified for the purpose of 3D search and retrieval tasks that are widely used in the augmented reality (AR) and virtual reality (VR) fields. Initially, features from Key points are extracted using different state of the art local descriptor algorithms and then they are joined to constitute the feature tuple for the respective key point. Additionally, a feature vocabulary for each descriptor is created that maps those tuples to the respective vocabularies using distance functions that applied among the newly created tuples of each Point Cloud. Subsequently, an inverted index table is formed that maps the 3D models to each tuple respectively. Therefore, for every query 3D model only the corresponding 3D models are retrieved as these were previously mapped in the inverted index table. Finally, from the retrieved list by comparing the local features frequency of appearance to the first vocabulary, the final re ranked list of the most similar 3D models is produced.

KEYWORDS

Point Cloud, local descriptor, features tuple, keypoint, cascaded fusion.

1. INTRODUCTION

With the increasingly growing technology of augmented reality (AR) and virtual reality (VR) the even greater growing technologies in the field of mobile devices, there is a major need of computationally inexpensive algorithms that should be efficiently executed in limited resource devices (mobiles ,tablets etc.) and seen as lightweight operators that perform AR and VR tasks. One of the core tasks in the AR and VR field is the search and retrieval of similar 3D models from a classified stored 3D repository. Therefore in this paper a lightweight matching method in terms of processing, power and memory consumption is proposed that could be applied in both limited resource devices or in cloud based infrastructures and could provide almost in real-time the functionality of 3D model matching against a known 3D model dataset. Similar to the very well-known internet text-based search engines where Keywords are matched with keywords from different web pages and lead to web page retrieval, the local features describing Key points are used in the same way. Therefore, Key point features are matched with Key point features from different Point Clouds and lead to Point Cloud retrieval where each point Cloud is a representation of a 3D model from a stored dataset. Providing a 3D model search and retrieval capability to a low energy power and memory device could have unlimited possibilities and could be applicable to several different scientific fields (i.e. robotics, healthcare etc.) that use 3D

DOI : 10.5121/ijcga.2016.6102

modelling for simulation of the real word Use Cases and scenarios. Furthermore, due to the decreasing cost of 3D sensors and in combination with the technological evolution of mobile devices more and more commercial end user applications along with scientific research fields like robotics and computer vision are observed to be built in such a manner that can support AR and VR system functionalities. For that reason and in order to perform AR and VR operations, beyond the accuracy of the performed tasks an important matter that should be always take into consideration, as much as possible, is the reduction in complexity of the implemented algorithm. Due to that in this paper we propose a method that takes into account the above limitations and performs the majority of all heavy computation operations in a pre-process phase so that the task of search and retrieval of 3D models will be performed in almost real time.

2. RELATED WORK

Numerous researches have been conducted and an equal number of methods have been proposed for the task of searching and retrieving 3D models. More specifically, following state of the art methods achieve high accuracy results evaluated against some of the most challenging datasets like SHREC'13[1], '14[2] and '15[3]. The methods proposed include the hydric shape descriptors [4] and meta similarity generation for non-rigid 3D model retrieval [6] that integrates geodesic distance–oriented global features and curvature-oriented local descriptors and in a more general approach can be applied to similar methods that integrates more than one features in order to develop a specific algorithm for both partial and rigid 3D models. Another efficient descriptor methods are the Histograms of Area Projection Transform, the Radical symmetry Detection and the Shape Characterization with Multiscale Area projection transform [5] and theR-BiHDM[2]method where it computes the project matrix k called reduced bi-harmonic distance matrix and the resulted equation is equal to:

