

Technical Report

Appearance-based Landmark Selection and Reliability Evaluation for Topological Navigation of an Aerial Vehicle

Lorenz Gerstmayr Alexandre Bernardino
José Santos-Victor*

March 4, 2004

Laboratorio de Visão[†]
Instituto de Sistemas e Robótica
Instituto Superior Técnico
Avenida Rovisco Pais
Torre Norte, Piso 7
1049-001 Lisboa
Portugal

*{lgerstmayr, alex, jasv}@isr.ist.utl.pt

[†]<http://www.isr.ist.utl.pt/vislab>

Abstract

The future aim of the work described in this report is appearance-based topological navigation of an autonomous aerial vehicle, namely a blimp, over an urban area. Here we only address the selection of good landmarks, which is one subproblem of the problems that still have to be solved in order to achieve topological navigation.

Landmark selection methods try to optimize navigation by selecting distinctive features therefore minimizing localization errors. Common landmark selection algorithms select good landmarks in two steps. In the first step they detect possible landmarks, in the second step, they discard landmarks that are not reliable. The algorithms proposed in this report only deal with discarding unreliable landmarks. The first algorithm we called “profile-based algorithm” evaluates pairwise image dissimilarity between possible landmarks and selects those landmarks that are as distinctive as possible. The second one called “IPCA-based algorithm” incrementally updates an existing eigenspace by adding that landmark that can be expressed worst. Therefore an algorithm for Incremental Principal Component Analysis (IPCA) is used. Although not addressed here, this algorithm is capable to perform on-line landmark selection and can be used for Simultaneous Localization and Mapping (SLAM).

In order to evaluate the quality of the selected landmarks, we propose a method to measure localization reliability with respect to small deviations in the orientation or the position of the blimp and small changes in image brightness. The method is based on computing a limit for the image dissimilarity for which an error-less localization is guaranteed.

An evaluation of the proposed landmark selection algorithms shows that none of the algorithms is superior to the other but that both algorithms select reliable and distinctive landmarks that are stable over a large range of deviations.

The remainder of the report is structured as follows. The first chapter will give an introduction to the theoretical things that are necessary to understand the report and to see the work in a context. Therefore we will define the concept of landmarks, point out the key aspects of localization and landmark selection algorithms and show related work in this fields. The second chapter introduces the proposed algorithms and methods and also gives an introduction to the mathematical methods that are necessary to understand our work. The third chapter shows the experiments, their results and the conclusions that can be drawn for each group of experiments. The experiments include visualizations to get familiar with the used eigenspaces, the landmark selection for both the profile-based and the IPCA-based algorithm and the reliability evaluation using the proposed method. In the next chapter some ideas and results for failed approaches

are shown. The ideas include landmark selection based on clustering and an alternative approach to reliability evaluation. The fifth and last chapter gives the overall results and conclusions and points out further working directions. The appendices give a comprehensive overview over the data that would intersect the natural flow of the document if shown in the experimental section, a skeleton for a visual path integration algorithm and an overview over the implemented MATLAB functions. Parts of this work, namely the essentials of chapters 1, 2, 3 and 5, will also be published in Gerstmayr et al. (2004).

Acknowledgments

This work is the result of a long but a very pleasant and enriching work. Enriching in technical or professional terms as well as in social terms: the main part of the work was done during a stay at the *Laboratorio de Visaão*, or in short *Vislab*, of the *Instituto de Sistemas e Robótica* at Lisbon. For me that time during the summer of 2003 is a very valuable one. Not only because I could reach a level of sun tan I never thought I could ever reach, but also because I made the great experience of living and working in a foreign country. Thanks a lot to everybody who made my stay possible.

In particular I would especially like to thank Prof. José Santos-Victor, the head of VisLab, and Alexandre Bernardino for giving me the chance for the stay, for helping me with the organization and for a lot of valuable and practical tips and feedback for my work. And of course I would like to thank my colleagues and friends Nuno Gracias, Etienne Grossman, Ricardo Oliveira, Sandra Nope Rodriguez, José Gaspar, Rodrigo Jacob, Plínio Moreno López, Manuel Lopes, Roger Castros Freitas, Filipe Metelo, Louís Campos, José Pedro Queiroz de Melo and the other “summer visitor” V. Javier Traver. Thanks a lot for making me feel comfortable, for welcoming me, for enjoying your free time with me, for assisting me... I am sure I will not forget the common experience – the all day university life as well as special events like the *Mini-Maratona* or the *Churrasco*. Thank you very much, Plínio, for organizing me the room in the house with the wonderful view over the *Tejo*, the harbor and the *Ponte 25 de Abril* to *Cristo Rei*. Thank you very much Prof. Hanspeter Mallot, head of the Department of Cognitive Neuroscience at the University of Tübingen, for supporting me, giving me the chance to finish this report and for allowing me to built up on the work I did during my stay.

Finally I would like to thank all the other people that made my stay a pleasant one: my parents for supporting me whenever and wherever, my girl-friend Nina Hillen – especially for visiting me and for the wonderful trips through the North and South of Portugal – and Ricarda Martin for unforgettable outdoor-picnics and hours of talking.

Finally I should also mention that the presented work was partially funded by the project RESCUE – Cooperative Navigation for Rescue Robots FCT SRI/32546/99-00.

Contents

List of Figures	10
List of Tables	12
List of Algorithms	13
1. Introduction	15
1.1. Aims of the project	15
1.2. Landmarks	17
1.3. Visual localization	18
1.3.1. Fundamentals	18
1.3.2. Related work	20
1.4. Landmark selection	24
1.4.1. Fundamentals	24
1.4.2. Related work	25
2. Our approach	31
2.1. Mathematical methods	31
2.1.1. Principal Component Analysis – PCA	31
2.1.2. Incremental Principal Component Analysis – IPCA	35
2.2. Clustering	37
2.2.1. Hierarchical Clustering	38
2.2.2. K-means clustering	40
2.3. Proposed landmark selection algorithms	40
2.3.1. Introductory annotations	41
2.3.2. Profile-based algorithm	41
2.3.3. IPCA-based algorithm	42
2.4. Robustness and reliability evaluation	44
3. Experiments and Results	47
3.1. Fundamentals	47
3.2. PCA-related experiments	49
3.2.1. Aim and description	49
3.2.2. Results	49
3.2.3. Conclusions	49

3.3.	Profile-based landmark selection	52
3.3.1.	Aim and description	52
3.3.2.	Results	52
3.3.3.	Conclusions	56
3.4.	IPCA-based landmark selection	57
3.4.1.	Aim and description	57
3.4.2.	Results	57
3.4.3.	Conclusions	59
3.5.	Reliability Evaluation	59
3.5.1.	Aim and description	59
3.5.2.	Results	60
3.5.3.	Conclusions	63
4.	Failed approaches	64
4.1.	Cluster-based landmark selection	64
4.1.1.	Aim and description	64
4.1.2.	Results	65
4.1.3.	Conclusions	69
4.2.	Localization matrices	71
4.2.1.	Aim and description	71
4.2.2.	Results	71
4.2.3.	Conclusions	72
5.	Final conclusions and further work	73
5.1.	Final conclusions	73
5.2.	Further work	74
5.2.1.	Color segmentation	74
5.2.2.	Line segmentation	76
5.2.3.	Preselection of landmarks	77
5.3.	Final summary	77
A.	Even more plots and data	79
A.1.	View positions	79
A.2.	Quality of Recovered images	80
A.3.	Distance Matrices	81
A.4.	Profile Matrices	87
A.5.	Landmark selection	90
A.5.1.	Results of profile based ranking	90
A.5.2.	Results of profile-based IPCA ranking	96
A.5.3.	Start lists for IPCA ranking with random start list	102
A.5.4.	Results of IPCA ranking with random startlist	103
A.6.	Reliability Evaluation	106
A.6.1.	Orientation	106
A.6.2.	Altitude	113

A.6.3. Position	120
A.6.4. Image Brightness	127
A.7. Landmark selection by hierarchical clustering	134
A.8. Landmark selection by k-means clustering	137
B. Tracking algorithm	140
C. MATLAB files for the experiments	143
Bibliography	147

List of Figures

1.1.	Practical setup for the experiments.	16
1.2.	Schematic drawing for topological navigation of an autonomous blimp.	16
2.1.	Idea of the PCA	32
2.2.	An example dendrogram	39
2.3.	Reliability evaluation	45
3.1.	Position and view number of the views	48
3.2.	Covered variance against number of dimensions of PCA space	50
3.3.	The first 25 eigenimages	51
3.4.	Example distance matrix.	52
3.5.	Example distance profiles.	53
3.6.	Example profile rankings.	55
3.7.	Example selections for the IPCA-based landmark selection	58
3.8.	Representative plots for the reliability evaluation.	61
3.9.	Extreme example for the relative size of catchment areas.	62
4.1.	Example dendrograms for clustering views	66
4.2.	scatter-plot of k-means clustering in 2D eigenspace	68
4.3.	Analysis of silhouette values	69
4.4.	Example results for the analysis of localization matrices	72
5.1.	Floormap images to visualize ideas for color segmentation	75
5.2.	Pattern formed by the streets	76
A.1.	Position of the views taken from the floormap	79
A.2.	Quality of recovered images	80
A.3.	Distance matrices 1/6	81
A.4.	Distance matrices 2/6	82
A.5.	Distance matrices 3/6	83
A.6.	Distance matrices 4/6	84
A.7.	Distance matrices 5/6	85
A.8.	Distance matrices 6/6	86
A.9.	Profile matrices 1/3	87
A.10.	Profile matrices 2/3	88
A.11.	Profile matrices 1/3	89

A.12. Visualization of the used start lists for IPCA-based ranking	102
A.13. Reliability evaluation for orientation errors 1/7	106
A.14. Reliability evaluation for orientation errors 2/7	107
A.15. Reliability evaluation for orientation errors 3/7	108
A.16. Reliability evaluation for orientation errors 4/7	109
A.17. Reliability evaluation for orientation errors 5/7	110
A.18. Reliability evaluation for orientation errors 6/7	111
A.19. Reliability evaluation for orientation errors 7/7	112
A.20. Reliability evaluation for altitude errors 1/7	113
A.21. Reliability evaluation for altitude errors 2/7	114
A.22. Reliability evaluation for altitude errors 3/7	115
A.23. Reliability evaluation for altitude errors 4/7	116
A.24. Reliability evaluation for altitude errors 5/7	117
A.25. Reliability evaluation for altitude errors 6/7	118
A.26. Reliability evaluation for altitude errors 7/7	119
A.27. Reliability evaluation for position errors 1/7	120
A.28. Reliability evaluation for position errors 2/7	121
A.29. Reliability evaluation for position errors 3/7	122
A.30. Reliability evaluation for position errors 4/7	123
A.31. Reliability evaluation for position errors 5/7	124
A.32. Reliability evaluation for position errors 6/7	125
A.33. Reliability evaluation for position errors 7/7	126
A.34. Reliability evaluation for image brightness 1/7	127
A.35. Reliability evaluation for image brightness 2/7	128
A.36. Reliability evaluation for image brightness 3/7	129
A.37. Reliability evaluation for image brightness 4/7	130
A.38. Reliability evaluation for image brightness 5/7	131
A.39. Reliability evaluation for image brightness 6/7	132
A.40. Reliability evaluation for image brightness 7/7	133
A.41. Example for landmark selection by hierarchical clustering	134
A.42. Example for landmark selection by hierarchical clustering continued . .	135
A.43. Example for landmark selection by hierarchical clustering continued . .	136
A.44. Example for landmark selection by k-means clustering	137
A.45. Example for landmark selection by k-means clustering continued	138
A.46. Example for landmark selection by k-means clustering continued	139
B.1. Regions for the tracking algorithm	140
B.2. Example tracking sequence	141
B.3. Example tracking sequence continued	142

List of Tables

1.1.	Characterization of related localization work	23
1.2.	Characterization of related landmark selection methods	29
1.3.	Characterization of related landmark selection methods continued	30
3.1.	Scale factors and resulting view sizes.	48
3.2.	Parameters for Gaussian smoothing.	48
3.3.	Differences between randomly chosen start lists.	57
4.1.	Clusters for a good dendrogram	67
4.2.	Example k-means clustering	70
A.1.	Results for profile ranking 1/6	90
A.2.	Results for profile ranking 2/6	91
A.3.	Results for profile ranking 3/6	92
A.4.	Results for profile ranking 4/6	93
A.5.	Results for profile ranking 5/6	94
A.6.	Results for profile ranking 6/6	95
A.7.	Results for IPCA-ranking 1/9	96
A.8.	Results for IPCA-ranking 2/9	97
A.9.	Results for IPCA-ranking 3/9	98
A.10.	Results for IPCA-ranking 4/9	99
A.11.	Results for IPCA-ranking 5/9	100
A.12.	Results for IPCA-ranking 6/9	101
A.13.	Results for IPCA-ranking 7/9	103
A.14.	Results for IPCA-ranking 8/9	104
A.15.	Results for IPCA-ranking 9/9	105
C.1.	Special files or functions	143
C.2.	Functions for fundamental computations	144
C.3.	Functions related to landmark selection	144
C.4.	Functions related with reliability evaluation	145
C.5.	Functions related to localization experiments	145
C.6.	Functions related with landmark clustering	146
C.7.	Auxiliary functions	146

List of Algorithms

2.1. Algorithm for agglomerative hierarchical clustering	38
2.2. Algorithm for k-means clustering	40
2.3. Algorithm for distance-profile based landmark selection	43
2.4. Algorithm for IPCA based landmark selection	44
B.1. Skeleton for visual path integration algorithm	141

1. Introduction

This chapter will describe the aim of the project. Then we will focus on important theoretical aspects that are necessary to understand the work and see it in a context of related work. This includes the concepts of landmarks and an introduction to the localization problem and landmark selection methods. Also a review of related work in the fields of localization and landmark selection is given.

1.1. Aims of the project

The future aim of the project described here is an appearance-based topological navigation strategy for an autonomous aerial vehicle flying over an urban area. The project is part of the RESCUE project, that deals with cooperative navigation for rescue robots and aims to develop robots that help humans in search and rescue missions in disaster areas like areas destroyed by earthquakes, floods or terrorist attacks. For a further overview over the aims and the first results of the project see the project homepage¹ or Lima et al. (2003) and Bernardino et al. (2003). One subproject is to design an autonomous blimp flying over the scenario. In Lima et al. (2003) is mentioned that the advantage of aerial robots is that they can provide a broad view from a bird's position and that they can therefore map areas of high destruction and guide human and machine rescue troops to these areas or that they can supply the operation controllers with very detailed information. In order to solve all these tasks, the used vehicle needs very robust and good navigation and localization abilities.

Since dealing with a complete catastrophic scenario, which is highly dynamic, is still too difficult to handle because of the moving rescue workers and the excavation of the debris, we only deal with navigation over an urban area. For easier testing of the techniques a practical setup was developed at VisLab: An indoor blimp (approximately with 0.8 m in diameter and 2 m in length) is flying over a huge poster of an aerial image of the neighborhood around *Instituto Superior Técnico* in Lisbon, Portugal. The poster is shown in figure 1.1(a). The blimp shown in figure 1.1(b) is equipped with a camera looking downwards to the poster. All the processing and control of the blimp is done at a host computer, the data is transmitted via a radio link. Beside easier testing the advantage of the setup is that daylight changes or weather conditions like rain or wind can be excluded.

For topological navigation the agent has to memorize several places and the links between these places in a graph-like structure, the topological map (Franz and Mallot,

¹<http://rescue.isr.ist.utl.pt>

2000). A schematic drawing visualizing the plans is shown in figure 1.2. Therefore one key aspect on the way to implement the navigation strategy is to find good landmarks that allow robust localization. In this report we address exactly that problem. The navigation from landmark to landmark is left for further work. Probably some kind of visual odometry like in Iida (2003) or based on the ideas described in appendix B will be used for roughly approximating the goal.



(a) The used aerial image

(b) The blimp flying over the poster (will be used in future work)

Figure 1.1.: Practical setup for the experiments.

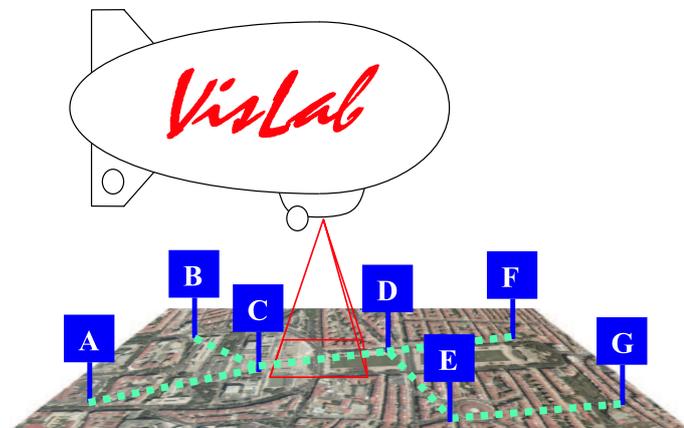


Figure 1.2.: Schematic drawing for topological navigation of an autonomous blimp.

In this work, we focus on the selection of good landmarks that make the localization more robust which is very important for achieving good navigation. All the proposed methods assume that a map of the environment is already given. It does not matter how it was obtained, aerial images or by mosaicking are just two example ways. As usually in the field of appearance-based navigation we use windows around image features as landmarks.

Landmark selection algorithms usually select the best landmarks from a set of possible landmarks. For our approach to the problem, the landmarks are transformed to an eigenspace. The first algorithm we called “profile-based algorithm” computes pairwise image dissimilarities of the landmarks in the eigenspace and selects landmarks that have the greatest average image dissimilarity to all other possible landmarks. The second algorithm called “IPCA-based algorithm” updates an existing eigenspace by iteratively adding exactly that image that can be expressed worst in the existing eigenspace. This grants that the added image that is selected as landmark, is as dissimilar as possible to all the other landmarks already selected. Because of using the IPCA the algorithm is capable of online landmark selection and can be used for Simultaneous Localization and Mapping (SLAM).

So far, only little work was done in measuring the quality of the selected landmarks. If at all, related works show experimental results but no quantitative measures for the robustness. For localization the robot has to compare its current camera view to the stored landmarks. Little deviations in the pose or position of the robot lead to current images that are different to the stored views of the selected landmarks resulting in a growing image dissimilarity between the views. A good landmark is very robust to these deviations. To measure the landmark’s quality we propose a method that is based on image distances in the eigenspace and that computes a maximal image dissimilarity for which an error-less localization is granted.

In order to show the context of the work it is necessary to deal with the most important theoretical aspects underlying the presented work. This includes the concept of landmarks and an introduction to localization and landmark selection.

1.2. Landmarks

Thrun (1998) states that the concept of landmarks is easy to understand, but very difficult to define. The definitions vary and are often application specific and often the term is used without being defined clearly. To allow an overview over the huge range of possible definitions, some definitions found in the reviewed literature will be cited here:

Leonard and Durrant-Whyte (1991): “A landmark is a naturally occurring environment feature that can be reliably observed in successive sensor measurements and can accurately be described in terms of concise geometric parameterization.”

Deng et al. (1996): “A landmark is a localized physical feature that the robot can sense and use for the estimation of its own position relative to a map that contains the landmarks absolute position.”

Little et al. (1998): “Landmarks are distinctive locations with identifiable appearances.”

Thrun (1998): “A landmarks is any element (object or feature) which can serve as reference.”

Mata et al. (2001): “A landmark is a localized physical feature that the robot can sense and use to estimate its own position in relation to the map that contains the landmark’s relative position and/or other mark characterization”.

From all cited definitions, the one given in Thrun (1998) is the most abstract one and it can also serve as a superordinate concept of the definition used for the proposed approach to landmark selection. In this work, a landmark is an image that is as dissimilar as possible to all the considered images. The works of Knappek et al. (2000) and Ohba and Ikeuchi (1997) use the same definition.

1.3. Visual localization

Whenever the blimp approaches a landmark it has to localize itself in order to determine if the landmark is reached. The localization problem, i.e. the problem to determine and track a robot’s position relative to its environment (Little et al., 1998; Thrun, 1998), is often addressed in the recent research because an effective localization is a key element for a successful navigation. At the first glance, the problem looks simple and Little et al. (1998) mention that it would not be any problem, if perfect odometry and the initial robot position would be available. Methods based on computer vision are often used to overcome the inaccuracy of odometry. All vision-based methods have in common that they somehow have to characterize the known places. Therefore, they all use landmarks.

Before characterizing some algorithms proposed in related work, it is necessary to discuss some fundamentals or possible characterizations of visual localization methods.

1.3.1. Fundamentals

Most of the approaches use single cameras, panoramic vision systems or range data obtained by laser range scanners or stereo vision systems as input. In the following paragraphs, the terms in which the methods can be characterized will be explained in detail.

Topological vs. geometrical localization

Referring to Ulrich and Nourbakhsh (2000) the approaches to visual localization can be divided into three groups: *geometric*, *topological* and *hybrid* approaches. In Gaspar et al. (2000) is mentioned that the method which performs best is often problem dependent and a trade-off between accuracy for short-range navigation and robustness for large-scale navigation has to be found. The key characteristics of the approaches are

Geometric: The geometrical methods have in common that they use a 2D grid or a 3D-model of the environment as map representing the environment. The robot’s position is tracked or determined exactly with respect to

the map's coordinate system. The main advantage of geometrical approaches is that they are more exact (Gaspar et al., 2000), but the methods are more error-prone, because small deviations in odometry or perception can lead to great localization errors (Mata et al., 2001).

Topological: The topological methods use an adjacency graph as representation of the environment. They only determine the node, which is closest to the robot's actual position but not the exact position. The advantages of topological navigation are that it is more stable but needs a powerful perception system to identify the elements of the environment (Mata et al., 2001; Gaspar et al., 2000; Ulrich and Nourbakhsh, 2000).

Hybrid: Hybrid methods try to combine geometric and topological approaches.

Representation of the environment

All the localization methods described in the previous paragraph need a map-like representation of the environment, in which the known places or landmarks are stored and which also has to be chosen application dependent. The most common approaches to represent the environment are:

Feature maps: In feature maps the global locations of a set of features and the description of these features are stored (Bailey and Nebot, 2001). Localization is performed by searching the closest match between stored features and visible features.

Occupancy grids: Occupancy grids are a matrix-like representation of the environment and the probability that the area corresponding to the matrix entry is occupied is stored in the matrix (Bailey and Nebot, 2001).

Topological maps: Topological maps are graph-like representations of the environment. The edges contain information for traveling from location to location, the nodes store the representation of the location (Bailey and Nebot, 2001). Vale and Ribeiro (2003) mention that topological maps do not provide metric informations but are useful and scalable abstractions of an environment.

Surface models: Surface models are a 3D-representation obtained by range data.

Iconic maps: Iconic maps are large images of the robot's known environment.

Matching

For localization the observation taken at the current robot position has to be somehow matched with the stored information about the environment to localize the robot. There are two matching strategies, namely local and global search, which search only

a part of the search space or the whole space respectively. Mainly, the following matching strategies are used:

Feature matching: Correspondences between the stored and the observed features are established and used for localization.

Image matching: The image dissimilarity between the current image and parts of the stored representation are computed and used to determine the robot's position. To speed up the computations needed for image matching, the images are – like in our work – often transformed to lower dimensional representations (e.g. using Principal Component Analysis, see section 2.1.1). These methods are referred to as appearance based localization methods.

Model matching: For model matching, the observed environmental model is matched with the stored model by computing similarity and using the similarity measure for localization.

1.3.2. Related work

The following paragraph is supposed to allow a rough overview over proposed visual localization methods. How these methods can be characterized in relation to the characteristics described above is shown in table 1.1. Since our approach is too far away from most of the reviewed works we do not comment on the advantages or disadvantages of the proposed work but just give a short review trying to characterize the work according to the possible methods introduced above.

Krotkov (1989): The algorithm detects edges in the input image and tries to match these edges with the landmarks marked in an a priori known feature map.

Leonard and Durrant-Whyte (1991): The proposed algorithm matches observed features with known features in an a priori built feature map. For an estimation of the search position in the feature map, an Extended Kalman Filter (EKF) is used.

Andersen et al. (1997): The algorithm builds a topological map from wide-angle images of the environment. While exploring images are added, when the dissimilarity between the images reaches a certain threshold, and the commands to step from one place to the next are stored. For navigating, the images at the robot's position are compared to the stored images around the robot's estimated position.

Betke and Gurvits (1997): The authors use an one-dimensional strip taken from an panoramic image as sensor input. They try to match landmarks, which are dark strips in the used sensor data and try to match them in an a priori known map of landmark positions. The exact robot position is determined by triangulation.

- Olson (1997): The method first generates a range map of the observed environment and then computes an occupancy map of the terrain surface. For localization, this surface model is matched against the stored model.
- de Verdiere and Crowley (1998): The authors propose an algorithm for topological localization or object recognition. The input image is divided into a grid of sub-images. For all the sub-images of all the training images, an eigenspace is computed. Then for each sample image all the sub-images are transformed to the lower dimensional subspace. The image is now represented as a surface. For recognition, the resulting surface of the input image is compared to all the surfaces of the sample images.
- Dudek and Jugessur (2000): The localization method uses attention operators to determine interesting points in the training images. Windows around these interesting points are taken and transformed to a rotation invariant image space. For these training images an eigenspace is computed. For recognition, the interesting points of the observation are detected and the resulting projections to the eigenspace are compared to the eigenimages. A voting algorithm is used for the final place recognition.
- Gaspar et al. (2000): The algorithm is based on building a topological map out of the eigenimages of the computed eigenspace. These eigenimages are used for localization. For navigation between the known places, a feature tracking method is used.
- Kelly (2000): The author proposes a navigation method for large scale environments with locally flat scenes. He constructs a large image of the known area and subsequently tracks the robots position using a visual tracking algorithm.
- Ulrich and Nourbakhsh (2000): This method uses several histograms to characterize images. A topological map is built by capturing representative images and associating the corresponding location. Locations are characterized by histograms for several color channels and color models. During navigation the currently visible image characterization is compared to the stored ones. For each color channel the location is determined. The final localization is done by voting between the several channels.
- Bailey and Nebot (2001): The authors propose a topologically feature map to combine the better tracking capabilities of feature maps with the better scalability of topological maps. Each node of the graph stores a feature map and for localization the robot position is determined according to the feature map's reference by tracking the robot's position. Places are characterized by the feature arrangement.
- Artac et al. (2002b): The authors use Incremental Principal Component Analysis (IPCA, see section 2.1.2) to built an eigenspace of images. The eigenimages are used

as landmarks and between these positions in eigenspace were interpolated to achieve a grid-like representation. For localization, the current view is transformed to the eigenspace and a nearest neighbor search is performed.

Shaw and Barnes (2002): The proposed algorithm uses a topographical map of the environment taken from an aerial vehicle. There typical surface features are detected and stored in a feature map of the environment. A feature map of the current sensor information is computed and matched with the map of the environment to determine the robot's position.

Freitas et al. (2003): This appearance-based approach fuses localization and exploration using IPCA. While exploring the robot adds landmarks, in this case eigenimages, to the topological map. The focus is not on navigational tasks but on a performance analysis of the method.

Table 1.1.: Characterization of related localization work

Authors	Image data	Localization	Map	Search	Matching	Place characterization
Krotkov (1989)	Single	Geometrical	Feature	Local	Features	Feature configuration
Leonard and Durrant-Whyte (1991)	Range	Geometrical	Feature	Local	Features	Feature configuration
Andersen et al. (1997)	Single / Stereo	Global	Topological	Local	Image	Distinct images
Betke and Gurvits (1997)	Panoramic (1D)	Geometrical	Feature	Global	Features	Landmark configuration
Olson (1997)	Stereo	Geometric	Surface	Global	Image	Specific sub-image
de Verdiere and Crowley (1998)	Single image	Topological	Feature	Global	Surface matching	Surface in eigenspace
Dudek and Juges-sur (2000)	Single	Geometric	Feature	Global	Features and voting	Configuration of interesting points
Gaspar et al. (2000)	Panoramic	Topological	Topological	Global	Image, appearance based	Eigenimages
Kelly (2000)	Single	Geometrical	Image	Local	Images	Images
Ulrich and Nourbakhsh (2000)	Panoramic	Topological	Feature	Global	Features and voting	Histograms of distinct images
Bailey and Nebot (2001)	Range	Hybrid	Topological feature map	Local	Features	Feature configuration
Artac et al. (2002b)	Panoramic	Geometric	Topological	Global	Image, appearance based	Eigenimages
Shaw and Barnes (2002)	Range	Geometrical	Feature	Global	Features	Surface features
Freitas et al. (2003)	Panoramic	Topological	Topological	Global	Image, appearance based	Eigenimages

1.4. Landmark selection

The work described in the previous section only uses landmarks to characterize places without taking into account, which landmarks allow good or robust localization or assumes that landmarks have already been selected. In the last couple of years some papers were published, that try to select landmarks to achieve better results and therefore are related to the algorithms presented here.

In the next paragraph some basics of landmark selection shall be pointed out. Then related approaches are reviewed.

1.4.1. Fundamentals

Thrun (1998) mentions three disadvantages if a robot can not perform landmarks selection and is fixed to a certain type of landmark.

Lack of Flexibility: Since a given feature is only useful if it exists in the robot's environment, the localization or navigation methods will fail if the robot has to navigate in a different environment.

Lack of optimality: Even for features that are generally applicable, it is not obvious what the best landmark would be.

Lack of autonomy: If a certain environment feature is selected as landmark this feature is not necessarily the best landmark since the sensors of humans and robots are too different. Therefore features that appear as good landmarks for humans might not be good landmarks for robots.

Landmark selection methods overcome these drawbacks at least partially. All landmark selection methods have in common that they try to minimize the localization uncertainty. Some approaches like for example Thrun (1998) or Olson (2002) use a probabilistic approach. Others like Sutherland and Thompson (1994) or Burschka et al. (2003) try to use landmarks as references for which it is known that they lead to small localization errors.

The approaches that are more related to our algorithms select the landmarks in two steps. In the first step, possible landmarks are detected. For this step most often attention operators are used. Attention operators detect low-level image features like edges, edge-density, edge orientation, corners or symmetry. For a review of attention operators the interested reader is referred to Sim et al. (2003) or Schmid et al. (2000). In a second step, the algorithms then discard landmarks that are neither reliable nor unique. Which landmarks are discarded and how the bad landmarks are detected is highly application or algorithm dependent and therefore cannot be introduced here.

Due to that two-step pattern that is shared by many landmark selection algorithms, one can state the following properties of good landmarks. The properties are proposed by Ohba and Ikeuchi (1997) for selecting image regions that are best chosen for recognizing an object and can also be applied to landmarks. The three properties are:

- Detectability: The landmark has to be detectable within the image. Usually the first step of the algorithms selects landmarks that are detectable by applying attention operators.
- Uniqueness: Since the detectability criterion only detects landmarks but does not grant the global uniqueness of a landmark a further property is needed. In order to select discriminative landmarks already detected landmarks that are too similar are excluded.
- Reliability: The last criterion selects landmarks that are unique but don not lead to robust localization. It ensures that the selected landmarks are robust against little errors in the robot's position or pose. Discarding landmarks that are neither unique nor reliable are usually discarded in the second step of the landmark selection methods.

1.4.2. Related work

After introducing the basic things about landmark selection the related work will be reviewed. The focus will be on the selection criterion and not on the methods how localization is performed, how the images are represented or on the advantages and drawbacks of the methods. The following paragraph is not restricted to landmark selection for visual navigation, because similar ideas are used in visual recognition and image retrieval too and the number of published papers is much smaller than for visual localization.

Sutherland and Thompson (1994): The authors analyze how measurement errors lead to localization errors. They propose a simple algorithm that uses geometric properties of a set of given landmarks to select an optimal subset. The subset minimizes the localization uncertainty. The algorithm also can propose an optimal area for detecting new landmarks.

Yeh (1995): The proposed algorithm selects these features from a list of given 3D features that will be most likely recognized from single image data and that are as distinctive as possible. As byproduct, the algorithm builds a topological map.

Murphy et al. (1997): The paper presents a homing algorithm that selects the landmarks in order to minimize the homing error. Heuristics are used to rank possible landmark triples. The best triple is then evaluated by applying several test cases. If the tests could not be completed satisfactory, another triple is taken.

Ohba and Ikeuchi (1997): In this paper an eigenwindow method is proposed that allows stable object recognition. At several interesting points, subimages are taken and an eigenspace is built for all subwindows of the training set. For recognition each subwindow is recognized and a voting algorithm is used. For selecting interesting points, three measurements, namely detectability, uniqueness and

reliability are used. Therefore, an attention operator detecting corners, a comparison of the pairwise uniqueness of all views and a measure for recognition quality under rotations are used respectively. This work is very related to the work presented here and the criterion for discarding influenced our approach for measuring a landmark's quality.

Schmid and Mohr (1997): In this paper an algorithm to characterize images for image retrieval is proposed. It is based on local gray value invariants which are computed for corner points detected with an attention operator. For retrieval a voting algorithm is used.

Takeuchi et al. (1997): In the paper a learning-algorithm for landmark recognition is presented. It uses a training set of images to group the images corresponding to the same distinctive landmark. Each landmark is described by a set of features and for recognition the current feature set is compared to the ones stored in the database. The algorithm can be used for topological robot localization as well as for image categorization.

Bourque et al. (1998): The authors propose a method for image acquisition with a mobile robot for building an environmental representation based on images. Images are only taken at representative places. The problem of detecting these views is solved by an attention operator evaluation of the edge structure.

Little et al. (1998): The authors use range data obtained from a trinocular camera system to build a 2D occupancy grid of the environment for navigation and obstacle avoidance. The occupancy grid is combined with sparse 3D landmarks that allow a better localization of the robot. For landmark detection a corner detector is used and afterwards corners are excluded where the local surface is not planar. This approach keeps corners that are visible from many directions and were not formed by the projection of areas separated in space.

Sim and Dudek (1998): The proposed algorithm detects landmarks by using an attention operator which detects image regions of high edge density. Then for subwindows an eigenspace is computed to get low-dimensional descriptions of the observed landmarks. For position estimation, the arrangement of interesting areas in the image is evaluated. For localization the detected landmarks are tracked over several image frames.

Simhon and Dudek (1998): The authors propose a method for building a collection of local coordinate frames to cover a large scale environment. Therefore they analyze the environment to find areas of high distinctiveness, i.e. areas where good sensor data is provided and sufficient spacial structure is available. The authors show examples for orthogonal lines detected with an ultrasonic range sensor and smooth areas with varying features detected by a vision system.

- Thrun (1998): The author proposes a learning algorithm that is very abstract but very powerful because it allows the autonomous selection of optimal landmarks and also provides environmental and sensor flexibility. The idea is to optimize the localization uncertainty with a Bayesian approach. The proposed algorithm enables the robot to learn which features or landmarks are best used for localization and provides the association between sensor readings and robot position. The author also proposes an artificial neural network to extract features from sensor readings.
- Bourque and Dudek (2000): The paper describes an approach to characterizing an environment with views that are as interesting as possible. To measure the interest of an image, they perform image statistics on the edge density and orientation of an image. They provide a way for online-selection of images that are unusual for the environment and therefore are of maximal interest.
- Johnson (2000): The proposed algorithm computes feature vectors to characterize surface points by combining pose dependent oriented surface points and pose invariant surface structures. Afterwards feature vectors that would decrease localization quality are excluded by checking for each feature vector, if it is very similar to others or if it is in an area, where small measurement errors would result in great localization errors. Therefore vectors in plains or respectively in areas with great slope are excluded.
- Jugessur and Dudek (2000): The approach is based on using an attention operator detecting corners to obtain good visual features. In a second step all these features are discarded, for which the standard deviation of a neighborhood around the feature is below a thresholds. The selected features serve as cues to obtain recognition. For all selected features of the training set, an image eigenspace is computed using PCA in the frequency domain. For recognition the features of the image to be recognized are selected and each feature is recognized for its own. The object is recognized using a consensus of all the feature recognitions. The approach allows robust localization even for rotated, scaled, partially occluded objects and for variable backgrounds.
- Knapek et al. (2000): In this paper a landmark selection algorithm is used to detect image features that are distinctive and salient. First salient features are detected with an attention operator. From this preselected set the most distinctive features are computed by comparing the similarity between features and selecting the most dissimilar ones. In the paper experimental proofs are provided to show that the algorithm is stable in selecting and recognizing landmarks. This work is closely related to the profile based landmark selection described in section 2.3.2. The algorithm computes a distance matrix for feature vectors.
- Se et al. (2001): The algorithm uses trinocular stereo for ego-motion tracking of a robot. For each of the three images, features that are invariant to image translation, scaling and rotation are computed. Correspondences are established

between the images, and the features are tracked between frames. For 3D map building, the coordinates of the points are computed.

Olson (2002): In this work a method for selecting optimally image areas for terrain matching by selecting good landmarks is proposed. Therefore a sensor error model is used for estimation of the probability distribution of the environment as seen from the current robot position. The estimated distribution is compared to a known map of the terrain and the optimal landmark is selected by minimizing the predicted uncertainty for the localization.

Burschka et al. (2003): The authors propose a method to optimally place or select landmarks for tracking to optimize vision based control of a mobile robot. For a setup using a single camera or a panoramic vision system, they show theoretically and experimentally that trying to minimize the distance between robot and tracked landmarks leads to better results. In future work, the authors want to apply the method for automatically selecting good landmarks for visual path following.

Sim et al. (2003): In the paper several attention operators for landmark selection and the resulting localization errors are compared.

The following tables are supposed to give an overview over the key aspects of the reviewed works.

Table 1.2.: Characterization of related landmark selection methods

Citation	Aim	Input	Landmarks	Landmark Selection	Localization, Recognition
Greiner and Isukapalli (1994)	Learning landmark selection	Panoramic image (1D)	Features	Experience-based learning algorithm	Triangulation
Sutherland and Thompson (1994)	Geometric localization	Single image	Features	Minimizing expected error	Triangulation
Yeh (1995)	Landmark selection	Single image	Features	Measure feature robustness	Future work
Murphy et al. (1997)	Homing	Single image / Laser data	Features	Optimize feature arrangement	Triangulation
Ohba and Ikeuchi (1997)	Recognition	Single image	Subwindows	Corner detection, uniqueness and reliability measures	Recognizing features, voting
Schmid and Mohr (1997)	Image retrieval	Single image	Features	Corner detection	Feature matching
Takeuchi et al. (1997)	Single image	Topological localization	Feature distribution	Distinctiveness of feature distribution	Comparing feature sets
Bourque et al. (1998)	Building environmental description	Single image	–	Attention operator evaluating edge structure	–
Little et al. (1998)	Geometric localization	Trinocular stereo	Features	Corner detection, excluding of corners not located on planar surfaces	Future work
Sim and Dudek (1998)	Geometric localization	Single Image	Subwindows	Detecting areas of high edge density	Recognizing interesting features, tracking, evaluation of the landmark arrangement

Table 1.3.: Characterization of related landmark selection methods continued

Citation	Aim	Input	Landmarks	Landmark Selection	Localization, Recognition
Simhon and Dudek (1998)	Map building for hybrid localization	Single Image / Ultrasonic Range data	Areas with high distinctiveness	Detecting areas of high distinctiveness	Map matching
Thrun (1998)	Geometric localization	Variable, single color camera, ultrasonic range sensors	Features extracted from sensor reading	Minimizing uncertainty	Bayesian / Probabilistic
Bourque and Dudek (2000)	Topological navigation (future work)	Panoramic images	Landmarks: Images of maximum interest	Image statistic of edge density and orientation	Future work
Johnson (2000)	Geometric localization	Range data	Feature vectors	Excluding feature vectors	Matching feature vectors
Jugessur and Dudek (2000)	Object Recognition	Single Image	Subwindows	Corner detection, gray value variance	Recognizing features, voting
Knapek et al. (2000)	Selection of recognizable landmarks	Single image	Image features	Corner detection and measure of distinctiveness	Feature matching
Se et al. (2001)	Geometric localization	Trinocular Stereo	Scale-invariant features	Detection of swift features	Feature tracking
Olson (2002)	Geometric localization	Stereo image	Features in surface map	Minimizing localization uncertainty	Terrain matching
Burschka et al. (2003)	Visual path following (Future work)	Single image / Panoramic Image	Selectable image features	Features a priori known	Future work
Sim et al. (2003)	Pose estimation	Single image	Features	Several attention operators	Feature matching

2. Our approach

In this chapter we give a detailed introduction to the profile-based and the IPCA-based landmark selection algorithms and the reliability measurement we propose in this report. Before describing the algorithms an introduction to the necessary mathematical background, namely to Principal Component Analysis (PCA), Incremental Principal Component Analysis (IPCA) and clustering is given for readers not familiar with these methods.

We tried to keep the same mathematical notation like in Murase and Nayar (1995) and in Freitas et al. (2003). Vectors are set in boldface lower case letters, matrices in capital boldface letters, sets in capital letters.

