Warsaw University of Technology

Faculty of Mechatronics

Institute of Automation and Robotics

Adam Kosiorek

3D Object Classification based on RGBD images

bachelor thesis written under supervision of

prof. dr. hab. Barbara Siemiątkowska

Warsaw 2014

Streszczenie

Properties of special generalized functions and short-range selectors which are indispensable in the Shina Tan description of interacting Fermi systems [Annals of Physics 323, 2952 (2008)] are analyzed in details. Also all details of the proof of the Shina Tan energy theorem and properties of the contact are presented. The manuscript should be interesting for researchers who want to understand all elements of the Shina Tan extraordinary work.

Abstract

Properties of special generalized functions and short-range selectors which are indispensable in the Shina Tan description of interacting Fermi systems [Annals of Physics 323, 2952 (2008)] are analyzed in details. Also all details of the proof of the Shina Tan energy theorem and properties of the contact are presented. The manuscript should be interesting for researchers who want to understand all elements of the Shina Tan extraordinary work.

Contents

Chapter 1

Introduction

1.1 Bag of Words image representation

As portrayed by Tsai in [17] the Bag of Words (BoW) model has originated from the text retrieval domain. The model enables text document representation in a form of a vector. Suppose we have a corpus of documents. A dictionary over the corpus is a set of words such that every word from the corpus can be matched to exactly one word in the dictionary and every two words in a dictionary are different. Given a dictionary, a histogram of incidence of words can be build. This histogram has the following form: for every entry in the dictionary there is a number of words in the document matching the entry. Such an approach obliterates any grammatical dependencies in order to retain only statistical information associated with words. It is believed that the words' frequencies encode enough data for further processing. The following questions emerge:

- How does this apply to the area of computer vision?
- How can one convert an image to a text document?

In order to answer the first question one has to consider what digital images are — they are nothing but strings of binary code. Translated to i.e. RGB colour space they can be displayed and understood by humans, but there are no direct methods that would let computers 'understand' what does a picture contain. Suppose two images of the same object were taken from different viewing angles. Being easy for a human, it is impossible for any computer to compare those two pictures (or objects they depict) in any simple manner. This example shows a need for an intermediate representation that would enable image processing. Advanced algorithms for characteristic region detection and description exist (SIFT, SURF, FAST among many others). Even then, however, the resulting representation is a low level one. Consider a scenario in which a robot is to adjust its behaviour depending on where it currently is (a type of a location, i.e. house, factory, mine). It order to define required adjustments a programmer would have to encode the low level representation of possibly many robot's destination environments. It is a variation of a problem described by Tsai [17], with respect to the Content Based Image Retrieval (*CBIR*), as *"the gap between the extracted and indexed low-level features by computers and the high-level concepts (or semantics) of user's queries"*. Suppose a text document can be created from an image. This document can be translated into a

BoW model. Since it preserves semantic information that can be extracted an impact of the semantic gap can be reduced.

As for the second question, there is a quite well established pipeline that enables creation of text documents from images. The steps of the BoW methodology are as follows:

- Keypoint detection — keypoints are local interest points or regions. They are usually computed in such a way so as to provide scale and location invariance. Rigid transformation and illumination invariance would be desired but it is somewhat harder to achieve
- Keypoint description — each keypoint have to be described in a manner that distinguishes the particular keypoint in some way.
- Vector quantization — clustering algorithms are used to find regions in the high dimensional space of keypoint descriptors. When clusters are found, each described keypoint can be assigned to a corresponding cluster. Then, an image can be represented by the numbers of clusters, to which the image's keypoints were assigned. The numbers themselves are called *'visual words'*.

1.2 BoW in Computer Vision

The Bag of Words image representation has been extensively used in the areas of scene [2, 3, 17] and object categorisation [19] as well as CBIR [8, 16] yielding state-of-the-art results. Advantages of this model are simplicity, computational efficiency and at least partial invariance to affine transformation, occlusion and lighting conditions.

The Bag of Words is a type of intermediate representation. Therefore it is not sufficient to compute a BoW model of an image in order to predict its category. Additional operations are essential if the model is to be used in one of the enumerated fields.