$$BiHDM\colon \lim_{m\to\infty} spectrum\,(K_{m,n}) \qquad (1)$$

Regarding the non-rigid a fast non-rigid retrieval method is the HKS-TS[7] where it constructs a context-based integral kernel operator on a manifold and then applies modal analysis to map this operator into low-frequency functional representation and lastly it computes the spectrum that forms the shape based signature. Another efficient descriptor is the High level Feature Learning for 3D Shapes, the Bag-of-Features approach with Augmented Point Feature Histograms, the BoF and SI-HKS and the Spectral Geometry [8]. Regarding the retrieval algorithms evaluated in more recent dataset the methods include the Cross-Domain Manifold for range Scan-based 3D Model Retrieval, the 2D-DTC Coefficient of Silhouettes for Range Scan Matching, the Fast point feature Histograms[3], the Bag of features and Modified Shape Distribution approach for Range Scan Matching[3], the Partial Shape retrieval with Fast Point Feature Histograms(FPFH) and Signature Quadratic Form Distance(SQFD) that is an adaptive similarity measure for content-based feature representations of data, the Depth Image Similarity based method for range Scan Matching and finally the Hough-voting in a Continuous Voting Space for Range Scan Matching[3].Regarding the selected descriptors even though the proposed method could be applied in any pair of local descriptors we select the PFH and the FPFH due to the fact that they were more suitable for the developed engine. Therefore, the studied features [11] presented in the following table.

Table 1. Descriptors important features

Descriptor	Number of points	Number of dimensions	Parallel	Normals
PFH	n	125	GPU	Y
FPFH	n	33	OMP+GPU	Y

3. OVERVIEW OF THE PROPOSED METHOD

The proposed method framework is shown in Figure 1. The presented pipeline is based on an image based method for logo retrieval over common pictures [13]. Based on the idea of the referenced method for logo retrieval that uses cascaded sparse localized colour information we suitably modify the necessary modules and with the addition of the following described components the proposed method is formed for search and retrieval over 3D models. Initially the local features are extracted from the entire dataset using both FPFH and PFH descriptor algorithms. Hence we create two distinct matrices for every 3D model, where each matrix is equal to the number of the Key points of the respective 3D model and the size of the respective descriptor's dimensions. Then, from the extracted features of the dataset we create two local feature vocabularies using the well-known BoF method. Subsequently, using a distance metric function like the Euclidean distance we replace each feature from the two matrices of every 3D model with the closest id from the features of the respective vocabulary. Subsequently an index is created where for every *id* of each vocabulary a mapping is performed between the 3D models that had at least once replace their feature with the respective *id*. By uniting the two indexes we build a unified inverted index where for every tuple of *ids* we map only the intersection of the 3D models that were previously mapped in each index.

Thereby for every query 3D model first features are extracted and then replaced with the closest id as previously described. Next, from the inverted index only the 3D models mapped in the corresponding tuples are retrieved. The final retrieved list is re ranked using the frequency of appearance of the *ids* from the first vocabulary and a cosine similarity between these frequencies.

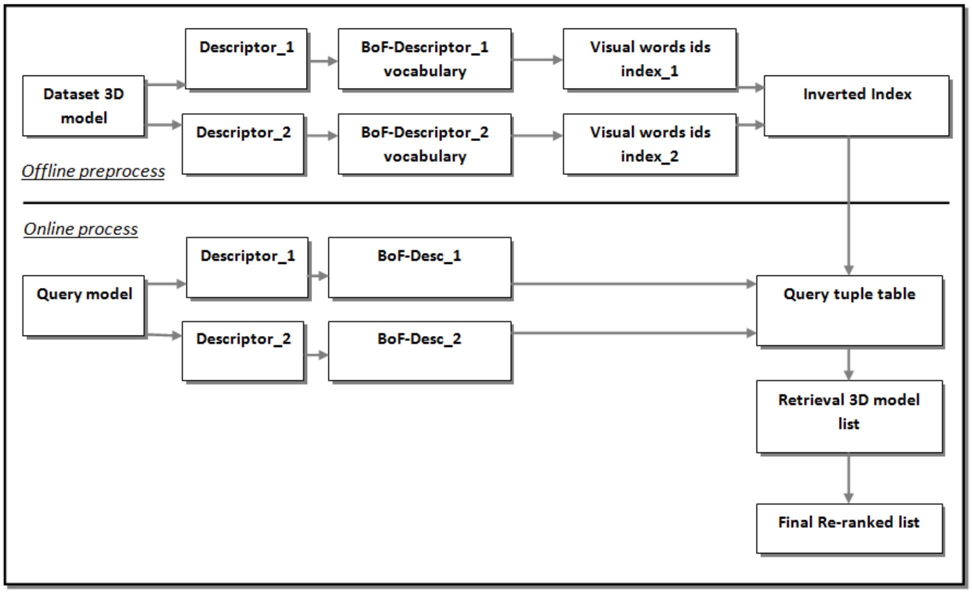


Figure 1.Cascaded retrieval framework

Even though the presented method could be applied in any local descriptors combination, for the purpose of this research only the point feature histograms [12] and the fast point feature histogram descriptors [9] are conducted.