2.1. Mathematical methods

2.1.1. Principal Component Analysis – PCA

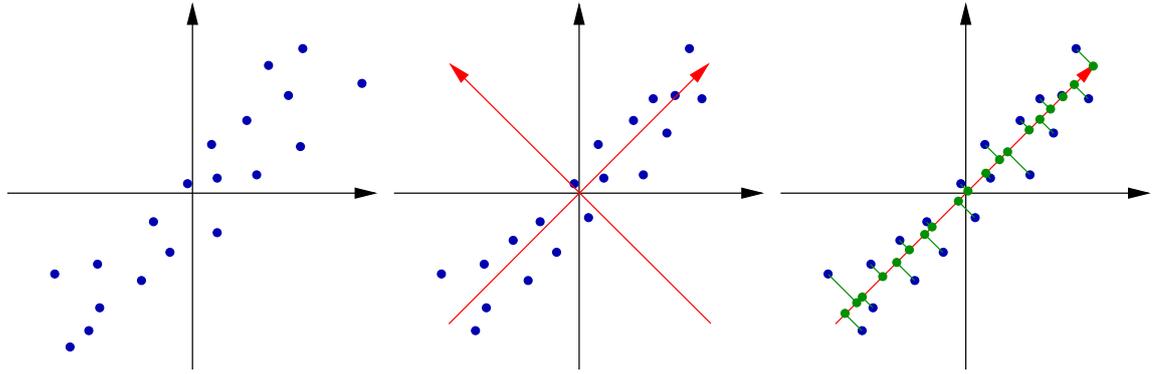
Principal Component Analysis (PCA) got an established method for finding patterns or reducing dimensionality in high dimensional datasets. Besides statistics, data mining and signal processing it is widely used in computer vision for face (Turk and Pentland, 1991) and object recognition (Murase and Nayar, 1995; Leonardis and Bischof, 2000) or in the field of robot vision for localization (Jogan and Leonardis, 2000) and robot navigation (Gaspar et al., 2000; Winters and Santos-Victor, 2002; Vasallo et al., 2002).

The following paragraphs are supposed to give an introduction to the mathematics behind the PCA and are based on Murase and Nayar (1995). The idea behind the PCA for a set of training images is to compute an orthogonal basis of eigenvectors for the training set, so that the origin of the new coordinate system equals the average image and the first axis points in the direction of the greatest variation in the data set, the second axis points in the direction orthogonal to the first covering the second greatest variation and so on. Figure 2.1 points out the idea of the PCA.

Although for an exact reconstruction often a large set of eigenvectors is required, only few eigenvectors or dimensions are sufficient to capture significant image characteristics. Especially for this work, where many computations of image similarity are needed, the reduction of the dimensionality results in faster computations.

Building the basis of eigenvectors

The images \mathbf{X}_i ; $i = 1 \dots n$ that are going to build the training set for the PCA, are reshaped to column vectors $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$ where m is the number of pixels in each image.



(a) The zero-mean data set with the original coordinate axis

(b) The data set with rotated axis, so that the first axis covers the greatest variance and the second axis is perpendicular covering the second greatest variance.

(c) The data set with reduced dimensionality. The points are projected to the first axis

Figure 2.1.: Idea of the PCA

In the next step the average image vector

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_i \bar{\mathbf{x}}_i \quad (2.1)$$

is computed and subtracted from every image \mathbf{x}_i

$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}. \quad (2.2)$$

These zero-mean vectors $\hat{\mathbf{x}}_i$ are put together to a $m \times n$ matrix

$$\mathbf{A} := [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_n]. \quad (2.3)$$

Then the $m \times m$ covariance matrix

$$\mathbf{C} := \mathbf{A}\mathbf{A}^\top \quad (2.4)$$

is computed. Solving the Eigenvalue Decomposition (EVD) for \mathbf{C} leads to the eigenvalues λ_i and the associated eigenvectors $\mathbf{e}_i \in \mathbb{R}^{m \times 1}$ with $i = 1 \dots m$. The eigenvectors belonging to the non-zero eigenvalues span a basis with at most $\kappa := \min(m, n)$ vectors, because the rank of \mathbf{C} can be smaller than κ and the matrix has at most κ non-zero eigenvalues.

Since \mathbf{C} is actually a matrix of the size $m \times m$ and therefore is very huge, the computation of the EVD is very expensive. In Murase and Nayar (1995) are several algorithms mentioned to overcome the problem. Here we focus on the algorithm used for this work which is based on Single Value Decomposition (SVD). If the number of images n is much smaller than the number of pixels m it is better to compute the implicit covariance matrix

$$\tilde{\mathbf{C}} := \mathbf{A}^\top \mathbf{A} \quad (2.5)$$

which is of the size $n \times n$. Then the eigenvectors $\tilde{\mathbf{e}}_i$; $i = 1 \dots n$ and eigenvalues $\tilde{\lambda}_i$ of $\tilde{\mathbf{C}}$ are computed by SVD of the matrix $\tilde{\mathbf{C}}$. The eigenvectors and eigenvalues of \mathbf{C} can be computed using

$$\lambda_i = \tilde{\lambda}_i \quad (2.6)$$

and

$$\mathbf{e}_i = \tilde{\lambda}_i^{-\frac{1}{2}} \mathbf{A} \tilde{\mathbf{e}}_i \quad (2.7)$$

For both ways the result is a set of eigenvalues λ_i where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_\kappa$ and a set of associated eigenvectors \mathbf{e}_i ; $i = 1 \dots \kappa$. Since \mathbf{C} and $\tilde{\mathbf{C}}$ are both positive definite, every eigenvalue λ_i is positive and is a measure for the variance or variation in the data set covered by \mathbf{e}_i . Up to that point, the PCA is nothing else than a rotation of the existing basis so that the first axis covers the greatest variation, the second axis covers the second greatest variation and so on. The axis of the new coordinate system are the eigenvectors.

Transformation to the eigenspace

Since in this thesis the number of pixels m in the images is always greater than the number of images n , only this case will be mentioned from now on. To project the images into the eigenspace a dimension k has to be chosen with $1 \leq k \leq n$. The more dimensions are chosen, the more accurate is the representation of the images in the eigenspace. A way to measure the covered variance is to compute the following ratio

$$t := \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i}, \quad (2.8)$$

which should be close but less than unity. For $k = n$ each image of the training set can be represented exactly by a vector of the length k . In this case the PCA only performs an image compression.

Then the transformation matrix

$$\mathbf{T} := [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k], \quad (2.9)$$

where the columns are the first k eigenvectors has to be built. At this step an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{T}, \mathbf{\Lambda}, n)$ has been computed and an image \mathbf{y} can be transformed to the eigenspace by using

$$\mathbf{g} := \mathbf{T}^\top (\mathbf{y} - \bar{\mathbf{x}}). \quad (2.10)$$

Recovering images

The transformation of a point \mathbf{g} in the eigenspace to an image can be expressed as a linear combination of all n eigenvectors weighted with the coordinates \mathbf{g}_i of the point:

$$\mathbf{y} = \sum_{i=1}^n \mathbf{g}_i \mathbf{e}_i + \bar{\mathbf{x}} \quad (2.11)$$

$$= \mathbf{T} \mathbf{g} + \bar{\mathbf{x}}. \quad (2.12)$$

In case of a reduced dimensionality the image can only be recovered with a certain residue vector \mathbf{r} :

$$\mathbf{y} = \sum_{i=1}^k \mathbf{g}_i \mathbf{e}_i + \bar{\mathbf{x}} + \mathbf{r}. \quad (2.13)$$

Image Distance in the eigenspace

For the computations described in this report, computing the similarity between images using the Sum of Squared Differences (SSD) or some normalized equivalent is very important. Therefore the proof given in Murase and Nayar (1995) will be recapitulated showing that the SSD of two images \mathbf{y}_1 , \mathbf{y}_2 in the image space is approximately the SSD of the corresponding points \mathbf{g}_1 , \mathbf{g}_2 in the eigenspace:

$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \approx \left\| \sum_{i=1}^k \mathbf{g}_1 \mathbf{e}_i - \sum_{i=1}^k \mathbf{g}_2 \mathbf{e}_i \right\|^2 \quad (2.14)$$

The right hand side can be simplified

$$\left\| \sum_{i=1}^k \mathbf{g}_1 \mathbf{e}_i - \sum_{i=1}^k \mathbf{g}_2 \mathbf{e}_i \right\|^2 = \left\| \sum_{i=1}^k (\mathbf{g}_{i_1} - \mathbf{g}_{i_2}) \mathbf{e}_i \right\|^2 \quad (2.15)$$

$$= \sum_{i=1}^k \sum_{j=1}^k \mathbf{e}_i^\top \mathbf{e}_j \times (\mathbf{g}_{i_1} - \mathbf{g}_{i_2})^2. \quad (2.16)$$

Using that the eigenvectors are orthogonal and therefore $\mathbf{e}_i^\top \mathbf{e}_j = 1$ for $i = j$ and 0 otherwise, we obtain

$$\left\| \sum_{i=1}^k \mathbf{g}_1 \mathbf{e}_i - \sum_{i=1}^k \mathbf{g}_2 \mathbf{e}_i \right\|^2 = \|\mathbf{g}_1 - \mathbf{g}_2\|^2. \quad (2.17)$$

And finally

$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \approx \|\mathbf{g}_1 - \mathbf{g}_2\|^2. \quad (2.18)$$

This equation guarantees that similar images will be transformed to similar points in the eigenspace. Because of this property of the PCA it is possible to compute image similarities in the eigenspace resulting in much faster computations.

Annotations

Before the Incremental Principal Component Analysis will be explained, we want to point out some properties of the PCA without proofing them.

- The columns of the transformation matrix \mathbf{T} , i.e. the eigenvectors, are also referred to as eigenimages.
- After Hall et al. (1998) the set of points can be understood as k -dimensional hyper-ellipse in the n -dimensional space. The center of the ellipse is the mean image $\bar{\mathbf{x}}$, the axis are the first k eigenvectors and the length of the axis is the square root of the associated eigenvalues.
- de Verdiere and Crowley (1998) mention that the reduction of dimensionality leads to a k -dimensional approximation of the dataset that is optimal in the least-square sense. So the first principal component can be understood as regression line minimizing the perpendicular distance to the data set.
- In Rupar et al. (2002) the authors state that spatial relation between pixels is not exploited because the image is reshaped to a vector. To overcome this drawback, the authors use the tensor rank method.
- Several authors, e.g. Ohba and Ikeuchi (1997), Dudek and Jugessur (2000), de Verdiere and Crowley (1998) and Jugessur and Dudek (2000) state that the PCA is very sensitive to image changes like rotations, shifts and differences in scale and changes in lightning conditions. When projected into the eigenspace the resulting points are not necessarily close to their corresponding training image points. To overcome these drawbacks, good segmentation and normalization methods have to be found.
- If an image is divided into a grid of sub-images and the PCA is computed using these sub-images, the method is called eigenwindow method. It was first proposed by Ohba and Ikeuchi (1996).

2.1.2. Incremental Principal Component Analysis – IPCA

The batch method for computing the eigenspace model described in the previous section has some drawbacks if tasks involving exploration behavior of the robot are

concerned. Exploration is strongly related to the problem of adapting an existing eigenspace model by adding unknown images. Therefore, the major drawbacks of the batch method are that learning and localization stage are strictly separated that all training images have to be known before the model can be constructed and that for adapting the model all images have to be kept in storage resulting in a need of huge storing capabilities and the re-computation of the model from scratch.

Because of these drawbacks, algorithms for Incremental Principal Component Analysis (IPCA) were proposed e.g. by Hall et al. (1998) and improved by Artac et al. (2002b). Like the batch method, IPCA is widely used in robot localization and navigation (Artac et al., 2002b; Freitas et al., 2003) or object recognition (Artac et al., 2002a). In the following paragraphs, the algorithm will be described more detailed. The description is based on Artac et al. (2002b) and Freitas et al. (2003).

Updating the eigenspace model

The algorithm described here assumes that there was already an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{T}, \mathbf{\Lambda}, n)$ built. To update the existing model by adding a new image \mathbf{x}_{n+1} the mean has to be updated:

$$\bar{\mathbf{x}}' = \frac{1}{n+1} (n\bar{\mathbf{x}} + \mathbf{x}_{n+1}). \quad (2.19)$$

Then the new image is projected to the current basis \mathbf{T}

$$\mathbf{g}_{n+1} = \mathbf{T}^\top (\mathbf{x}_{n+1} - \bar{\mathbf{x}}). \quad (2.20)$$

The residual vector \mathbf{r} caused by projecting \mathbf{x}_{n+1} into a lower dimensional subspace and recovering, can be computed according to equation (2.13)

$$\mathbf{r}_{n+1} = (\mathbf{T}\mathbf{g}_{n+1} + \bar{\mathbf{x}}) - \mathbf{x}_{n+1}. \quad (2.21)$$

The residual vector \mathbf{r}_{n+1} is orthogonal to all other basis vectors of \mathbf{T} and so its normalized equivalent

$$\hat{\mathbf{r}}_{n+1} = \begin{cases} \frac{\mathbf{r}_{n+1}}{\|\mathbf{r}_{n+1}\|} & \|\mathbf{r}_{n+1}\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

will be used to update the existing basis \mathbf{T} . The new $m \times (k+1)$ sized basis \mathbf{T}' is obtained by appending $\hat{\mathbf{r}}_{n+1}$ to the current basis \mathbf{T} and applying a rotation \mathbf{R}

$$\mathbf{T}' = [\mathbf{T}, \hat{\mathbf{r}}_{n+1}] \mathbf{R}, \quad (2.23)$$

where \mathbf{R} is of the size $(k+1) \times (k+1)$. To update the existing eigenimages \mathbf{g}_i ; $i = 1 \dots n$ it is necessary to reconstruct each image \mathbf{g}_i using equation (2.12) and to transform it again to a new low dimensional representation

$$\mathbf{g}'_i = \left(\mathbf{T}'^\top \right) (\mathbf{x}_i - \bar{\mathbf{x}}'); \quad i = 1 \dots n + 1. \quad (2.24)$$

After updating the basis the original images and the old basis can be discarded. The $n + 1$ images are now represented in a $k + 1$ dimensional eigenspace.

The rotation matrix

Since the computation of the matrix \mathbf{R} is not so essential for the proposed landmark selection algorithm, only a short description will be given here. Interested readers are referred Hall et al. (1998). The rotation matrix \mathbf{R} is a solution for the eigenproblem

$$\mathbf{D}\mathbf{R} = \mathbf{D}\mathbf{\Lambda}' \quad (2.25)$$

where $\mathbf{\Lambda}'$ is a diagonal Matrix with the new eigenvalues λ'_i ; $i = 1 \dots k + 1$ and \mathbf{D} is a matrix consisting of known components of λ and \mathbf{x}_{n+1}

$$\mathbf{D} = \frac{n}{n+1} \begin{bmatrix} \mathbf{\Lambda} & 0 \\ 0 & \mathbf{x}_{n+1} \end{bmatrix} + \frac{n}{(n+1)^2} \begin{bmatrix} \mathbf{g}_{n+1}\mathbf{g}_{n+1}^\top & \gamma\mathbf{g}_{n+1} \\ \gamma\mathbf{g}_{n+1}^\top & \gamma^2 \end{bmatrix} \quad (2.26)$$

with

$$\gamma = \hat{\mathbf{h}}_{n+1}^\top (\mathbf{x}_{n+1} - \bar{\mathbf{x}}). \quad (2.27)$$

Following these steps a new eigenspace model $\Omega' = (\bar{\mathbf{x}}', \mathbf{T}', \mathbf{\Lambda}', n + 1)$ was computed.

Model's dimensionality

The algorithm described above allows an exact representation of all $n + 1$ images because the spanned subspace is $k + 1$ dimensional. So with every image added, the subspace grows which is – according to the application – not desired. On the other hand the computed model Ω' allows an representation with the same accuracy as Ω and the images $x_n + 1$ is also represented accurately.

Several methods were proposed in the literature to find a trade-off between accuracy and keeping dimensionality low. These solutions include adding a new vector whenever the size of the residue vector exceeds a certain threshold, whenever the sum of the reconstruction errors of the image representations exceeds a certain threshold or whenever the smallest eigenvalue exceeds a certain threshold.

The reduction of dimensionality takes place before the images are transformed to the new basis by keeping only the first k columns of the transformation matrix \mathbf{T}' and the k greatest eigenvalues of $\mathbf{\Lambda}'$.

2.2. Clustering

Cluster analysis tries to find an optimal grouping of data into clusters so that every observation in the cluster is very similar to the other observations in the same cluster but all the clusters are dissimilar to each other. The two most often used methods for clustering are Hierarchical Clustering and K-means Clustering. These methods will be described below. Our description is based on Rencher (2002) and Duda et al. (2001).

2.2.1. Hierarchical Clustering

Hierarchical clustering methods can be divided into agglomerative and divisive methods. Here we focus on agglomerative hierarchical clustering, which starts with one cluster per observation and iteratively joins clusters into a new cluster leading to a single cluster containing all observations in the final step. Therefore the pairwise observation distance is computed and in the joining step the two closest clusters are melted. Therefore the distance function is essential for the results of the clustering. Once the clusters are melted, they remain together resulting in an hierarchical sequence that can be represented as tree. The underlying algorithm is given in algorithm 2.1.

Algorithm 2.1: Algorithm for agglomerative hierarchical clustering

```
1: procedure clusterTree=hierarchicalClustering( $Z$ )
2:  $c = |Z|$ 
3: while  $c > 1$  do
4:    $c = c - 1$ 
5:   find  $i, j$  for which  $\text{dist}(Z_i, Z_j)$ ;  $i, j \in [1, c]$  gets minimal
6:    $Z = Z \setminus Z_i$ 
7:    $Z = Z \setminus Z_j$ 
8:   merge  $Z_i$  and  $Z_j$  to a new cluster  $Z_{ij}$ 
9:    $Z = Z \cup Z_{ij}$ 
10: end while
11: return  $Z$ 
```

For hierarchical clustering the resulting tree is called dendrogram. It shows all the steps in the hierarchical procedure including the distances at which the clusters are merged. The dendrogram often gives a hint for the optimal number of clusters. The number of clusters is given by cutting all the edges in the tree at a certain distance level. The cut should be done between two levels that are far away from each other. An example dendrogram and several possible cuts are shown in figure 2.2.

Distance functions

So far we did not mention the distance function used to determine which clusters have to be melted. For well separated clusters all functions will compute the same clusters, but for badly separated clusters the choice of the distance function can have an influence on the results. The distance functions used most often are:

Single Linkage

For the single linkage or nearest neighbor method the distance between two clusters A and B is defined as the minimum distance between a point in A and B

$$\text{dist}_{\min}(A, B) = \min\{d(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \in A, \mathbf{x}_j \in B\} \quad (2.28)$$

where d is any distance function.

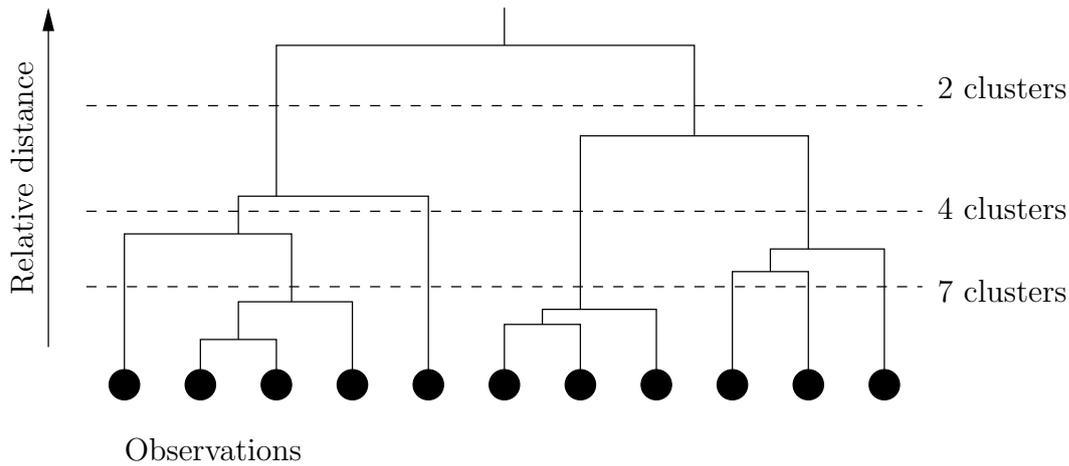


Figure 2.2.: An example dendrogram with cuts resulting in 2, 4 or 7 clusters.

Complete Linkage

The complete linkage or farthest neighbor method defines the distance between A and B as the maximum distance between a point of A and B:

$$\text{dist}_{\max}(A, B) = \max\{d(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \in A, \mathbf{x}_j \in B\}. \quad (2.29)$$

Average Linkage

For the average linkage method the distance between A and B is defined as the average of the $n_A n_B$ distances between the n_A points in A and the n_B points in B with any distance function d :

$$\text{dist}_{\text{avg}}(A, B) = \frac{1}{n_A n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} d(\mathbf{y}_i, \mathbf{y}_j). \quad (2.30)$$

Centroid

The centroid method defines the distance between A and B as the distance between the mean vectors of A and B:

$$\text{dist}_{\text{cent}}(A, B) = d\left(\frac{1}{n_A} \sum_{i=1}^{n_A} \mathbf{y}_i, \frac{1}{n_B} \sum_{j=1}^{n_B} \mathbf{y}_j\right) \quad (2.31)$$

The mean vector of the vectors of one cluster is often referred to as centroid.

There are a lot of other approaches that can be used to measure the distance, e.g. the ward's method, the median method or the flexible beta method. Which of the methods leads to the best results or separates the clusters optimally is highly application dependent.

2.2.2. K-means clustering

K-means clustering assumes an initial clustering and then reallocates the observation in order to minimize some error function. The class of algorithms K-means clustering belongs to is called partitioning or optimization clustering. A good strategy for partitioning would be to test every possible partition of the data set and find the optimal partition according to some error criterion. Since the number of possible partitions grows very fast with the number of observations it is not possible to test all of them.

The K-means algorithm first selects c observations where c is the number of clusters. There are several strategies how to choose the seed, for example the observations can be chosen randomly, the first c items of the data set can be chosen or those points that are farthest away from each other can be used as seed. Independent on which method is used, the number of clusters has to be specified before the clustering begins. After choosing the seeds, each remaining observation is assigned to the nearest seed. As soon as a cluster has more than one member, the seed is replaced by the cluster's centroid. After assigning each element to one cluster, each item is examined again if it is closer to another cluster's centroid than to its own cluster's centroid. If so, it is moved to the corresponding cluster and the centroids are updated. This process is updated until the clustering can not be improved any more. A pseudo-code notation of the algorithm can be found in algorithm 2.2.

Algorithm 2.2: Algorithm for k-means clustering

```
1: procedure clusters=kMeansClustering( $Z, c$ )
2: initialize seeds  $\mu_1, \mu_2, \dots, \mu_c$ 
3: repeat
4:   classify the samples according to nearest  $\mu_i$  (Nearest Neighbor Search)
5:   recompute  $\mu_i$ 
6: until no change in  $\mu_i$ 
7: return  $\{\mu_1, \mu_2, \dots, \mu_c\}$ 
```

The k-means procedure is sensitive to the initial choice of the seeds. If choosing different seeds that lead to different results or if the clustering converges only very slowly, there may be no natural clusters in the data or the number of clusters is not chosen appropriately.

2.3. Proposed landmark selection algorithms

In this section, the proposed algorithms for landmark selection will be described. The basic idea of the proposed algorithms is to select landmarks that are as dissimilar as possible to all other landmarks. The profile-based algorithm uses pairwise image dissimilarity between sub-images as ranking criteria, the IPCA-based algorithm uses Incremental Principal Component Analysis. Before introducing the algorithms we point out some things that are supposed to outline the purpose of our studies.

2.3.1. Introductory annotations

As we have shown in section 1.4.1 most of the landmark selection algorithms preselect possible landmarks in a first step and then – in the second step – discard unreliable or un-unique landmarks. Here we do not propose methods for preselecting possible landmarks. The proposed algorithms assume that a list of preselected landmarks is already given. This is due to the fact that in the used aerial image the distribution of features like corners, edges, edge-density or symmetry detected by standard attention operators (Sim et al., 2003) is rather uniform and does not constrain the number of landmarks or the location of possible landmarks to distinguishable image regions. Therefore we do not achieve any advantages of preselecting possible landmarks.

However, nothing prevents the proposed algorithms from having such a prior attentional module. For future work and for larger environments we think it is necessary to build such an attentional module to speed up the necessary computations.

Our approach assumes that an image of the environment as seen from the blimp is given as map of the environment. For example, the map can be obtained by mosaicking or can be an aerial image like in this case. For a review of mosaicking approaches see Gracias (2002). Since we do not preselect possible landmarks the image is divided into a grid and all the sub-images are used as possible landmarks. The algorithms then select the most promising landmarks according to the algorithm’s criterion. Both of the algorithms operate on a set of sub-images transformed to an eigenspace. The eigenspace is computed using PCA-methods. PCA coefficients lead to a compact representation and – if only few dimensions are used – also to generalization. Although pure appearance based methods have been proposed for localization and object recognition, there are only few works like Ohba and Ikeuchi (1997) that use appearance-based methods to compute the ranking between possible landmarks in order to select the best ones.

2.3.2. Profile-based algorithm

The key aspect of the algorithms is the computation of a distance matrix of pairwise image dissimilarities. For each possible landmark, the average dissimilarity to all other possible landmarks is computed and those landmarks that are most dissimilar are selected.

In detail this algorithm takes a set G of images \mathbf{g}_i ; $i = 1..n$ in k -dimensional eigenspace and the number of landmarks l that should be selected as input. In the next step it computes a distance matrix \mathbf{D} with the pairwise image distances as entries

$$D_{i,j} = \text{dist}(\mathbf{g}_i, \mathbf{g}_j). \quad (2.32)$$

As a distance measure any kind of dissimilarity function ρ that holds the following requirements

$$\rho(\mathbf{g}_i, \mathbf{g}_j) \geq 0 \text{ and } \rho(\mathbf{g}_i, \mathbf{g}_j) = 0 \Leftrightarrow \mathbf{g}_i = \mathbf{g}_j \quad (\text{non-negative}) \quad (2.33)$$

$$\rho(\mathbf{g}_i, \mathbf{g}_j) = \rho(\mathbf{g}_j, \mathbf{g}_i) \quad (\text{symmetrical}) \quad (2.34)$$

can be used. Due to the properties of the similarity function the matrix is symmetrical with zeros at the diagonal. For this work a normalized Sum of Squared Differences (nSSD) is used:

$$\text{nSSD}(\mathbf{g}_i, \mathbf{g}_j) = \frac{\sum_{\nu=1}^k \left[\left(g_1^{(\nu)} - \bar{g}_1^{(\nu)} \right) - \left(g_2^{(\nu)} - \bar{g}_2^{(\nu)} \right) \right]^2}{2 (\|\mathbf{g}_1 - \bar{\mathbf{g}}_1\| + \|\mathbf{g}_2 - \bar{\mathbf{g}}_2\|)}, \quad (2.35)$$

where $g^{(\nu)}$ denotes the ν -th element of vector \mathbf{g} and $\bar{\mathbf{g}}$ is the $k \times 1$ vector

$$\bar{\mathbf{g}} = \frac{1}{k} \sum_{\nu=1}^k \mathbf{g}^{(\nu)} \cdot (1, \dots, 1)^\top. \quad (2.36)$$

The nSSD guarantees a normalization to $0 \leq \text{nSSD}(\mathbf{g}_i, \mathbf{g}_j) \leq 1$. A proof for the normalization is given in Stürzl (2003).

For the further evaluation a distance profile vector p is computed by averaging over the rows of column j of the distance matrix

$$p_j = \frac{1}{n-1} \sum_{i=1; i \neq j}^n D_{i,j}. \quad (2.37)$$

For this approach sub-images that are very dissimilar to all other views are considered as landmarks. Since $D_{i,j}$ is close to unity if the views $\mathbf{g}_i, \mathbf{g}_j$ are very dissimilar, a sub-image that is very dissimilar to all other sub-images will have a profile value close to unity. The ranking is done by sorting the profile values in an descending order and selecting the l most dissimilar views.

Algorithm 2.3 gives a formal description of the algorithm described above. The algorithm is of squared complexity and storage requirement, since the distance matrix has to be computed and stored.

2.3.3. IPCA-based algorithm

The second algorithm uses IPCA to compute the image ranking. It iteratively updates an existing eigenspace by adding that image that can be expressed worst in the existing eigenspace. The image that can be expressed worst is exactly that image, for which the norm of the residue vector is maximal. Because of using the IPCA the algorithm is capable of online-landmark selection and can therefore be used for Simultaneous Localization and Mapping (SLAM). For an overview over current SLAM approaches see Bailey and Nebot (2001) and Frese and Hirzinger (2001). Most of the approaches to landmark selection require an exploration phase during which the map is built and a selection phase to select the best landmarks from the map. The advantage of the online-algorithm is that the both steps are combined to a single step. Anyway, here we do not deal with online landmark selection.

Algorithm 2.3: Algorithm for distance–profile based landmark selection

```

1: procedure landMarkList=profileRanking( $G, l$ )
2:  $\mathbf{p} = 0 \in \mathbb{R}^n$ 
3: for  $i = 1$  to  $n$  do
4:   for  $j = i$  to  $n$  do
5:      $D_{i,j} = D_{j,i} = \text{dist}(\mathbf{g}_i, \mathbf{g}_j)$ 
6:   end for
7: end for
8: for  $j = 1$  to  $n$  do
9:   for  $i = 1$  to  $n$  do
10:     $p_j = p_j + \frac{1}{n-1} D_{i,j}$ 
11:   end for
12: end for
13: (values, indices)=sortDescending( $\mathbf{p}$ )
14: return indices[1 :  $l$ ]

```

In detail, the algorithm requires a set of images S , the number of landmarks l and the set of all possible sub–images X . It then computes a k –dimensional eigenspace model Ω using the batch method described in section 2.1.1 for the set of start images S , which will be updated later. The images of S are used as the the first landmarks.

Then every image \mathbf{x}_i is transformed to the eigenspace according to equation (2.10) and the resulting residue vectors \mathbf{r}_i are computed using equation (2.21). Since the norm of the residue vector is a measure how good the considered image can be expressed with the eigenvectors of the current basis, the maximal norm of the residue vectors is computed and the corresponding image is selected as new landmark. Therefore the added landmark is exactly that image that is as different as possible to the already selected landmarks.

Then the picked image has to be removed from X and the eigenspace model is updated. The algorithm terminates if the number of selected landmarks is greater than l . Alternatively some threshold for the residue vector can be used although we did not try it here to achieve a better comparability to the profile–based algorithm.

One aspect of the IPCA–based ranking that has to be mentioned before describing the experiments are the methods how the start set of images S and the number of start images are chosen. Since the images contained in the start set are used as landmarks, the further ranking is dependent on the selected images. For this work a fixed number of start images and two methods for selecting start views were considered:

1. Choosing the start set according to the profile ranking. The advantage of this method is that the first landmarks are very dissimilar to all other views and should therefore be good landmarks to start the further ranking. The disadvantage is that the profile ranking has also to be computed.
2. Choosing images randomly. The advantage of the method is that there is no

Algorithm 2.4: Algorithm for IPCA based landmark selection

```

1: procedure landMarkList=ipcaRanking( $S, l, k, X$ )
2:  $X = X \setminus S$ 
3:  $k = |S|$ 
4:  $(\bar{\mathbf{x}}, \mathbf{T}, \mathbf{\Lambda}, n) = \text{computeEigenspaceModel}(S, k)$ 
5: while  $k \leq l$  do
6:    $\mathbf{r} = \mathbf{0} \in \mathbb{R}^{|X|}$ 
7:   for  $i = 1$  to  $|X|$  do
8:      $\mathbf{g} = \mathbf{T}^\top (\mathbf{x}_i - \bar{\mathbf{x}})$ 
9:      $\mathbf{r}_i = (\mathbf{T}\mathbf{g} + \bar{\mathbf{x}}) - \mathbf{x}_i$ 
10:  end for
11:   $m = \arg \max\{\|\mathbf{r}_i\| \mid i \in 1 \dots |X|\}$ 
12:   $X = X \setminus \{\mathbf{x}_m\}$ 
13:   $S = S \cup \{\mathbf{x}_m\}$ 
14:   $(\bar{\mathbf{x}}, \mathbf{T}, \mathbf{\Lambda}, n) = \text{updateEigenspaceModel}(\bar{\mathbf{x}}, \mathbf{T}, \mathbf{\Lambda}, n)$ 
15: end while
16: return  $S$ 

```

computational overhead. The disadvantage is that it is not predictable how the start images influence the further ranking. The influence of the further ranking, if very similar images are chosen, is also unpredictable (or would require experiments that were not performed for this work).

2.4. Robustness and reliability evaluation

After presenting the landmark selection algorithms a detailed overview over the reliability evaluation will be given. The method computes an upper limit of the image dissimilarity for which an error-less localization is guaranteed.

When the blimp has to localize itself in the topological map a nearest neighbor search has to be done in order to determine that landmark that is the closest one. For each known landmark the camera image has to be compared with every known landmark and the landmark to which the current image is most similar is selected as the blimp's current position.

To measure a landmark's reliability we assume the blimp is at the landmark's position except for small errors in the orientation, the position or the altitude. These small errors result in views that are either rotated, translated or have a different scaling respectively. Additionally we consider changes in the image brightness. The current view is transformed to the eigenspace resulting in a point with coordinates differing from those of the landmark. The task is now to find out, how much the view can be changed so that the landmark is still selected as nearest neighbor.

Therefore we propose to compute for a set $L := \{\mathbf{g}_{i'} \mid i' = 1 \dots l\}$ of selected landmarks the maximal image dissimilarity ϵ so that the correct landmark is still selected. We

A similar method was used by Ohba and Ikeuchi (1997) to select the most stable and therefore the most reliable landmarks. The authors have chosen a fixed limit and a fixed rotation angle. If the image dissimilarity between the landmark and the landmark rotated by the angle is not within the threshold, the landmarks is discarded as unreliable landmark. Most other related works only show experimental data to proof the reliability of the selected landmarks.

3. Experiments and Results

In this chapter we describe the experiments that were done within this work. The experiments include some basic experiments in order to get familiar with the PCA and the eigenspaces. Then the experiments and the results for the landmark selection for the used aerial image and the reliability evaluation are presented.

3.1. Fundamentals

For the experiments described in this chapter, an aerial image of the area around *Instituto Superior Técnico* at Lisbon, Portugal, was used. This image is referred to as “floormap”. The size of the original image I is 1600×1200 pixels with a resolution of approximately one square meter per pixel.

The floormap was divided into sub-images, in further referred to as “views”. 42 views were taken along a grid with six rows and eight columns. The chosen arrangement results in a view size of $v_1 = 200 \times 200$ pixels. Another 35 views were taken at these points, where the other views are adjoining, resulting in a total amount of 83 views with an overlap between two views of 25%. Figure 3.1 shows the floormap and the positions of the 83 views taken. Figure A.1 shows the same more clearly arranged.

The experiments were run with several levels of scaling and smoothing. For a given scale factor s_i the view size v_i was computed by

$$v_i = \lfloor v_1 s_i \rfloor. \tag{3.1}$$

In order to cover the whole image with views and to keep the number of views constant, the original image I was down-scaled to an image size of $8v_i \times 6v_i$ which is coupled with small changes in the aspect ratio. The view sizes for the considered scaling factors are listed in table 3.1. After picking out of the floormap and down-scaling, the sub-views were smoothed with several levels of Gaussian smoothing (in further referred to as “blurring”), listed in Table 3.2.

To speed up computations all the necessary computations on the images and views were done in the eigenspace. Therefore, all the views were transformed according to section 2.1.1. For the experiments a set E of dimensions was used:

$$E = \{1, 2, 3, 5, 7, 10, 12, 15, 18, 20, 25, 35, 50, 65, 83\} \tag{3.2}$$

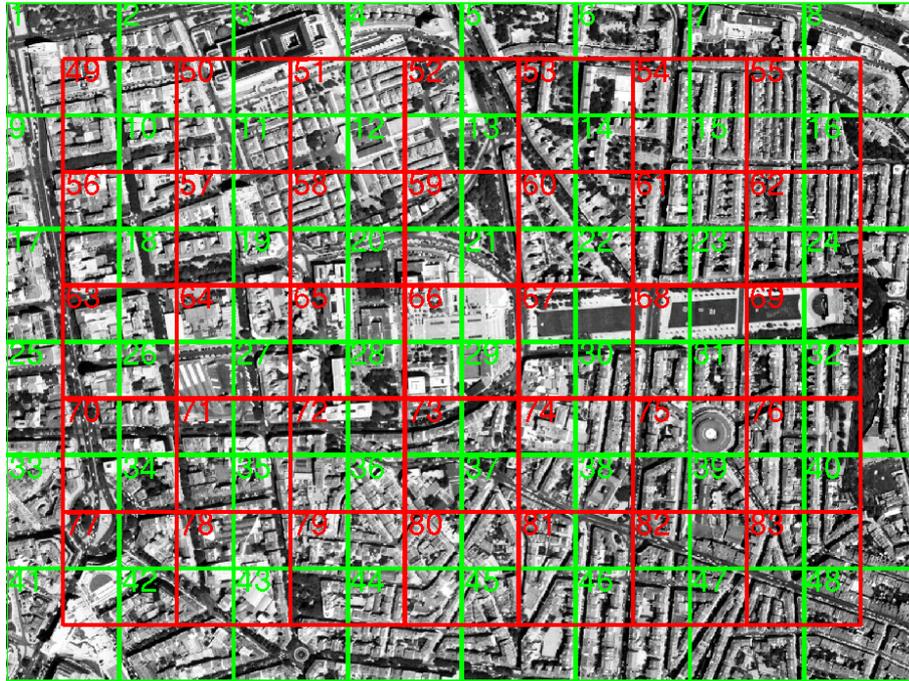


Figure 3.1.: Position and view number of the views taken out of the floormap image.

Table 3.1.: Scale factors and resulting view sizes.

scale factor s_i^{-1}	1	2	3	4	5	6	8	10	12	15	20
view size v_i [pixels]	200	100	66	50	40	33	25	20	16	13	10

Table 3.2.: Parameters for Gaussian smoothing.

Blurring:	neighborhood-size	Standard deviation σ
none	–	–
gauss1	3	0.48
gauss2	5	0.65
gauss3	7	0.76

3.2. PCA-related experiments

3.2.1. Aim and description

The aim of the PCA-related experiments was to visualize how many percent of the image information are covered by the first n dimensions in the PCA-space. Therefore the first n eigenvalues were summed. Another aim was to visualize eigenimages, in order to get a feeling for them and for the influence of the dimension to the accuracy of the representation.

3.2.2. Results

In figure 3.2 the plots of the covered variance against the number of dimensions for several scale factors are shown. The plots show that

- the more the views are smoothed, the greater is the variance covered by the first principal component (i.e. the greatest eigenvalue). Therefore the resulting graph gets steeper.
- the more the views are down-scaled, the greater is the greatest eigenvalue.

To summarize, the plots show that there are less dimensions needed to represent the views if the views were more down-scaled and smoothed.

Figure 3.3 shows the first 25 eigenimages. The first three eigenimages are very generalized, there do not appear much shapes like crossroads or streets. If an image in the eigenspace is understood as linear combination of the eigenimages, it is clear that the image can only be represented very bad. Then in the following eigenimages structures like blocks or streets appear. These structures get more and more detailed. So it is visible that a huge number of eigenimages can be necessary to represent an image accurately, but only few eigenimages are sufficient for a generalized representation.

3.2.3. Conclusions

The results are especially interesting, because a small view size and a small number of principal components both speed up the necessary computations. But the more the views are down-scaled and smoothed the more difficult will be the localization in the map, because the number of similarities will rise. Therefore it is necessary to find a good trade-off between localization quality and fast computation or view size and smoothing.

The visualization of the eigenimages shows clearly that a certain number of dimensions is needed in order to achieve a clear and unique representation of the view allowing reliable or robust localization. In order to visualize this fact again, the quality of recovered views for several dimensions is shown in figure A.2.