1.2.1 Content Based Image Retrieval

One of the most notorious use-case of *CBIR* is to search a database in order to find an object fulfilling certain conditions. As [16] outlines, several criteria have to be met for the task to be performed efficiently. Firstly, all entries should be indexed in a concise way.

Secondly, some (dis)similarity measure should be provided. Finally, an efficient search algorithm should be available.

There are numerous methods suitable for computation of objects' signatures. Many of them can be used in the indexing step. All the methods were divided into three general categories in [16], specifically: feature based methods, graph based methods and other methods.

Feature based methods can be either global or local. The former takes the form of a single vector or a point in a d dimensional space — the similarity measure being a point-wise distance in that space. The latter gives multiple such points for each object, rendering the computation of a similarity measure slightly more complicated. The global features takes forms of the models' volume, their mass or mass distributions. Others might incorporate the global features' distribution — one conceivable approach would be to compute global features' distributions and summarise them into a histogram. These features have the advantage of being easy to compute and straightforward to implement. However, they are insensitive to any local shape variations, thus being ill-suited for detailed comparisons. On the other hand, they might be exploited in the preprocessing of the data with more sophisticated methods being used afterwards. Partial matching is not possible, since the global features do not encode any relations between parts of the objects.

As for the local features, [16] discusses only features describing neighbourhood of the points on boundaries of objects. Any settings that does not match this criteria are neglected. Moreover, the authors state that the local feature based methods are inefficient and indexing is rather complex. A Bag of Words based approach, described in [8] has no such drawbacks.

Bag of Words techniques can be regarded as feature distributions, even though they are local feature based — the local features being visual words. When all the image's visual words are summarised into a histogram a distribution is created. Being similar to a point in a multi-dimensional space, it is similar to a global feature. Consequently, similar methods apply, with the distinction being that the histogram is rather a vector then a point. Thus metrics well-suited to vector comparison, such as a cosine distance, can be used.

1.2.2 Scene Categorisation

One of the first works employing the BoW for the purpose of scene categorisation is [2]. The authors suggested a general framework. What is more, algorithms performing each main pipeline's step were proposed and evaluated as well as a codebook construction method was developed. The *Harris Affine Detector* was used for feature extraction and the *SIFT* for the feature description step. The visual vocabulary has been generated by clustering only a limited number of keypoints from each category. After summarising every image in the training set with a histogram of visual words, supervised learning was used to train two classifiers: Naìve Bayes and a Support Vector Machine. As expected the Naìve Bayes resulted in a lower accuracy than SVM (72 vs. 85% on 7 categories). An analysis on the number of centroids in the clustering step depicts that the greater the size of the visual vocabulary the greater the accuracy.

Li *et al* further refined the above approach by examining several keypoint detectors and descriptors. The main contribution of their work is, however, the development of a genuine classification algorithm based on a probabilistic graphical model. Accuracy of 76% on a large 13 category dataset was achieved.

1.2.3 Object Categorisation

In many cases object categorisation might be addressed as a scene categorisation problem. There are following differences: (1) an object should be localised on an image (i.e. its bounding box has to be found) and (2) in object categorisation task information about a scene type might be used. The second case is symmetric, for in the scene categorisation problem information about objects present in the scene can be utilised as well. Having said that, it is possible to consider an object categorisation problem where images of singled-out objects are provided — e.g. a single object covers the majority of an image's area. Such an approach was exploited in a recent work by Zhang *et al* [19].

1.3 BoW based classification

The Bag of Words intermediate image representation provides a concise way of summarising images. It is invariant to affine transformations, partial occlusion and, in some extent, to changing lighting conditions. What is more, it is easy to implement and computationally efficient. The efficiency can be enhanced even further with gpu based implementations as shown in [18]. In order to incorporate BoW into an object classification framework one has to feed the resulting histograms into a classification algorithms. Either generative or discriminative methods can be used. Below the main steps of BoW classification processing pipeline will be discussed.

1.3.1 Detection

Characteristic point detection is the first step in any Bag of Words framework. Numerous detection methods have been developed, but choosing the right one for the particular case might prove tricky. A good overview of various mechanisms is available in [17]. The most common detectors make use of a Harris corner detector or image's first or second derivatives. A technique taking advantage of the Harris corner detector is for example a Harris-Laplace detector — the Harris function is scale adapted and its outcome is a subject to a Laplacian-of-Gaussian operator, which selects relevant points in the scale space. Images' regions' 2$^{\text{nd}}$ derivatives — namely the regions' Hessians — can be combined with a LoG operator. This combination allows selection of points significant in the two spaces: the scale space and the Hessian's determinant space. The latter entails the speed at which pixel intensities change in the neighbourhood of a point.