Point feature histogram:

The Point feature Histogram (PFH) descriptor approach is based on computation of both surface normal and also curvature estimation *(xyz* 3D data). To compute the relative difference between two points$p$ and q, first a reference frame is built on p using three vectors *(u, v, w)* where:

i. *vector u is equal to the surface normal at p.*

ii. *vector v is equal to*$v = u \times (p - q)/d$ *(2)and*

iii. *vector w is equal to* $w = u \times v$ *(3)*

Hence using the *(u, v, w)* the distance between *p(np)* and *q(nq)* is estimated by computing the angles *(α,φ,θ)* where:

i. *α is equal to*$\alpha = arccos\ (v\ n\_p)$ *(4)*

ii. *φ is equal to* $\varphi = arccos\ (u\ (p - q)/d)$ *(5)*and

iii. *θ is equal to* $\theta = arctan\ (w \cdot n\_p\ , u \cdot n\_p\ )$ *(6)*

Computing angles *(α, φ, θ)* for all the pairs in the *k*-neighbourhood of *p*, will result into a 125 bin-histogram for each point *p*.

Fast point feature histogram:

Fast point feature histogram (FPFH) is the resulted descriptor algorithm after reduction complexity of the PFH from *O(nk2)* to *O(nk).* This is achieved by compute the angles *(α,φ,θ)* only between *p* and its *k*-nearest neighbours. This will result into the SPFH (simplified point feature histogram) where the final FPFH is produced by the sum of the SPFH and the weighted based on the distance to *p* value of its *k*-neighbour.

$$\mathrm{FPFH_p} = \mathrm{SPFH_p} + \frac{1}{\mathrm{k}}\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{SPFH\ (i)/w_i} \qquad (7)$$

This way, each one of the three angles is described through an *11-bin* histogram, where the concatenation resulted in a *33-bin* histogram. The above ensures that FPFH is invariant to scale orientation and position

3.1. Pre-process

3.1.1 Phase 1: feature extraction and creation of vocabularies

Based on the well-known FPFH descriptor [9] in this phase first every 3D model is converted into Point Cloud and then local features are extracted from all Point Clouds that are stored in our database. This results into an *NxM* matrix where *N* is the number of key points and *M* is the size of the descriptor which for fast point feature histogram (FPFH) is equal to 33 [9]. Having the above matrix for every 3D model in our dataset a vocabulary is generated of visual words. These visual words are the centroids of a clustering obtained using *k-means*, to the entire set of the extracted features. For the *k-means* clustering, our experimental results show that the best optimization is achieved when*k* equals to *20%* of the total features. Repeating the above step using a different local descriptor we create a second *NxM* matrix where again *N* is the same number of key points and *M* is the size of the descriptor which is 125 bins for Point Feature

Histogram [12].Additionally as previously described a second vocabulary of visual words is created again by obtaining *k-means* clustering to the newly created features dataset.It is worth mentioning here that the selected descriptors could be replaced with any other local feature descriptors

3.1.2 Phase 2: mapping of the 3D models against the vocabularies' ids

Mapping the two matrices of each 3D model to the two vocabularies created in phase 1 the matrices are encoded by replacing each one of the feature with the closest visual word of the respective vocabulary. Hence the two matrices are replaced with the ids of the closest visual word from the vocabulary. If *N* is the total number of the first descriptor set and K is the total number of the respective visual word vocabulary, then for every *ith (i=1,2,3,....N)* we perform a brute force matching against the *K* visual words and we select the *jth(j=1,2,3....K)* visual word *id* that has the minimum distance *d* to the *ith* feature. As a distance metric we use the simple Euclidean *L* distance where:

$$\mathrm{L} = \mathrm{d}(\mathrm{p},\mathrm{q}) = \sqrt{(\mathrm{p}_1-\mathrm{q}_1)^2+(\mathrm{p}_2-\mathrm{q}_2)^2+\cdots+(\mathrm{p}_\mathrm{m}-\mathrm{q}_\mathrm{m})^2} \qquad (8)$$

Therefore every *ith* feature is replaced with the *id* of the *jth* visual word.