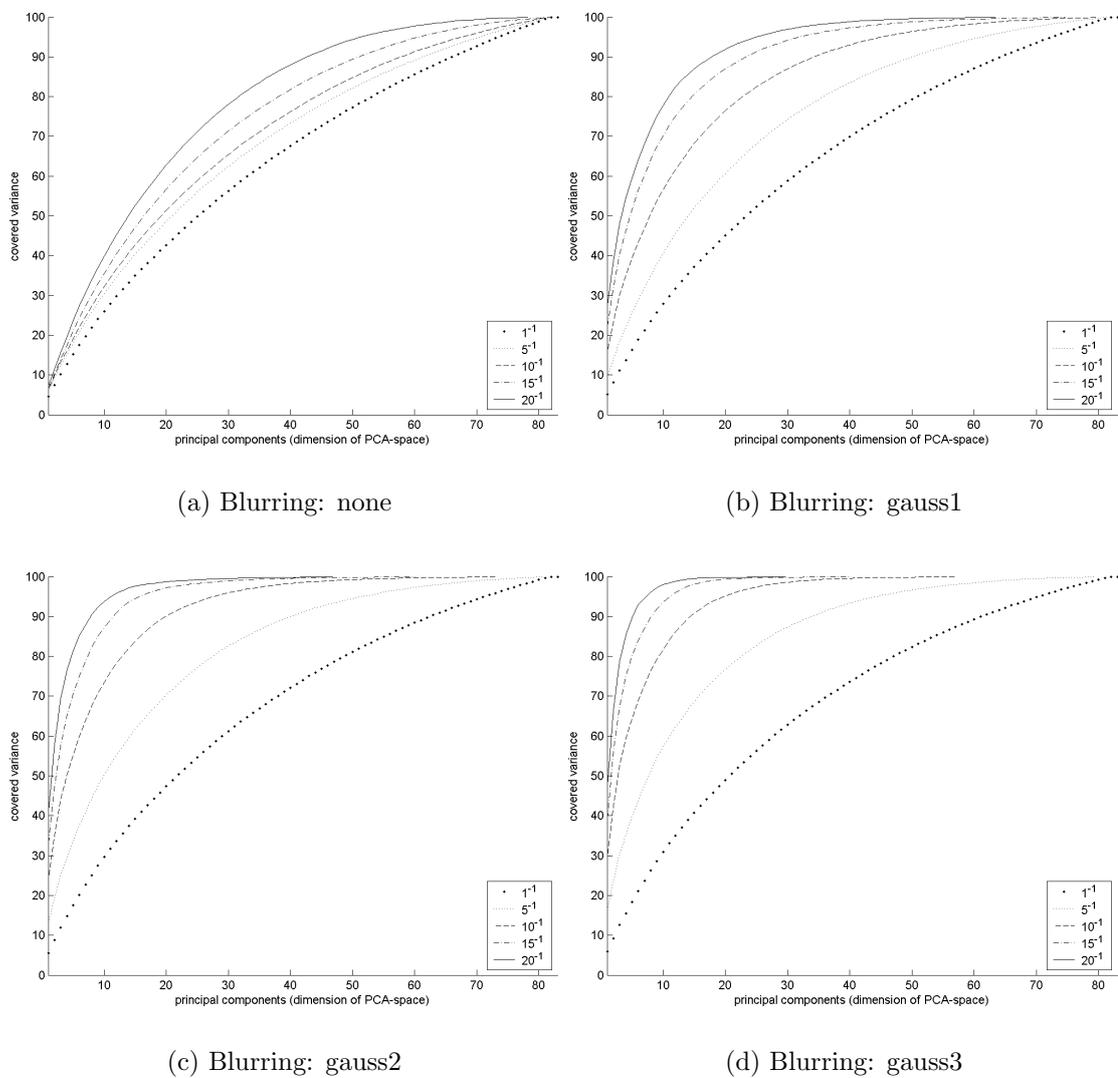


Figure 3.2.: Covered variance against number of dimensions of PCA space for several scale factors and blurrings.

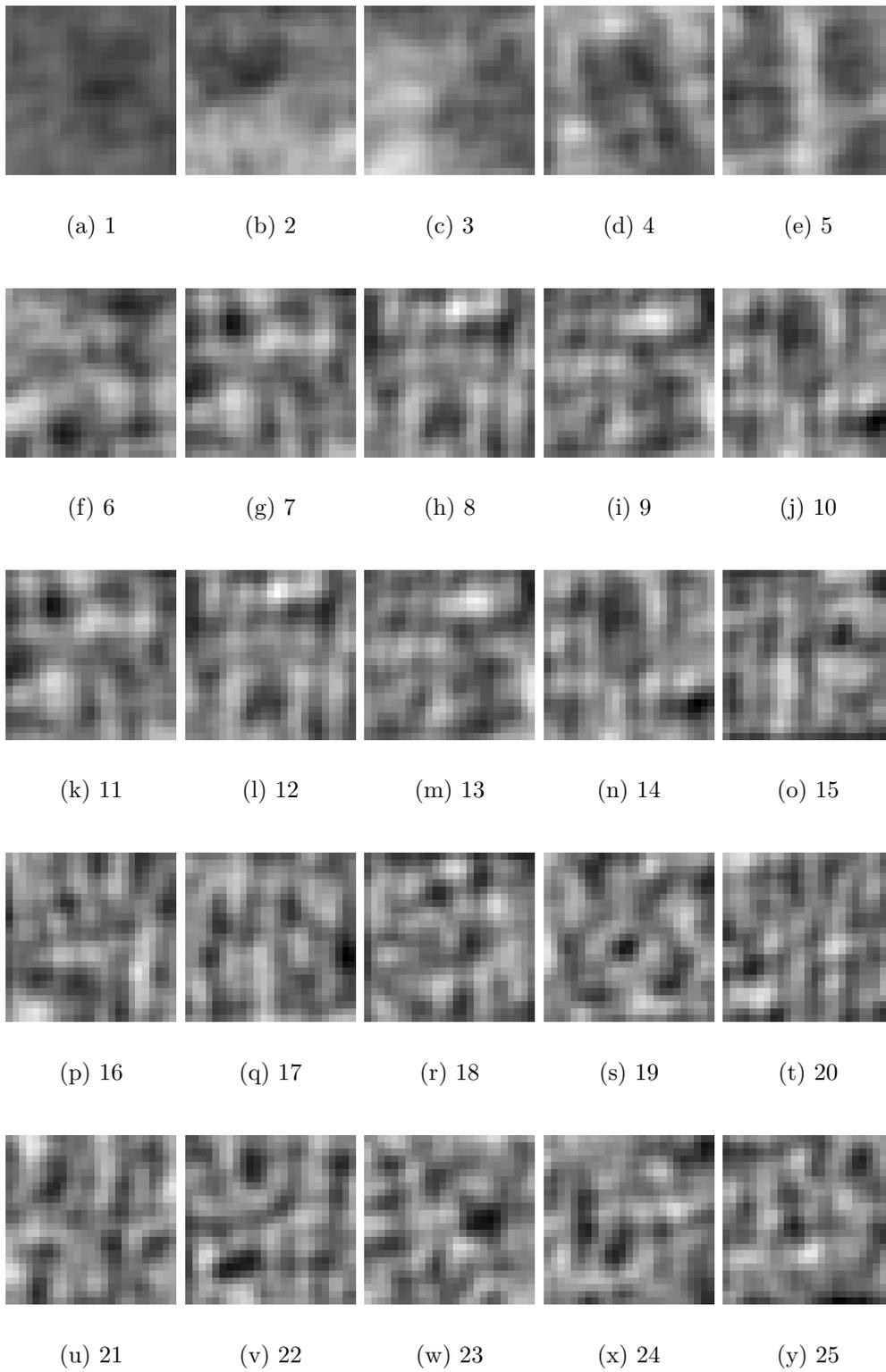


Figure 3.3.: The first 25 eigenimages. Scale: 10^{-1} , Blurring: gauss1

3.3. Profile-based landmark selection

3.3.1. Aim and description

The aim of this experiment was to compute the profile-based landmark selection for the floormap and all the possible combinations of parameters and to find out which parameters lead to the best results and how the parameters influence the landmark selection and all the sub-steps needed for computing the selection. In the following we present the results for each of the sub-steps, namely the computed distance matrices, the computed distance profiles and finally the result of the algorithm, the landmark ranking.

3.3.2. Results

Distance matrices

An example distance matrix¹ is shown in figure 3.4, for a more comprehensive overview see figures A.3 to A.8. In the plots white entries mark very dissimilar and black entries mark identical views.

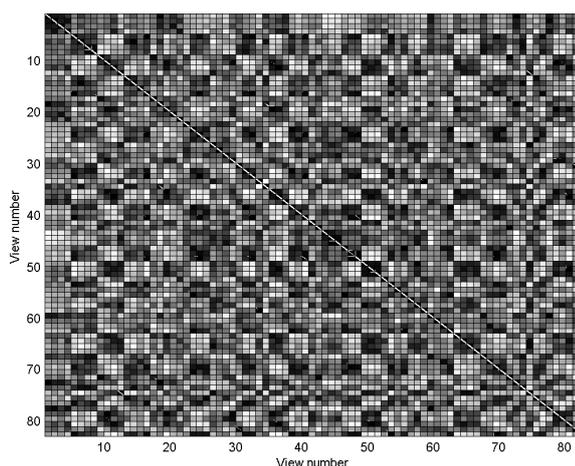


Figure 3.4.: Example distance matrix. Scale factor: 10^{-1} , dimensions: 3, blurring: gauss1

The surface plots show that

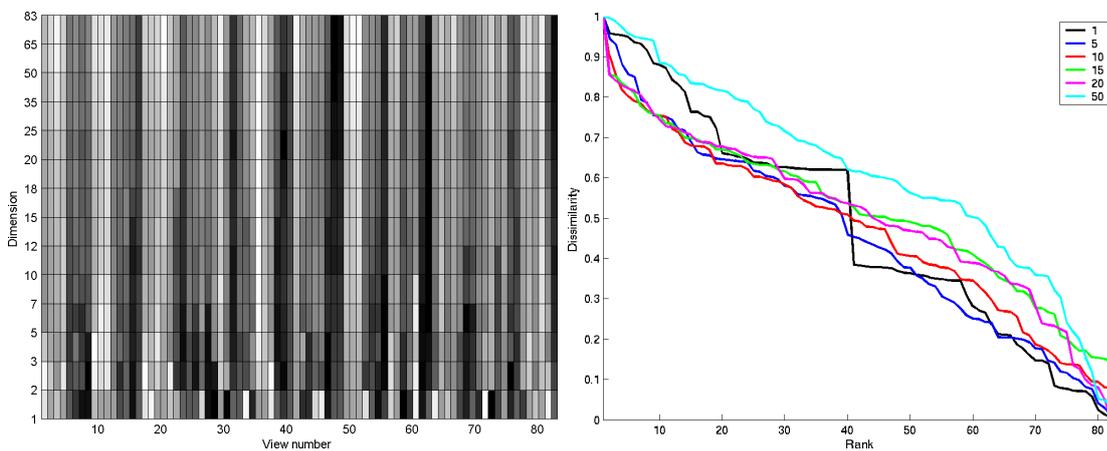
- there are – especially in low dimensional eigenspace – views that are quite similar to each other. These views show out as dark blocks in the distance matrices and usually contain a region of adjacent views. There are also a couple of white blocks that show regions of views that are very dissimilar to other regions. These blocks disappear if more dimensions in the eigenspace are used.

¹The white diagonal is a plotting error which is due to MATLAB

- there are views that are very dissimilar to all other views. These views show out in the distance matrices as white rows or columns. These views are very difficult to detect visually in the shown matrices.
- the more dimensions in PCA-space are used, the more homogeneous the distance matrices get, i.e. the most entries are approximately 0.5. The less dimensions are used, the wider is the range of gray values.
- the more the views are downsampled the greater gets the difference between smoothed and unsmoothed views, especially in higher dimensional eigenspace.

Distance profiles

For easier evaluation the computed profiles were stuck together in a matrix with $|D| = 15$ rows and 83 columns and visualized as matrix plots. An example matrix is shown in figure 3.5(a), for a more comprehensive overview over the profile matrices see figures A.9 to A.11. The profiles for each dimension were normalized to $[0, 1]$ and again white codes dissimilar and black similar.



(a) Example profile matrix. Scale factor: 8^{-1} , blurring: gauss1

(b) Sorted profile values for several dimensions. Scale factor 8^{-1} , blurring: gauss1

Figure 3.5.: Example distance profiles.

The profile matrices show that

- the dimensions can be roughly divided into three groups. The first for 1D and 2D, the second for three to ten dimensional PCA-space, the third for more than 10 dimensions. The profiles for the first group are always different then the profiles of the other groups. The ones for the second group, are more similar to those of the third group, but minima and maxima change more often then in the third group. For the third group, the extreme values seldom change and many views have values around an average gray.

- especially for the smoothed views and for views with a scale factor smaller than 4^{-1} there are many columns that have very similar entries for all the dimensions.
- at a first glance these columns or views that show out to be very similar or very dissimilar to others, seem to be very stable for all the different possible parameters.
- there are a lot of views with average dissimilarities and only little outstanding or very similar views.

Additionally figure 3.5(b) visualizes the dissimilarities for different dimensions. The plots show better than the profile matrix that there are only less views that are very dissimilar or very similar to all other views and there is a huge number of views with an average similarity.

Landmark selection

Some examples for possible profile rankings are shown in figure 3.6, tables with all the results are shown in tables A.1 to A.6.

The tables and figures show that

- the division into three groups according to the number of used dimensions mentioned in the paragraph about the profile matrices is also visible in the rankings. These results can be compared to the grouping found in the eigenimages.
- the rankings get more stable for smoothed images and more than five dimensions. That means that there are less changes in the order of the view numbers and that there are very often the same views selected. The selected images are often the same for a constant scale factor and variable dimensions and vice versa. The first couple of selected landmarks seem to be more stable than the others.
- the selected views really look different. They often contain one unique image region, like for example the roads crossing in an acute angle² in view 29, the black square³⁴ in view number 20 or the big building shaped like an eight⁵ in view number 3.
- no views of neighborhoods that look very similar or even grid-like were selected. These areas include the left part⁶ and the right part of the floormap⁷.

²For insiders: *Avenida Rovisco Pais* and *Avenida Almirante Reis*

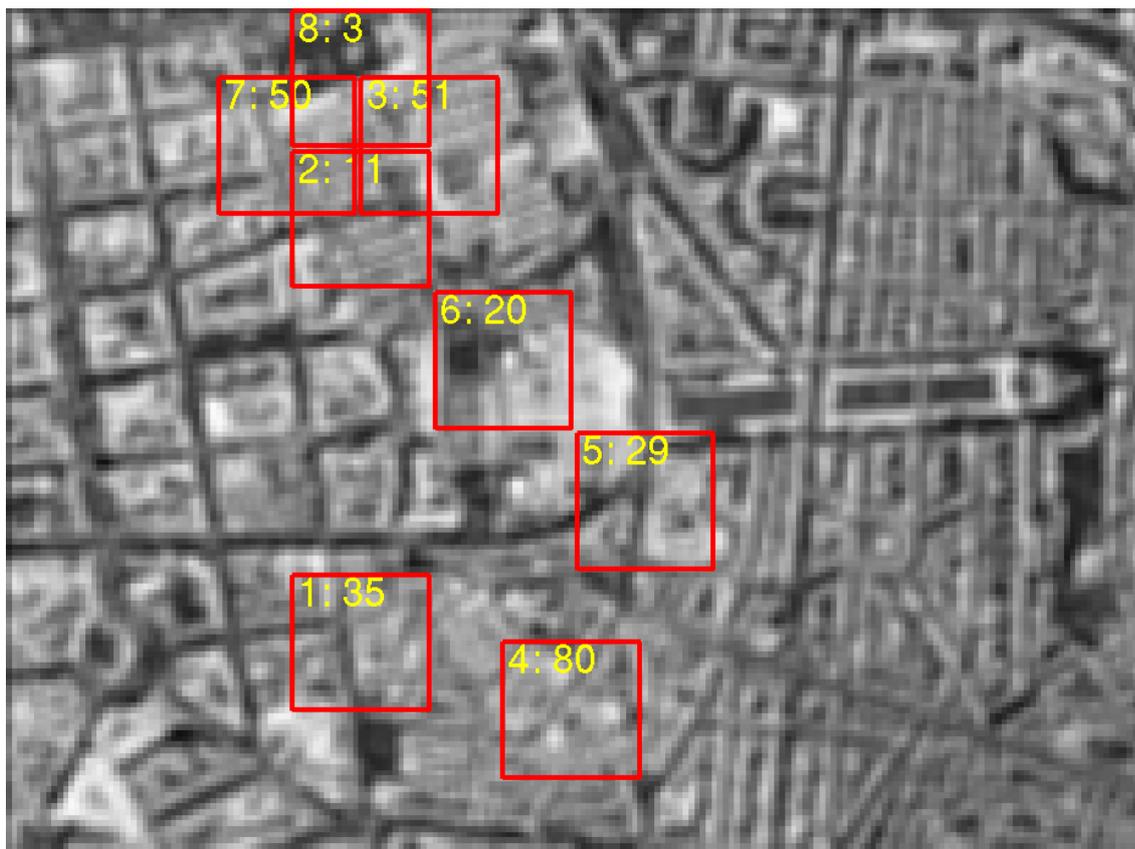
³The north Tower of *Instituto Superior Técnico*

⁴Watch out that the algorithm does not select the lab's building as best landmark ;-)

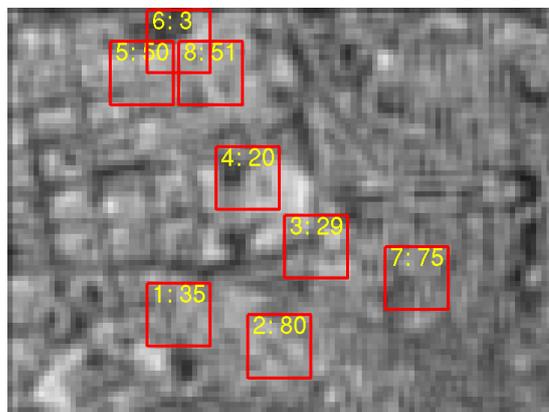
⁵The building of *Caixa Geral de Depósitos*

⁶The neighborhoods between *Saldanha* and *Campo Pequeno*

⁷*Anjos, Areeiro* and *Alameda Almirante Reis*



(a) Scale factor: 8^{-1} ; Dimension: 18; Selected Views: 35, 11, 51, 80, 29, 20, 50, 3



(b) Scale factor: 15^{-1} ; Dimension: 20; Selected Views: 35, 80, 29, 20, 50, 3, 75, 51

Figure 3.6.: Example profile rankings. Blurring for both images: gauss1

3.3.3. Conclusions

Like in the work of Knapek et al. (2000) and Ohba and Ikeuchi (1997) computing the distance matrix leads to a robust landmark selection. The averaging over the profile seems to weight landmarks that are very dissimilar to all other landmarks. It seems like for the chosen arrangement of landmarks the views are already too dissimilar because many views have an average dissimilarity and there are only view very similar or dissimilar views. This conclusion is based on figure 3.5(b). It is left for further work to find out how the overlap between images influences the distance matrices and profiles. Here we will focus on the conclusions that can be drawn from the sub-experiments.

Distance matrices

The fact that the matrices get more homogeneous the more dimensions in PCA-space are used, shows that generalization is needed for landmark detection, although it is expected that more dimensions in PCA-space are needed for a better localization. So there is another hint that it is necessary to find a trade-off between robust localization and computation speed and robust landmark detection.

The blocks representing adjacent views that are only visible in matrices for low dimensional eigenspace, are a hint that the views are only similar if there is a huge degree of generalization. If the number of dimensions raises, the blocks disappear and the views get more and more dissimilar to each other. Further experiments could evaluate, how this result changes with changing the overlap between the images.

Since some views are quite dissimilar to all other views the proposed method of averaging over the dissimilarities is a good method for further evaluation of the data.

Distance profiles

The fact that the views that are very similar or very dissimilar to all the other views seem to be constant for all possible parameters might be a hint that the chosen method is an appropriate way to select and characterize landmarks, although further tests have to show that the selected views are really better landmarks than the others. This fact could also be used to average over the columns of the profile matrices to put more weight on those views that have great entries for a lot of dimensions.

Landmark selection

Promising results are that the landmark selection seems to be stable and the selected landmarks all look different. Another good result is that the views were only taken from regions of the floorplan that look very dissimilar. So the landmark distribution of the shown figures is not a problem of the algorithm, but is due to the geometry of the shown neighborhoods.

3.4. IPCA–based landmark selection

3.4.1. Aim and description

The landmark selection with the IPCA–based algorithm was computed systematically for all possible parameters. The several start lists used are listed in table 3.3 and visualized in figure A.12. They were chosen in order to see how the landmarks are selected if a certain case for the arrangement of the start list occurs. Every start list contained three landmarks and again eight landmarks were selected to allow a better comparison to the profile–based algorithm.

Table 3.3.: Differences between randomly chosen start lists.

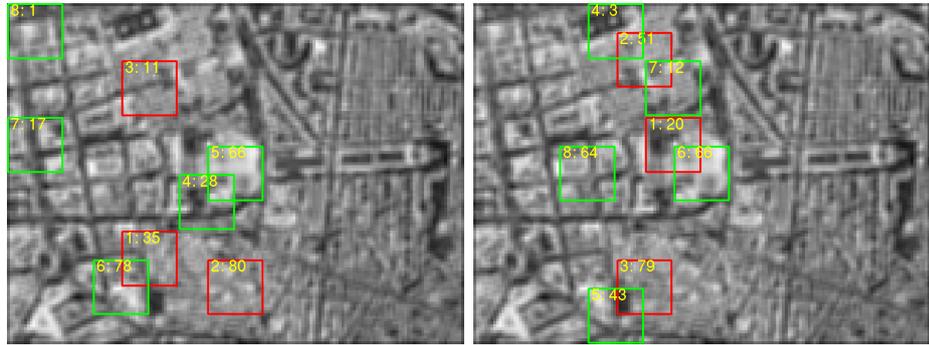
Method	Views	Description
rnd1	20, 52, 79	Very different views
rnd2	2, 29, 69	Very different views
rnd3	9, 36, 62	Two views in the same grid–like area, the third in another grid–like area
rnd4	37, 77, 78	Each view is distinctive, but maximal overlap between views
rnd5	15, 62, 77	Two of the views are very similar, the third is different and distinctive to all other possible views

3.4.2. Results

Some example selections are shown in figure 3.7, a comprehensive overview over the results can be found in tables A.7 to A.12 for the selection with profile–based start list and in tables A.13 to A.15 for the selection based on random start lists. Since the IPCA–based ranking is independent of the dimension of the used eigenspace. The dimension only gets important if the start list is based on the profile–based ranking or if further computations like the reliability evaluation have to be done. That is the reason why the tables of the results for the IPCA–based selection with profile startlist include several dimensions and the results for the selection based on random startlists do not.

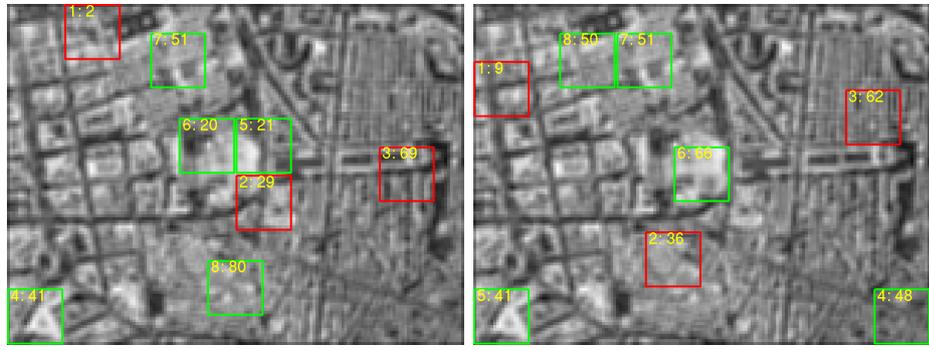
Looking at the results it strikes out that:

- also for the IPCA–based algorithm all the selected landmarks look different. They also show one characteristic pattern per landmark.
- Independent from the start list, the ranking selects landmarks that are dissimilar to every other selected landmark.
- the ranking is very dependent on the selected start list. For different startlists with very similar views the results will be similar. Vice versa, if the start lists only contain distinctive views, the selection will be completely different.



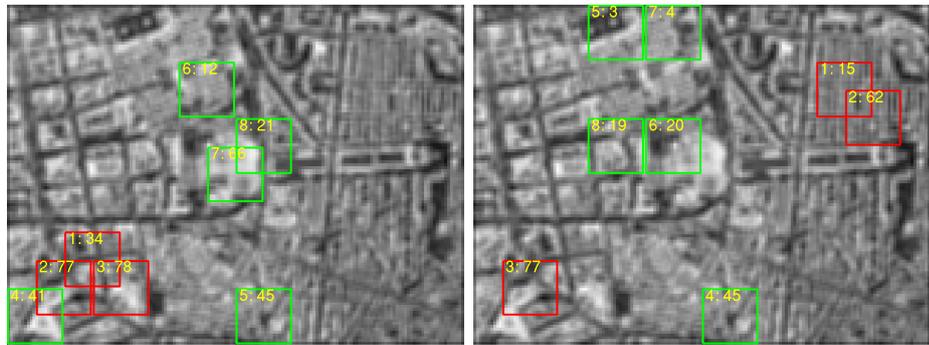
(a) Profile startlist

(b) rnd1



(c) rnd2

(d) rnd3



(e) rnd4

(f) rnd5

Figure 3.7.: Example selections for the IPCA-based landmark selection. Scale factor 10^{-1} , dimension of figure 3.7(a): 15; Blurring: gauss1

- for some parameter combinations numerical instabilities appeared. Since these stabilities are rather seldom compared to the number of performed rankings, the affected data is marked in the tables showing the results with dashes (“-”).

3.4.3. Conclusions

The results show that the IPCA-based algorithm also selects – except for the cases when numerical instabilities appeared that will be fixed in further work – good and distinctive landmarks. Anyway it is not as clear as for the profile-based algorithm which landmarks are selected and what characterizes the selected landmarks. With the selected landmarks it is not visible, which startlist leads to a good or to a bad ranking or if the start list has any influence on the quality of the selected landmarks. What can be concluded from the data is that if there is an influence it is not as great as we expected so that also a start list containing very similar landmarks can lead to a good ranking. It is far beyond the scope of this report to do experiments investigating the influence of the start lists more detailed.

3.5. Reliability Evaluation

3.5.1. Aim and description

In this section we are going to describe the results of the reliability evaluation. For each considered parameter, namely orientation, position, altitude and image brightness, we computed a lower and an upper bound for which equation (2.39) still holds. Values between the upper and lower limit curves of the presented plots do not produce erroneous localizations.

Orientation

To analyze deviations in the orientation of the blimp the image dissimilarity between the landmark and a view rotated leftwards or rightwards respectively was computed. Therefore a sub-image that is centered at the landmark’s position, was cut out of the floormap. This sub-image was sufficiently big, so that after rotating it, a view of the view size v_i , could have been inscribed without losing image information at the corners of that view. For comparison with the landmark, a view of the appropriate size was cut out of the sub-image. From step to step the rotation angle was changed by 1° .

Altitude

Deviations in the altitude of the blimp show out in views that contain a bigger or smaller area of the underlying floormap. This effect can be modelled by taking subimages of different sizes out of the floormap and scaling them all to the same view size

v_i . For each step, the size of the subimages was changed by 1% compared to the view of view size v_i .

Position

To measure the quality of a landmark with respect to deviations in the blimp's position the size of the catchment area around the correct landmark position was computed by stepwise spiraling around the correct position and counting the number of positions for which a correct localization is possible and for which another position in a 3×3 neighborhood has also been marked as successful.

Image brightness

Changes in the illumination conditions were modelled by applying a gamma-correction to the original image. The change of the gamma value between two steps was 0.5.

3.5.2. Results

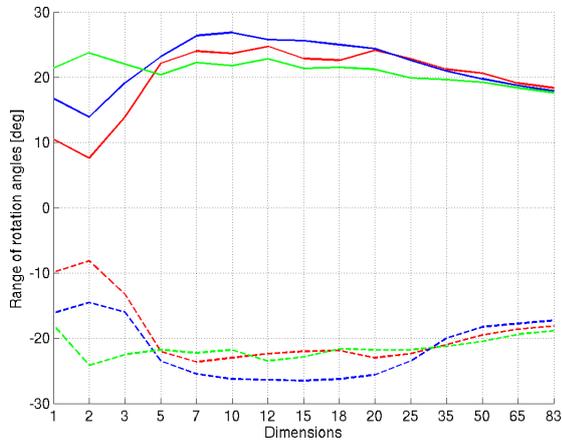
Representative plots showing the reliability for a constant scale and a varying dimension of the eigenspace are shown in figure 3.8.

Orientation

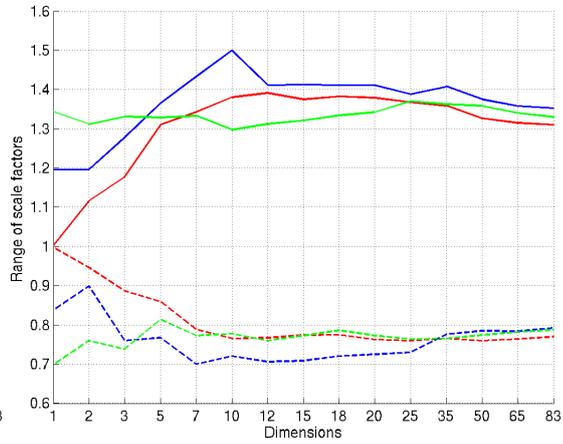
For up to ten dimensions an increase of stability is visible, i.e. the absolute values of the upper and the lower limit are increasing. Then, for more than ten dimensions, the stability slowly decreases. The maximal reliability is reached for ten to 15 dimensions allowing – for the representative data shown in figure 3.8(a) – a deviation of approximately $\pm 25^\circ$ from the correct orientation. The level of smoothing seems to have only little influence on the reliability, the values are roughly identical. It seems that the views that have been smoothed more, seem to be a little bit more stable. Independent of the dimension of the eigenspace, the difference between the levels of blurring gets greater for images that have been downscaled a lot. A comprehensive overview over the data is given in figures A.13 to A.19.

Altitude

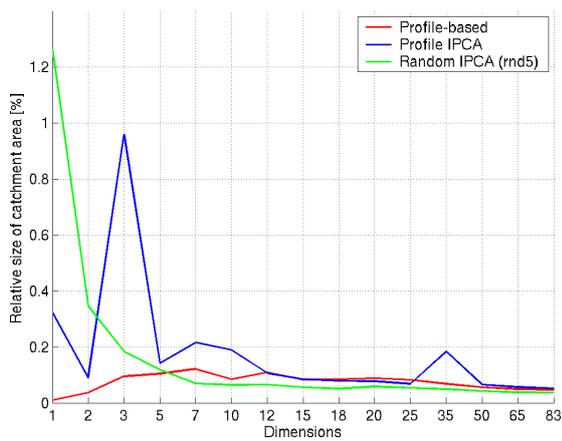
The results for the altitude look very similar as those for the orientation: Again stability increases for one to ten dimensions, then it keeps at a constant level or slightly decreases again depending on the selection. The range gets much smaller for images that were downscaled for more than one tenth of the original size. For the example plot of figure 3.8(b) the maximal scale factor is approximately 1.35 for the upper bound scale factor and 0.825 for the lower bound. The limits for the different levels of blurrings are again close together. An overview over the complete data is given in figures A.20 to A.26.



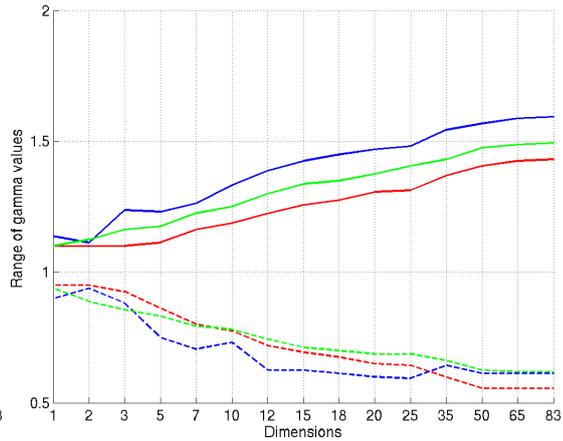
(a) Orientation



(b) Altitude



(c) Position



(d) Image brightness

Figure 3.8.: Representative plots for the reliability evaluation. Scale factor: 10^{-1} , blur-ring: gauss1, dimension for the profile-based ranking: 15.

Position

For presenting the results, the size of the catchment area was normalized by the total number of pixels in the floormap. So the plots show the relative size of the catchment area compared to the floormap size. At the first glance, the percentages seem to be rather small but one has to keep in mind that one view only covers approximately 2% of the floormap image and that landmarks are characterized by unique patterns that have to be visible and at approximately the same position of the view in order to recognize the place properly.

The sizes of the catchment areas seem to be bigger if the dimensions of the eigenspace are low. For more than ten dimensions, it decreases or keeps at a constantly low level. Although the images blurred stronger very often have larger catchment areas, they are often much larger if less than ten dimensions are used. Generally the percentages vary a lot for different selections and parameter combinations. Plots for position evaluation are given in figure 3.8(d) and in figures A.27 to A.33. If the dimensions are kept constant and the scale factor varies, differences between the several rankings show out: for the profile-based ranking and the IPCA-based ranking with profile-based start list, things look quite regular and predictable. The size of the catchment area grows with the level of smoothing and the level of down-scaling. For the randomly selected start lists, the relative sizes get rather unpredictable.

An extreme example is given in figure 3.9. There the views 15, 62 and 66 are surrounded by very large catchment areas and dominate the mean size. For more than ten dimensions, the images are represented more accurately and the sizes of the catchment areas are comparable to those of other landmark selections.

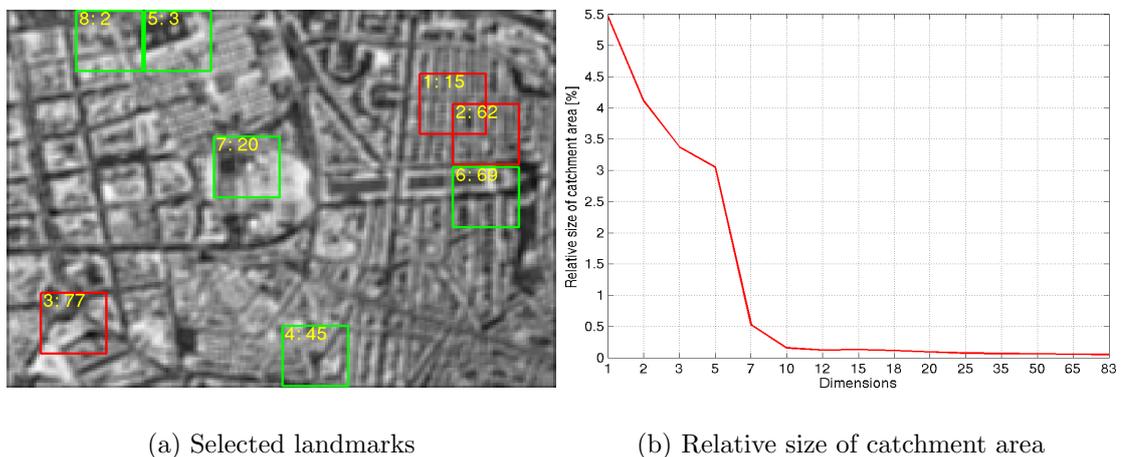


Figure 3.9.: Extreme example for the relative size of catchment areas. Scale factor: 8^{-1} , blurring: gauss1, start list for IPCA-based ranking: rnd5

Image brightness

The plots for the possible range of deviations of the gamma value presented in figure 3.8(d) and in figures A.34 to A.40 all show an approximately linear increase of stability with the number of dimensions. All the plots show that the more the views have been smoothed the less reliable they are. For more than 25 dimensions, the differences can get quite large. If the dimension is kept constant and the scale factor varies, it shows out that the range gets much smaller for images that were downsampled and smoothed a lot.

3.5.3. Conclusions

The results – especially those for the evaluation of image deviations with respect to errors in the blimp’s altitude and orientation – show that for achieving reliable localization at least ten dimensions are needed. The range between ten to 15 dimensions seems to be the optimal dimension of the eigenspace for the used setup. Also the optimal level of down-scaling seems to be around 8^{-1} to 12^{-1} . So the views can be compressed to approximately 38% of the originally view size, if we assume a scale factor of 12 and a 15-dimensional eigenspace. For the considered levels of Gaussian smoothing, the landmarks get more reliable if the images are smoothed more. Anyway, it is left for further work to find an optimal level of smoothing, because it seems that the used level “gauss3” is not some kind of threshold level.

The only parameter that does not produce optimal results in the mentioned range of scale factors and dimensions seems to be the image brightness. Therefore for real world experiments with changing light conditions, it is necessary to think about contrast or gray value normalization.

The relative size of the catchment area seems to give a good measure if a landmark is located in a region where surrounding views look very similar or not. In case it is, then the size of the catchment areas is very huge or even bigger than the size of the landmark, otherwise it is rather small. That results in a great variance of the data contributing to the mean size. In further work, we have to find a suitable method to solve that problem, e.g. by normalizing the data so that landmarks with a huge catchment area contribute more than landmarks with only a small area or vice versa. Another important aspect of further work is to find a suitable measure that combines all the parameters and therefore reducing the amount of data.

4. Failed approaches

In this chapter a short overview of approaches and experiments that did not lead to promising results shall be given. Namely, these are experiments that were done to select landmarks by clustering views in the eigenspace and experiments to measure a landmark’s reliability by computing and analyzing localization matrices.

4.1. Cluster-based landmark selection

We had the idea to cluster the points in the eigenspace using the clustering methods described in section 2.2 to detect landmarks. We thought that it might be a good approach to use clustering methods because the views within a neighborhood should be similar to each other but very dissimilar to views from other neighborhoods.

Unfortunately, the approach did not lead to promising results, so we only show the first experiments and their results that made us conclude that there are better approaches to the landmark selection problem.

4.1.1. Aim and description

For clustering the points the hierarchical method as well as the k-means algorithm have been used.

Hierarchical clustering

To find out how many clusters or landmarks should be used, we visualized the hierarchical clustering by plotting and analyzing dendrograms. Also we systematically computed the cophenetic correlation coefficient (MathWorks, 2002; Everitt et al., 2001). It measures the distortion of the dendrogram by comparing distances in the dendrogram and in the observation. Values close to unity signal a good correlation. The cophenetic correlation coefficient (CCC) is defined as

$$\text{CCC} = \frac{\sum_{i < j} (O_{i,j} - \bar{o}) (C_{i,j} - \bar{c})}{\sqrt{\sum_{i < j} (O_{i,j} - \bar{o})^2 (C_{i,j} - \bar{c})^2}} \quad (4.1)$$

where $O_{i,j}$ is the pairwise distance between two observations, $C_{i,j}$ is the pairwise distance of two observations in the dendrogram and \bar{o} , \bar{c} are the mean distances in the observations and the dendrogram respectively. Depending on the distance measure used to cluster the observations, $O_{i,j}$ is not the same as the corresponding entry of the distance Matrix D .

K-means clustering

Some fundamental experiments with the k-means clustering were to visualize some clusterings by marking those views that have been assigned to the same cluster.

The drawback of the k-means clustering is that you have to know the number of clusters in advance. Therefore we systematically tried each value from 1 to 83. As distance measure for all experiments with k-means clustering the sum of squared differences (SSD) was used.

To compare the validation of the different clusterings, the silhouette value was computed for each view. After Bolshakova and Azuaje (2003) the silhouette value (SIL) has a value between -1 and 1 and is a confidence indicator on the membership of the i -th observation of cluster $C_j; j = 1 \dots c$ and all the other observations contained in clusters $C_k; k = 1 \dots c; k \neq j$. It is defined as

$$\text{SIL}(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}} \quad (4.2)$$

where $a(i)$ is the average distance between the i -th observation and all other samples included in C_j and $b(i)$ is the minimum average distance between the i -th observation and all of the samples clustered in C_k . If $\text{SIL}(i)$ is close to 1 , one can assume that the i -th sample was assigned to the cluster correctly, if the value is close to -1 it indicates that the sample has been misclassified.

In order to find out the optimal parameter combination, we analyzed the mean and the median SIL values of all the views for a given number of clusters, scale factor, dimension and blurring. Additionally we computed the percentage of negative silhouette values.

4.1.2. Results

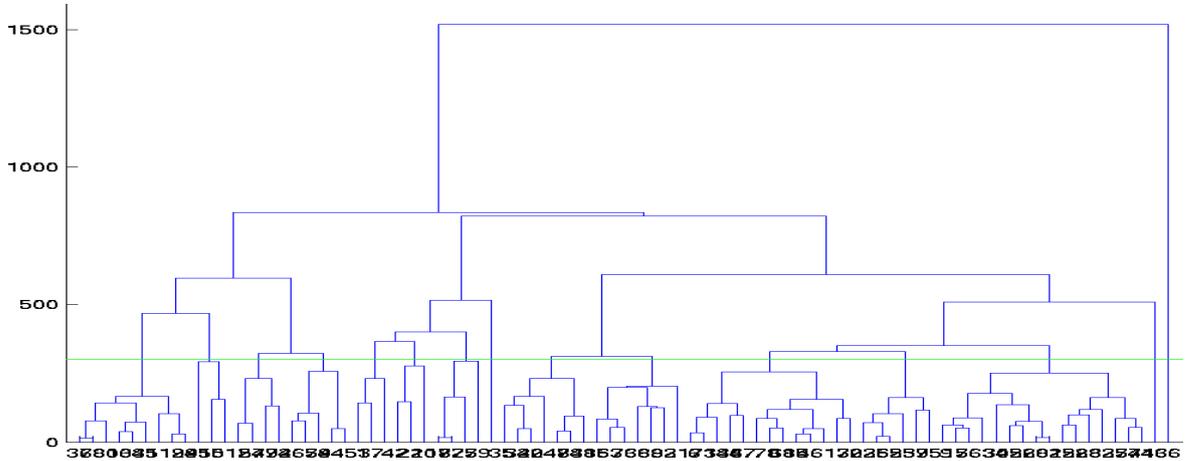
Since we only want to give a short overview why we think the clustering of the views did not lead to the results we expected, we do not show all the results and data. We only pick out those results that looked promising but made us decide to focus on other approaches.