A number of more complicated recipes for salient region localisation have been developed and implemented. These include Scale Invariant Feature Transform [9], SUSAN [15] and Intrinistic Shape Signatures [20]. The majority of keypoint detection formulas is being developed for the 2D domain. A number of them have been adapted to 3D, however. A comparative evaluation of detection algorithms available in PCL can be find in [4]. Another comprehensive study is [14].

All these formulas, called sparse feature detectors, resort to selection of maxima in specific state spaces. An entirely different scheme is to use a dense feature detector. This particular form requires users to specify a uniformly sampled grid from which points are taken. Dense detectors have an advantage of taking points from slow changing regions. A sparse detector might be unable to summarise a slow changing region such as clouds, sky or ocean. Li *et al* showed that dense detectors outperforms the sparse ones.

1.3.2 Description

Computing localisation of a salient point is not enough. If a keypoint is to be affine transform invariant, it has to be described in more general terms. Such description, usually in a form of coordinates in a multi-dimensional space, is provided by specialised algorithms. 128-dimensional SIFT [10] is a 3D histogram of gradient locations and orientations. It is the most often extracted as well as one of the most effective descriptors [17]. Other methods include various colour descriptors, binary descriptors such as 512-dimensional GIST [11]. There are techniques designed for 3D exclusively. Among them one can find Persistent Point Feature Histogram [13] and its faster alternative Fast Point Feature Histogram [12], both implemented in PCL.

1.3.3 Vector Quantization

The final step of extracting Bag of Words features is vector quantization. Generally clustering with the kMeans algorithm is used [17]. The kMeans algorithm was developed in the 50's and a variety of modifications have been developed since [6]. Some of them are: faster than the original *approximate kmeans*, *hierarchical kmeans*, which automatically chooses the resulting number of clusters and a *soft kmeans* — a variation of the algorithm that allows a fuzzy alignment (*i.e.* each point can belong to several clusters with different weights. The soft kMeans is a compromise between kMeans' (relative to GMM, discussed below) low computational cost and GMM's precision. The number of resulting visual words depends on the number of clusters.

In order to improve performance multitude of pre- or postprocessing techniques can be resorted to. A weighting scheme such as Term Frequency (TF) or Term Frequency - Inverse Document Frequency (TF-IDF) can be used. Spatial information might be encoded so as to capture spatial relations of the extracted features.

It has been shown that the vector quantization step is the computationally most expensive step in any Bag of Words framework. Fortunately, the majority of the cost is associated with the training part of the computation. When all the models are trained the only operation required in case of vector quantization is keypoint – visual vocabulary matching. Even so, diverse algorithms have been used in order to minimise the impact of clustering on the overall efficiency. The *approximate kMeans* is faster but insufficiently so as to solve the problem. Random Forests algorithm is considerably less expensive and can provide better Mean Average Precision (MAP) score.

Quite a different approach is using a Gaussian Mixture Model (GMM) algorithm. It is many times more expensive than kMeans, the tradeoff being higher precision. The GMM finds not only clusters' centroids but the Gaussian distributions of points in each cluster. Therefore, in the keypoint-centroid matching stage a set of probabilities is obtained instead of a simple single-cluster assignment. These probabilities encode likelihoods of the keypoint belonging to each cluster. It is up to the user how to utilise this additional information.

1.3.4 Classification

Predicting a class associated with an image requires feeding the visual words histograms into a classifiers. A simple example of a classifier would be a K-nearest neighbours algorithm. In this setting every histogram is treated as a point in a multi-dimensional space. A class of an image is determined by classes of the nearest neighbours. Manifold of distinct metrics can be used: L1, L2 and others. The number of nearest neighbours (K) has to be determined. More advanced classifiers can be divided into the two main classes: generative models and discriminative models.