3.1.3 Phase 3: Inverted indexing of the 3D models dataset

In this phase we extend the created vocabularies by mapping the 3D models that have at least one appearance in the respective visual word of every vocabulary. Therefore having the resulted matrices of the previous phase we link the 3D models of our dataset with respect to the *ids* of the visual words. Finally, from the above we create an inverted index table consisted of visual word id tuples and the intersection of 3D models between the two visual word ids. If for instance {*id1, id7, id567,...,idm*} is the vi encoded descriptor of the *ith* 3D model the inverted index will map the ith 3D model only to the visual words with the *ids: ida1, ida7, ida567,...,idam* respectively. Doing that for all the 3D models of the dataset and then form the inverted index will result in a table of *ids* where each *id* will be linked with the 3D models that has at least one appearance, for example let *idai* is the following set *{3Dmodel1,3dmodel5,..., 3DmodelN etc.}*. Repeating the above for the second (b) visual word vocabulary and joining the two inverted indexes will result into an *NxM* table, where *N* is the size of the first vocabulary and *M* is the size of the second. To form the final inverted index table we only need to include for every possible tuple of *ids* the intersection of their linked 3D models. For instance, if*{3Dmodel1, 3Dmodel5, 3Dmodel897}*is the *idai* set and*{3Dmodel7, 3Dmodel11, 3Dmodel5}*is the *idbi* set. Then, the intersection of these two sets is described by the following equation:

$$(\mathrm{Idai} \cap \mathrm{Idbi}) = \{x : x \in idai \wedge x \in idbi\} \qquad (9)$$

Hence, in the final inverted index table only the 3Dmodel5 will be linked to that tuple because it's the only 3D model left after the performed intersection of the two sets. Therefore, equation (9) will be equal to:

$$(\mathrm{Idai} \cap \mathrm{Idbi}) = \{\mathrm{3Dmodel5}\} \qquad (10)$$

3.2 Phase 4: Query 3D model processing

Having a query model its set of features is extracted using the two different descriptors thatwere also used in the first phase for the entire dataset and then the two matrices are encoded by replacing the features with the closest visual word id from the respective vocabularies. Hence by the union of these two matrixes we constitute a new query table that consists of visual word ids tuples.By comparing this query table to the inverted index table we retrieve only the 3D models that appear in the respective tuples eliminating the rest of the dataset.For instance if the closest tuple of *ids* for a single keypoint from the query 3D model is: (*idan, idbm*) then from the inverted index table we find the corresponding tuple and retrieve the linked 3D models.

3.3. Phase 5: Re ranking of the retrieved list

The final list of 3D models is re-ranked using as a distance metric the cosine similarity function betweenthe frequencies of appearance histograms based on the visual words vocabulary created from the first descriptor.

$$s(A,B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \qquad (11)$$

Adding the weighted class distance

As an additional final step and using the information of the predefined categorization of the database 3D models we can achieve significantly better optimization in the task of category recognition by taking into account the distance between the query model and each and every class of the dataset. Adopting the proposed method [15] we compute this step and weight it accordingly so that the final list generated based on the distance described above and the class distance computed in this phase. As proposed, to measure the class distance we can use the centroid the average or the minimum distance. Since average and centroid methods need more computations we use the minimum distance class.

$$d_c(q, C_j) = \min_{m \in C_j}\{d(q,m)\} \qquad (12)$$

The selection of the minimum against the other two methods is based on the fact that since it is the distance between the query 3D model and the closest 3D model from the respective class it has already been computed in the previous phases. Hence without adding any complexity to our method we integrate the class distance information in our final results. Lastly, for the computations of the final distance in order to identify the query 3D model category based on the proposed method [16] we adopt the following equation.

$$d(q, C_j) = [\left(\frac{1}{k}\right) \sum_{i=1,2,\ldots,k} d_c \min(q, C_j)] \qquad (13)$$

Where Cj is the *jth* class, *k* is the *kth* 3D model of the ranked list and dc is the respective class distance. The weights *(k=1/k)* are linearly decreasing having as a starting point the first 3D model in order to give more weight in the class of the closest 3D models.