Hierarchical clustering

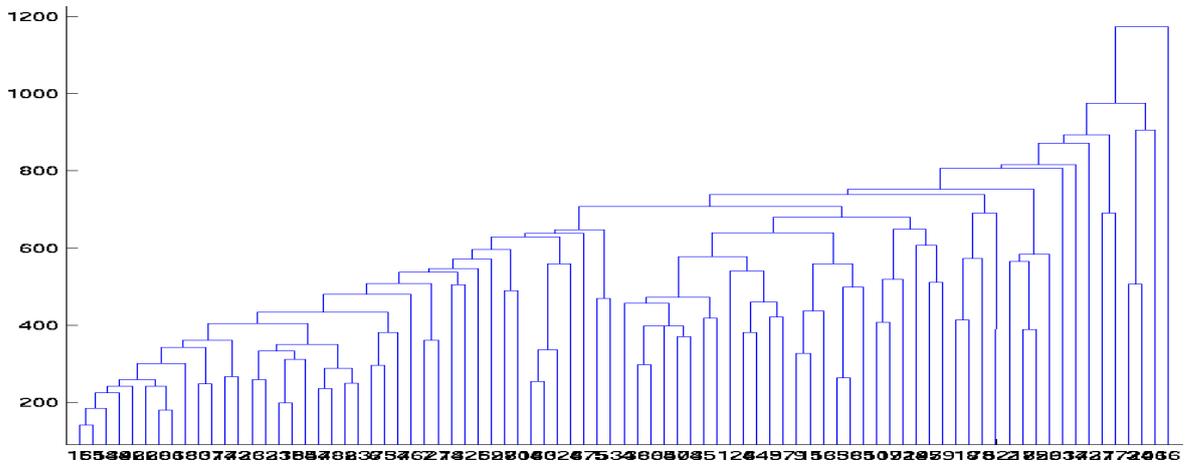
The systematical analysis of the CCC showed that the use of the Euclidean distance function to measure the distance between two observations and the average, centroid or single linkage method gave the best results. For this clustering parameters, the CCC is often in a range between 0.7 and 0.8 .

Some example dendrograms are shown in figure 4.1. The upper one shows a good dendrogram that can be used to cluster the views. There it is easy to set a cut-off value and to determine the number of clusters. In the example, 300 was chosen as cut-off value resulting in 15 clusters. Table 4.1 lists which views are within the same cluster, figures A.41 to A.43 visualize the selections. These plots show, that only sometimes

views of one neighborhood are within the same cluster. Often the clusters are very big but the views in the clusters are distributed all over the floormap. Sometimes the clusters look like they could easily be divided into two clusters, like e.g. cluster 11. Choosing a smaller cut-off value would result in smaller clusters, but also the amount of clusters that only contain a single view would raise.



(a) Example for a good dendrogram; scale factor: 6^{-1} , dimensions: 2, blurring: gauss3, CCC: 0.7173



(b) Example for a useless dendrogram; scale factor: 10^{-1} , dimensions: 10, blurring: gauss1, CCC: 0.7673

Figure 4.1.: Example dendrograms for clustering views. Clustering parameters for both dendrograms: Average linkage and Euclidean distance

If more dimensions of the eigenspace are used, the dendrograms look rather like figure 4.1(b). For these dendrograms it is very difficult to set an appropriate cut-off value, because the distance level at which the clusters are fused is more or less the

Table 4.1.: Clusters for a good dendrogram; scale factor: 6^{-1} , dimensions: 2, blurring: gauss3. The clusters are marked in figures A.41 to A.43.

cluster	views
1	5, 24, 32, 40, 47, 48, 83
2	16, 31, 38, 53, 69, 76, 82
3	12, 49, 57, 74
4	26, 44, 45, 65, 79
5	6, 7, 8, 13, 14, 15, 34, 46, 61, 67, 68, 70, 73
6	23, 25, 39, 55, 62, 75
7	9, 17, 22, 27, 28, 30, 33, 42, 52, 54, 56, 58, 60, 63, 71, 81
8	1, 4, 37
9	2, 20, 21
10	18, 59, 72, 77
11	10, 11, 19, 29, 35, 36, 64, 78, 80
12	41, 50, 51
13	43
14	3
15	66

same and the views within one cluster are very dissimilar to each other. So there would either be view very big clusters or many clusters containing only a single view.

K-means clustering

Figure 4.2 shows a scatter-plot of the points corresponding to views in a 2D eigenspace. The plot shows, that there are no obvious clusters in the data, the views are somehow close to the other views assigned to the same cluster, but other borders between the clusters also seem suitable.

A single silhouette plot is shown in figure 4.3(a). It shows for each cluster the SIL values of its members. It can be seen that there are only 5 observations with negative silhouette values. There are a couple of observations – including the negative ones – whose absolute silhouette value is close to 0, showing that these observations are approximately equally far away from the cluster’s centroid it was assigned to and its closest neighbor. There are only view observations with SIL values greater then 0.75 showing that the clustering is not too good. This plot is meant as a help to better understand what we have done by systematically computing the mean and the median SIL values for several parameter combinations.

A representative plot of this analysis is shown in figure 4.3(b). The analyzed values get worse if more dimensions of the eigenspace are used. There are several optimal parameter combinations, but there does not seem to be a range, for which the clustering leads to the best results. So the optimal combinations have to be found individually. For the shown example one such optimal parameter combination is the one used for

the shown plots. Of course the analyzed values get better, i.e. the mean and average SIL value are closer to unity and the percentage of negative values decreases to 0, if the number of clusters is increased and many clusters only contain one view.

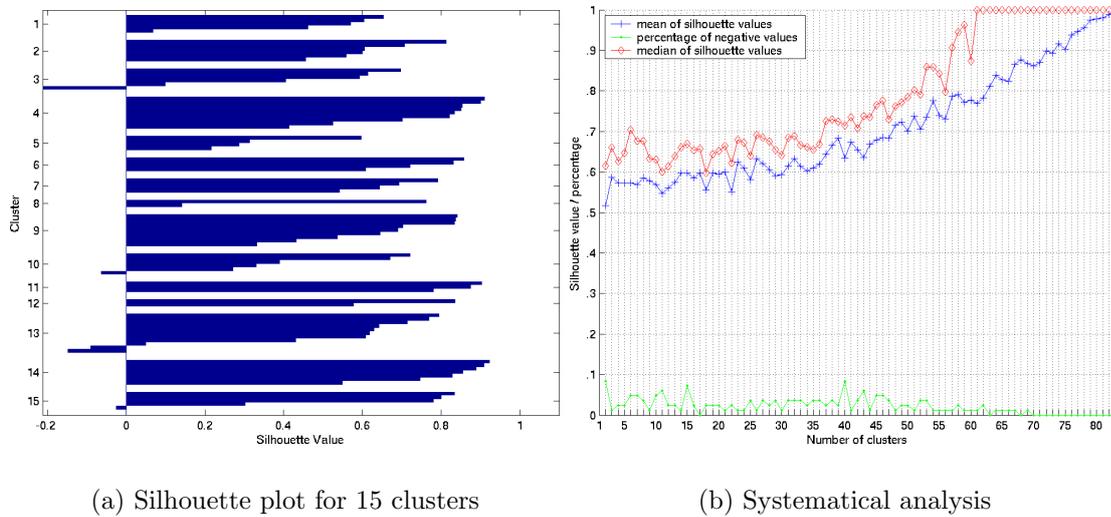


Figure 4.3.: Analysis of silhouette values; scale factor: 12^{-1} , dimensions: 2, blurring: gauss1

To round up the experiments related with k-means clustering, the clustering was performed on the views. The results for the sample parameter combination used in this section is shown in table 4.2 and is visualized in figures A.44 to A.46. The resulting partition has the same problems as the one based on hierarchical clusterings: Some clusters contain views that are distributed all over the floormap, some clusters look like they could easily be divided into to clusters looking more suitable and so on.

4.1.3. Conclusions

The most important conclusion is that there are no obvious or natural clusters in the views. So it is difficult to find an appropriate number of clusters. There also exists a trade-off between the number of clusters and the quality of clustering. If the number of clusters is kept small, the quality is low, i.e. there are many clusters that seem to be unreasonable and can easily be divided into two better clusters. On the other hand, if the number of clusters increases, there are many clusters that include only a single view.

Another problem showing out from the results is the difficulty to find an appropriate landmark representing the views that were assigned to a cluster. Since often the views of one cluster are distributed all over the floormap, using the centroid view is not an appropriate method. For example if the views in the upper left and the lower right corner are clustered together as well as the views in the lower left and the upper right,

Table 4.2.: Example k-means clustering; scale factor: 12^{-1} , dimensions: 2, blurring: gauss1. The clusters are marked in figures A.44 to A.46

cluster	views
1	7, 8, 15, 27, 46, 52, 54, 61, 68, 75
2	22, 28, 33, 39, 43, 45, 58
3	5, 24, 32, 40, 47, 48, 69, 82, 83
4	12, 26, 44, 49, 57, 65, 74, 79
5	13, 23, 25, 38, 55, 62
6	20
7	2, 3, 21
8	10, 11, 19, 29, 35, 36, 64, 78, 80
9	16, 31, 53, 76
10	41, 66
11	18, 59, 72, 77
12	6, 14, 30, 34, 67, 70, 73
13	9, 17, 42, 56, 60, 63, 71, 81
14	1, 4, 37
15	50, 51

both cluster's centroids are located at the image center resulting in two more or less identical landmarks that are both far away from the images they are representing.

Also the clustering methods are not selective for neighborhoods as we expected them to be. It is obvious that the clustering is done on PCA-coefficients and not on image coordinates and that views which are far away from each other in the floormap but which share a common pattern are projected to similar points in the eigenspace. But landmark selection for topological navigation is dependent on representing areas by characteristic AI landmarks. Therefore we conclude from the outcome of our experiments that clustering views is – at least for our setup – not an appropriate approach for the landmark selection problem.

Possible strategies to achieve a neighborhood selectivity by clustering could be the use of a special distance function that is based on image dissimilarity and on the landmark's distances in the floormap, which for us seems to be a valuable information to achieve region selectivity. Another approach could be to use fuzzy k-means clustering (Duda et al., 2001) so that these observations that cannot be assigned confidently are partially assigned to more clusters.

After considering the problems that were posed by the results we want to point out several conclusions. The hierarchical clustering iteratively merges the two most similar clusters. So the most distinctive landmarks are merged last. Therefore the hierarchical clustering is strongly related to the profile-based landmark selection, although in our case different distance measures have been used.

Another point to be mentioned is that the results seem to show that the views as possible landmarks are too dissimilar. Maybe clusters would show out and the ideas

would work if we would use more views with greater overlap. It seems that the clustering approach needs much more possible landmarks than the proposed algorithms. This is due to the fact that by clustering similar patterns in the image have to be found.

4.2. Localization matrices

If the image similarity between a certain view and every possible position of the floorplan is computed, a minimum will show out at the view's position. Around the view's position, the values will increase again and in an ideal case the minimum will be the only one. In the usual case – and especially if the landmark is not a very distinctive one – there will show out several other minima. In an extreme situation, the global minima could even appear at a position, that is not the view's position. The idea for an alternative approach to reliability evaluation was to compute these localization matrices and to analyze the shape, the number and several other parameters of the resulting peaks to characterize good and bad landmarks.

In order to measure properties of the peaks in the image, it is necessary to detect the peaks. Therefore we use an algorithm that is similar to the water shed algorithm invented by Vincent and Soille (1991). Our algorithm computes the position of the peaks, i.e. the local minimum, as well as the pixels belonging to a peak. The algorithm takes a localization matrix, sorts the entries in ascending order and iterates over every entry. Each entry is a possible peak. So the algorithm checks if the corresponding pixel is in a neighborhood region of another peak. If it is, then a neighborhood region around the point is marked as belonging to the peak. Therefore the pixel only contributes to the size of the peak. If the current pixel is not located in an already visited neighborhood region, then its position is added to the list of peaks and the pixels in its neighborhood are marked as belonging to the new peak. The algorithm stops if the maximal number of peaks is reached or if the dissimilarity values are above a threshold for which it can be assumed that they are not part of a peak.

4.2.1. Aim and description

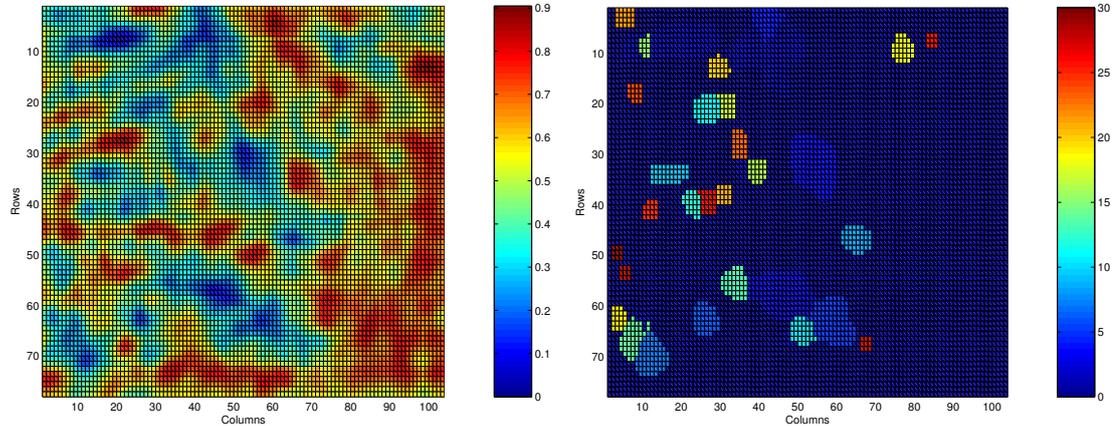
Since we decided not to follow up this approach at an early stage only fundamental experiments are shown. These include the visualization of localization matrices and some test runs for the algorithm described above.

4.2.2. Results

The localization matrices show several peaks as we expected. The variance of the similarity values and in correlation the number of local minima and maxima is greater if the images were downscaled a lot and smoothed a lot. An example localization matrix is shown in figure 4.4(a). The view for which the dissimilarities were computed is view number 20 which shows the institute's building approximately at the center of the floorplan. It seems like the most similar position was detected at the upper left

corner of the matrix. In this case a correct localization based on detecting the global minimum of the image dissimilarities is not successful.

Figure 4.4(b) shows the results of the peak analysis. The colder the color temperature is, the earlier the peaks have been detected. Since the figure is only supposed to be an example, no further results shall be pointed out.



(a) Localization matrix

(b) Peak matrix. „Blue“ shows peaks that have been detected first

Figure 4.4.: Example results for the analysis of localization matrices; scale factor: 15–1, dimension: 18, blurring: gauss1, view number 20

4.2.3. Conclusions

One big problem of this approach is the time needed for computations. So we decided to focus on the reliability measurements described in section 3.5. The trade–off that is visible for the localization matrices between down–scaling a lot and the capability for a robust localization is interesting for real world experiments.

The peak–detection algorithm seems to work but there are still some situations that should be handled better, e.g. at those pixels where regions of different peaks meet. Another aspect to think about is the further analysis of the data computed by the algorithm. A lot of things from the field of multivariate data analysis could be done to characterize the peaks and the landmark’s quality. Among these, we think the size of the regions, the slope of the peaks, the depth of the minima and the difference between the global and the local minima could be of interest. Anyway, one has to keep in mind that beside the needed computational times the analysis would also produce a huge amount of data so it would probably be necessary to find a combined measure for the landmark’s quality.

5. Final conclusions and further work

In this section the results of the work will be summarized and future working directions will be pointed out.

5.1. Final conclusions

The most conclusions have already been pointed out. So here only the main conclusions and the conclusions that can be drawn from all experiments will be mentioned.

The most important conclusion is that the proposed algorithms work and that both algorithms select distinctive landmarks. The landmarks often contain one characteristic feature like a certain pattern formed by the streets or an unique building. At least for the experiments that were done for this report it is not possible to differentiate when it is better to use the profile-based landmark selection algorithm and when it is better to use the IPCA-based one. The profile-based algorithm is a little bit more transparent, because it is not clear how the views contained in the start list of the IPCA-based algorithm influence the further selection. On the other hand the advantage of the IPCA-based algorithm is that it can be used for on-line landmark selection and therefore does not need an exploring phase before it is possible to select the landmarks like the profile-based algorithm.

The reliability evaluation showed that the selected landmarks were all stable over a huge range of image deviations which are due to simulated deviations in the blimp's pose or simulated changes in the light conditions.

A problem influencing the results presented in section 3 as well as the failed approaches presented in section 4 seems to be the chosen arrangement of the landmark candidates. The drawback of grid-like arrangement is that outstanding features of the floorplan can be located at the boarder of the view and not in the center like it would be the case if the views have been selected by an attention operator. The drawback of the small overlap of 25% is, that two neighbored landmarks are already very dissimilar. This is due to the observations that the clustering approach did not lead to good results and that there are only view very dissimilar and view almost identical landmarks. Especially for the clustering approach a huge amount of landmark candidates seems to be necessary in order to group views of similar neighborhoods. Although the chosen arrangement has drawbacks for the clustering approach and the proposed algorithms, it was sufficient to show that both algorithms are capable of selecting reliable landmarks. Anything else and especially a preselection of views that is optimal for the given floorplan is far beyond the scope of this report.

5.2. Further work

Several ideas for future work have already been proposed in the concluding sections of chapter 3 or in section 1.1 where we described how the presented work fits in the context of the RESCUE project. Here we want to point out the future working directions for the part dealing specifically with the appearance-based landmark selection.

If a human has a look at the floormap image shown in figure 5.1(a), the striking things are on the one hand the streets and patterns that are formed by the streets and on the other hand the color transitions between black streets and red roofs. In both cases, humans have to perform some kind of image preprocessing or segmentation. Since further work will be done at the department of Cognitive Neuroscience at the University of Tübingen¹, Germany, we will focus on biologically motivated algorithms for preprocessing or segmentation.

5.2.1. Color segmentation

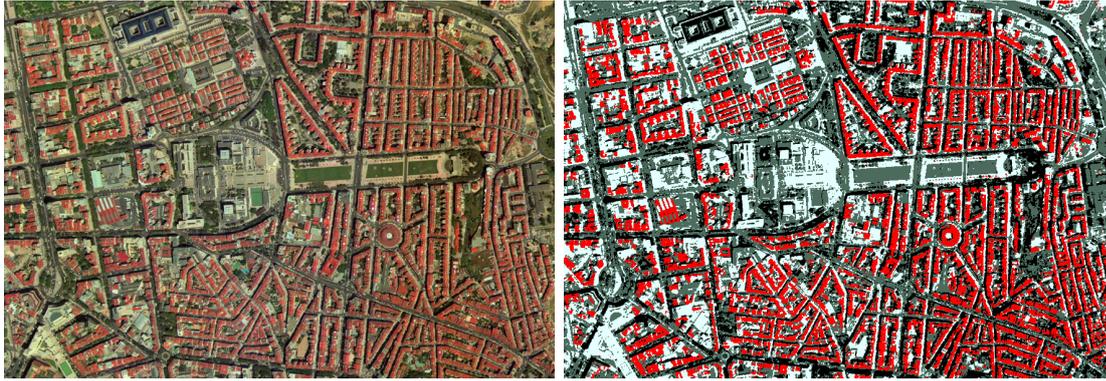
In Lucchese and Mitra (2003) color segmentation is defined as the partition of an image by determining disjoint and homogeneous regions or by determining boundaries between different regions. The authors also define properties for good segmentations: the segmented regions should be as simple as possible without having small holes in the interior. Adjacent regions should differ significantly with respect to the feature on which they are uniform. And finally the boundaries between regions should be simple, not ragged and should be spatially accurate.

In the last years a lot of algorithms were proposed for color segmentation. From the viewpoint of modeling human perception, the algorithms of Yoon et al. (2000) and Yoon and Kweon (2001) are especially interesting. In both approaches k-means clustering (see section 2.2.2) is performed in the CIE Lab color space. One advantage of the CIE Lab color space is that the Euclidean distance of two colors is proportional to the difference perceived by the human visual system. The algorithms also take into account that the human visual system has different sensitivities according to the spatial color pattern of the image. Therefore they compute weights according to the color variance in the neighborhood that is taken into account for clustering the color values. The output of the algorithms is a palette image where similar colored regions have been merged to bigger regions of one color.

The output should be comparable to the image shown in figure 5.1(b), which was created by simply reducing the number of colors. Using color segmentation hopefully reduces the number of small holes inside regions.

For further processing the palette values can be mapped to gray values in order to apply standard image processing methods. Since color segmentation produces homogeneous regions, edges will only appear at the borders of regions. So color segmentation can be an appropriate preprocessing for edge detection, because only important edges will be detected. Since in the floormap image edges most often appear between streets

¹<http://www.uni-tuebingen.de/cog>



(a) Colored floormap. See how transitions from red to dark gray correspond to transitions from buildings to streets.

(b) The floormap as palette image with four colors (black, dark gray, light gray and red). In a good segmentation approach small holes in the interior of regions should disappear.

Figure 5.1.: Floormap images to visualize ideas for color segmentation

and buildings, the palette value for streets and for roofs can be mapped to gray scales with maximal contrast which will lead to better responses of edge detecting filters. For example in figure 5.1(b) black and dark gray could be mapped to black and light gray and white could be mapped to white ensuring a maximal contrast between streets and houses.

Mapping color values back to gray values has also advantages for computing image similarities. Since palette indices can not be chosen so that color differences are proportional to the differences of the indices (which is due to the fact that it is not possible to build an order of color values by sorting them), the whole color information (usually one triple for each pixel) has to be used to compute image similarity. Computing the image similarity of gray scale images is much faster, because only each pixel is represented by only one value.

Especially for localization, when an image taken by the camera of the blimp has to be compared with the stored floormap, changes in brightness or contrast or noise can be a problem. Therefore performing color segmentation on the input image and mapping the palette color values to known gray scales can help to minimize the differences between the stored floormap and the input image. By segmenting a huge amount of images taken at different illumination conditions it should be possible to compute some average color values resulting from color segmentation of these images and to define a mapping from these color values to gray scales. Then it is possible to segment an input image, perform a nearest neighbor search between the known average color vectors and the segmented color vectors and finally do the mapping to gray scales.

5.2.2. Line segmentation

Another approach is to detect the streets in the gray scale image and to use the network or pattern formed by the streets instead of the pure image. In an ideal case, the algorithm for the line segmentation would output something like the blue lines in figure 5.2.



Figure 5.2.: Pattern formed by the streets

A lot of work related to biologically motivated edge detection and enhancement was published (Grigorescu et al., 2002; Kolesnik et al., 2002; Hugues et al., 2002; Thielscher et al., 2002). There are also some works modeling computational neural network models for line segmentation in the visual cortex of humans that seem to be more suitable to our problem. Here we will focus on the work of Li (1999) and Neumann and Sepp (1999).

Both methods assume oriented edge segments, how they can be computed using Gabor filters, as input stimulus. Again the change from red roofs to black or dark gray streets could be used for edge detection. Therefore it would be necessary to create a color sensitive edge detector or to use the gray value mapping described above.

Both networks enforce edge segments that form good line segments and inhibit edge segments that do not build line segments. In Li (1999) horizontal links between the neurons of the primary visual cortex V1 are modeled to achieve the inhibition and excitation. In Neumann and Sepp (1999) recurrent feedforward and feedbackward links between V1 and the secondary visual cortex V2 are modeled. The excitation pattern of the V1 neurons is compared to “curvature templates”, which are models of good line segments. In the feedback step, the neurons in V1 are inhibited or excited according to the correlation with the curvature templates.

In an ideal case, the proposed networks would compute some similar segmentation like the one shown in figure 5.2. A possible adaptation of the network proposed in Neumann and Sepp (1999) is to model crossroads or rotaries and to use these models as curvature templates. Then landmarks can be selected at image positions corresponding to the neurons that are excited most.

5.2.3. Preselection of landmarks

Another approach that seems worth thinking about is the preselection of possible landmarks. As the results show, taking possible landmarks at grid positions with only 25% of overlap between adjacent landmarks as proposed for this work has some drawbacks.

So one possibility would be to raise the overlap and therefore select more possible landmarks. This method is extremely expensive, because of the complexity of the proposed algorithms. In order to save computational costs it will be necessary to preselect good landmarks according to some criterion and later select only the best ones. As table 1.2 and 1.3 show a lot of proposed methods use attention operators to overcome this problem.

Attention or interest operators often detect low-level image features like edge density and orientation, contours or corners and therefore act like a spatial filter for these image features. There are a lot of standard attention operators that are reviewed in Sim et al. (2003) that can be applied for a large number of problems. But finding the best one or designing an operator that is most suitable for the detection of landmarks in the aerial image can be quite hard. The greatest problem (and the reason why we did not address one for this work) related with the use of attention operators is the stability of feature detection. The operator has to detect exactly the same features in the current camera image of the blimp and in the floormap image. To achieve this, a good image preprocessing reducing scene clutter and color variances is helpful or even necessary.

Another approach for designing an application specific attention operator would be to identify distinctive parts of the floormap or the street pattern, like for example specially shaped crossroads, and to match models of these in the floormap. Then the preselected landmarks are chosen at these places, where the matching resulted in a good correlation.

5.3. Final summary

In this work we present two algorithms for appearance-based landmark selection. Landmark selection algorithms try to optimize the navigation and localization abilities by selecting good landmarks. The algorithms, which will in future work be used to select landmarks for topological navigation of an autonomous blimp, are called “profile-based algorithm” and “IPCA-based algorithm”. The first algorithm evaluates image dissimilarities between possible landmarks represented by PCA-coefficients. The second one enlarges an existing eigenspace stepwise by adding that possible landmark, that can be represented worst in the existing eigenspace. Both algorithms tend to select those landmarks that are as dissimilar as possible to all other landmark candidates. To measure a landmark’s quality a reliability criterion was proposed. It uses the image dissimilarities between all the landmarks to compute an upper limit of image dissimilarity for which a correct localization to a given landmark is guaranteed. Additionally we presented the ideas for some approaches that did not lead to promising results.

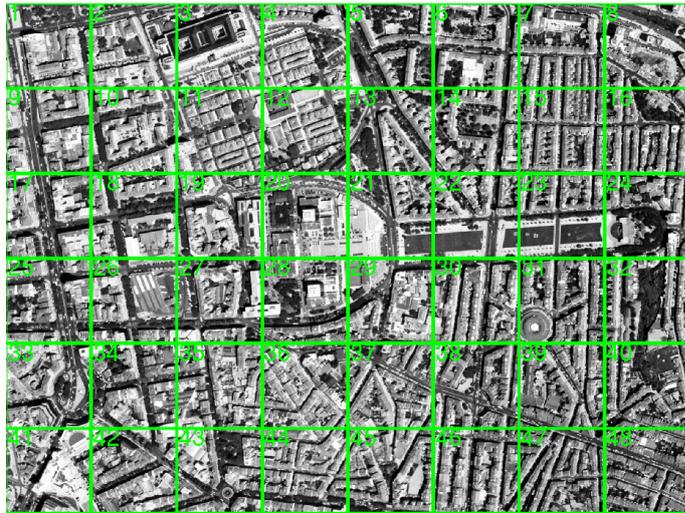
Namely this were ideas for selecting landmarks by clustering the possible landmarks and to globally evaluate the image dissimilarities between a landmark and the known map of the environment.

The results show, that none of the proposed algorithms is superior to another. In case a more than 10–dimensional eigenspace is used, all methods select landmarks that are robust over a large range of image deviations and that all have a very similar averaged reliability.

Since the proposed methods will be applied in large scale environments, there will be the need for discarding possible landmarks in order to speed up the analysis. Therefore we want to apply image preprocessing like color segmentation or contour grouping to bring out the underlying geometrical pattern. We think that these overall geometrical aspects of roads and buildings are the most relevant features for human observers. Another future working directory could be to implement a biologically inspired attention mechanism that is combined with the image preprocessing steps we plan to use.

A. Even more plots and data

A.1. View positions



(a) Position of the views 1–48



(b) Position of the views 49–83

Figure A.1.: Position of the views taken from the floormap

A.2. Quality of Recovered images

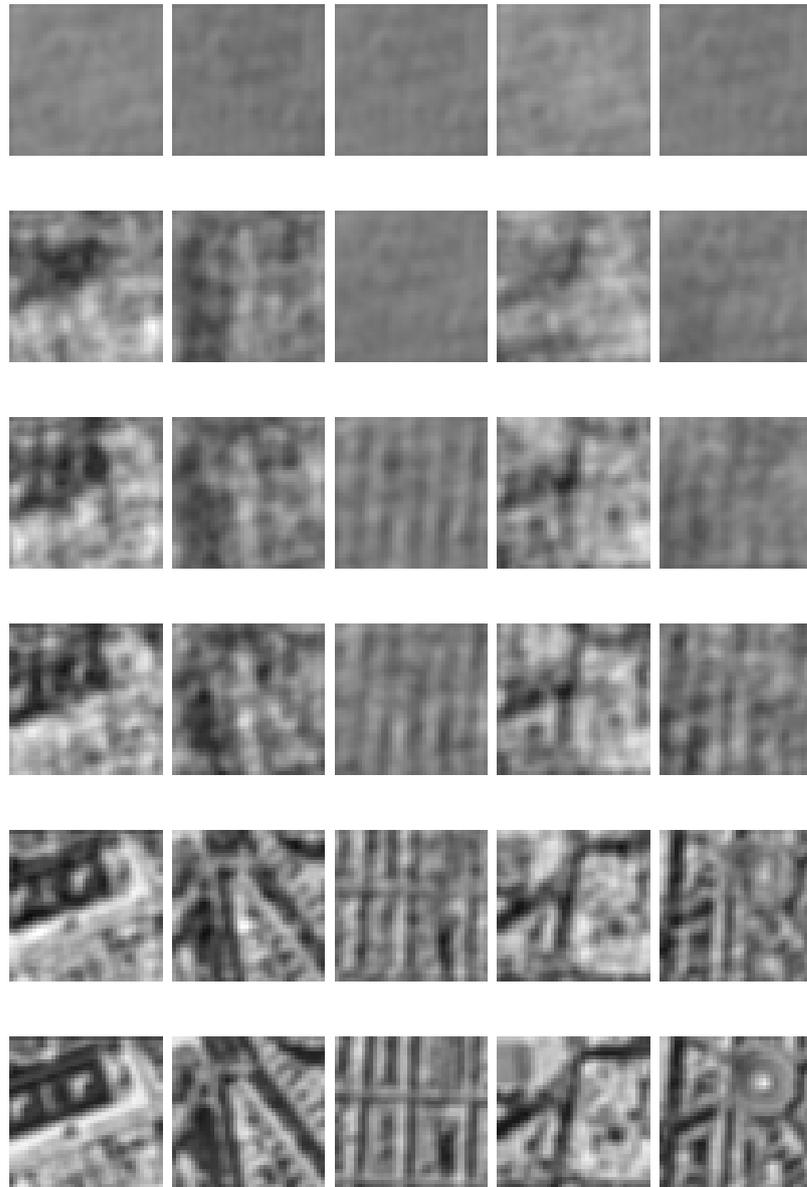
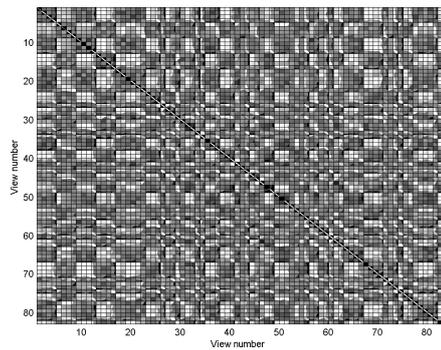
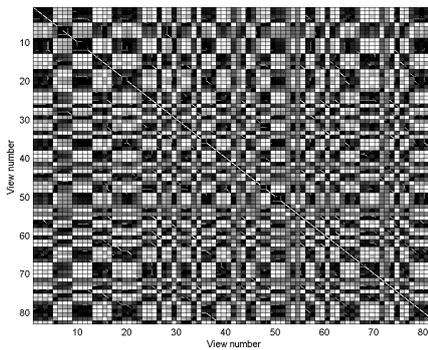


Figure A.2.: Quality of recovered images. The columns show the landmarks 3, 13, 15, 29, 75 transformed and recovered from an eigenspace with 1, 5, 20, 50 and 83 dimensions. Scale Factor: 10^{-1} , blurring: gauss1

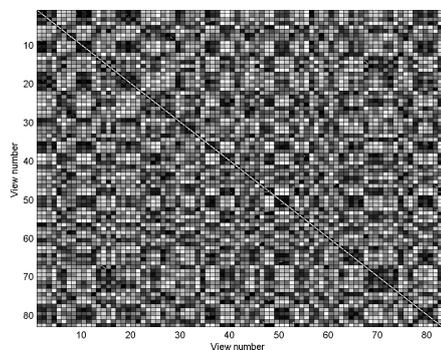
A.3. Distance Matrices



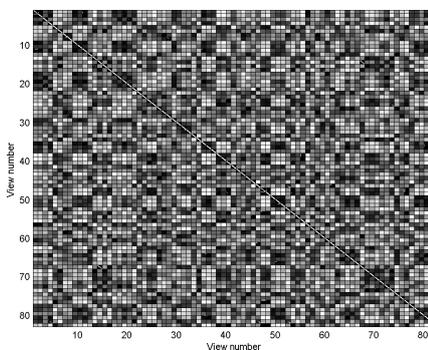
(a) Dim.: 1, blurr.: none



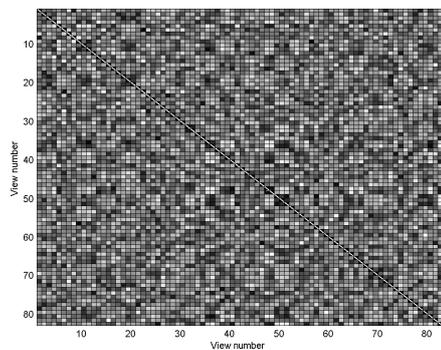
(b) Dim.: 1, blurr.:gauss3



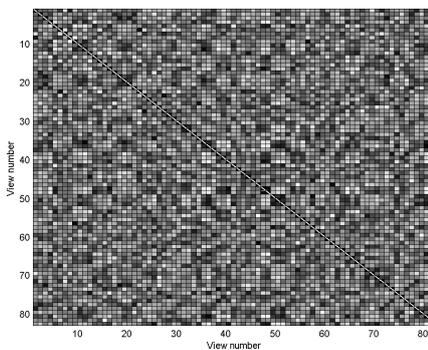
(c) Dim.: 3, blurr.: none



(d) Dim.: 3, blurr.:gauss3

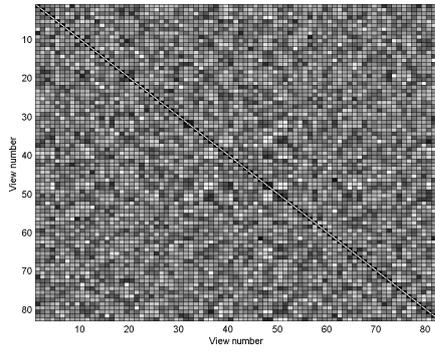


(e) Dim.: 7, blurr.: none

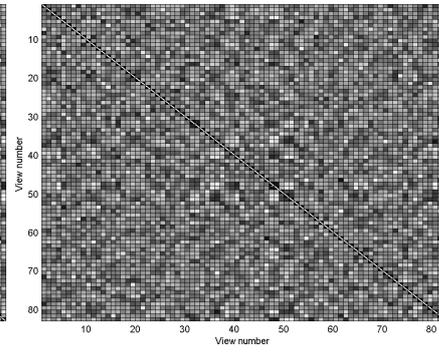


(f) Dim.: 7, blurr.:gauss3

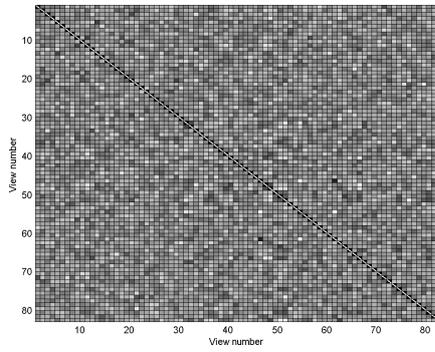
Figure A.3.: Distance matrices; scale factor: 1^{-1}



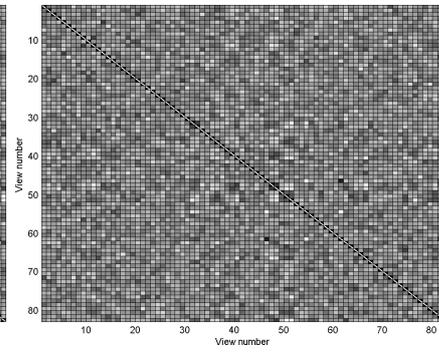
(a) Dim.: 12, blurr.: none



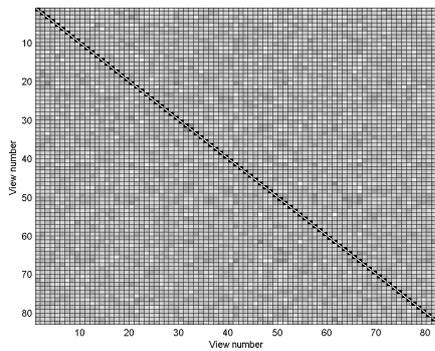
(b) Dim.: 12, blurr.:gauss3



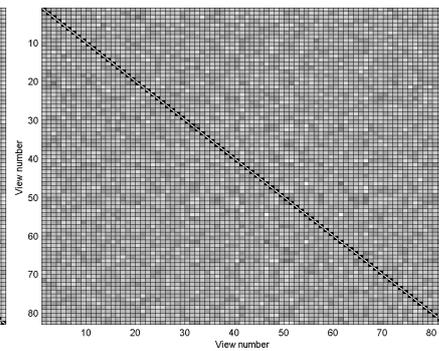
(c) Dim.: 25, blurr.: none



(d) Dim.: 25, blurr.:gauss3

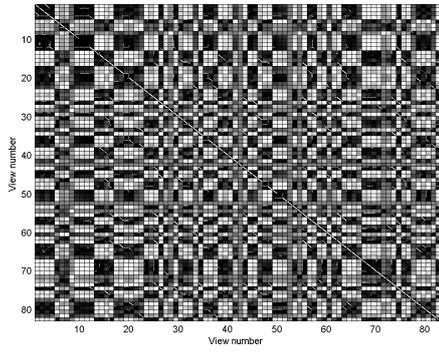


(e) Dim.: 83, blurr.: none

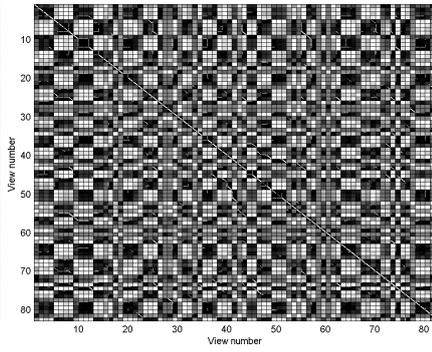


(f) Dim.: 83, blurr.:gauss3

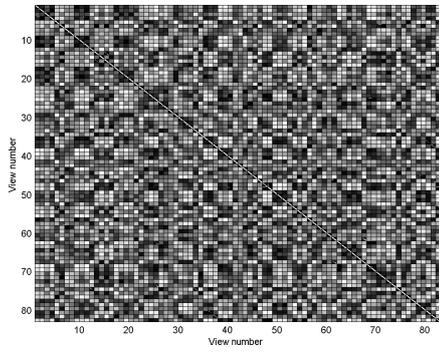
Figure A.4.: Distance matrices; scale factor: 1^{-1} continued