Construction of a discriminative model requires a supervised learning approach. It can be seen as a learning by example method. The general aim is to compute an approximate mapping between representations of examples (the train set). The purpose of this mapping is correct (e.g. with some success rate) prediction of labels for inputs which label is unknown. The model computation stage is called *training* of a model. After the model has been trained previously unseen examples can be fed into the classifier. The classifier calculates some similarity measure between the unlabelled input example and all the modeled classes. The resulting label is a label of which class had the highest score in terms of the similarity measure. The KNN is an example of a discriminative model. The most widely used discriminative classificator is a Support Vector Machine.

The generative models are usually Bayesian text-based models. Many of them heavily depends on the Probabilistic Graphical Models concept. They include *Latent Semantic Analysis* with *probabilistic Latent Semantic Analysis*, *Latent Dirichlet Allocation* and others. These models allow discovery of topic distribution in documents. In the image domain an image can be though of as a document and an object category as a topic.

Under this conditions an image (and its category) can be modeled as a mixture of topics. Learning consists of finding topics distributions for each specific category.

1.4 Libraries

1.4.1 PointCloud Library

1.4.2 OpenCV

1.4.3 Boost

1.5 Datasets

Multiple RGBD datasets have been created. Unfortunately, only a few of them are fit for the problem being discussed. The majority of available databases are constructed around a theme of object tracking. They usually contain (short) video sequences depicting either specific objects or complex environments containing these objects. Some of the unfit databases can be fitted for the purpose of single-image object classification, but it would require additional computation time. Should that be of no concern, Microsoft Kinect Fusion technology (or it's open source pcl implementation) could be used to construct a detailed point cloud from an RGBD move sequence.

Recently Zhang *et al* compiled a large dataset of good quality RGBD data of objects from 10 categories [19]. Another suitable dataset is the Berkely 3D Object dataset (B3DO) [7]. While they are not perfect, they were used to evaluate the proposed method after preprocessing that will be discussed below.