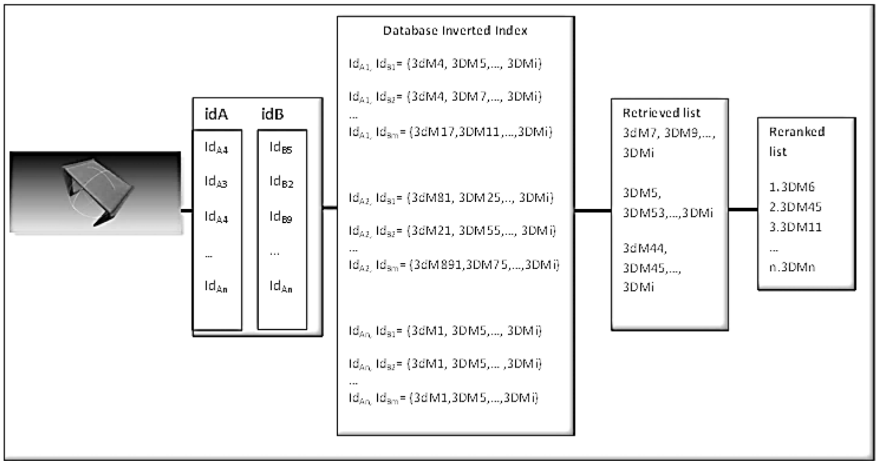
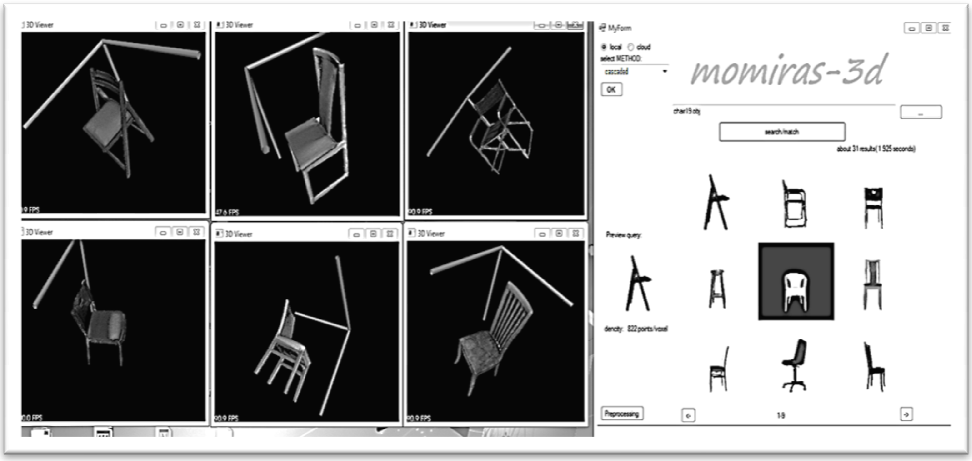

idA
idB
Id_{A4}
Id_{A3}
Id_{A4}
...
Id_{An}
Id_{B5}
Id_{B2}
Id_{B9}
...
Id_{An}
Database Inverted Index
$Id_{A1,}$ Id_{B1}= (3dM4, 3DM5,..., 3DMi)
$Id_{A1,}$ Id_{B2}= (3dM4, 3DM7,..., 3DMi)
...
$Id_{A1,}$ Id_{Bm}= (3dM17,3DM11,...,3DMi)
$Id_{A2,}$ Id_{B1}= (3dM81, 3DM25,.., 3DMi)
$Id_{A2,}$ Id_{B2}= (3dM21, 3DM55,..., 3DMi)
...
$Id_{A2,}$ Id_{Bm}= (3dM891,3DM75,...,3DMi)
$Id_{An,}$ Id_{B1}= (3dM1, 3DM5,..., 3DMi)
$Id_{An,}$ Id_{B2}= (3dM1, 3DM5,... ,3DMi)
...
$Id_{An,}$ Id_{Bm}= (3dM1,3DM5,...,3DMi)
Retrieved list
3dM7, 3DM9,..., 3DMi
3DM5, 3DM53,...,3DMi
3dM44, 3DM45,..., 3DMi
Reranked list
1.3DM6
2.3DM45
3.3DM11
...
n.3DMn