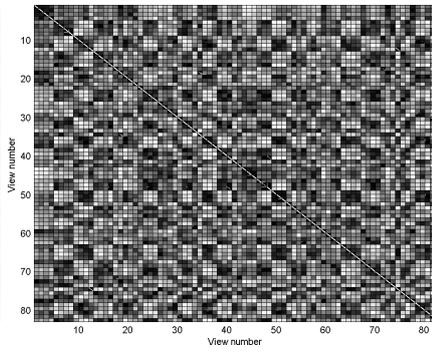
(a) Dim.: 1, blurr.: none



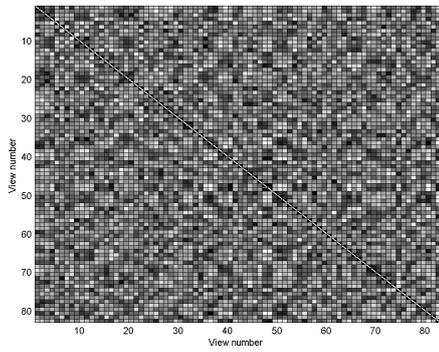
(b) Dim.: 1, blurr.:gauss3



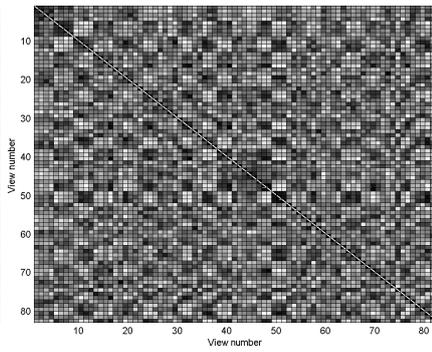
(c) Dim.: 3, blurr.: none



(d) Dim.: 3, blurr.:gauss3

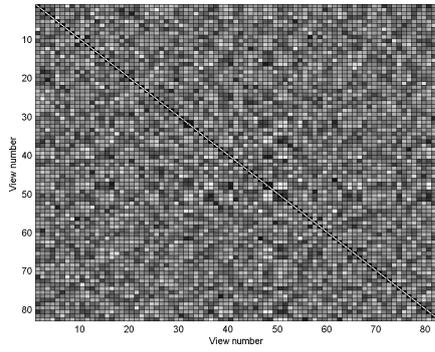


(e) Dim.: 7, blurr.: none

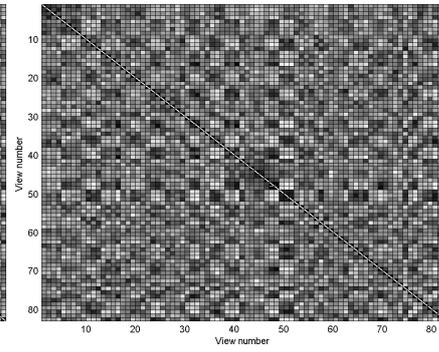


(f) Dim.: 7, blurr.:gauss3

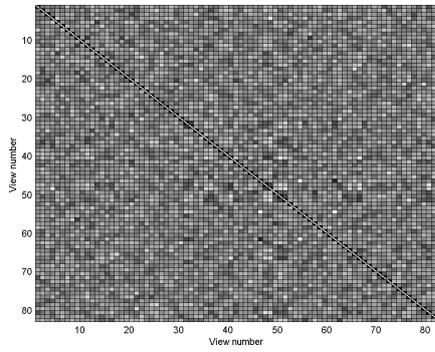
Figure A.5.: Distance matrices; scale factor: 10^{-1}



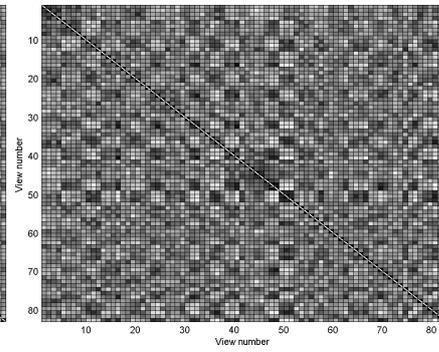
(a) Dim.: 12, blurr.: none



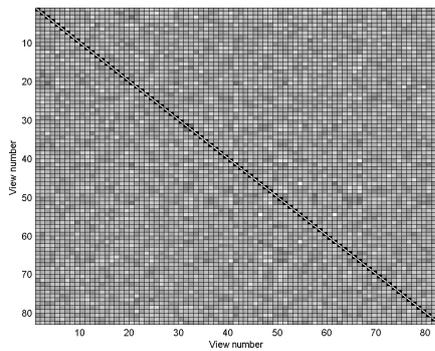
(b) Dim.: 12, blurr.:gauss3



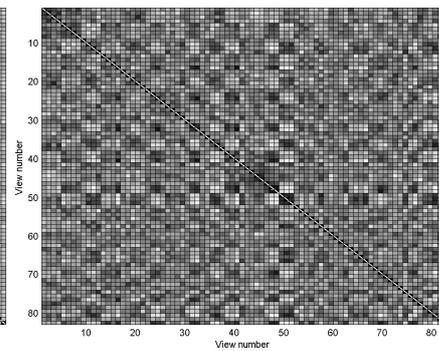
(c) Dim.: 25, blurr.: none



(d) Dim.: 25, blurr.:gauss3

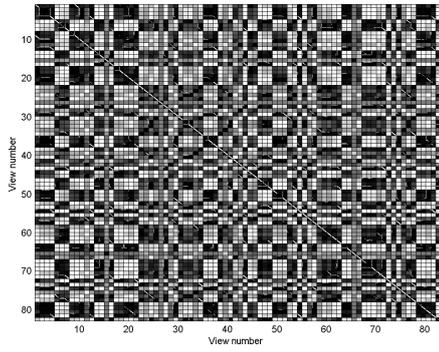


(e) Dim.: 83, blurr.: none

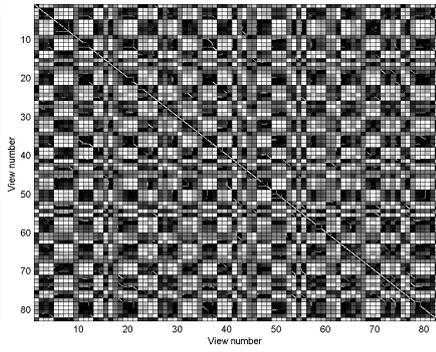


(f) Dim.: 83, blurr.:gauss3

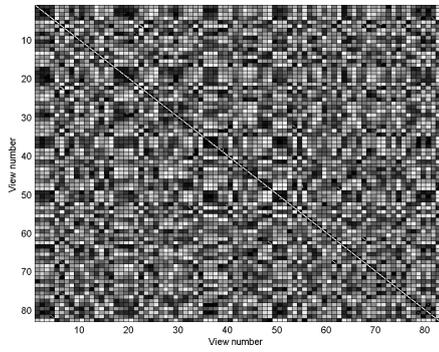
Figure A.6.: Distance matrices; scale factor: 10^{-1} continued



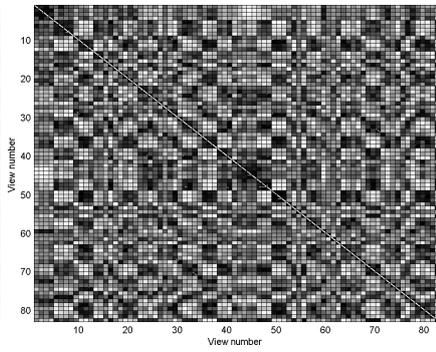
(a) Dim.: 1, blurr.: none



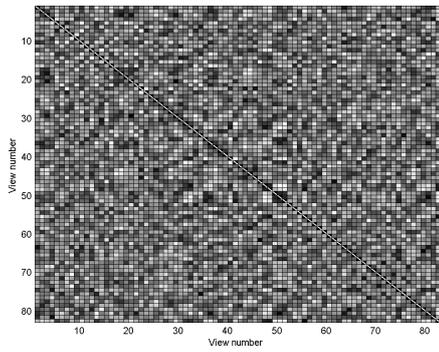
(b) Dim.: 1, blurr.:gauss3



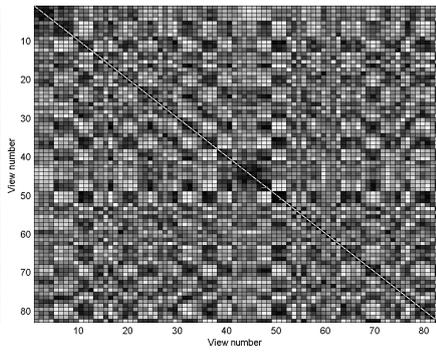
(c) Dim.: 3, blurr.: none



(d) Dim.: 3, blurr.:gauss3

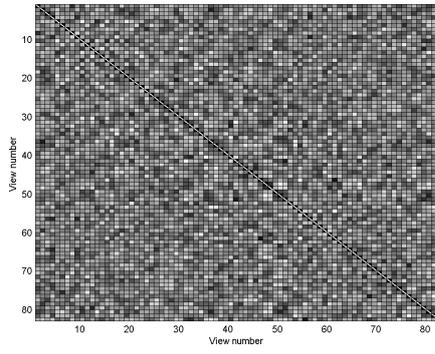


(e) Dim.: 7, blurr.: none

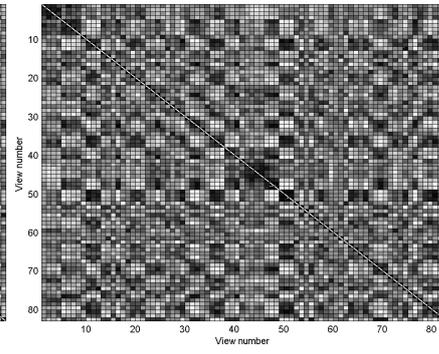


(f) Dim.: 7, blurr.:gauss3

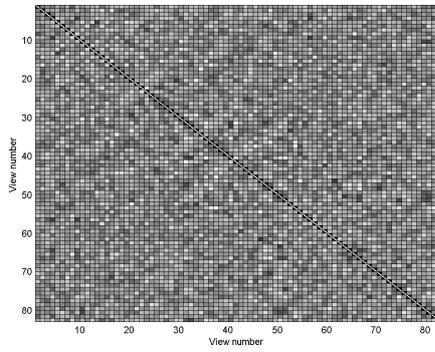
Figure A.7.: Distance matrices; scale factor: 20^{-1}



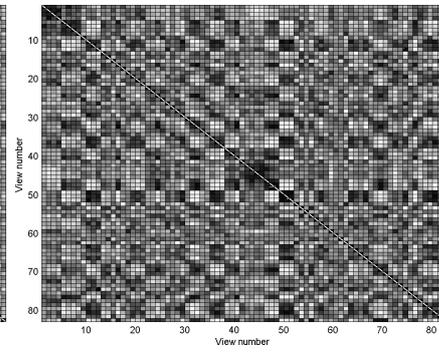
(a) Dim.: 12, blurr.: none



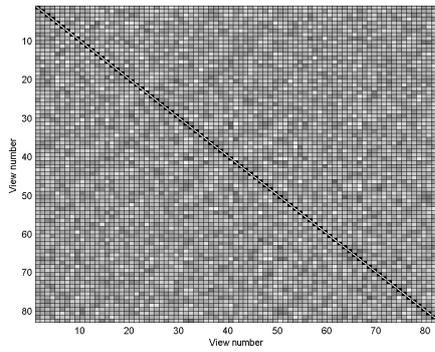
(b) Dim.: 12, blurr.:gauss3



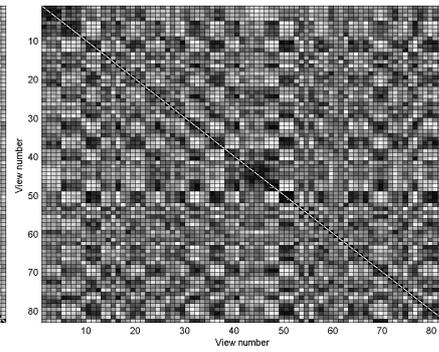
(c) Dim.: 25, blurr.: none



(d) Dim.: 25, blurr.:gauss3



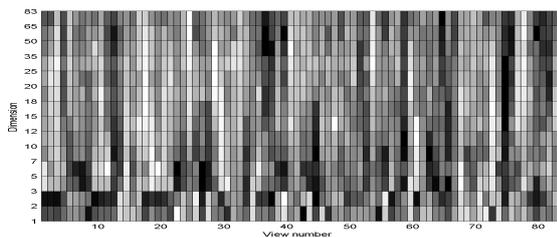
(e) Dim.: 83, blurr.: none



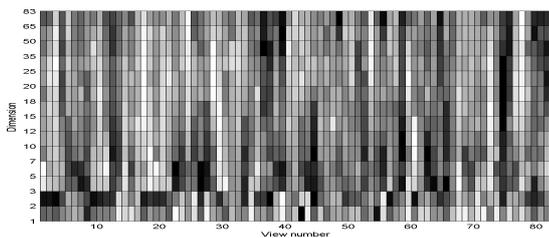
(f) Dim.: 83, blurr.:gauss3

Figure A.8.: Distance matrices; scale factor: 20^{-1} continued

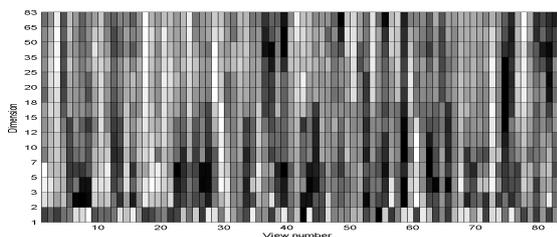
A.4. Profile Matrices



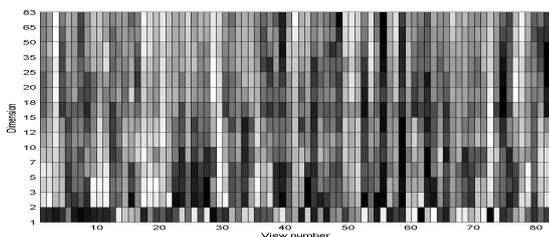
(a) Scale factor: 1^{-1}



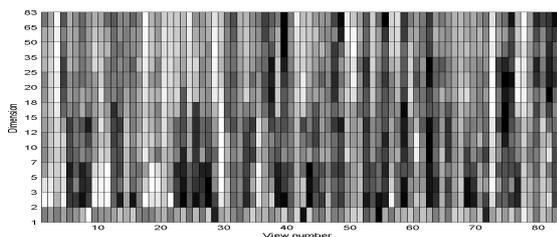
(b) Scale factor: 2^{-1}



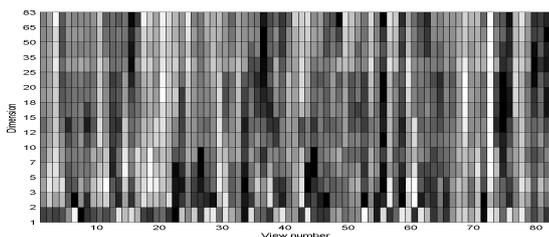
(c) Scale factor: 4^{-1}



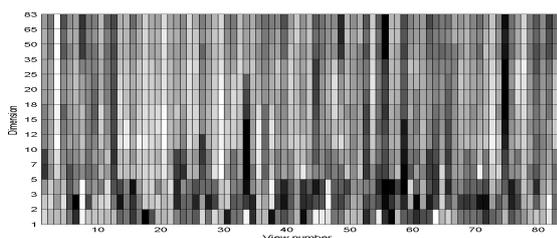
(d) Scale factor: 8^{-1}



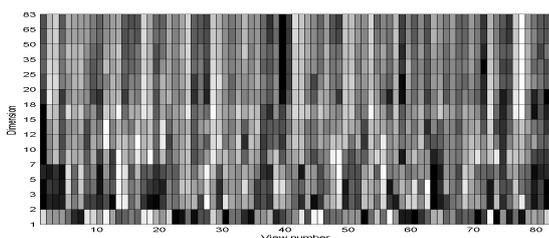
(e) Scale factor: 10^{-1}



(f) Scale factor: 12^{-1}

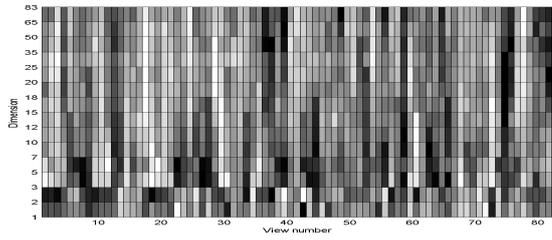


(g) Scale factor: 15^{-1}

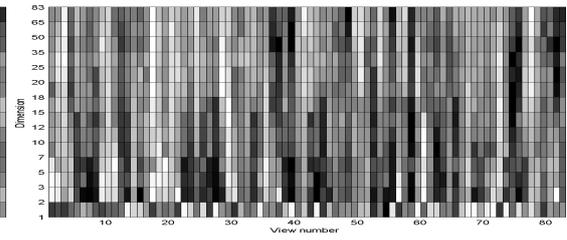


(h) Scale factor: 20^{-1}

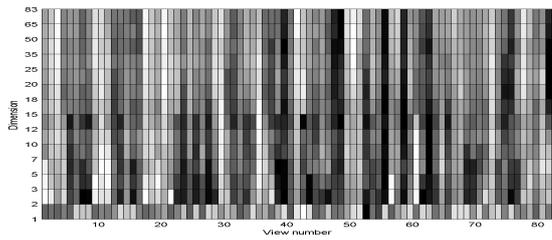
Figure A.9.: Profile matrices; blurring: none



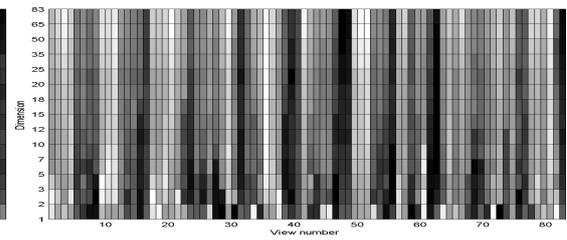
(a) Scale factor: 1^{-1}



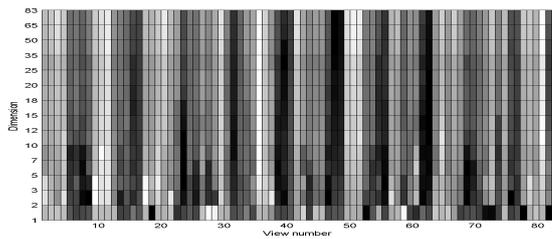
(b) Scale factor: 2^{-1}



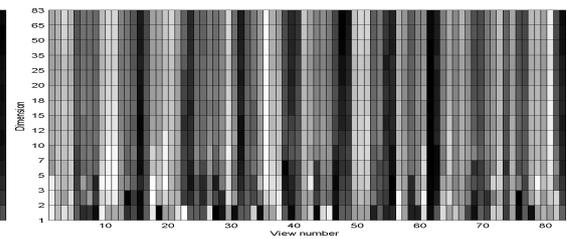
(c) Scale factor: 4^{-1}



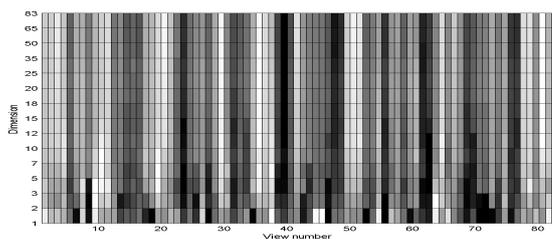
(d) Scale factor: 8^{-1}



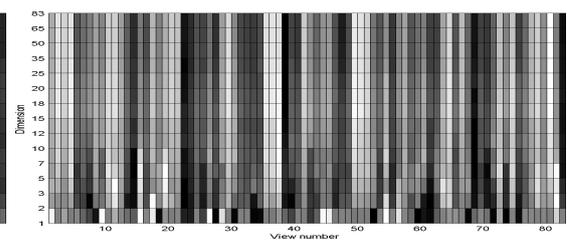
(e) Scale factor: 10^{-1}



(f) Scale factor: 12^{-1}

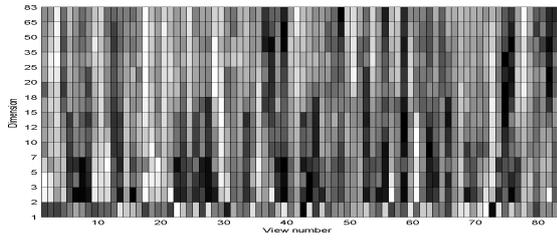


(g) Scale factor: 15^{-1}

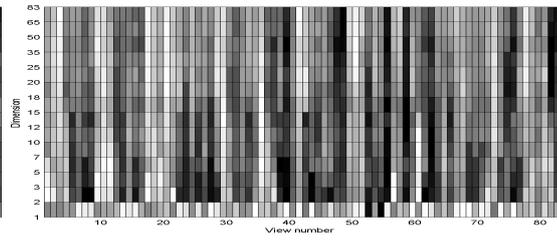


(h) Scale factor: 20^{-1}

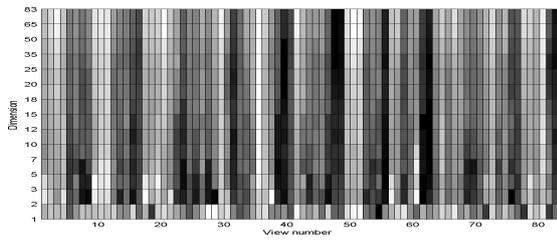
Figure A.10.: Profile matrices; blurring: gauss1



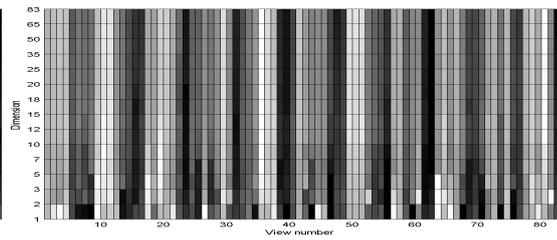
(a) Scale factor: 1^{-1}



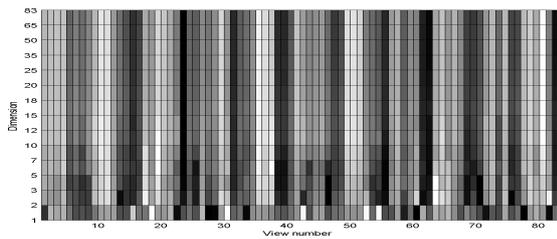
(b) Scale factor: 2^{-1}



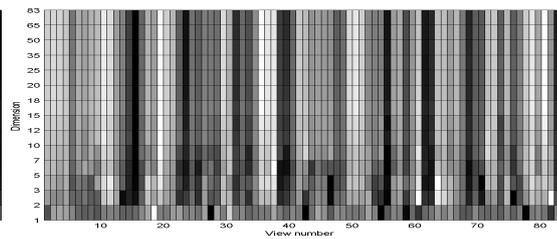
(c) Scale factor: 4^{-1}



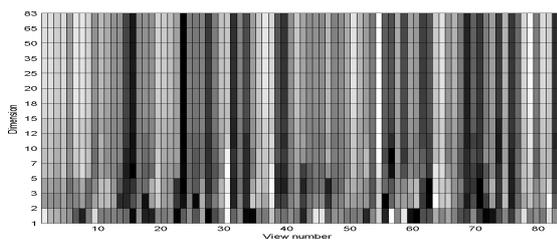
(d) Scale factor: 8^{-1}



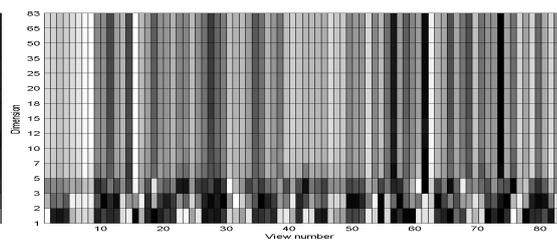
(e) Scale factor: 10^{-1}



(f) Scale factor: 12^{-1}



(g) Scale factor: 15^{-1}



(h) Scale factor: 20^{-1}

Figure A.11.: Profile matrices; blurring: gauss3

A.5. Landmark selection

A.5.1. Results of profile based ranking

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	39	68	27	59	34	73
2	39	25	55	70	13	59	67	43
3	67	30	18	72	59	1	14	29
5	11	19	67	18	35	72	14	60
7	18	29	60	72	3	20	11	17
10	60	72	29	18	16	17	14	20
12	47	60	29	17	16	72	18	76
15	72	17	29	24	76	67	10	77
18	17	28	10	72	29	59	77	76
20	17	28	59	77	21	2	53	76
25	28	53	17	3	24	77	40	72
35	40	28	17	53	24	32	9	77
50	24	77	17	53	3	40	5	18
65	24	77	53	32	16	17	5	40
83	24	53	77	3	32	17	41	69

(a) Scale factor: 1^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	27	39	34	68	59	43
2	39	55	25	70	13	34	43	59
3	67	18	30	72	1	29	59	14
5	1	19	11	67	35	18	72	29
7	18	60	29	11	3	72	17	20
10	60	29	72	18	17	20	10	14
12	29	60	17	72	16	47	20	18
15	72	29	17	10	24	77	76	20
18	17	28	77	10	59	29	72	67
20	17	28	59	77	53	21	3	76
25	28	3	17	53	77	72	24	32
35	28	17	53	40	24	77	3	9
50	77	24	17	3	53	18	20	41
65	77	24	53	17	41	3	5	32
83	24	3	77	17	53	41	32	69

(b) Scale factor: 2^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	8	43	34	27	15	75	39	61
2	60	81	56	17	1	37	30	21
3	18	29	72	1	67	19	2	30
5	1	19	11	35	18	72	29	60
7	60	29	18	17	3	11	72	10
10	60	29	72	20	18	17	10	3
12	29	17	72	20	10	60	3	35
15	29	72	17	20	77	10	2	24
18	28	17	77	42	20	59	3	50
20	28	17	42	20	72	77	3	59
25	3	28	17	72	77	53	21	24
35	17	28	3	53	50	9	24	77
50	77	17	3	24	20	41	53	49
65	77	17	41	3	24	53	50	57
83	3	17	77	41	24	53	28	57

(c) Scale factor: 4^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	63	60	71	52	13	55	67	73
2	17	2	30	11	21	9	19	18
3	18	19	29	35	9	10	11	78
5	29	78	1	11	72	35	2	18
7	29	17	72	10	80	81	3	64
10	72	29	20	3	17	35	10	18
12	29	72	20	3	10	17	49	35
15	72	3	29	20	76	28	35	77
18	28	76	3	72	20	50	77	29
20	28	76	3	50	29	77	72	20
25	28	3	76	53	20	11	17	72
35	3	17	28	53	77	18	50	49
50	3	77	41	20	28	17	50	57
65	3	41	77	17	20	28	50	57
83	3	41	17	77	57	24	53	20

(d) Scale factor: 8^{-1}

Table A.1.: Results for profile ranking; blurring: none

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	8	77	51	41	50	66	34	39
2	21	9	17	1	11	56	10	2
3	18	29	11	72	10	19	9	35
5	11	29	18	35	72	4	10	19
7	29	60	17	81	10	49	3	72
10	29	60	35	72	20	3	4	17
12	29	72	10	20	35	17	3	80
15	72	29	3	20	77	76	42	2
18	72	76	3	20	77	28	42	29
20	72	50	42	76	28	3	20	17
25	72	3	28	76	17	77	50	20
35	50	28	3	17	72	77	24	20
50	49	77	17	72	50	28	3	24
65	17	3	77	41	24	49	28	72
83	3	17	41	77	24	28	49	72

(a) Scale factor: 10^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	52	6	72	77	58	28	75	43
2	60	56	30	42	37	17	21	11
3	60	11	18	59	19	29	2	56
5	72	18	14	29	70	3	60	2
7	72	18	14	3	16	70	11	29
10	72	29	68	10	60	20	19	3
12	72	68	29	20	60	10	19	3
15	72	68	20	29	76	3	10	21
18	72	68	20	42	29	59	76	10
20	68	72	20	59	76	42	3	29
25	72	68	3	20	59	24	56	53
35	72	3	20	17	49	21	24	42
50	72	20	24	3	17	56	49	77
65	72	3	17	20	24	56	49	77
83	72	24	3	20	17	77	21	56

(b) Scale factor: 12^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	63	7	33	44	54	45	81	46
2	81	46	7	37	4	11	2	10
3	7	37	46	21	11	59	18	30
5	29	70	42	35	20	28	64	17
7	29	35	20	3	72	28	16	17
10	20	3	16	72	29	35	17	14
12	20	35	29	3	14	37	72	19
15	29	20	3	72	17	13	19	77
18	29	3	53	20	13	72	24	76
20	3	29	53	20	77	76	24	47
25	3	77	29	24	20	53	17	32
35	3	20	77	53	17	32	29	25
50	3	20	77	29	53	17	24	41
65	3	20	24	29	77	17	53	41
83	3	24	20	77	17	29	53	32

(c) Scale factor: 15^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	56	45	72	11	44	18	17	42
2	27	13	14	77	30	62	48	55
3	62	13	48	38	68	55	60	39
5	48	13	55	27	82	14	5	7
7	13	18	48	47	16	73	28	70
10	16	11	55	20	82	29	28	32
12	11	20	28	72	35	49	77	29
15	77	53	28	13	76	17	24	72
18	77	2	76	28	17	72	42	53
20	77	17	28	29	3	2	76	50
25	77	28	76	2	24	17	3	41
35	77	24	17	76	53	28	3	42
50	77	17	41	3	53	24	28	76
65	77	41	24	53	17	3	76	28
83	77	41	24	53	17	3	76	28

(d) Scale factor: 20^{-1}

Table A.2.: Results for profile ranking; blurring: none

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	39	68	59	27	45	73
2	42	39	34	70	55	25	13	73
3	18	67	1	30	72	29	60	14
5	11	19	35	18	72	60	67	29
7	18	60	29	11	72	3	20	17
10	60	72	29	18	17	20	10	14
12	29	60	17	72	16	47	20	3
15	72	29	17	24	10	20	77	76
18	28	17	77	59	29	10	76	35
20	28	17	59	77	3	2	21	50
25	28	17	3	53	77	72	24	2
35	17	28	53	40	9	24	3	77
50	77	17	24	3	53	20	41	18
65	77	24	17	41	53	3	28	57
83	24	3	17	77	41	53	32	69

(a) Scale factor: 1^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	71	68	39	22	59	27	67	34
2	60	56	17	21	81	1	9	37
3	1	17	18	60	29	9	19	56
5	19	11	35	1	18	60	72	29
7	60	29	18	11	17	10	72	3
10	60	29	72	18	20	10	17	3
12	29	17	72	60	20	10	3	35
15	29	72	17	20	10	35	2	24
18	28	17	77	10	20	41	59	29
20	28	17	50	3	77	72	2	29
25	17	3	28	72	77	53	2	50
35	17	28	3	50	9	20	53	77
50	17	3	77	20	41	24	50	49
65	17	77	41	3	24	50	57	53
83	17	3	41	77	24	53	20	28

(b) Scale factor: 2^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	42	41	66	51	59	39	67	68
2	17	1	21	60	56	9	2	37
3	9	56	11	19	1	35	17	60
5	10	35	11	60	1	29	80	19
7	60	10	29	35	11	17	18	72
10	35	60	29	72	20	3	10	18
12	35	29	10	20	17	3	72	60
15	35	29	20	72	3	17	10	2
18	20	42	10	77	17	3	41	35
20	20	77	50	3	42	41	17	10
25	50	3	17	20	35	9	72	77
35	50	3	20	17	9	2	77	35
50	20	50	3	17	41	77	49	29
65	41	3	17	20	50	77	29	9
83	3	41	20	17	50	77	29	9

(c) Scale factor: 4^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	71	33	18	81	60	45	63	37
2	17	37	1	9	21	60	11	4
3	9	35	19	10	11	78	17	56
5	35	10	60	11	19	80	29	36
7	35	29	10	11	80	60	51	19
10	35	80	51	11	4	49	10	29
12	35	51	80	29	11	20	10	4
15	35	11	51	80	20	29	50	10
18	35	11	51	80	29	20	50	3
20	35	11	51	80	29	50	20	3
25	35	50	51	11	20	80	3	29
35	35	51	11	50	20	3	29	80
50	51	35	20	50	29	3	80	41
65	51	20	50	35	3	41	29	80
83	51	20	50	3	35	41	29	80

(d) Scale factor: 8^{-1}

Table A.3.: Results for profile ranking; blurring: gauss1

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	27	58	28	42	63	65	21	44
2	17	9	37	21	10	11	4	1
3	9	17	19	35	11	10	78	1
5	35	10	11	80	19	64	60	36
7	35	10	80	11	64	19	29	4
10	35	80	51	64	11	49	4	10
12	35	80	11	10	29	51	49	50
15	35	80	11	10	50	51	29	49
18	35	80	11	50	51	10	29	49
20	35	80	11	50	29	51	10	20
25	35	80	11	50	51	20	10	49
35	35	80	50	11	51	29	20	49
50	80	35	50	11	51	20	29	41
65	80	35	50	51	11	20	29	41
83	80	35	50	51	11	20	41	29

(a) Scale factor: 10^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	56	45	37	1	9	17	26
2	17	56	60	11	1	37	10	29
3	19	10	11	17	78	9	35	1
5	35	11	10	19	80	60	29	36
7	35	10	11	19	29	80	50	51
10	35	80	19	11	10	29	51	64
12	35	80	11	10	51	29	50	49
15	35	80	11	10	29	50	3	19
18	35	80	11	10	29	50	51	3
20	35	80	11	10	50	29	20	51
25	35	80	11	10	50	29	20	3
35	35	80	10	11	50	29	20	3
50	35	80	10	11	29	20	50	3
65	35	80	10	11	29	20	50	3
83	35	80	10	20	29	11	50	3

(b) Scale factor: 12^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	37	45	44	81	54	63	9	1
2	63	9	11	81	37	19	10	29
3	9	19	10	35	78	11	7	37
5	19	35	37	10	29	80	4	9
7	10	19	29	35	80	64	4	37
10	35	80	29	64	10	51	37	4
12	80	35	29	20	51	50	19	78
15	80	35	29	50	20	64	36	51
18	80	35	29	20	50	3	78	51
20	35	80	29	20	50	3	78	51
25	35	80	29	20	50	3	51	78
35	80	35	29	20	50	3	51	78
50	80	29	35	20	50	3	51	78
65	80	29	35	20	50	3	51	78
83	80	29	35	20	50	3	51	78

(c) Scale factor: 15^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	54	15	26	44	9	1	45	17
2	81	17	11	19	54	64	78	15
3	11	19	81	63	15	37	78	64
5	19	81	37	15	10	4	80	72
7	37	35	49	80	19	4	10	36
10	37	49	80	35	4	64	36	2
12	37	49	80	35	64	4	50	36
15	80	37	49	4	35	50	2	36
18	80	37	4	49	36	35	50	64
20	80	49	37	4	36	64	50	35
25	80	49	4	50	36	35	37	2
35	80	49	4	50	36	2	35	37
50	80	49	4	50	36	2	35	37
65	80	49	4	50	36	2	35	37
83	80	49	4	50	36	2	35	37

(d) Scale factor: 20^{-1}

Table A.4.: Results for profile ranking; blurring: gauss1

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	68	39	59	67	27	73
2	60	56	17	81	21	1	37	9
3	1	17	60	9	18	29	19	56
5	11	19	35	60	18	72	10	29
7	60	18	29	11	10	35	72	3
10	60	29	72	18	20	17	10	3
12	29	17	60	72	20	3	10	35
15	29	17	72	20	10	35	24	2
18	28	17	77	35	29	41	10	20
20	28	17	50	3	2	29	77	72
25	17	3	28	77	2	53	72	50
35	17	28	9	3	50	20	77	53
50	17	3	77	20	41	24	50	49
65	17	77	41	3	24	50	53	57
83	17	3	41	77	24	53	20	28

(a) Scale factor: 1^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	68	43	39	34	59	67	75	27
2	17	21	60	1	56	9	2	37
3	9	56	19	11	17	1	35	60
5	35	11	10	60	29	19	80	1
7	60	10	35	29	11	18	17	72
10	60	35	29	72	18	20	10	3
12	35	29	10	20	17	3	72	41
15	35	29	20	72	3	17	10	2
18	35	20	10	17	77	29	3	72
20	20	50	17	77	42	3	10	41
25	50	17	3	35	20	9	77	72
35	50	20	3	9	17	2	28	77
50	20	3	50	41	17	77	9	29
65	41	3	17	20	50	77	29	10
83	41	3	17	20	77	50	29	10

(b) Scale factor: 2^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	27	51	28	50	41	66	80	58
2	17	37	60	56	9	1	4	21
3	9	19	35	11	17	10	78	56
5	35	60	11	10	80	29	19	36
7	35	60	80	11	29	10	51	36
10	35	80	51	11	10	64	4	29
12	35	80	29	10	51	11	20	64
15	35	11	20	80	51	29	10	50
18	35	80	20	11	10	50	29	51
20	35	11	80	50	29	20	10	51
25	35	50	80	20	11	29	3	10
35	35	50	80	20	51	11	3	29
50	35	50	20	51	3	80	29	41
65	50	20	35	51	41	3	29	80
83	20	50	35	51	3	41	29	80

(c) Scale factor: 4^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	63	3	26	9	65	56	44	37
2	63	81	17	9	11	60	29	64
3	63	17	9	35	19	10	11	78
5	35	19	10	80	9	11	37	17
7	35	10	19	80	11	29	49	17
10	35	80	10	19	49	11	51	29
12	80	35	10	49	11	19	51	29
15	80	35	11	10	49	29	50	36
18	35	80	11	49	10	29	36	51
20	35	80	10	49	11	29	36	50
25	35	80	10	11	49	50	29	36
35	35	80	10	11	49	50	29	36
50	80	35	10	11	50	49	29	51
65	80	35	10	11	50	49	29	51
83	80	35	10	11	50	49	29	51

(d) Scale factor: 8^{-1}

Table A.5.: Results for profile ranking; blurring: gauss3

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	30	18	52	54	42	15	21	2
2	63	17	81	64	11	29	19	10
3	63	19	35	11	17	36	10	78
5	19	37	35	80	11	10	17	50
7	19	35	10	80	17	11	37	50
10	80	19	35	37	10	49	50	36
12	80	35	10	19	49	11	37	50
15	80	35	10	49	50	11	36	29
18	80	35	10	50	49	36	11	29
20	80	35	10	50	49	11	36	29
25	80	35	10	50	11	49	36	19
35	80	35	10	50	49	11	36	19
50	80	35	10	50	49	11	36	19
65	80	35	10	50	49	11	36	29
83	80	35	10	50	49	11	36	29

(a) Scale factor: 10^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	18	30	3	44	37	17	26	4
2	81	60	63	64	11	37	19	10
3	63	11	19	37	60	35	78	10
5	37	19	78	35	80	60	63	30
7	19	37	60	78	80	10	71	35
10	19	10	37	80	35	71	78	36
12	19	80	35	37	10	36	50	78
15	80	35	19	37	10	78	50	36
18	80	19	35	10	37	36	78	50
20	80	19	35	10	37	36	78	50
25	80	35	19	10	37	36	78	50
35	80	19	35	10	37	78	36	50
50	80	19	35	10	37	78	36	50
65	80	19	35	10	37	78	36	50
83	80	19	35	10	37	78	36	50

(b) Scale factor: 12^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	56	30	1	58	44	45	77	9
2	63	81	30	37	54	65	19	64
3	37	30	63	19	81	64	35	78
5	30	37	63	54	64	7	78	19
7	54	30	78	7	64	19	6	36
10	54	78	37	64	6	30	36	7
12	54	78	6	36	64	50	7	2
15	54	78	6	7	36	64	50	1
18	54	78	6	36	7	64	50	1
20	54	78	6	36	7	64	50	2
25	54	78	6	36	7	64	50	2
35	54	78	6	36	7	64	50	2
50	54	78	6	36	7	64	50	2
65	54	78	6	36	7	64	50	2
83	54	78	6	36	7	64	50	2

(c) Scale factor: 15^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	75	1	23	22	14	52	60	42
2	31	53	7	8	5	16	76	61
3	30	60	15	67	18	62	8	77
5	15	7	62	8	81	52	6	78
7	15	7	62	8	81	6	52	46
10	15	8	7	62	6	52	46	5
12	8	15	7	62	6	46	5	1
15	8	15	7	62	6	46	1	5
18	8	15	7	62	6	46	1	5
20	8	15	7	62	6	46	1	5
25	8	15	7	62	6	46	1	5
35	8	15	7	62	6	46	1	5
50	8	15	7	62	6	46	1	5
65	8	15	7	62	6	46	1	5
83	8	15	7	62	6	46	1	5

(d) Scale factor: 20^{-1}

Table A.6.: Results for profile ranking; blurring: gauss3

A.5.2. Results of profile-based IPCA ranking

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	39	-	-	-	-	-
2	39	25	55	49	45	16	36	63
3	67	30	18	14	53	22	20	2
5	11	19	67	24	40	57	41	51
7	18	29	60	3	44	54	58	16
10	60	72	29	26	59	68	3	21
12	47	60	29	41	66	20	2	3
15	72	17	29	3	59	68	48	21
18	17	28	10	65	61	49	1	45
20	17	28	59	65	2	21	72	76
25	28	53	17	65	41	10	29	2
35	40	28	17	34	66	50	2	51
50	24	77	17	19	59	66	50	78
65	24	77	53	19	59	4	66	41
83	24	53	77	19	59	4	66	41