1.5.1 B3DO

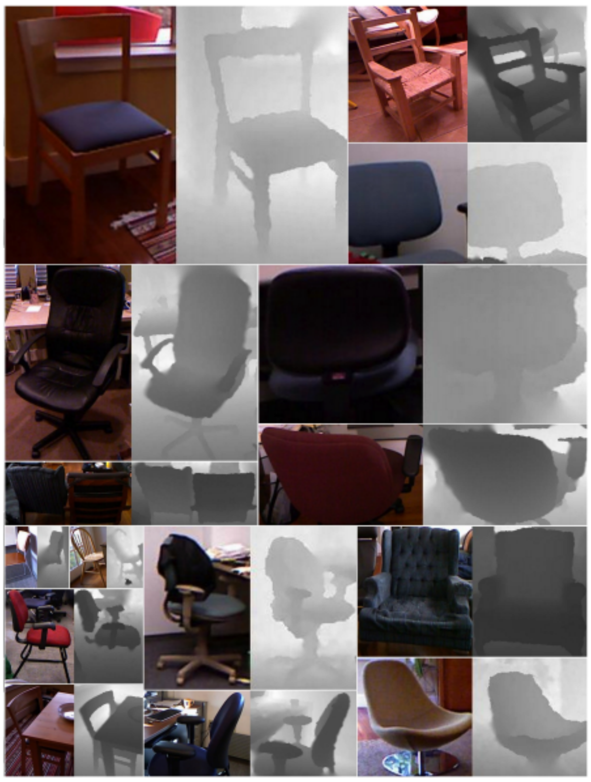

Figure 1.1: Berkely 3D Object Dataset

1.5.2 tokyo

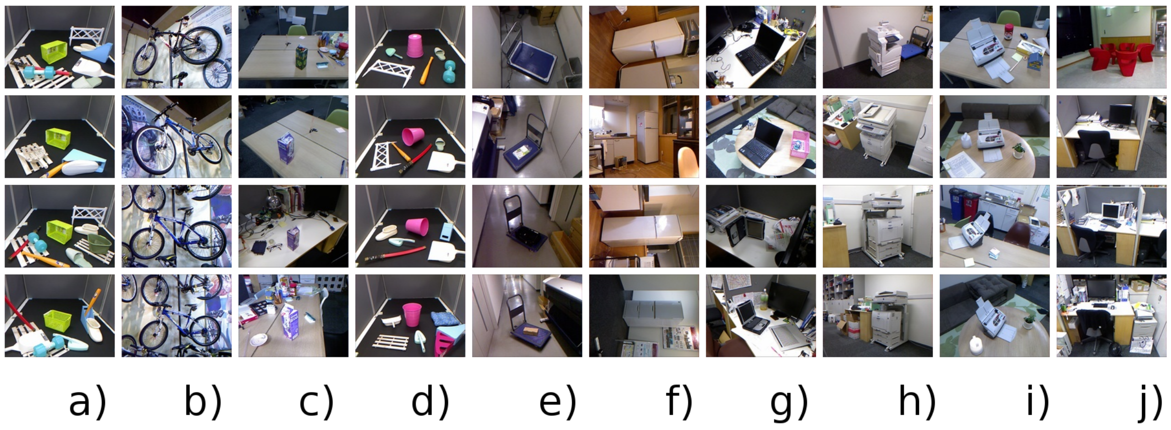

Figure 1.2: The

Chapter 2

Design

2.1 Software Functionality and Architecture

The programme resulting from this work is called “Tagger3D”, for it’s purpose is to tag an RGBD image of an object with a category label. The Bag of Words classification pipeline consists of four main parts discussed previously. In order to determine the best combination of algorithms for any given task an extensive research and evaluation has to be performed. In order to simplify the process the software system should be configurable, extensible and easy to maintain. Moreover, if it is to be used in any realistic setting it has to be efficient as well.

2.1.1 Required functionality

There are three major functions that have to be supplied by the programme: (1) estimation, (2) batch inference and (3) inference. The first one is responsible for the model training. In order to perform this operation the following steps have to be completed: (i) detection and description of keypoints on the training set, (ii) codebook generation, (ii) classifier model training. The functions (2) and (3) are somewhat similar. Both require: (i) detection and description of keypoints on the test set, (ii) parsing keypoints’ description into a visual vocabulary description and (iii) classification. The only is in the (i) step, e.g. the batch inference requires every image from the test set to be processed, whereas the inference can be done on any single image. The (3) have to be distinguished because it is the function that is to be used in a fully functional system mounted on i.e. a mobile robot.

Input/Output operations

Additional functionality have to be provided in the field of input/output operations. Before any processing can be done each image have to be read into memory. Both batch inference and estimation are dataset dependant. What I mean by this is that a previously compiled database have to be accessible. The inference, on the other hand, is meant to be used in a production environment i.e. with a Kinect connected only. While the OpenNI Grabber framework from the PointCloud Library is essentially what is required for the inference, a little more elaboration has to be done on the demands of the estimation and batch inference environments. The two third party datasets used in the project have different formats and need to be tackled separately. The B3DO object dataset contains multiple objects in a single image. The locations of the images are available in xml files, while rgb and depth data are stored in separate files, 8-bit 3 channell jpeg and 16 bit one channel png respectively. Single objects have to be extracted from both pictures using the provided annotations and point cloud have to be build. The Zhang's dataset contain single objects. The rgb data is stored in 8 bit 3 channel jpegs and csv files contains euclidean coordinates of every rgb pixel. Both databases contain objects' labels that have to be loaded separately.

A different problem is serialisation. If the models from the training stage are to be used during inference, they have to be serialised. This means that a standardised way of saving and loading of kMeans and SVM models have to be provided.

Configuration

Every algorithm employed in the project have several configuration parameters. If the parameters were to be compiled with the code, every change would require a rebuild of the application or its parts. In order to address this issue a means of configuration from outside the code should be provided.

Interchangeability of algorithms

Multitude of available algorithms makes it time consuming and computationally expensive to evaluate every available option. It should be possible to compile the programme with many algorithms suitable for performing the same task and decide which one should be used at run-time.

2.1.2 Architecture

The described functionality can be achieved in the following way. The utilisation of Object Oriented Programming (OOP) can lead to production of modular, flexible and extensible software. Using C++ as the language of choice can result in high efficiency. The excellent metaprogramming tools provided by the C++ language help minimise the length of resulting hand-written code.