Figure 2. Cascaded retrieval framework

4. EXPERIMENTAL SETUP AND BASELINE

Database and setup

In order to evaluate the proposed method, a subset from a large 3D models dataset was used. The initial set was created by CERTH/ITI [16] one of the leading Institutions of Greece in the fields of Informatics, Telematics and Telecommunications, with long experience in numerous European and national R&D projects. Additionally, using the Structure Sensor [15]a second smaller dataset was created with higher quality 3D models in order to evaluate our method in a second more challenging dataset. The first dataset consist of a training set of 565 3D models from 10 different categories and the second training set consist of 110 high quality 3D models. For the test sets we use two sets, approximately equal to the one third of the training sets, namely 100 and 45 3D models respectively. The two complete sets are publicly available through the website of the Momiras project [14]. Regarding the down sampling process, all features were extracted after sampling the Point Clouds with a grid size of 1.0cm.


● local ○ cloud

select METHOD: cascaded

chair15.obj

about 31 results(1.525 seconds)

density: 822 points/voxel

1-9

Figure 3.Retrieval results from a search performed using a chair 3D model as a query

Baseline and metrics

In order to evaluate our proposed method we compare our results with the retrieval results, achieved by each of the descriptors algorithm separately. Hence since our proposed method uses the combination of FPFH and PFH descriptor methods the performance of the two descriptors separately was measured as well. Moreover, the same pipeline used for the proposed method was adopted again with the only difference that after the extracted descriptors direct similarity operations were performed without the further pre-process steps. Finally to define a similarity metric between two sets of extracted descriptors the following distance metric was used where that is the sum of the distance between the centroids and the distance of their standard deviation respectively [11].

$$\mathrm{D}(\mathrm{A},\mathrm{B}) = \mathrm{L}_1(\mathrm{c}_\mathrm{A},\mathrm{c}_\mathrm{B}) + \mathrm{L}_1(\mathrm{std}_\mathrm{A},\mathrm{std}_\mathrm{B}) \tag{14}$$

$$\mathrm{std}_\mathrm{A}(\mathrm{i}) = \sqrt{\frac{1}{|\mathrm{A}|}\sum_{\mathrm{j}=1}^{|\mathrm{A}|}\left(\mathrm{a}_\mathrm{j}(\mathrm{i}) - \mathrm{c}_\mathrm{A}(\mathrm{i})\right)^2}\,, \mathrm{i} = 1,2,3,..\mathrm{n} \tag{15}$$

$$\mathrm{L}_1(\mathrm{a},\mathrm{b}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{a}(\mathrm{i}) - \mathrm{b}(\mathrm{i})| \tag{16}$$

Evaluation metrics

To evaluate our proposed method we compare the methods using the Mean Average Precision where the Average precision achieved by each query is computed for the total number of the performed queries

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{17}$$

5. RESULTS

Table 1&2 presents the experiments' results where the retrieval methods were ranked according to the mAP score they achieve.

Table 1. : First set mean average precision *(MAP)* scores.

Method	Map score
PFH	0.386
FPFH	0.357
Cascaded local descriptor	0.486

Table2. : Second set mean average precision *(MAP)* scores.

Method	Map score
PFH	0.537
FPFH	0.515
Cascaded local descriptor	0.607

Table 3&4 presents the category recognition based on the overall accuracy of the performed queries

Table 3. : Category recognition accuracy of the first set.

Method	Category Recognition
PFH	39%
FPFH	36%
Cascaded local descriptor	49%

Table 4. : Category recognition accuracy of the second set.

Method	Category Recognition
PFH	54%
FPFH	52%
Cascaded local descriptor	61%

6. CONCLUSION

Having a 2D-based methodological approach for coloured logo retrieval over image datasets as a reference point, this retrieval approach is expanded in the 3D field by applying the suitable transformations and the appropriate additions in order to develop the proposed 3D search and retrieval method. According to this, a shape based retrieval approach for 3D models was presented. The proposed method takes advantage of more than one local descriptor algorithms and by creating several lists of 3D models for every tuple of local features, it decreases the final number of retrieved 3D models. Hence by performing similarity metrics only to the final significantly smaller list of 3D models we achieve fast and accurate results. Additionally the pre-processed inverted index ensures fast retrieval without a one on one (linear complex) comparison on all of the stored 3D models but instead a selective retrieval based only on the corresponding subsets. Furthermore, the usage of the classification information of the initial models in the two newly created 3D model datasets that are publicly available under the Momiras project official page ensures a significant optimization in the task of object category recognition. Finally, in future workthe presented methodology will be applied using different local feature extraction algorithms and using also different 3D model datasets in order to confirm and refine the presented approach.