(a) Scale factor: 1^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	27	41	1	8	30	6
2	39	55	25	49	45	16	36	63
3	67	18	30	14	53	22	20	2
5	1	19	11	46	71	2	5	24
7	18	60	29	44	3	54	58	16
10	60	29	72	26	68	59	3	21
12	29	60	17	3	65	34	54	20
15	72	29	17	3	59	68	21	48
18	17	28	77	45	65	62	21	4
20	17	28	59	65	2	21	72	24
25	28	3	17	34	20	23	33	2
35	28	17	53	65	41	10	2	29
50	77	24	17	19	59	66	50	2
65	77	24	53	19	59	4	66	41
83	24	3	77	69	20	19	59	66

(b) Scale factor: 2^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	8	43	34	69	20	30	73	78
2	60	81	56	30	61	9	13	66
3	18	29	72	68	26	59	21	16
5	1	19	11	9	5	71	46	2
7	60	29	18	44	3	54	16	58
10	60	29	72	26	21	68	3	59
12	29	17	72	3	68	59	21	48
15	29	72	17	3	68	59	21	48
18	28	17	77	45	65	21	3	2
20	28	17	42	65	34	8	2	13
25	3	28	17	34	20	23	33	65
35	17	28	3	34	20	23	33	65
50	77	17	3	45	20	33	34	59
65	77	17	41	45	57	66	78	20
83	3	17	77	45	20	33	34	59

(c) Scale factor: 4^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	63	60	71	70	50	41	28	5
2	17	2	30	65	53	34	40	24
3	18	19	29	67	10	24	2	65
5	29	78	1	-	-	-	-	-
7	29	17	72	34	68	3	16	21
10	72	29	20	43	68	3	41	51
12	29	72	20	43	68	3	41	51
15	72	3	29	33	59	68	16	21
18	28	76	3	23	20	41	51	9
20	28	76	3	23	20	41	51	9
25	28	3	76	23	20	41	51	9
35	3	17	28	34	20	23	2	21
50	3	77	41	23	66	57	51	78
65	3	41	77	23	66	57	51	78
83	3	41	17	24	23	66	78	34

(d) Scale factor: 8^{-1}

Table A.7.: Results for IPCA-ranking. The startlist is determined by the best three landmarks selected with profile ranking; blurring: none

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	8	77	51	66	20	57	45	41
2	21	9	17	65	56	34	41	18
3	18	29	11	66	61	21	52	65
5	11	29	18	66	61	21	52	65
7	29	60	17	65	3	20	34	41
10	29	60	35	66	3	52	13	41
12	29	72	10	68	51	59	21	49
15	72	29	3	33	68	59	21	67
18	72	76	3	20	33	41	11	19
20	72	50	42	66	63	51	57	64
25	72	3	28	33	23	12	68	44
35	50	28	3	63	33	66	36	64
50	49	77	17	57	45	65	41	3
65	17	3	77	20	45	34	33	57
83	3	17	41	24	20	66	34	80

(a) Scale factor: 10^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	52	6	72	20	14	21	24	8
2	60	56	30	53	5	9	25	3
3	60	11	18	44	46	66	3	9
5	72	18	14	–	–	–	–	–
7	72	18	14	–	–	–	–	–
10	72	29	68	41	46	26	20	21
12	72	68	29	41	46	26	20	21
15	72	68	20	41	3	51	29	75
18	72	68	20	41	3	51	29	75
20	68	72	20	41	3	51	29	75
25	72	68	3	33	20	26	21	9
35	72	3	20	41	55	21	76	51
50	72	20	24	41	3	19	51	78
65	72	3	17	33	34	20	49	26
83	72	24	3	19	20	78	2	59

(b) Scale factor: 12^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	63	7	33	29	18	24	43	17
2	81	46	7	18	78	27	32	68
3	7	37	46	–	–	–	–	–
5	29	70	42	41	72	7	4	50
7	29	35	20	37	57	30	41	3
10	20	3	16	32	41	66	29	10
12	20	35	29	37	57	30	41	3
15	29	20	3	45	10	79	54	7
18	29	3	53	40	45	41	51	20
20	3	29	53	40	45	41	51	20
25	3	77	29	45	51	57	23	54
35	3	20	77	41	45	10	23	75
50	3	20	77	41	45	10	23	75
65	3	20	24	41	19	10	66	65
83	3	24	20	41	19	10	66	65

(c) Scale factor: 15^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	56	45	72	–	–	–	–	–
2	27	13	14	41	57	24	22	6
3	62	13	48	16	66	53	36	41
5	48	13	55	70	66	41	69	62
7	13	18	48	24	83	2	66	51
10	16	11	55	2	66	20	41	36
12	11	20	28	72	29	5	23	33
15	77	53	28	–	–	–	–	–
18	77	2	76	66	68	74	51	79
20	77	17	28	8	67	34	39	23
25	77	28	76	68	69	61	35	23
35	77	24	17	66	19	3	78	2
50	77	17	41	31	66	34	57	49
65	77	41	24	66	78	3	19	20
83	77	41	24	66	78	3	19	20

(d) Scale factor: 20^{-1}

Table A.8.: Results for IPCA-ranking. The startlist is determined by the best three landmarks selected with profile ranking; blurring: none

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	39	14	30	1	78	41
2	42	39	34	49	3	17	53	8
3	18	67	1	71	52	68	36	2
5	11	19	35	9	66	41	24	2
7	18	60	29	44	3	54	58	16
10	60	72	29	26	68	59	3	21
12	29	60	17	3	65	34	54	20
15	72	29	17	3	59	68	21	48
18	28	17	77	45	65	61	21	4
20	28	17	59	65	2	21	72	76
25	28	17	3	34	20	23	33	65
35	17	28	53	65	41	10	29	2
50	77	17	24	19	59	66	50	78
65	77	24	17	19	59	66	50	78
83	24	3	17	20	66	19	78	41

(a) Scale factor: 1^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	71	68	39	75	46	41	78	1
2	60	56	17	65	34	9	43	50
3	1	17	18	71	34	52	65	66
5	19	11	35	9	66	41	24	55
7	60	29	18	44	52	3	54	16
10	60	29	72	21	54	68	59	3
12	29	17	72	3	59	68	21	48
15	29	72	17	3	59	68	21	48
18	28	17	77	45	65	21	3	4
20	28	17	50	65	66	63	36	64
25	17	3	28	34	20	23	65	33
35	17	28	3	34	20	23	65	33
50	17	3	77	45	20	33	34	59
65	17	77	41	45	66	57	9	78
83	17	3	41	23	24	66	57	12

(b) Scale factor: 2^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	42	41	66	40	53	59	5	24
2	17	1	21	71	34	13	6	78
3	9	56	11	63	8	20	59	10
5	10	35	11	9	18	30	73	70
7	60	10	29	70	21	52	54	3
10	35	60	29	66	52	3	21	33
12	35	29	10	66	33	52	37	9
15	35	29	20	33	52	66	21	37
18	20	42	10	3	43	5	29	18
20	20	77	50	66	63	57	51	45
25	50	3	17	66	63	65	33	34
35	50	3	20	63	66	33	17	12
50	20	50	3	63	66	33	17	12
65	41	3	17	24	23	66	20	12
83	3	41	20	66	51	12	57	80

(c) Scale factor: 4^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	71	33	18	3	20	41	1	66
2	17	37	1	65	70	58	34	12
3	9	35	19	10	24	28	26	63
5	35	10	60	66	41	9	3	20
7	35	29	10	52	33	21	66	9
10	35	80	51	21	76	72	69	57
12	35	51	80	21	76	72	69	57
15	35	11	51	53	72	66	28	79
18	35	11	51	53	72	66	28	79
20	35	11	51	53	72	66	28	79
25	35	50	51	72	70	71	63	41
35	35	51	11	53	72	66	28	79
50	51	35	20	33	12	66	52	3
65	51	20	50	3	66	33	12	21
83	51	20	50	3	66	33	12	21

(d) Scale factor: 8^{-1}

Table A.9.: Results for IPCA-ranking. The startlist is determined by the best three landmarks selected with profile ranking; blurring: gauss1

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	27	58	28	21	67	41	20	17
2	17	9	37	65	52	1	59	63
3	9	17	19	65	56	24	34	10
5	35	10	11	9	66	71	28	17
7	35	10	80	21	66	9	11	2
10	35	80	51	21	76	72	69	79
12	35	80	11	28	66	78	17	1
15	35	80	11	28	66	78	17	1
18	35	80	11	28	66	78	17	1
20	35	80	11	28	66	78	17	1
25	35	80	11	28	66	78	17	1
35	35	80	50	72	70	71	40	63
50	80	35	50	72	70	71	40	63
65	80	35	50	72	70	71	40	63
83	80	35	50	72	70	71	40	63

(a) Scale factor: 10^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	56	45	77	59	21	8	25
2	17	56	60	65	58	70	34	9
3	19	10	11	55	9	18	65	28
5	35	11	10	66	9	17	65	50
7	35	10	11	66	9	17	65	50
10	35	80	19	21	28	69	64	66
12	35	80	11	66	69	17	1	21
15	35	80	11	66	69	17	1	21
18	35	80	11	66	69	17	1	21
20	35	80	11	66	69	17	1	21
25	35	80	11	66	69	17	1	21
35	35	80	10	21	66	2	65	64
50	35	80	10	21	66	2	65	64
65	35	80	10	21	66	2	65	64
83	35	80	10	21	66	2	65	64

(b) Scale factor: 12^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	37	45	44	2	72	3	77	18
2	63	9	11	1	21	56	51	66
3	9	19	10	32	16	18	76	63
5	19	35	37	5	16	41	66	29
7	10	19	29	63	52	18	41	20
10	35	80	29	21	66	2	52	58
12	80	35	29	21	66	2	52	58
15	80	35	29	21	66	2	52	58
18	80	35	29	21	66	2	52	58
20	35	80	29	21	66	2	52	58
25	35	80	29	21	66	2	52	58
35	80	35	29	21	66	2	52	58
50	80	29	35	21	66	2	52	58
65	80	29	35	21	66	2	52	58
83	80	29	35	21	66	2	52	58

(c) Scale factor: 15^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	54	15	26	44	2	3	62	59
2	81	17	11	62	18	58	21	52
3	11	19	81	16	8	18	20	3
5	19	81	37	43	3	20	17	77
7	37	35	49	1	76	6	59	18
10	37	49	80	32	76	66	21	36
12	37	49	80	32	76	66	21	36
15	80	37	49	32	76	66	21	36
18	80	37	4	76	66	41	78	12
20	80	49	37	32	76	66	21	36
25	80	49	4	28	41	21	78	52
35	80	49	4	28	41	21	78	52
50	80	49	4	28	41	21	78	52
65	80	49	4	28	41	21	78	52
83	80	49	4	28	41	21	78	52

(d) Scale factor: 20^{-1}

Table A.10.: Results for IPCA-ranking. The startlist is determined by the best three landmarks selected with profile ranking; blurring: gauss1

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	22	71	68	14	57	30	41	1
2	60	56	17	65	34	9	43	50
3	1	17	60	71	34	13	52	31
5	11	19	35	9	66	41	24	55
7	60	18	29	44	52	3	54	21
10	60	29	72	21	54	68	59	3
12	29	17	60	65	3	34	20	54
15	29	17	72	3	59	68	21	20
18	28	17	77	45	65	61	21	3
20	28	17	50	65	66	63	36	64
25	17	3	28	34	20	23	33	65
35	17	28	9	65	34	56	64	18
50	17	3	77	45	20	33	34	59
65	17	77	41	45	66	57	9	51
83	17	3	41	53	23	66	57	12

(a) Scale factor: 1^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	68	43	39	69	75	20	48	14
2	17	21	60	65	75	34	59	3
3	9	56	19	10	77	41	42	24
5	35	11	10	9	18	73	70	28
7	60	10	35	66	9	3	41	13
10	60	35	29	66	21	3	52	33
12	35	29	10	66	33	9	21	37
15	35	29	20	33	66	52	21	37
18	35	20	10	33	43	3	9	52
20	20	50	17	3	66	65	72	34
25	50	17	3	66	63	65	34	33
35	50	20	3	63	66	33	17	12
50	20	3	50	63	66	33	17	12
65	41	3	17	53	23	66	12	57
83	41	3	17	53	23	66	12	57

(b) Scale factor: 2^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	27	51	28	41	66	36	64	20
2	17	37	60	65	34	52	13	8
3	9	19	35	24	10	78	28	17
5	35	60	11	66	41	3	28	78
7	35	60	80	66	41	50	63	78
10	35	80	51	21	76	69	72	57
12	35	80	29	21	66	52	58	8
15	35	11	20	33	52	12	3	66
18	35	80	20	33	21	66	3	12
20	35	11	80	78	28	66	17	71
25	35	50	80	72	70	71	63	59
35	35	50	80	72	70	71	63	59
50	35	50	20	33	66	3	63	21
65	50	20	35	33	66	3	63	21
83	20	50	35	33	66	3	63	21

(c) Scale factor: 4^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	63	3	26	12	33	20	77	24
2	63	81	17	21	24	65	52	58
3	63	17	9	65	56	24	42	70
5	35	19	10	28	55	9	24	79
7	35	10	19	28	55	9	24	79
10	35	80	10	21	2	66	36	50
12	80	35	10	21	2	66	36	50
15	80	35	11	66	28	50	1	36
18	35	80	11	66	28	50	1	36
20	35	80	10	21	2	66	36	50
25	35	80	10	21	2	66	36	50
35	35	80	10	21	2	66	36	50
50	80	35	10	21	2	66	36	50
65	80	35	10	21	2	66	36	50
83	80	35	10	21	2	66	36	50

(d) Scale factor: 8^{-1}

Table A.11.: Results for IPCA-ranking. The startlist is determined by the best three landmarks selected with profile ranking; blurring: gauss3

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	30	18	52	20	3	53	44	25
2	63	17	81	24	21	32	52	65
3	63	19	35	28	66	41	9	1
5	19	37	35	5	66	41	40	29
7	19	35	10	28	79	55	21	66
10	80	19	35	28	21	66	64	50
12	80	35	10	21	66	2	4	50
15	80	35	10	21	66	2	4	50
18	80	35	10	21	66	2	4	50
20	80	35	10	21	66	2	4	50
25	80	35	10	21	66	2	4	50
35	80	35	10	21	66	2	4	50
50	80	35	10	21	66	2	4	50
65	80	35	10	21	66	2	4	50
83	80	35	10	21	66	2	4	50

(a) Scale factor: 10^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	18	30	3	48	43	20	52	1
2	81	60	63	24	21	41	32	66
3	63	11	19	21	1	18	67	56
5	37	19	78	32	41	24	66	29
7	19	37	60	32	24	41	66	20
10	19	10	37	5	41	21	29	66
12	19	80	35	21	66	4	64	50
15	80	35	19	21	66	4	64	50
18	80	19	35	21	66	4	64	50
20	80	19	35	21	66	4	64	50
25	80	35	19	21	66	4	64	50
35	80	19	35	21	66	4	64	50
50	80	19	35	21	66	4	64	50
65	80	19	35	21	66	4	64	50
83	80	19	35	21	66	4	64	50

(b) Scale factor: 12^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	56	30	1	28	3	58	2	21
2	63	81	30	21	59	24	41	1
3	37	30	63	24	21	67	66	41
5	30	37	63	24	21	67	66	41
7	54	30	78	32	66	41	51	3
10	54	78	37	32	24	66	41	51
12	54	78	6	66	51	41	12	49
15	54	78	6	66	51	41	12	49
18	54	78	6	66	51	41	12	49
20	54	78	6	66	51	41	12	49
25	54	78	6	66	51	41	12	49
35	54	78	6	66	51	41	12	49
50	54	78	6	66	51	41	12	49
65	54	78	6	66	51	41	12	49
83	54	78	6	66	51	41	12	49

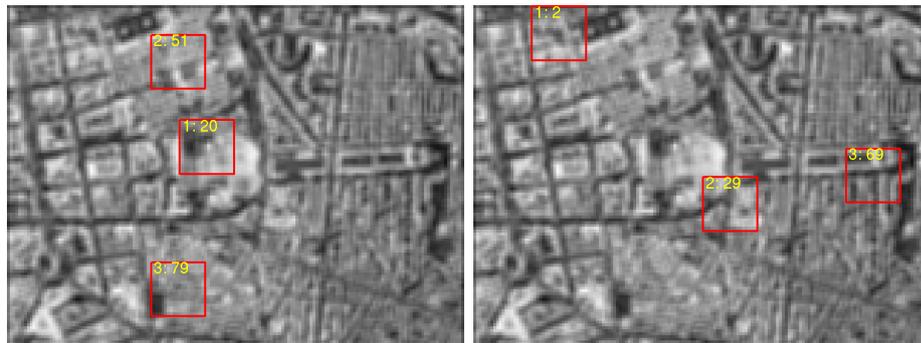
(c) Scale factor: 15^{-1}

Dim.	Rank							
	1	2	3	4	5	6	7	8
1	75	1	23	44	3	2	6	43
2	31	53	7	41	66	20	3	1
3	30	60	15	44	2	3	21	4
5	15	7	62	20	51	66	41	29
7	15	7	62	20	51	66	41	29
10	15	8	7	66	51	29	10	79
12	8	15	7	66	51	29	10	79
15	8	15	7	66	51	29	10	79
18	8	15	7	66	51	29	10	79
20	8	15	7	66	51	29	10	79
25	8	15	7	66	51	29	10	79
35	8	15	7	66	51	29	10	79
50	8	15	7	66	51	29	10	79
65	8	15	7	66	51	29	10	79
83	8	15	7	66	51	29	10	79

(d) Scale factor: 20^{-1}

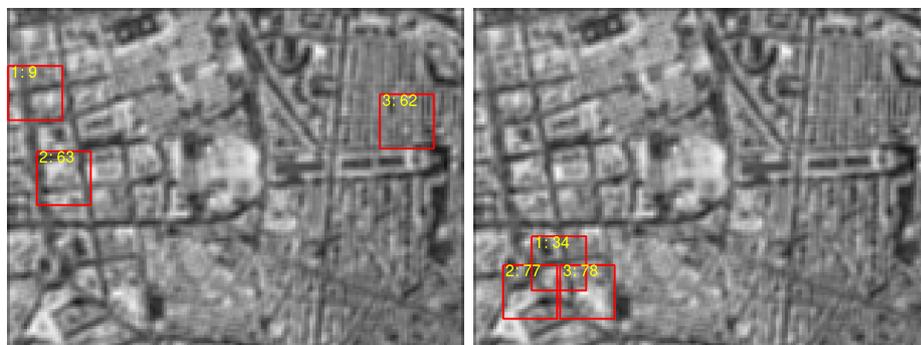
Table A.12.: Results for IPCA-ranking. The startlist is determined by the best three landmarks selected with profile ranking; blurring: gauss3

A.5.3. Start lists for IPCA ranking with random start list



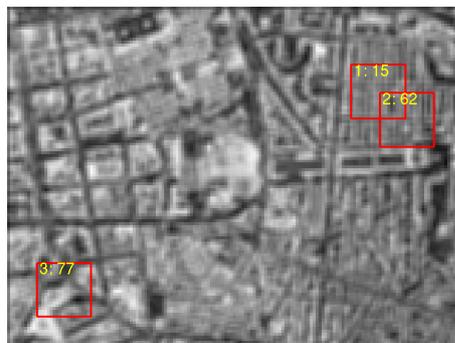
(a) Method 1: Views 20, 51, 79

(b) Method 2: Views 2, 29, 69



(c) Method 3: Views 9, 36, 62

(d) Method 4: Views 34, 77, 78



(e) Method 5: Views 15, 62, 77

Figure A.12.: Visualization of the used start lists for IPCA-based ranking

A.5.4. Results of IPCA ranking with random startlist

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	66	43	64
2	2	29	69	41	17	51	1	21
3	9	36	62	50	2	41	4	63
4	34	77	78	69	45	17	41	51
5	15	62	77	45	69	24	4	3

(a) Scale factor: 1^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	43	66	64
2	2	29	69	41	17	21	1	64
3	9	36	62	50	2	63	4	41
4	34	77	78	45	69	17	41	51
5	15	62	77	45	69	24	4	3

(b) Scale factor: 2^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	43	66	64
2	2	29	69	41	21	17	64	1
3	9	36	62	63	50	2	41	4
4	34	77	78	45	41	17	12	51
5	15	62	77	45	69	24	3	4

(c) Scale factor: 4^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	43	64	72	66
2	2	29	69	41	21	19	50	51
3	9	36	62	50	2	41	66	18
4	34	77	78	41	45	36	51	17
5	15	62	77	45	69	31	3	57

(d) Scale factor: 8^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	63	64	41
2	2	29	69	41	21	18	59	50
3	9	36	62	57	56	50	41	66
4	34	77	78	45	51	12	17	41
5	15	62	77	45	69	3	31	59

(e) Scale factor: 10^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	55	43	50
2	2	29	69	41	64	66	20	46
3	9	36	62	57	41	66	50	18
4	34	77	78	24	41	12	45	17
5	15	62	77	45	57	69	3	7

(f) Scale factor: 12^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	77	40	41	72
2	2	29	69	80	18	17	41	67
3	9	36	62	31	50	4	41	49
4	34	77	78	32	54	45	41	29
5	15	62	77	-	-	-	-	-

(g) Scale factor: 15^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	6	72	3	23	12
2	2	29	69	46	20	77	3	19
3	9	36	62	56	66	41	29	49
4	34	77	78	45	80	41	17	24
5	15	62	77	57	72	3	31	7

(h) Scale factor: 20^{-1}

Table A.13.: Results for IPCA-ranking. The startlist is determined by randomly choosing three landmarks; blurring: none

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	43	66	64
2	2	29	69	41	17	21	1	51
3	9	36	62	2	50	63	41	4
4	34	77	78	69	45	17	41	51
5	15	62	77	45	69	24	4	3

(a) Scale factor: 1^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	43	66	64
2	2	29	69	41	17	21	1	51
3	9	36	62	63	2	50	51	41
4	34	77	78	45	69	41	17	51
5	15	62	77	45	69	24	4	3

(b) Scale factor: 2^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	43	66	72	12
2	2	29	69	41	21	17	50	51
3	9	36	62	48	2	50	63	41
4	34	77	78	45	41	17	12	51
5	15	62	77	45	69	57	3	4

(c) Scale factor: 4^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	43	3	66	12	64
2	2	29	69	41	21	80	50	66
3	9	36	62	48	63	50	66	41
4	34	77	78	41	45	66	36	51
5	15	62	77	45	3	69	20	2

(d) Scale factor: 8^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	43	66	12	64
2	2	29	69	41	21	20	51	80
3	9	36	62	48	41	66	51	50
4	34	77	78	41	45	12	66	21
5	15	62	77	45	3	20	4	19

(e) Scale factor: 10^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	43	66	12	64
2	2	29	69	48	41	20	66	51
3	9	36	62	–	–	–	–	–
4	34	77	78	41	45	12	21	66
5	15	62	77	45	3	20	2	21

(f) Scale factor: 12^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	40	21	66	12
2	2	29	69	24	41	20	66	51
3	9	36	62	24	2	66	51	50
4	34	77	78	41	66	45	21	12
5	15	62	77	45	3	20	21	79

(g) Scale factor: 15^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	32	3	66	21	12
2	2	29	69	48	66	20	21	36
3	9	36	62	66	41	51	80	20
4	34	77	78	41	21	3	80	66
5	15	62	77	41	3	20	51	29

(h) Scale factor: 20^{-1}

Table A.14.: Results for IPCA-ranking. The startlist is determined by randomly choosing three landmarks; blurring: gauss1

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	72	43	66	64
2	2	29	69	41	17	21	1	51
3	9	36	62	2	50	63	41	51
4	34	77	78	69	45	41	17	51
5	15	62	77	45	69	24	4	3

(a) Scale factor: 1^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	43	66	72	64
2	2	29	69	41	21	17	51	50
3	9	36	62	48	2	50	63	51
4	34	77	78	45	41	17	51	12
5	15	62	77	45	69	24	3	4

(b) Scale factor: 2^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	43	3	66	12	64
2	2	29	69	41	21	20	50	51
3	9	36	62	48	63	2	50	51
4	34	77	78	45	41	12	51	66
5	15	62	77	45	69	3	20	4

(c) Scale factor: 4^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	3	66	12	21	64
2	2	29	69	48	41	66	51	21
3	9	36	62	48	66	41	51	50
4	34	77	78	41	66	21	12	51
5	15	62	77	45	3	20	66	4

(d) Scale factor: 8^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	40	3	66	21	12
2	2	29	69	48	41	20	66	51
3	9	36	62	48	66	41	51	50
4	34	77	78	41	21	66	12	51
5	15	62	77	20	3	51	21	66

(e) Scale factor: 10^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	48	3	66	21	12
2	2	29	69	48	66	51	41	20
3	9	36	62	24	66	51	41	20
4	34	77	78	–	–	–	–	–
5	15	62	77	20	3	21	2	29

(f) Scale factor: 12^{-1}

	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	48	3	66	21	36
2	2	29	69	48	66	51	41	20
3	9	36	62	24	66	51	21	2
4	34	77	78	66	21	41	51	50
5	15	62	77	48	3	20	21	51

(g) Scale factor: 15^{-1}

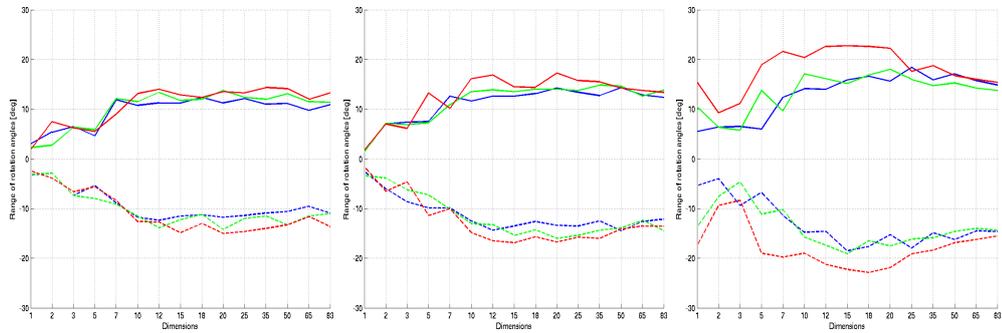
	Rank							
	1	2	3	4	5	6	7	8
1	20	51	79	32	48	66	3	36
2	2	29	69	48	66	51	3	36
3	9	36	62	48	66	51	21	80
4	34	77	78	66	21	41	80	51
5	15	62	77	20	41	51	3	66

(h) Scale factor: 20^{-1}

Table A.15.: Results for IPCA-ranking. The startlist is determined by randomly choosing three landmarks; blurring: gauss3

A.6. Reliability Evaluation

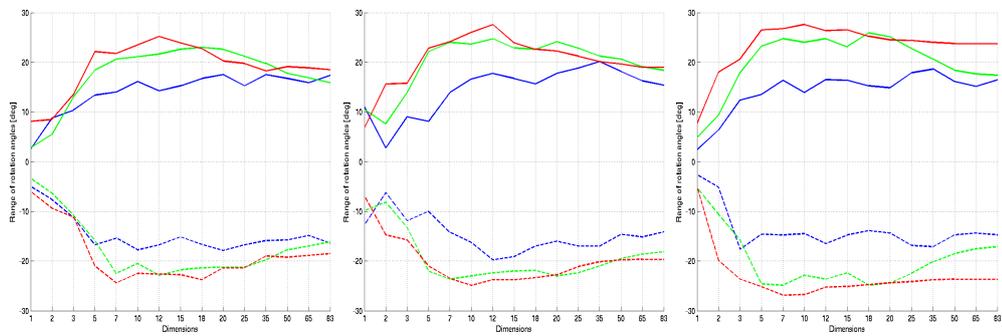
A.6.1. Orientation



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

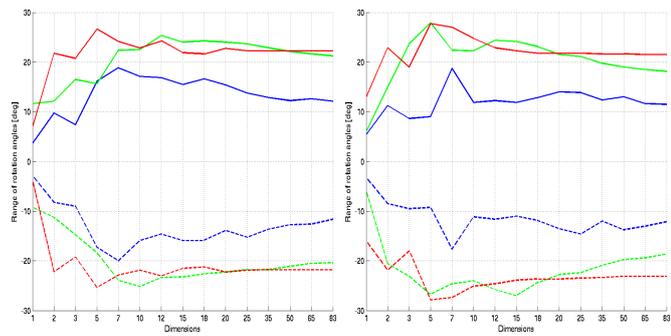
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

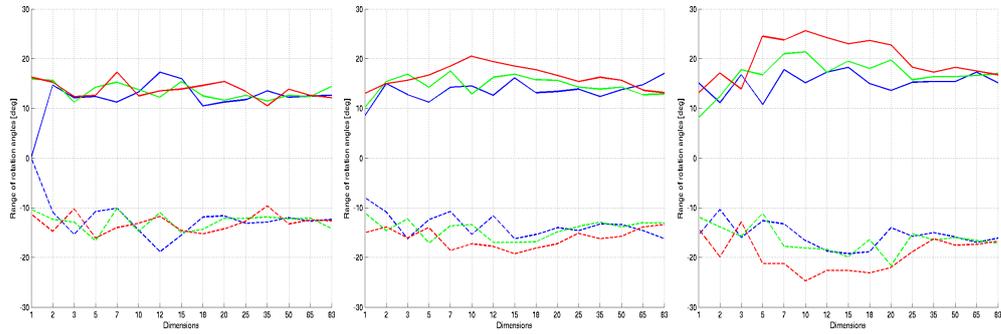
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

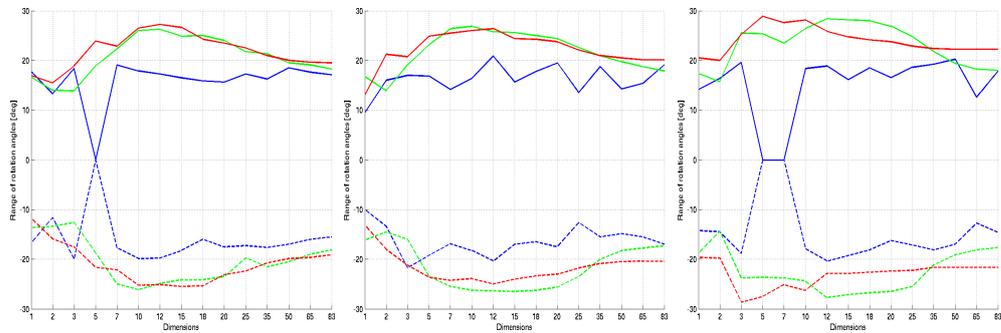
Figure A.13.: Possible deviations of the blimp's orientation for profile-based ranking. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

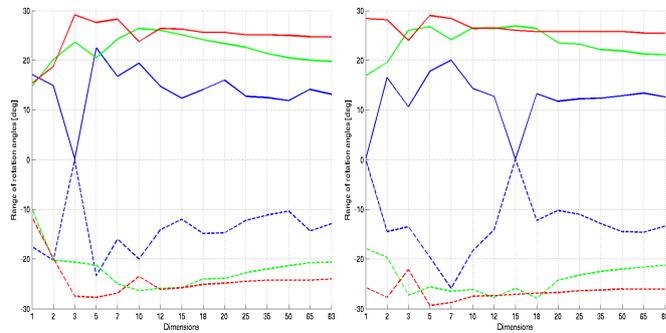
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

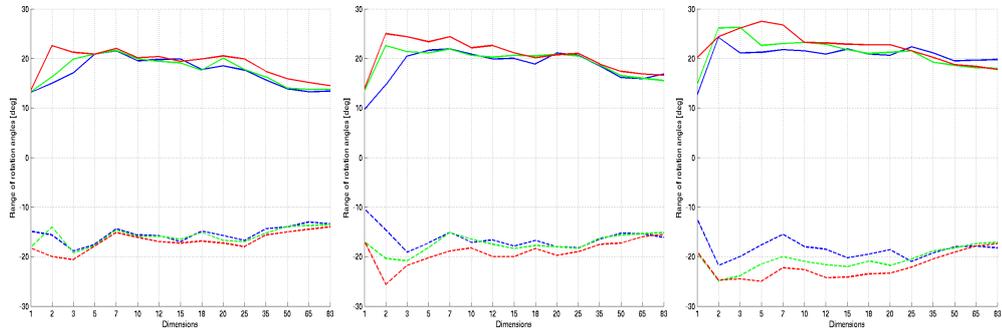
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

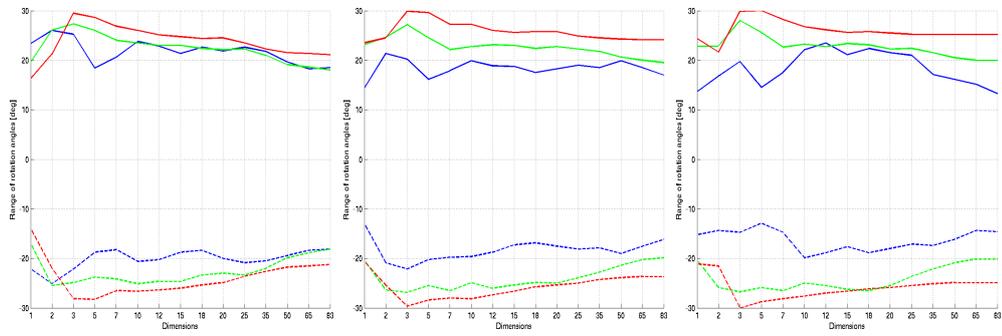
Figure A.14.: Possible deviations of the blimp's orientation for IPCA-based ranking with profile-based start list. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

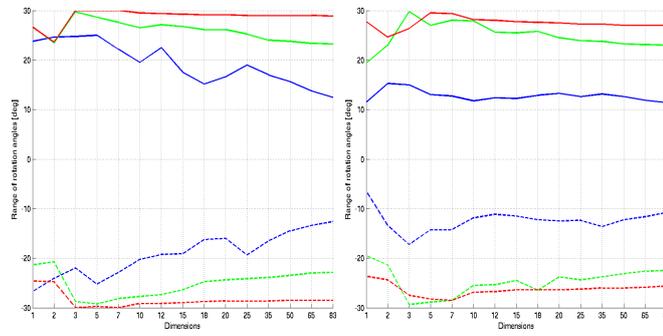
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

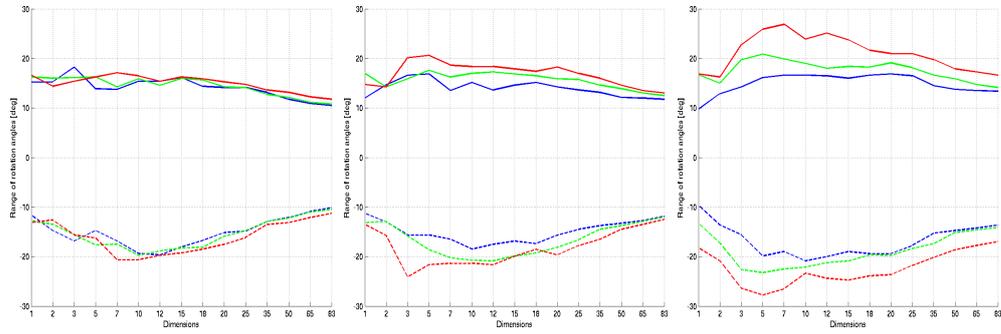
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

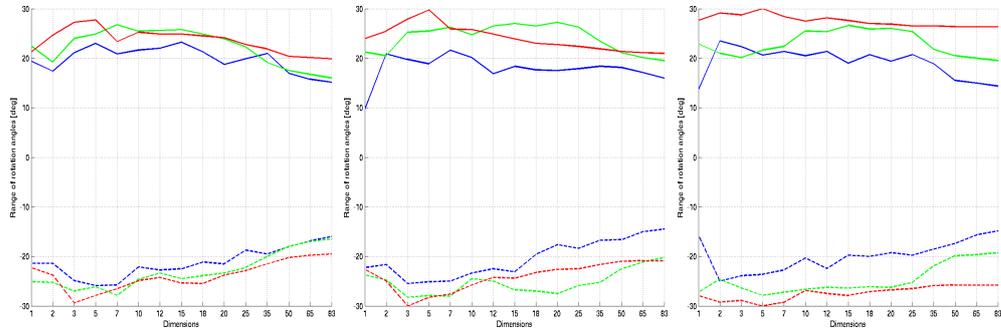
Figure A.15.: Possible deviations of the blimp's orientation for IPCA-based ranking "rnd1". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

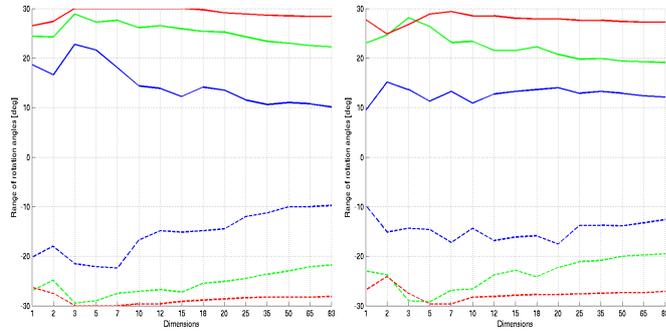
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

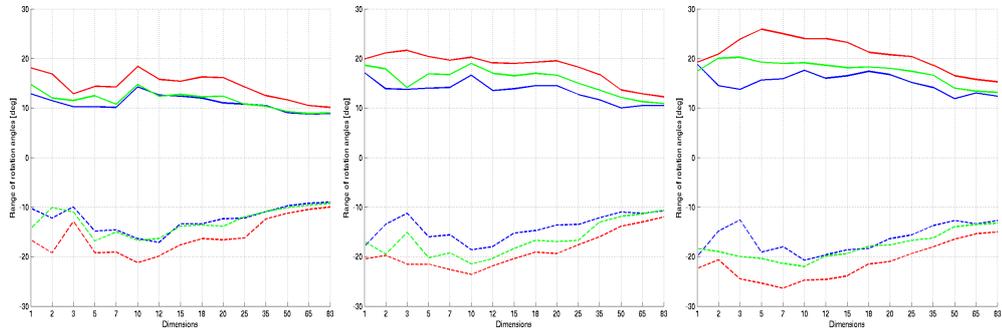
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

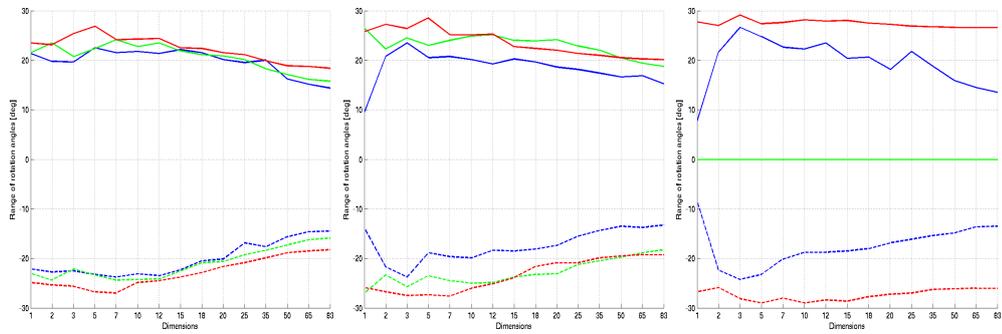
Figure A.16.: Possible deviations of the blimp's orientation for IPCA-based ranking "rnd2". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

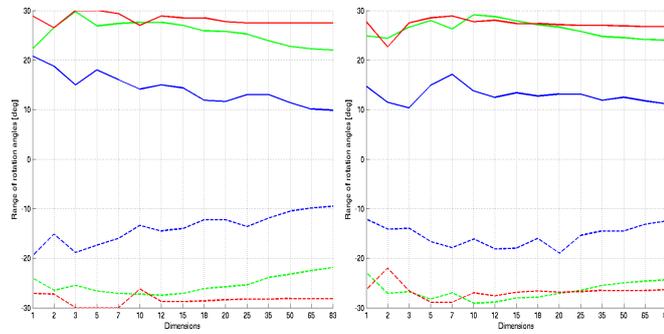
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

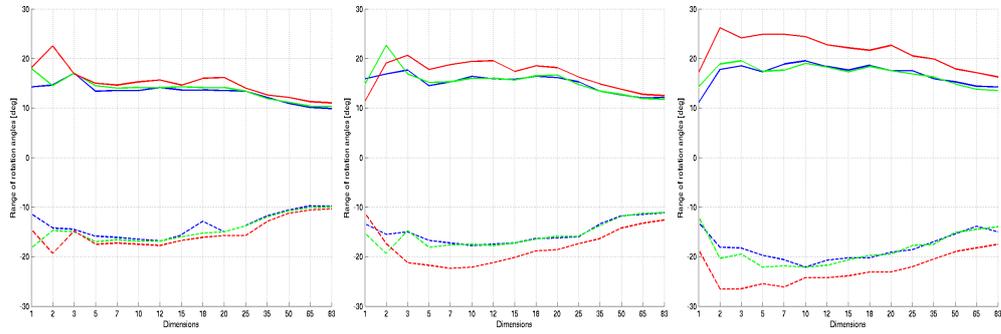
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