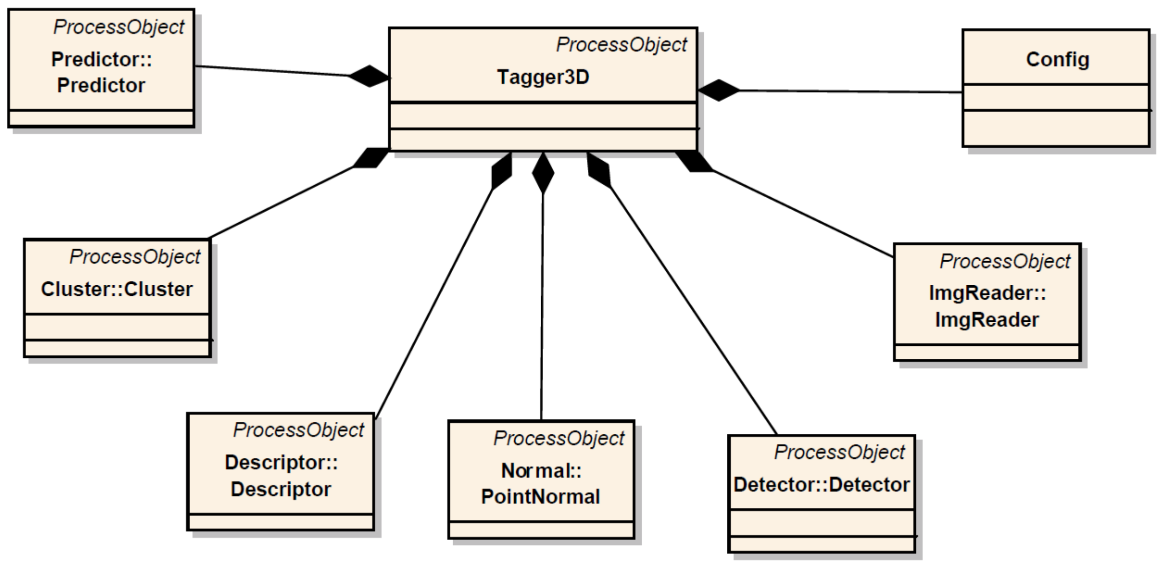

Figure 2.1: Tagger3D class diagram

Configurability

In order to achieve configurability all the configuration parameters are stored in a configuration file. A separate object has been constructed for configuration file parsing. Simply called ‘Config’, it makes use of the Boost.program_options library to parse both the configuration file and command line arguments. It returns an associative array of (parameter, value) pairs. The convention is that if an object has to be configured the configuration map has to be passed in the object’s constructor (other constructors are private).

Strategy Design Pattern

Design patterns has been introduced for the very first time by Gamma *et al* in [5]. They are best practices that should be used to solved common, recurring problems in software design. One of the patterns is the strategy pattern. It enables implementation of multiple algorithms handling some task and selection the desired behaviour at run-time. It is the exact problem encountered here. It is stated that an interface should be defined for each task and multiple classes encapsulating different algorithms for handling the tasks should realise those interfaces. One modification has been done: Algorithms have often some common operations, thus an idea of an interface (a class without any implemented methods) was dropped in favour of a virtual class.

The strategy pattern was used for every processing step of the pipeline so as to make