Acknowledgements

The research leading to these results has received funding from the Momiras bilateral Greece-Israel 2013-2015 research and technology collaboration project.

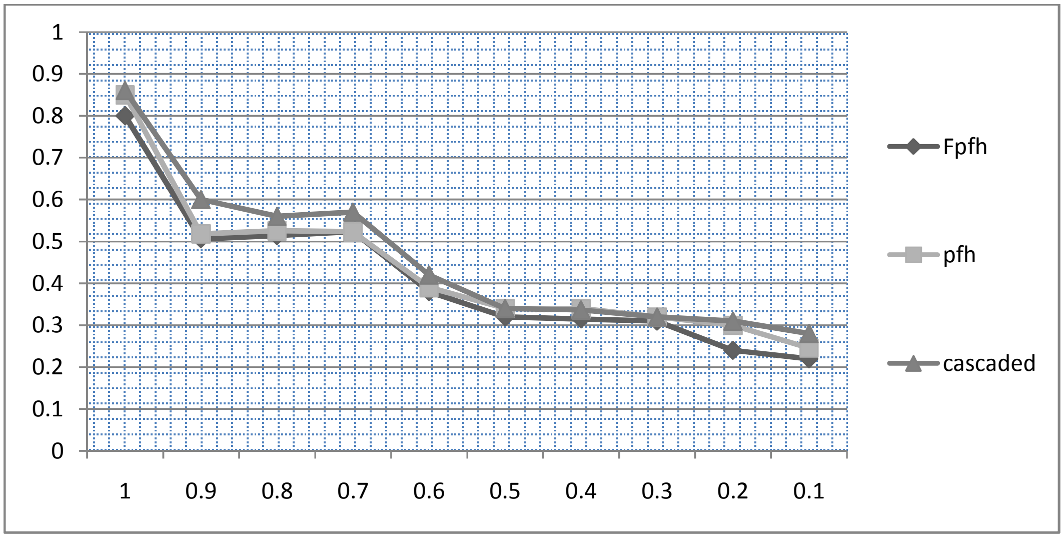


Figure 4. Precision Recall on the first dataset

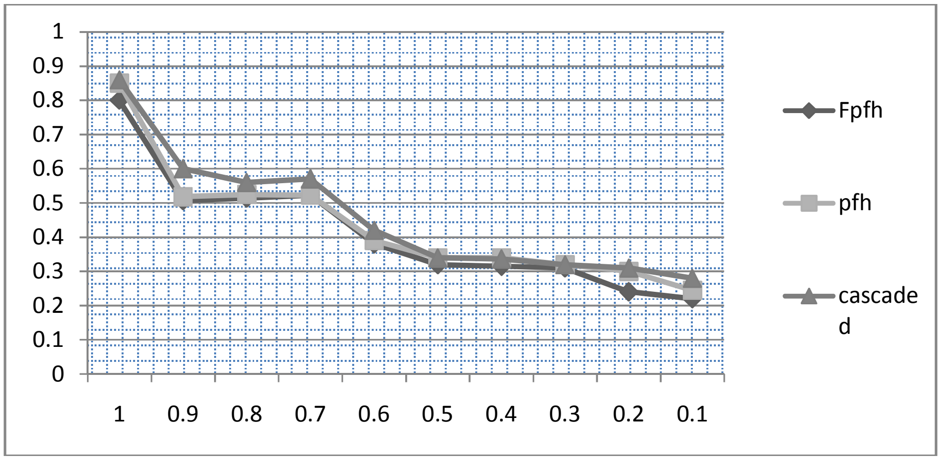


Figure 4. Precision Recall on the second dataset

REFERENCES

[1] LI, Bo, et al. SHREC'13 track: large scale sketch-based 3D shape retrieval. In: Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval. Eurographics Association, 2013. p. 89-96.