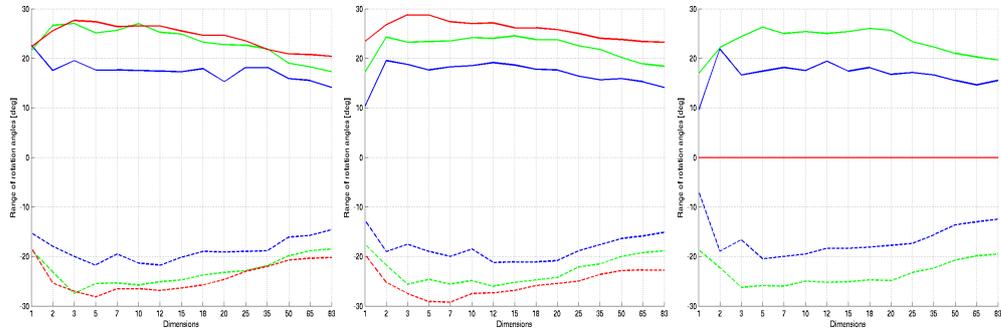
Figure A.17.: Possible deviations of the blimp's orientation for IPCA-based ranking "rnd3". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

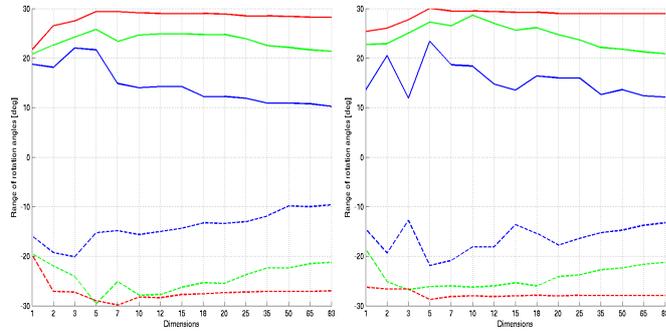
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

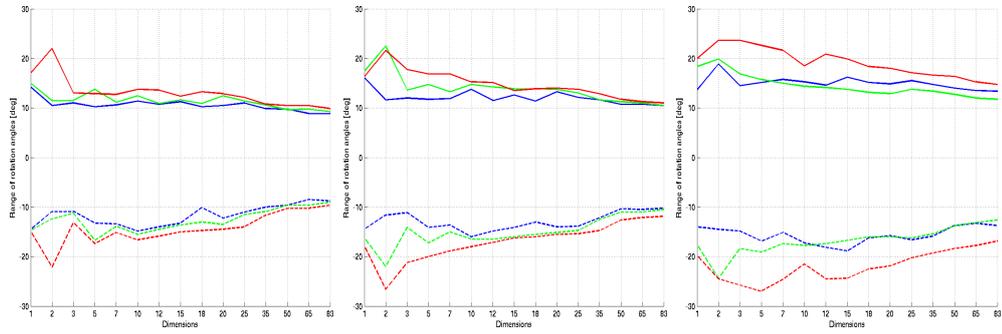
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

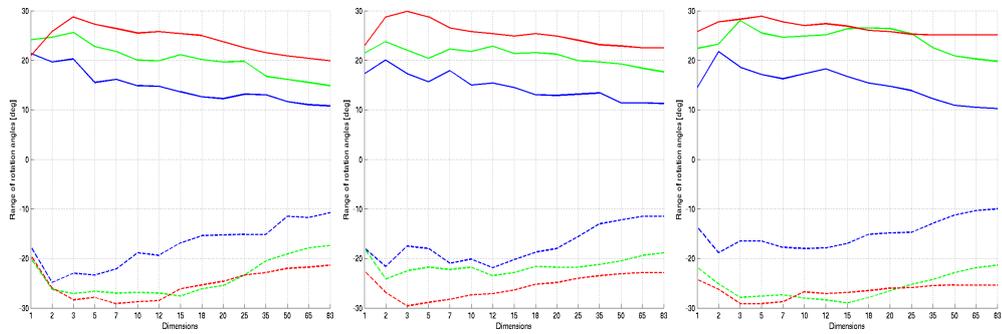
Figure A.18.: Possible deviations of the blimp's orientation for IPCA-based ranking "rnd4". Blue lines: un-blurred, green: gauss1, red: gauss4



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

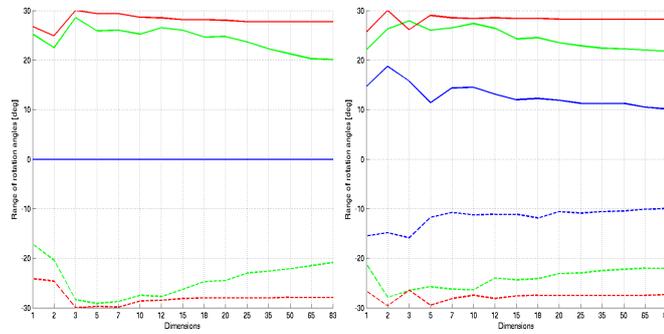
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

(f) Scale factor: 1^{-12}

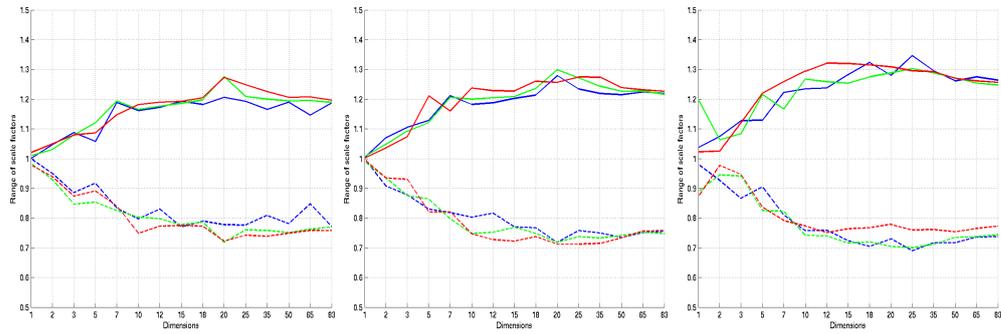


(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

Figure A.19.: Possible deviations of the blimp's orientation for IPCA-based ranking "rnd5". Blue lines: un-blurred, green: gauss1, red: gauss5

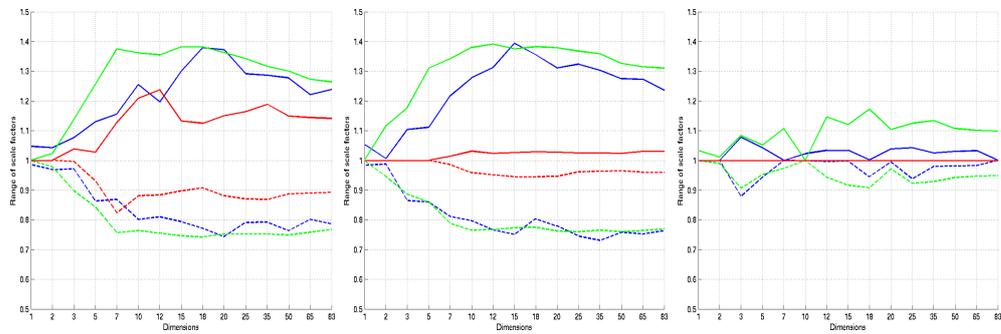
A.6.2. Altitude



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

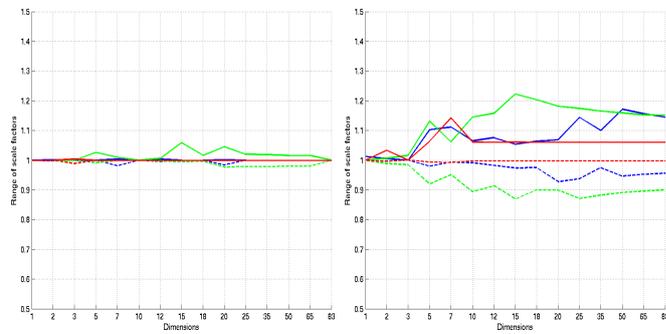
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

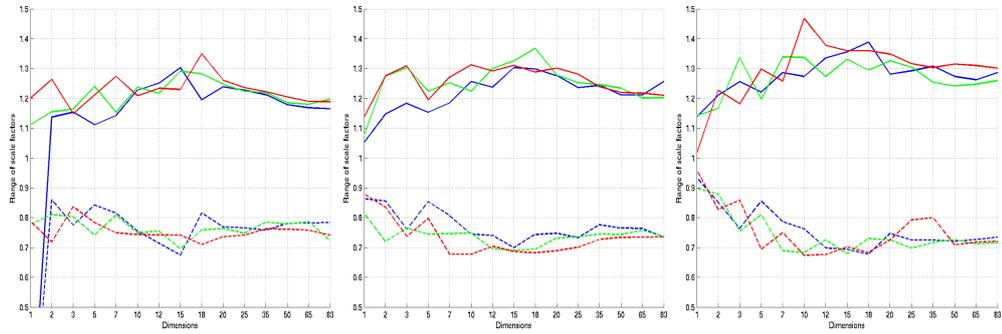
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

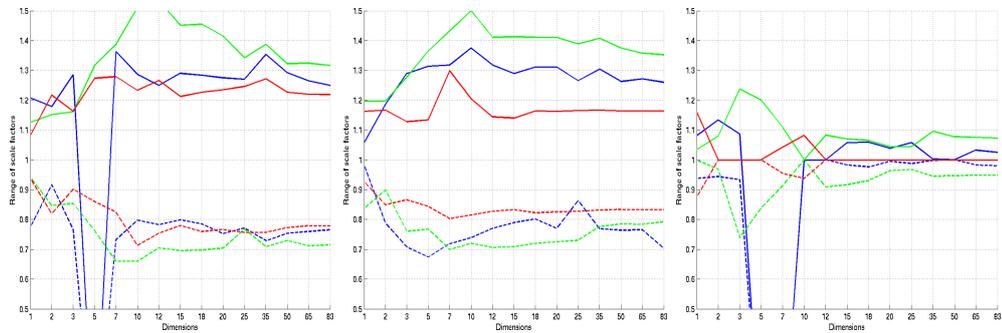
Figure A.20.: Possible deviations of the blimp's altitude for profile-based ranking. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

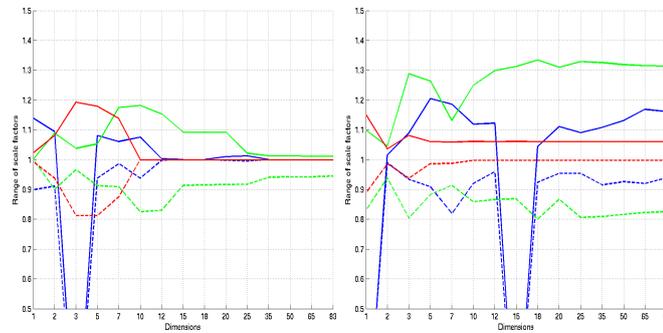
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

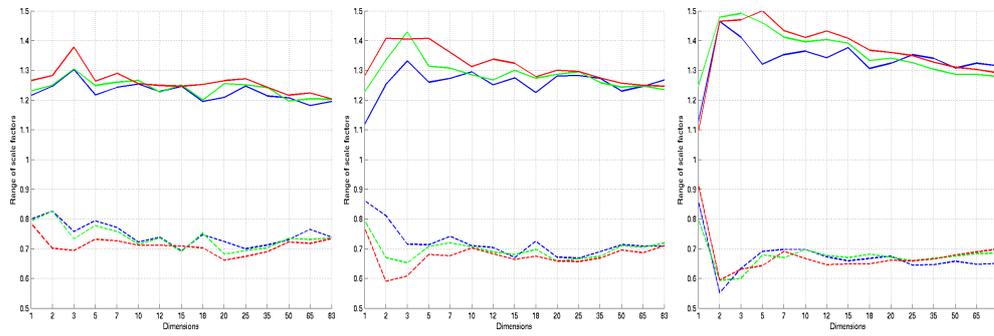
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

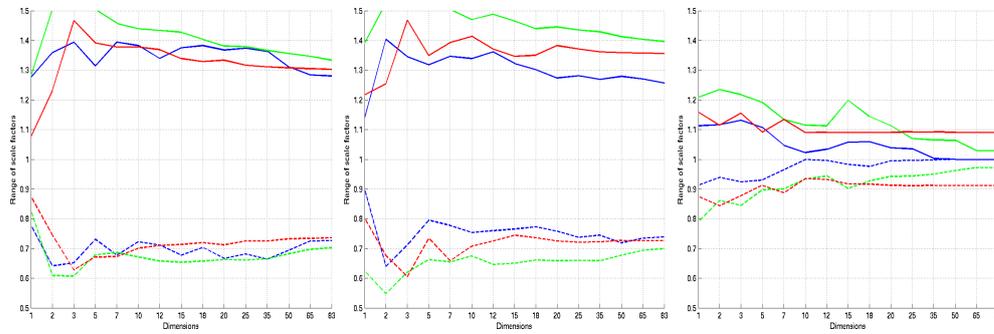
Figure A.21.: Possible deviations of the blimp's altitude for IPCA-based ranking with profile-based start list. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

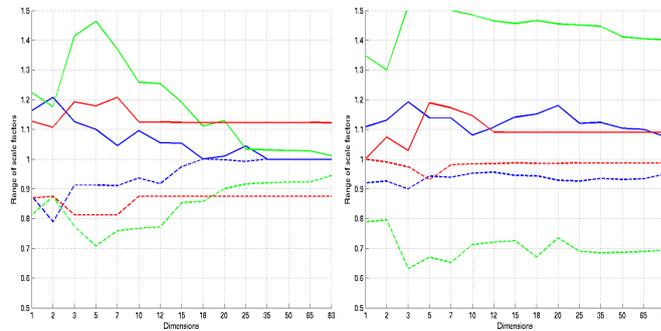
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

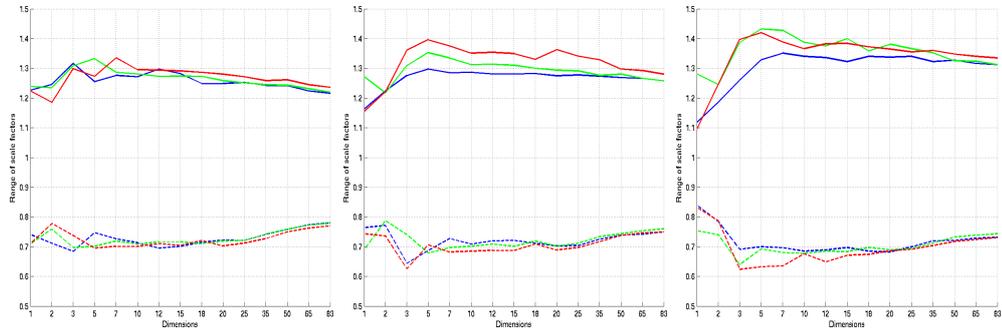
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

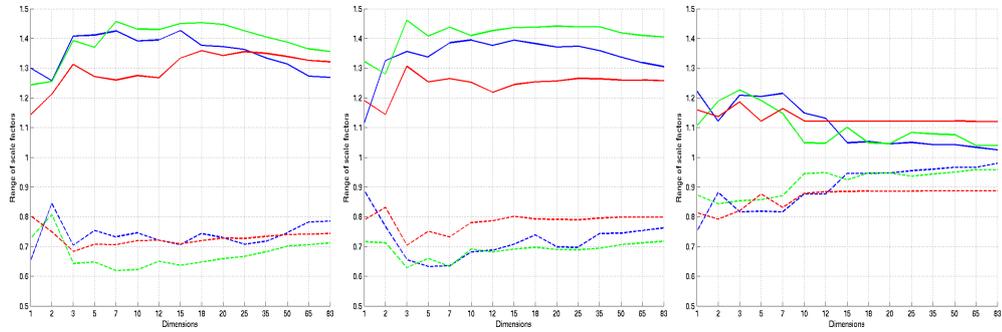
Figure A.22.: Possible deviations of the blimp's altitude for IPCA-based ranking "rnd1". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

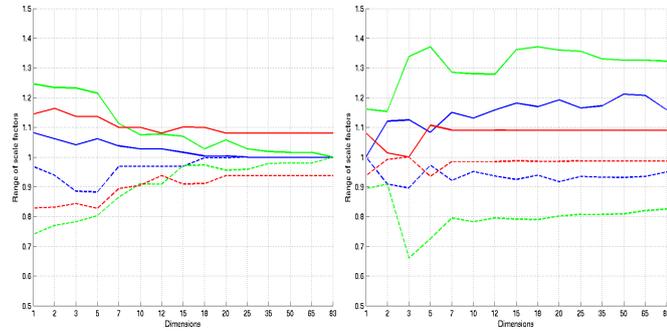
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

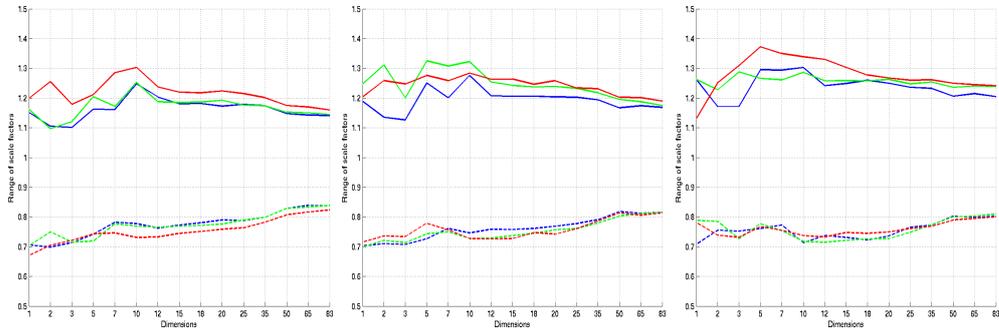
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

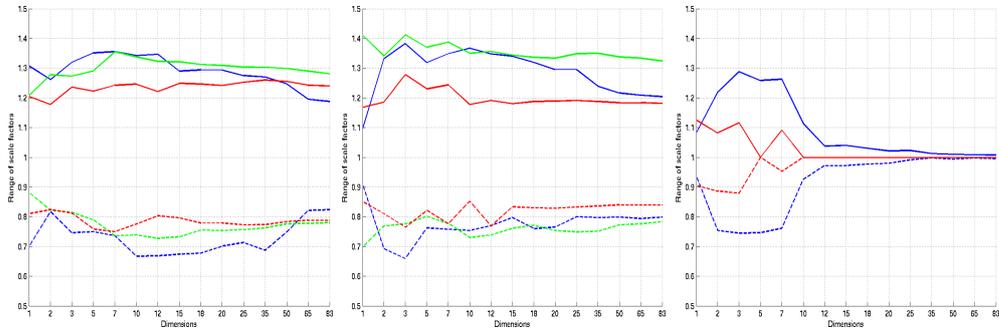
Figure A.23.: Possible deviations of the blimp's altitude for IPCA-based ranking "rnd2". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

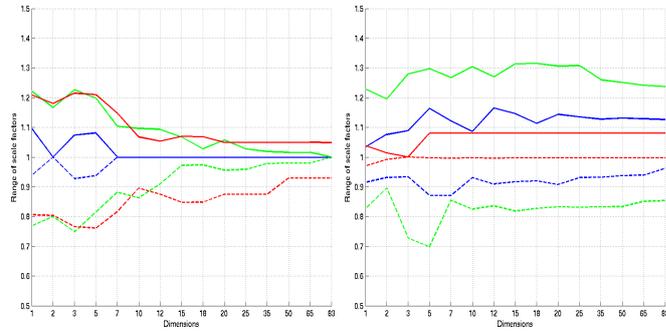
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

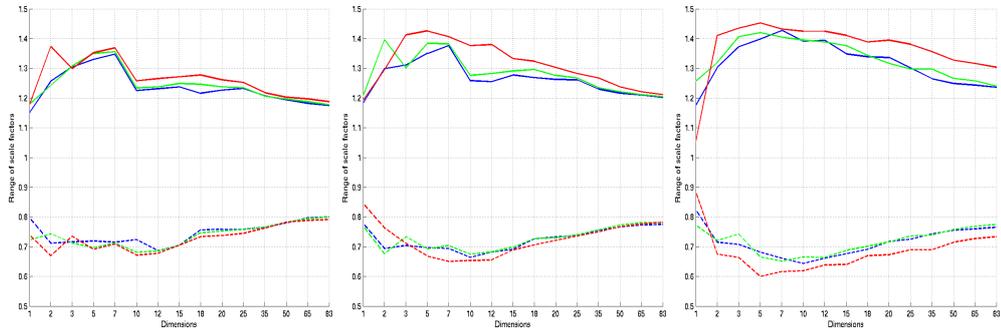
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

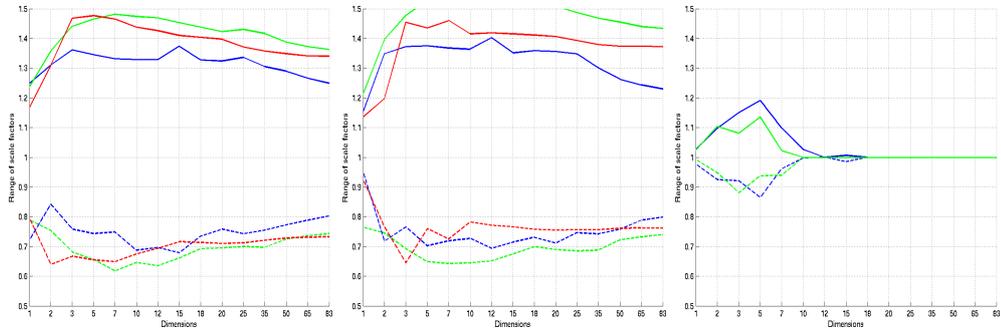
Figure A.24.: Possible deviations of the blimp's altitude for IPCA-based ranking "rnd3". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

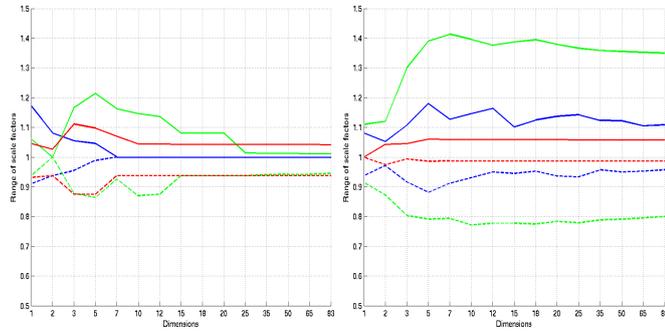
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

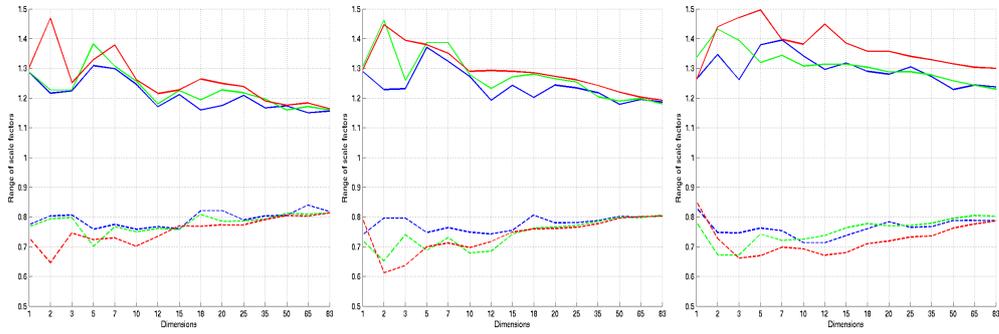
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

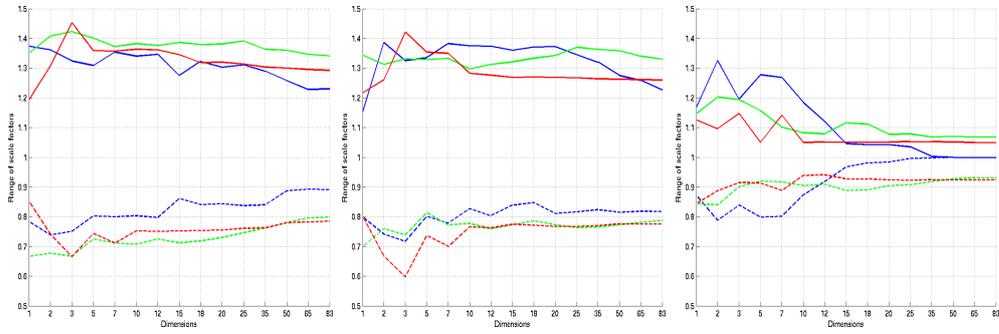
Figure A.25.: Possible deviations of the blimp's altitude for IPCA-based ranking "rnd4". Blue lines: un-blurred, green: gauss1, red: gauss4



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

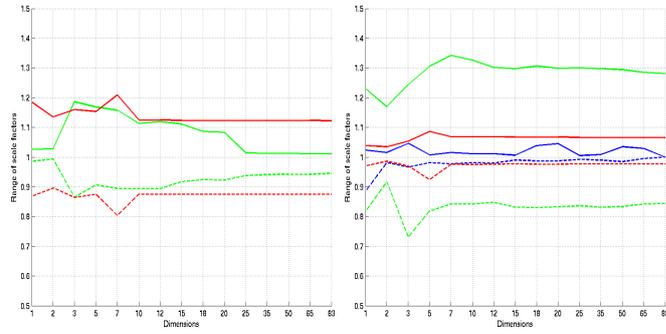
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

(f) Scale factor: 1^{-12}

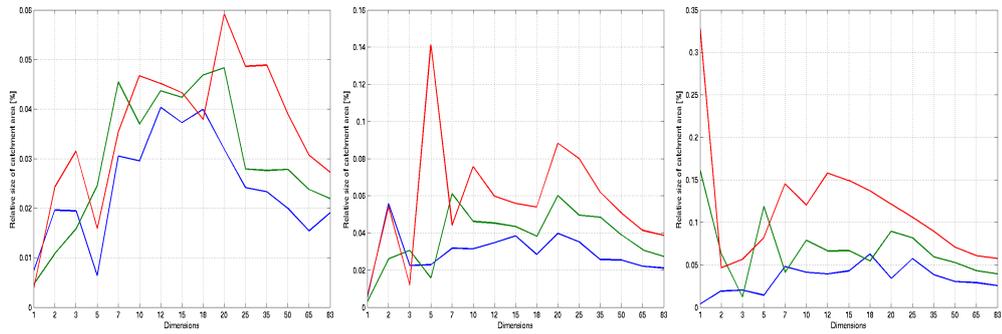


(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

Figure A.26.: Possible deviations of the blimp's altitude for IPCA-based ranking "rnd5". Blue lines: un-blurred, green: gauss1, red: gauss5

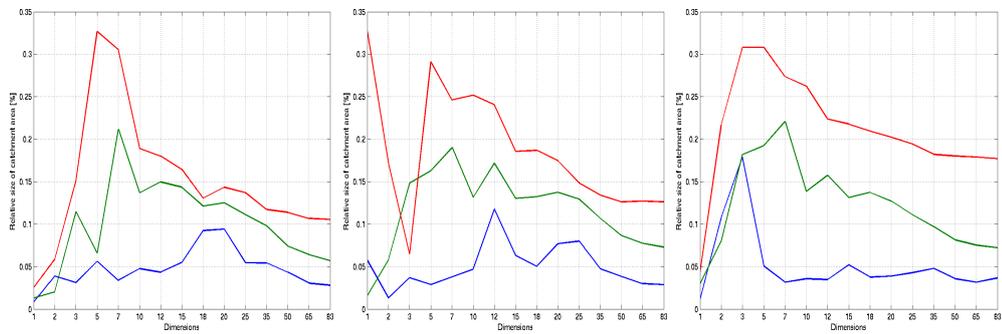
A.6.3. Position



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

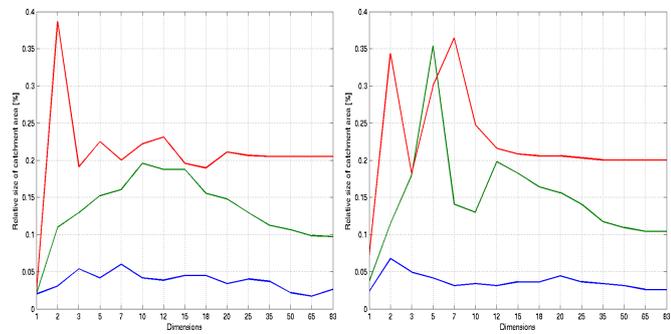
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

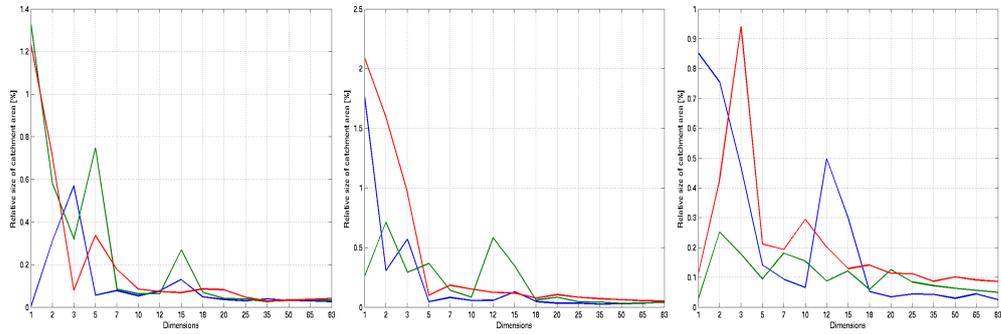
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

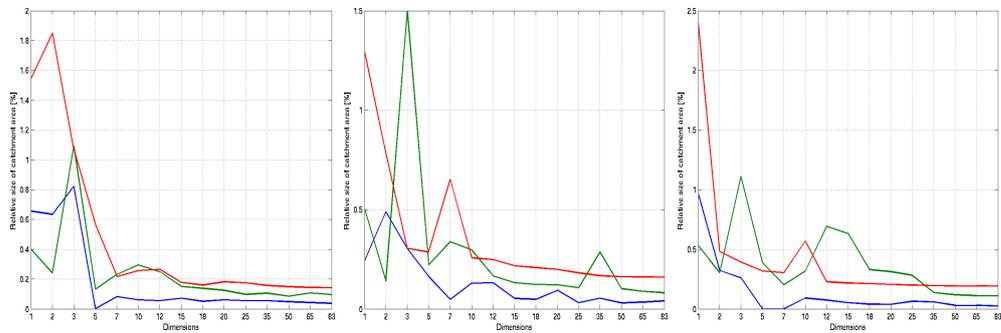
Figure A.27.: Possible deviations of the blimp's position for profile-based ranking. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

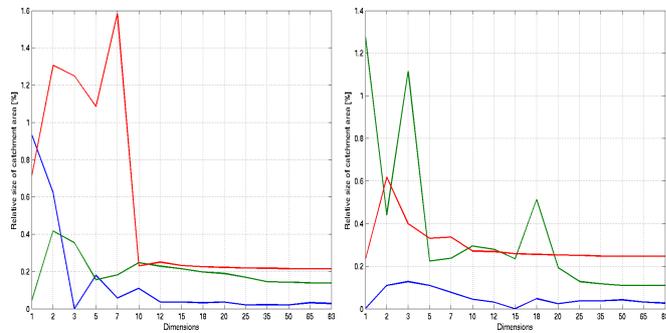
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

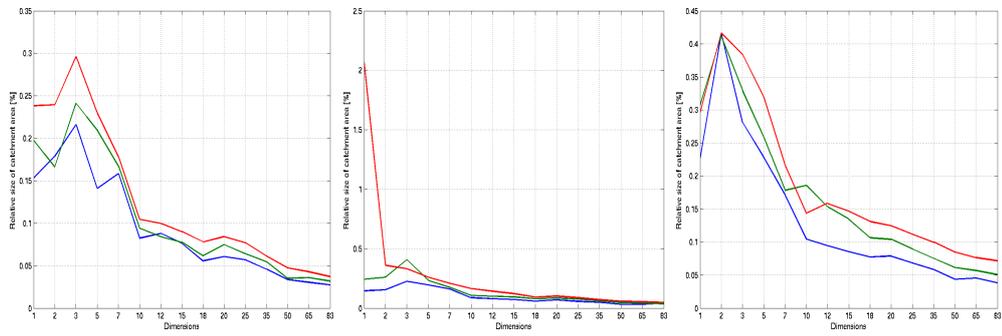
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

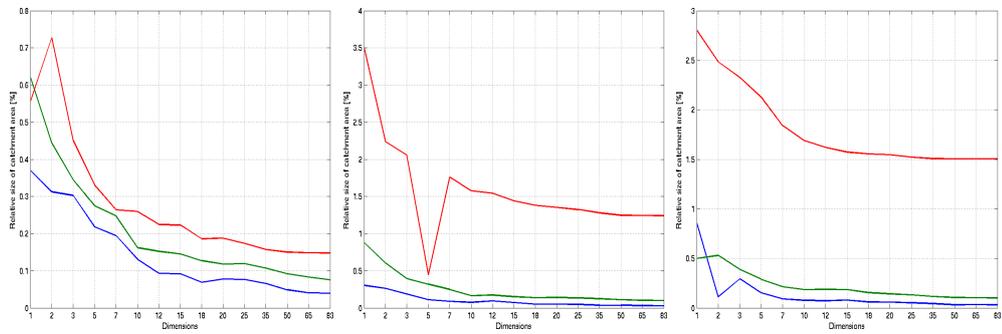
Figure A.28.: Possible deviations of the blimp's position for IPCA-based ranking with profile-based start list. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

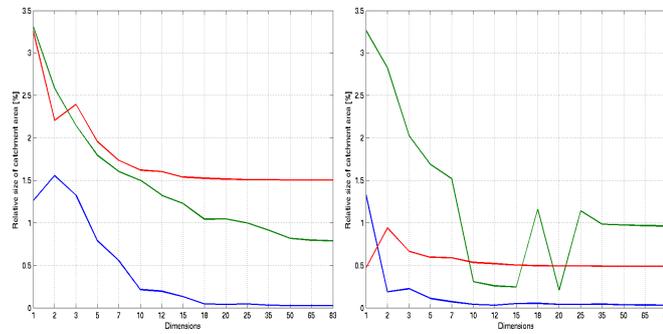
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

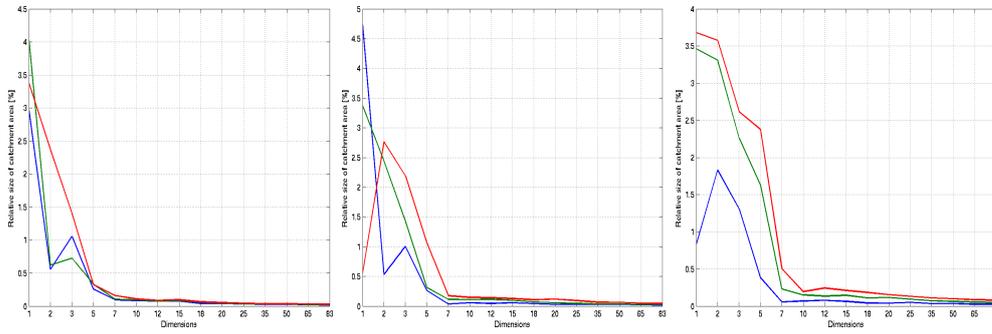
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

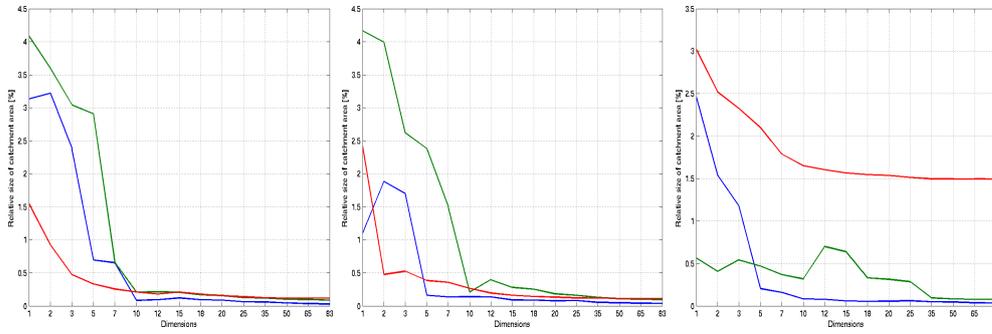
Figure A.29.: Possible deviations of the blimp's position for IPCA-based ranking "rnd1". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

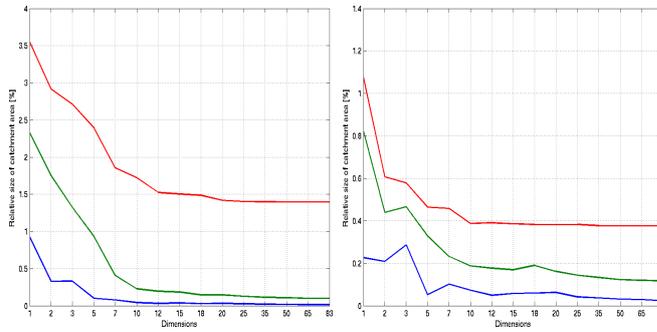
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

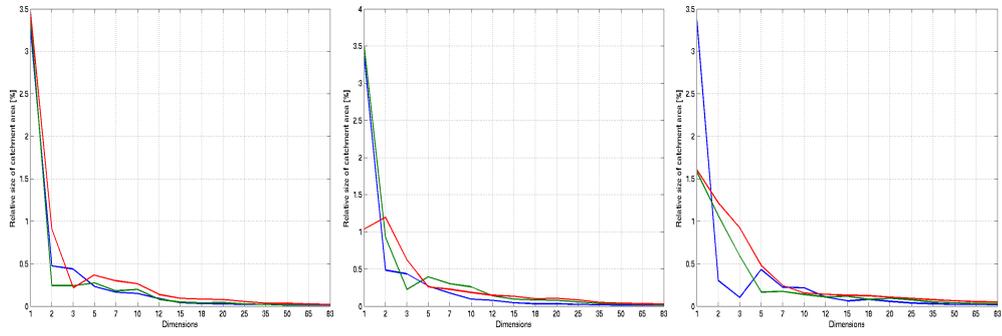
(f) Scale factor: 1^{-12}



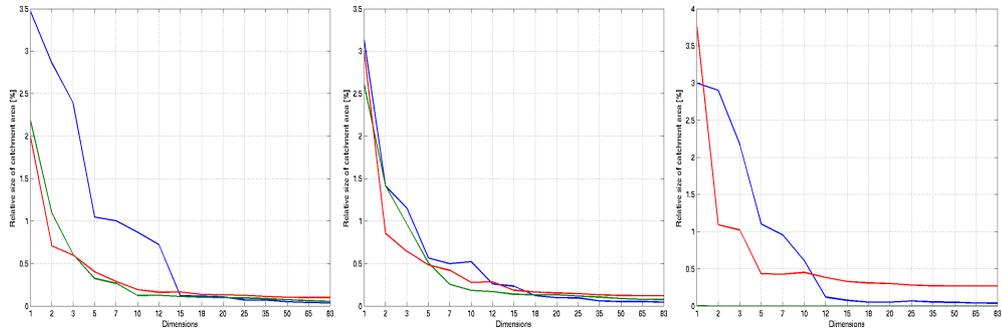
(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

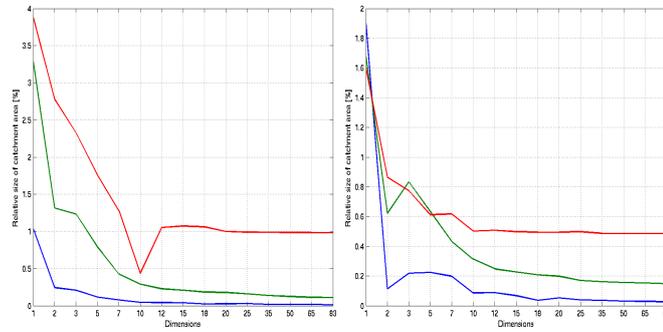
Figure A.30.: Possible deviations of the blimp's position for IPCA-based ranking "rnd2". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1} (b) Scale factor: 1^{-2} (c) Scale factor: 1^{-4}