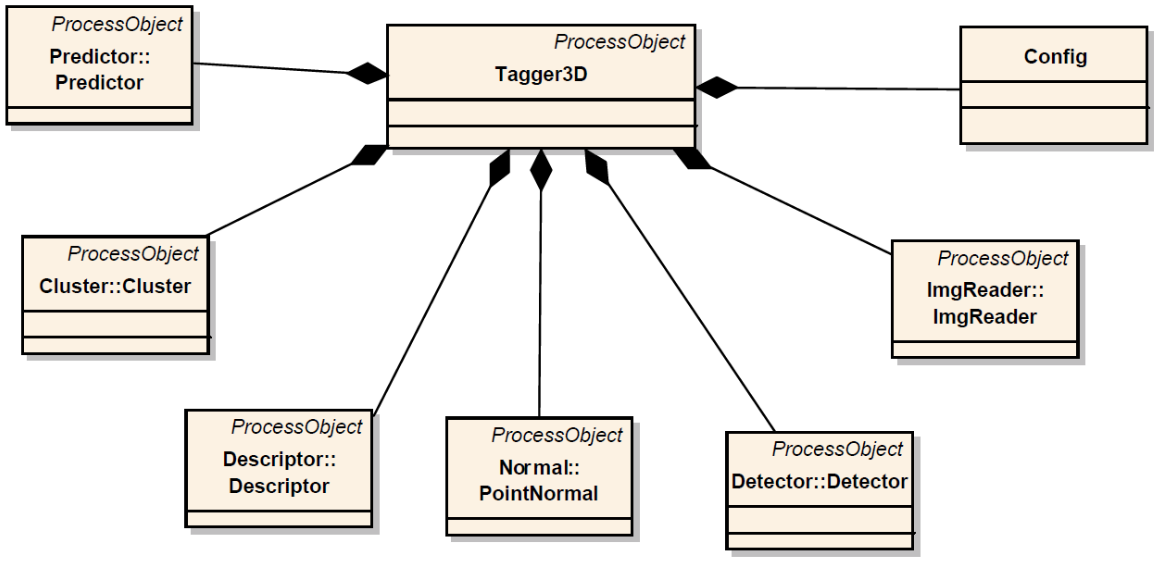

Figure 2.2: Strategy pattern

it extensible. One might want to enhance this solution by using an abstract factory design pattern, which has not been done in this case. It might make the code easier to maintain and read. However, the project is relatively small and the bulk of an abstract factory is unnecessary.

2.2 Experimental setup

Chapter 3

Experiments and results

3.1 Experiments

The Bag of Words classification pipeline consists of four previously discussed steps. For there are many different algorithms suitable for each part of the pipeline, an extensive research is required in order to discover the best possible combination in a particular setting. The number of options grows combinatorial with number of candidates for each processing step. Moreover, many algorithms have different parameters that should be adjusted for the given conditions. Evaluation of every possibility is unfeasible due to the lack of time and computational power required. Therefore it was chosen to evaluate algorithms with their default parametrs under fixed conditions. Only the winning methods qualify for the parameter tuning.

3.1.1 Detectors

ISS3D is generally better than SIFT

3.1.2 Descriptors

FPFH is faster than both PFH and PFHRGB. PFH does not give any advantages over FPFH. PFHRGB delievers considerably higher accuracy.

3.1.3 Codebook

Only kmeans tested with varying numbers of clusters i.e. vocabulary size

3.1.4 B3DO

3.1.5 tokyo

3.2 Conclusion

List of Figures

List of Tables

Bibliography

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[3] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[4] S. Filipe and L. A. Alexandre. A comparative evaluation of 3d keypoint detectors.

[5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: Abstraction and reuse of object-oriented design.* Springer, 1993.

[6] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[7] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013.

[8] X. Li and A. Godil. Investigating the bag-of-words method for 3d shape retrieval. *EURASIP Journal on Advances in Signal Processing*, 2010:5, 2010.

[9] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[11] J. Ponce, D. Forsyth, E.-p. Willow, S. Antipolis-Méditerranée, R. d'activité RAweb, L. Inria, and I. Alumni. Computer vision: a modern approach. *Computer*, 16:11, 2011.

[12] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[13] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Persistent point feature histograms for 3d point clouds. *Intelligent Autonomous Systems 10: Ias-10*, page 119, 2008.

[14] S. Salti, F. Tombari, and L. D. Stefano. A performance evaluation of 3d keypoint detectors. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 236–243. IEEE, 2011.

[15] S. M. Smith and J. M. Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[16] R. Toldo, U. Castellani, and A. Fusiello. A bag of words approach for 3d object categorization. In *Computer Vision/Computer Graphics CollaborationTechniques*, pages 116–127. Springer, 2009.

[17] C.-F. Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012, 2012.

[18] K. E. van de Sande, T. Gevers, and C. G. Snoek. Empowering visual categorization with the gpu. *Multimedia, IEEE Transactions on*, 13(1):60–70, 2011.

[19] Q. Zhang, X. Song, X. Shao, R. Shibasaki, and H. Zhao. Category modeling from just a single labeling: Use depth information to guide the learning of 2d models. In *Computer Vision and Pattern Recognition, 2013. CVPR 2013. IEEE Computer Society Conference on*. IEEE, 2013.

[20] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696. IEEE, 2009.