[2] LI, Bo, et al. Shrec'14 track: Large scale comprehensive 3D shape retrieval. In: Eurographics Workshop on 3D Object Retrieval. 2014. p. 131-40.

[3] GODIL, A., et al. SHREC'15: Range Scans based 3D Shape Retrieval.

[4] LI, Bo; GODIL, Afzal; JOHAN, Henry. Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3D model retrieval. Multimedia tools and applications, 2014, 72.2: 1531-1560.

[5] GIACHETTI, Andrea; LOVATO, Christian. Radial symmetry detection and shape characterization with the multiscale area projection transform. In: Computer Graphics Forum. Blackwell Publishing Ltd, 2012. p. 1669-1678.

[6] YE, Jianbo; YAN, Zhicheng; YU, Yizhou. Fast nonrigid 3D retrieval using modal space transform. In: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval. ACM, 2013. p. 121-126.

[7] SUN, Jian; OVSJANIKOV, Maks; GUIBAS, Leonidas. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In: Computer graphics forum. Blackwell Publishing Ltd, 2009. p. 1383-1392.

[8] LI, Chunyuan. Spectral Geometric Methods for Deformable 3D Shape Retrieval. 2013. PhD Thesis. Concordia University.

[9] RUSU, Radu Bogdan; BLODOW, Nico; BEETZ, Michael. Fast point feature histograms (FPFH) for 3D registration. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009. p. 3212-3217.

[10] OHBUCHI, Ryutarou; FURUYA, Takahiko. Accelerating bag-of-features sift algorithm for 3d model retrieval. In: Proc. SAMT 2008 Workshop on Semantic 3D Media (S-3D). 2008. p. 23-30.

[11] ALEXANDRE, Luís A. 3D descriptors for object and category recognition: a comparative evaluation. In: Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal. 2012.

[12] RUSU, Radu Bogdan, et al. Aligning point cloud views using persistent feature histograms. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE, 2008. p. 3384-3391.

[13] PANDEY, Rohit, et al. Cascaded sparse color-localized matching for logo retrieval. In: Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, 2014. p. 2207-2211.

[14] 160.40.50.50/Momiras,. "Momiras Content Import Toolkit". N.p., 2015. Web. 3Nov. 2015.

[15] Structure.io,. "Structure Sensor - 3D Scanning, Augmented Reality, And More For Mobile Devices". N.p., 2015. Web. 3Nov. 2015.

[16] Iti.gr,. "ΙνστιτούτοΤεχνολογιώνΠληροφορικήςΚαιΕπικοινωνιών". N.p., 2015. 3 Nov 2015.

Authors

Short Biography

Mochament Konstantinos is a postgraduate student in Information and Communication Technology (ICT) Systems of International Hellenic University (IHU) and from November 2014 is working as a Research Associate in Informatics and Telematics Institute of the Centre for Research and Technology Hellas in Thessaloniki (CERTH/ITI). His main research interests are in the areas of 3D data processing.

Thanos Tsakiris is a Research Associate in the Informatics and Telematics Institute. He received the Diploma in Computer Science from the Aristotle University of Thessaloniki, Greece and the MSc in Computer Games Technology from the University of Abertay Dundee, UK in 2000 and 2001 respectively.His main research activities include 3D rendering, VR/AR/MR technologies and applications, Multimedia Authoring and Multi-modal Human-Computer Interfaces. His involvement in these research areas has led to his participation in the authoring of 2 articles in refereed journals, 2 papers in international conferences and the co-authoring of 1 book chapter.

Mr. Dimosthenis Ioannidis is working from October 2006 as a Research Associate in Informatics and Telematics Institute of the Centre for Research and Technology Hellas in Thessaloniki.His main research interests are in the areas of biometrics (gait and activity-related recognition), 3-D data processing, and web semantics.

Dimitrios Tzovaras is a senior researcher (Grade A’) and the directorof the Information Technologies Institute (ITI) and a visiting professorof the Imperial College London. His main research interests includevisual analytics, computer vision,knowledge management, multimodalinterfaces, and virtual reality. His involvement with those researchareas has led to the co-authoring of over 80 articles in refereed journalsand 200 papers in international conferences. He has served as a regular reviewer for a number of international journals andconferences.