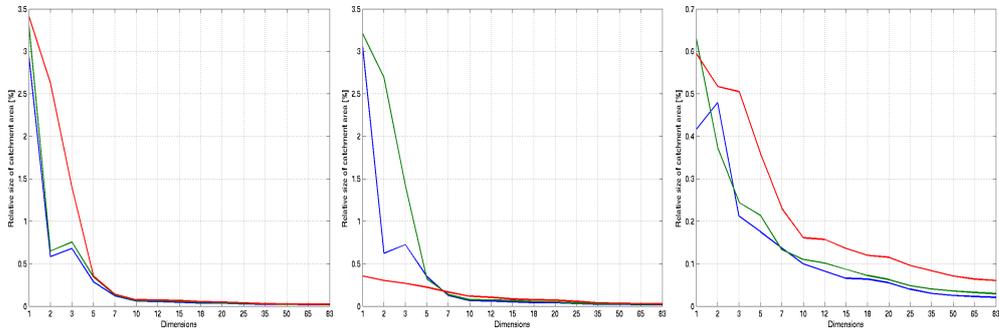


(d) Scale factor: 1^{-8} (e) Scale factor: 1^{-10} (f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15} (h) Scale factor: 1^{-20}

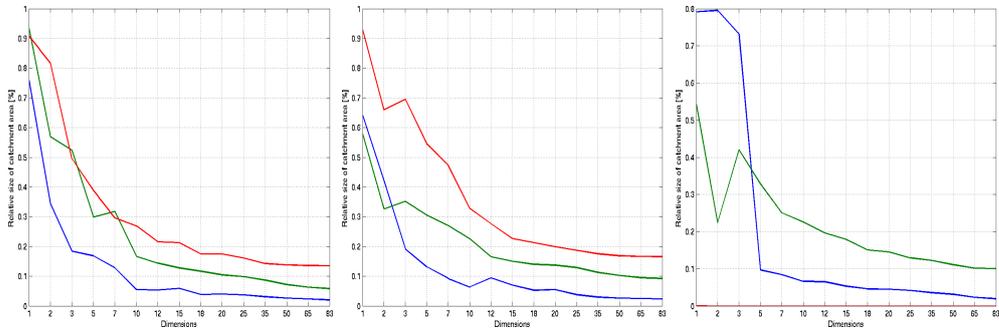
Figure A.31.: Possible deviations of the blimp's position for IPCA-based ranking "rnd3". Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

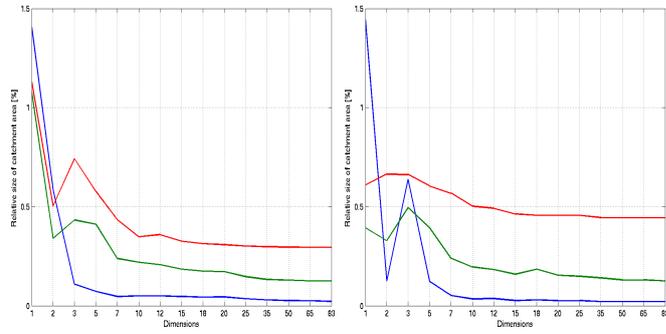
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

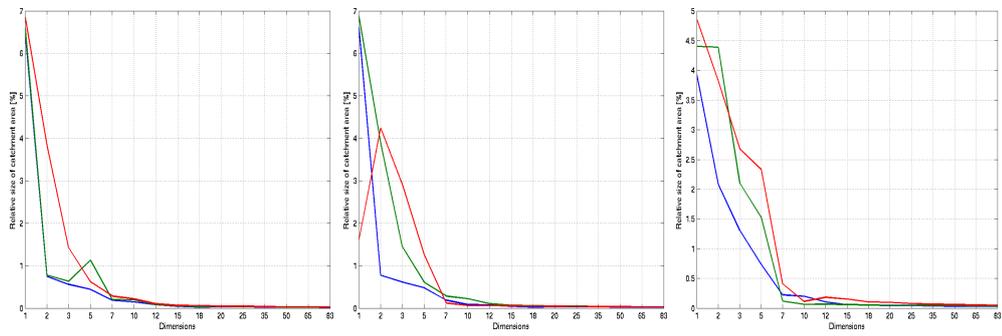
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

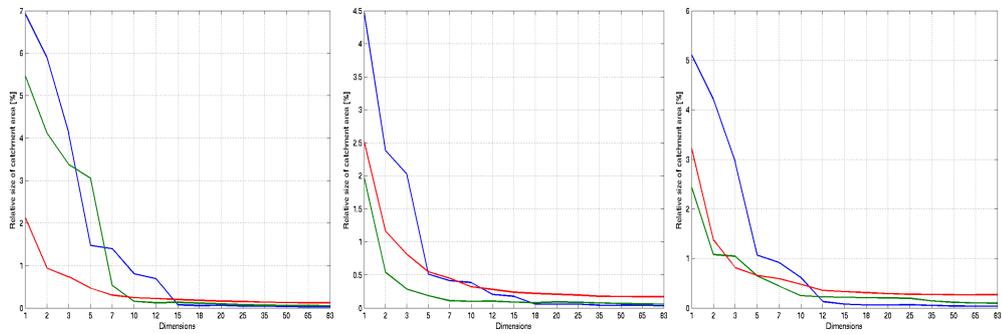
Figure A.32.: Possible deviations of the blimp's position for IPCA-based ranking "rnd4". Blue lines: un-blurred, green: gauss1, red: gauss4



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

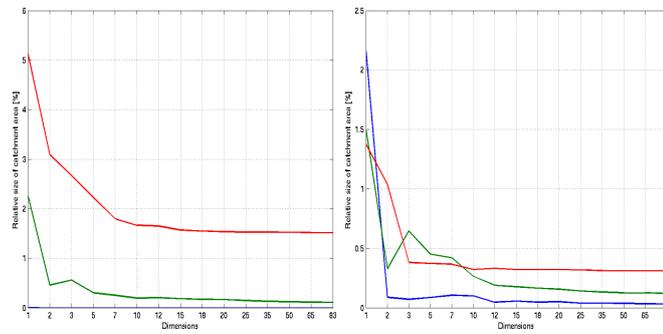
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

(f) Scale factor: 1^{-12}

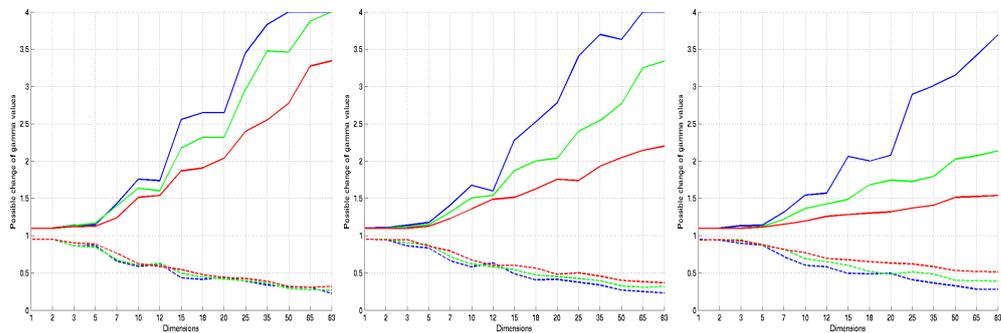


(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

Figure A.33.: Possible deviations of the blimp's position for IPCA-based ranking "rnd5". Blue lines: un-blurred, green: gauss1, red: gauss5

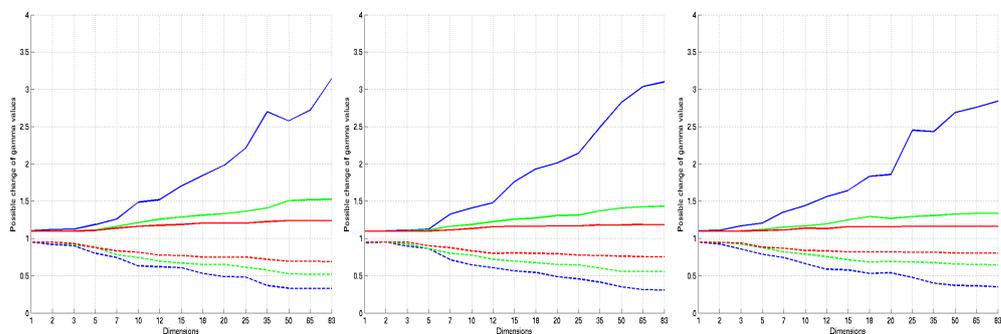
A.6.4. Image Brightness



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

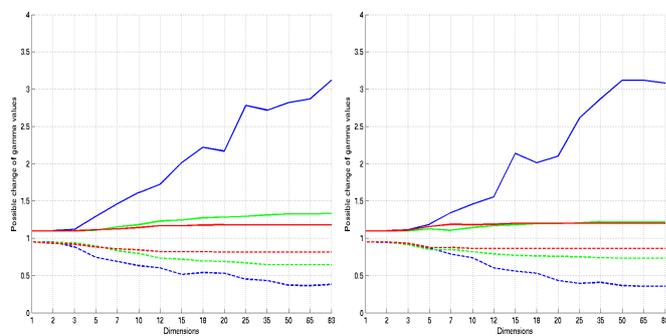
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

Figure A.34.: Possible deviations of the image brightness for profile-based ranking. Blue lines: un-blurred, green: gauss1, red: gauss3

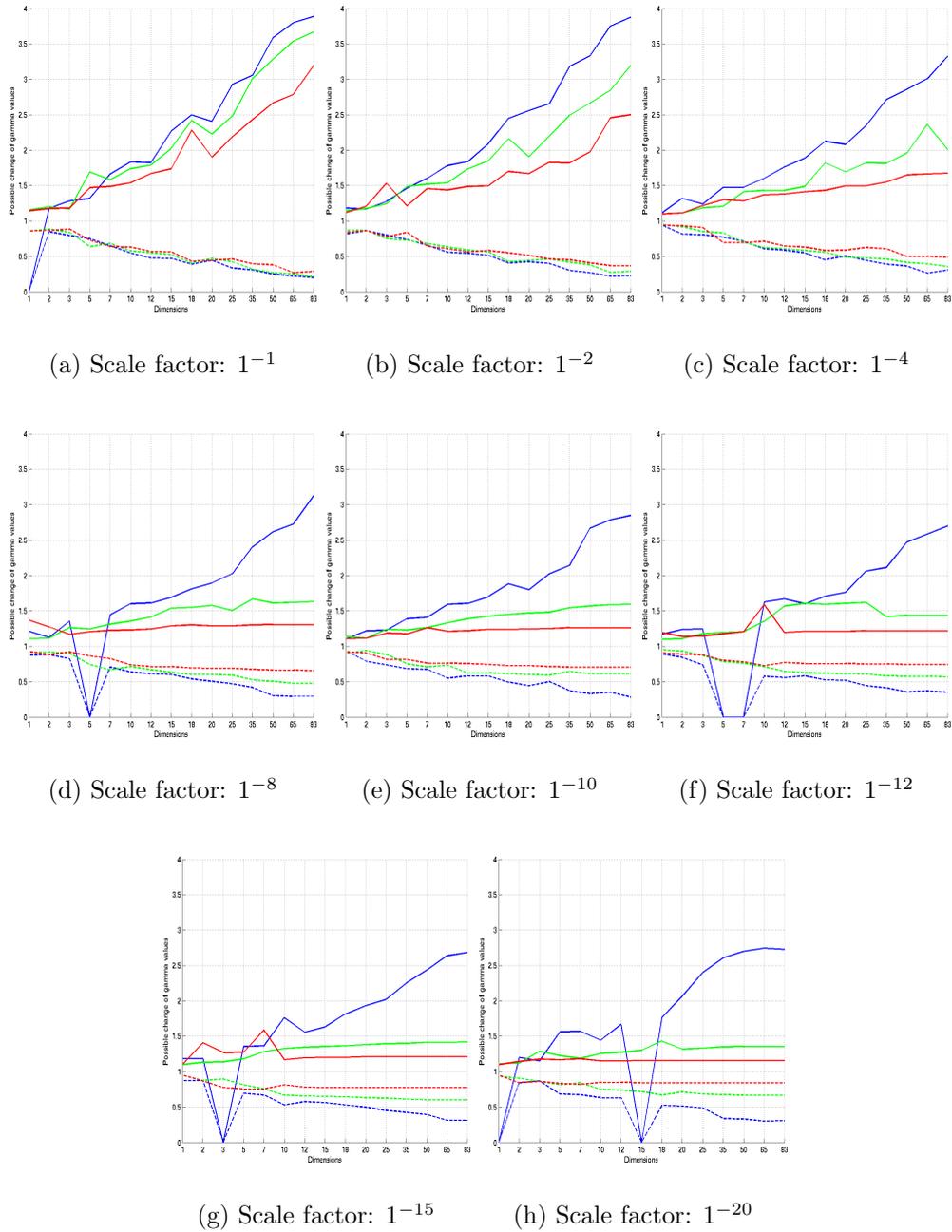
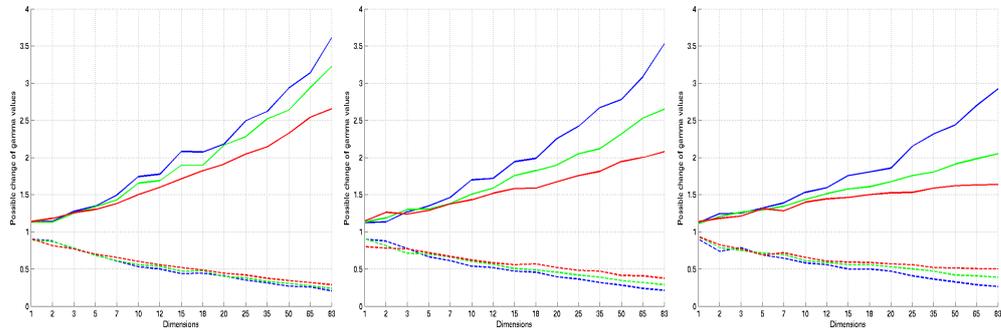


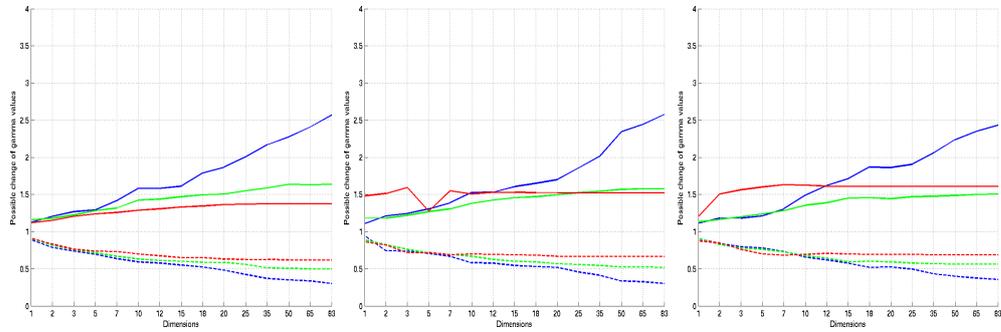
Figure A.35.: Possible deviations of the image brightness for IPCA-based ranking with profile-based start list. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

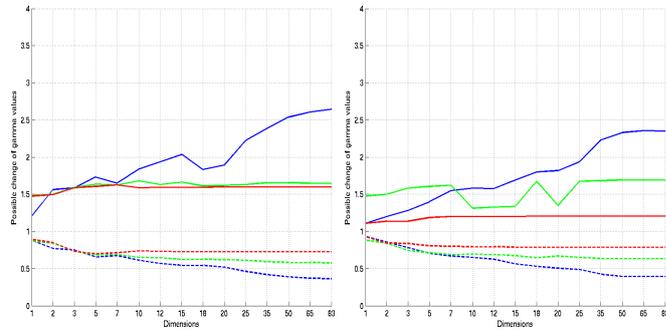
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

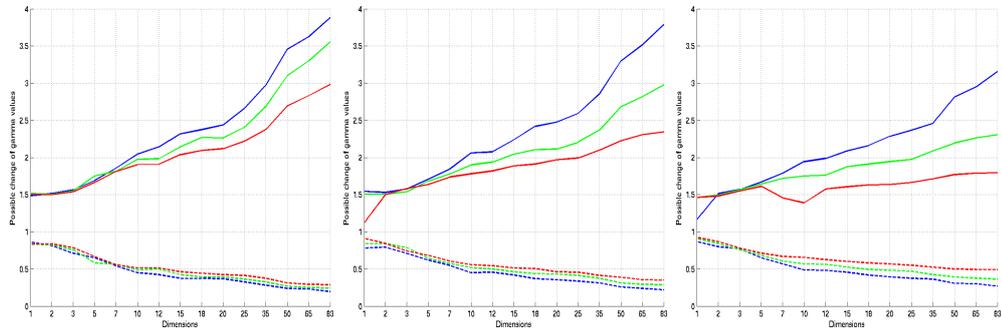
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

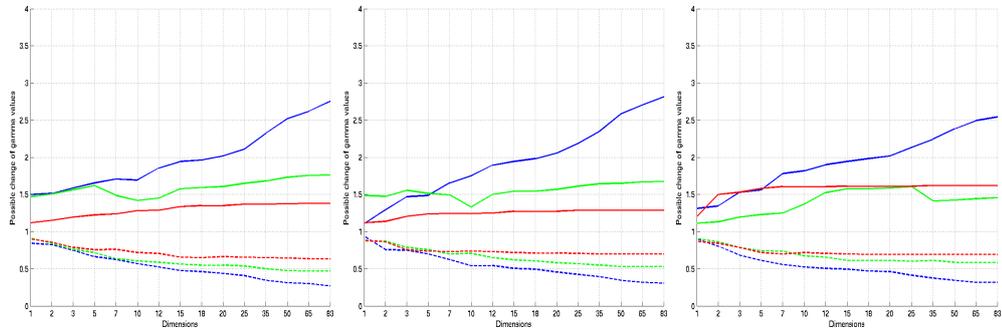
Figure A.36.: Possible deviations of the image brightness for IPCA-based ranking “rnd1”. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

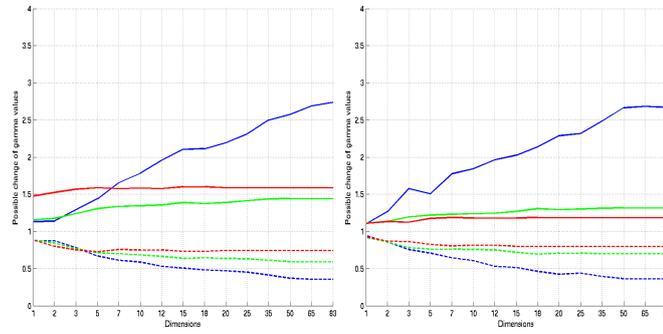
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

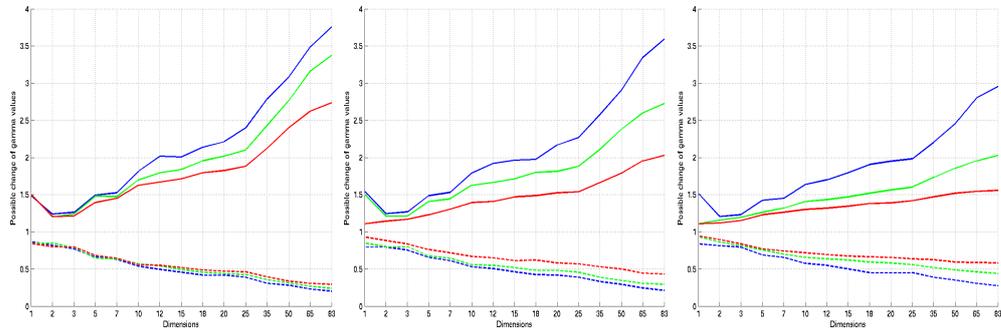
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

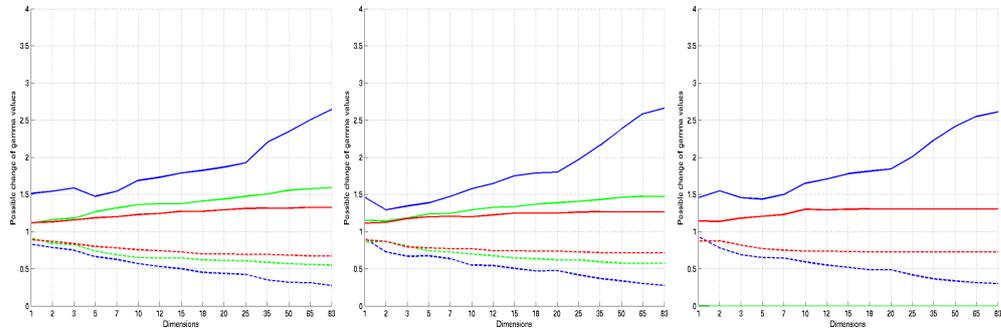
Figure A.37.: Possible deviations of the image brightness for IPCA-based ranking “rnd2”. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

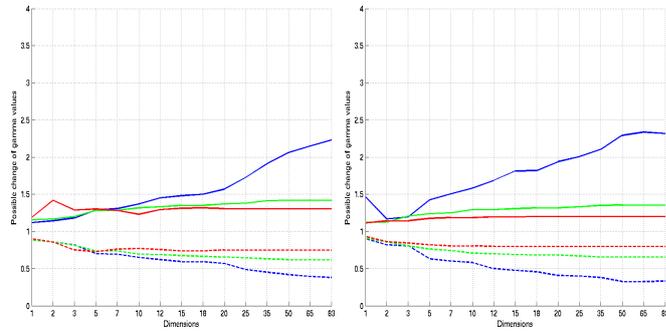
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

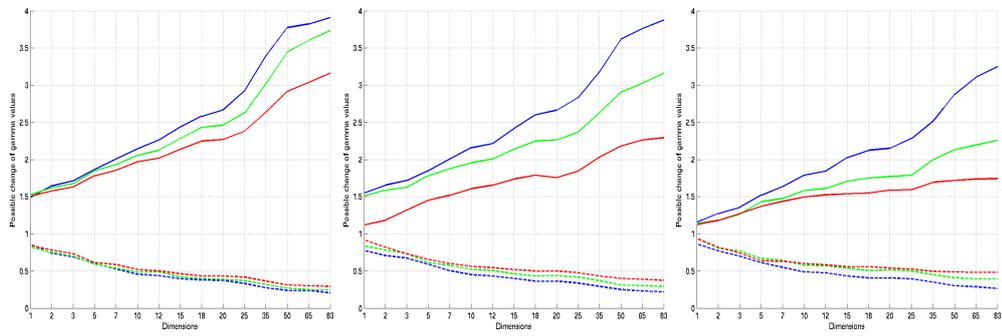
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

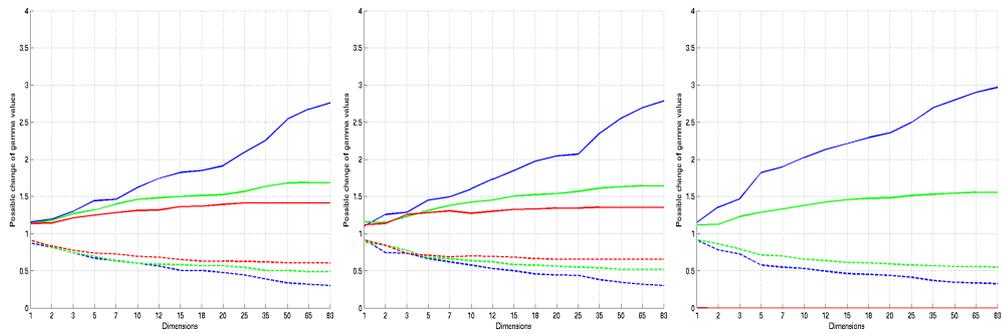
Figure A.38.: Possible deviations of the image brightness for IPCA-based ranking “rnd3”. Blue lines: un-blurred, green: gauss1, red: gauss3



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

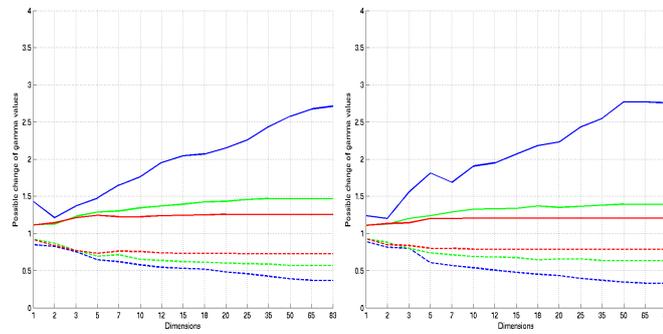
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

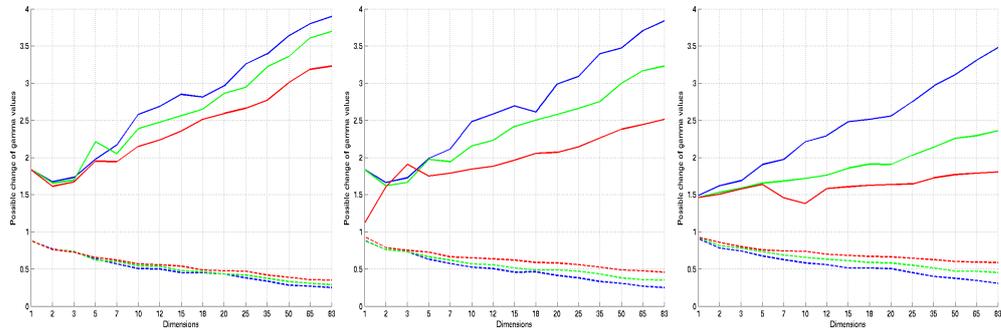
(f) Scale factor: 1^{-12}



(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

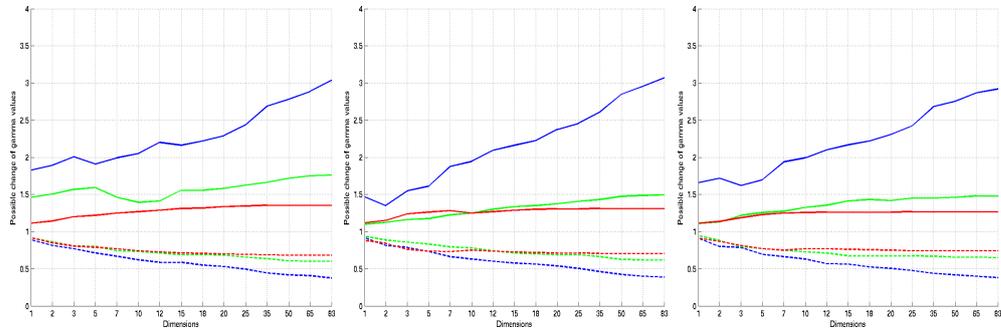
Figure A.39.: Possible deviations of the image brightness for IPCA-based ranking “rnd4”. Blue lines: un-blurred, green: gauss1, red: gauss4



(a) Scale factor: 1^{-1}

(b) Scale factor: 1^{-2}

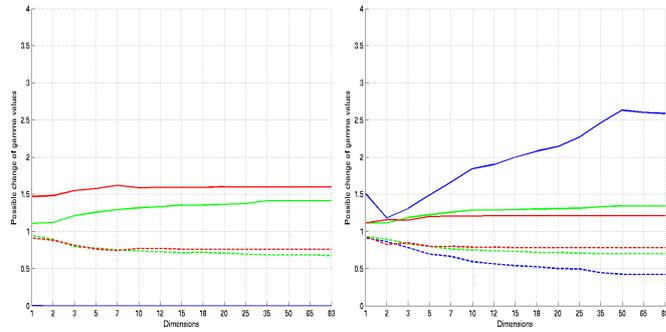
(c) Scale factor: 1^{-4}



(d) Scale factor: 1^{-8}

(e) Scale factor: 1^{-10}

(f) Scale factor: 1^{-12}

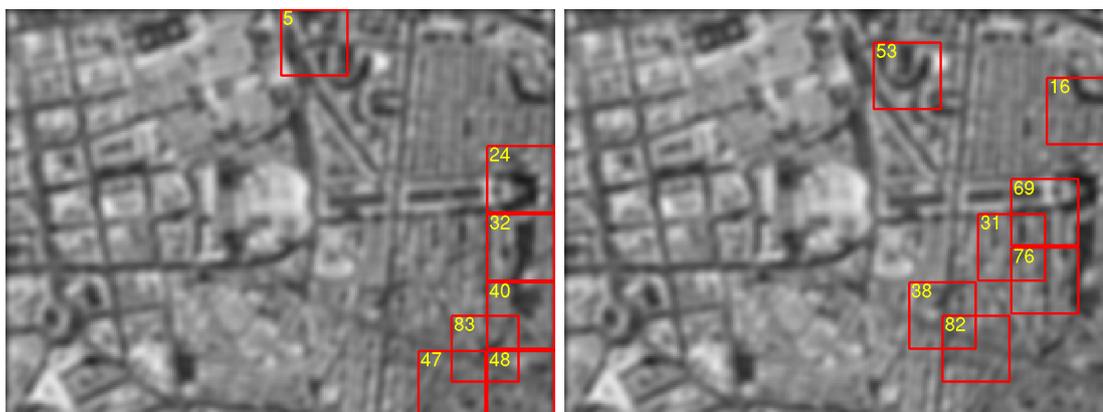


(g) Scale factor: 1^{-15}

(h) Scale factor: 1^{-20}

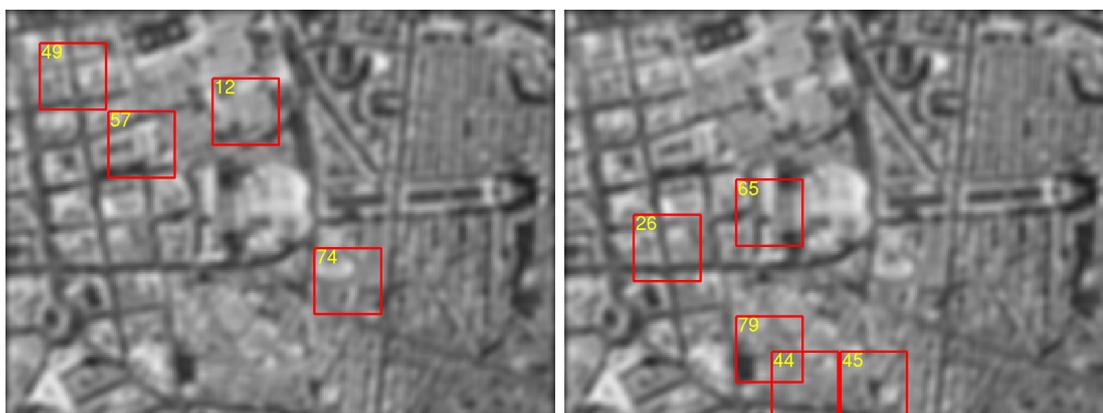
Figure A.40.: Possible deviations of the image brightness for IPCA-based ranking “rnd5”. Blue lines: un-blurred, green: gauss1, red: gauss5

A.7. Landmark selection by hierarchical clustering



(a) Cluster 1

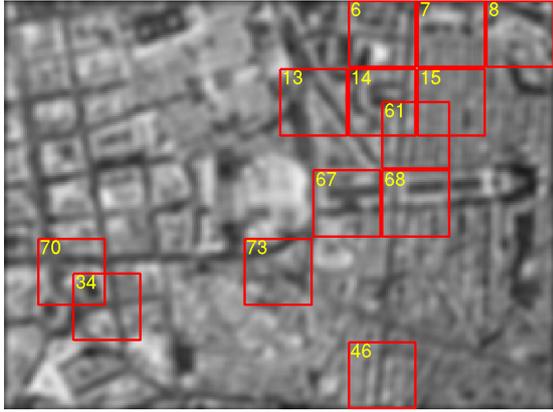
(b) Cluster 2



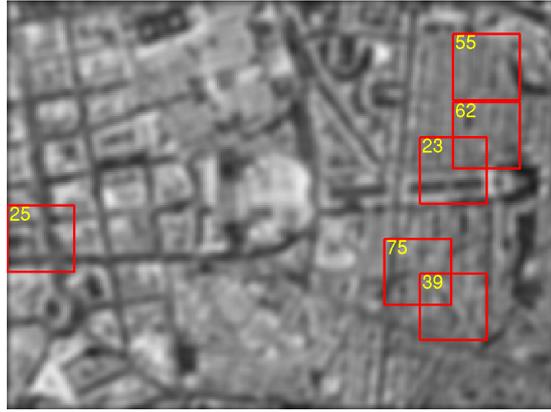
(c) Cluster 3

(d) Cluster 4

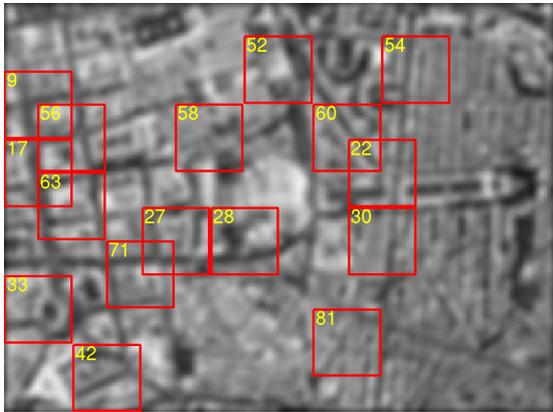
Figure A.41.: Example for landmark selection by hierarchical clustering; scale factor: 6^{-1} , dimensions: 2, blurring: gauss3, 15 clusters



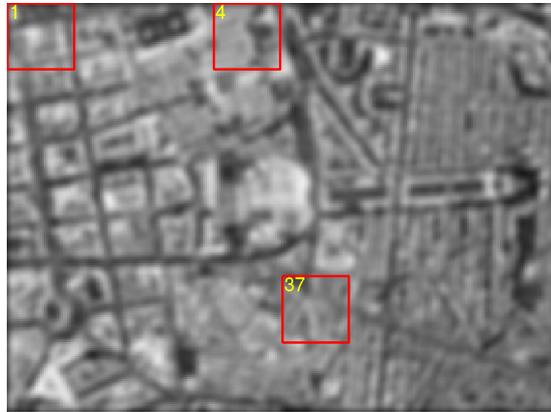
(a) Cluster 5



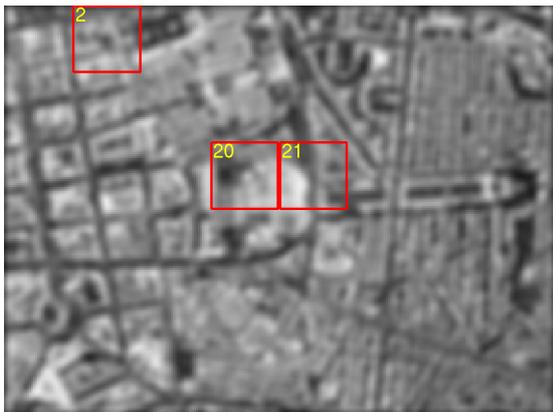
(b) Cluster 6



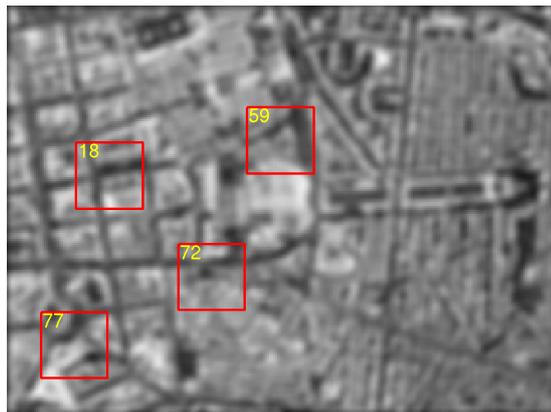
(c) Cluster 7



(d) Cluster 8

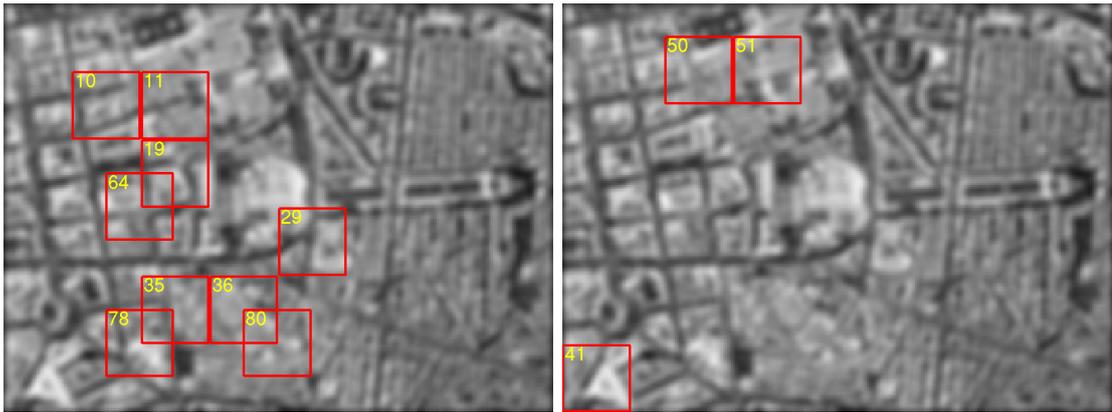


(e) Cluster 9



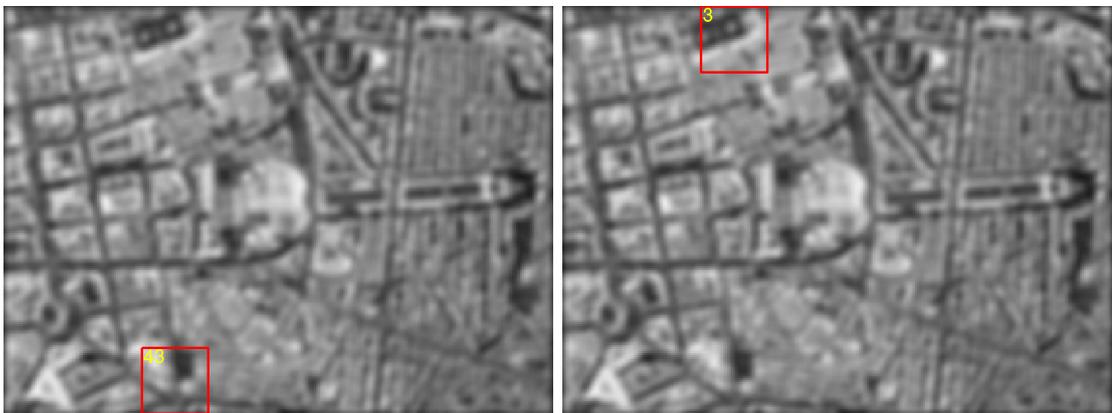
(f) Cluster 10

Figure A.42.: Example for landmark selection by hierarchical clustering continued



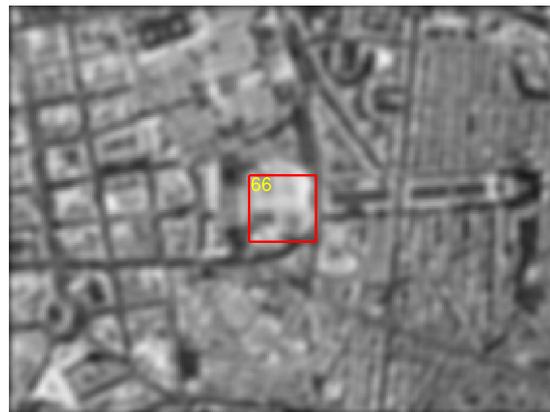
(a) Cluster 11

(b) Cluster 12



(c) Cluster 13

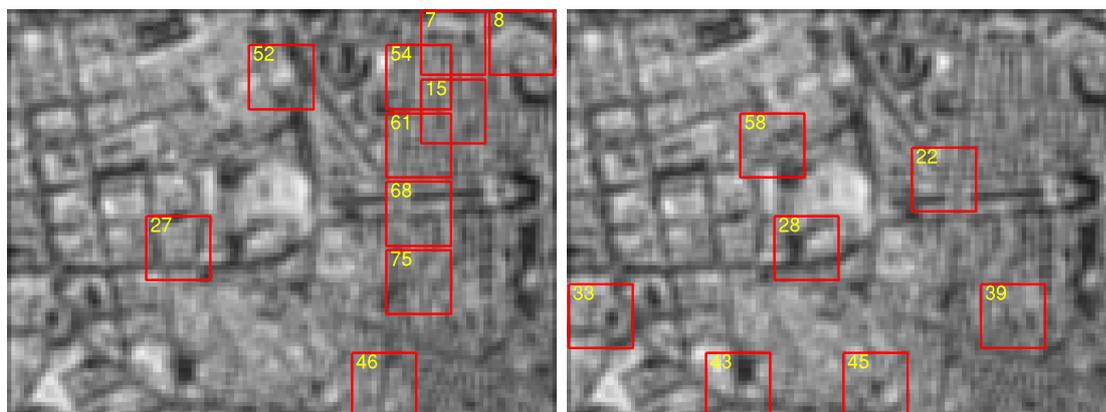
(d) Cluster 14



(e) Cluster 15

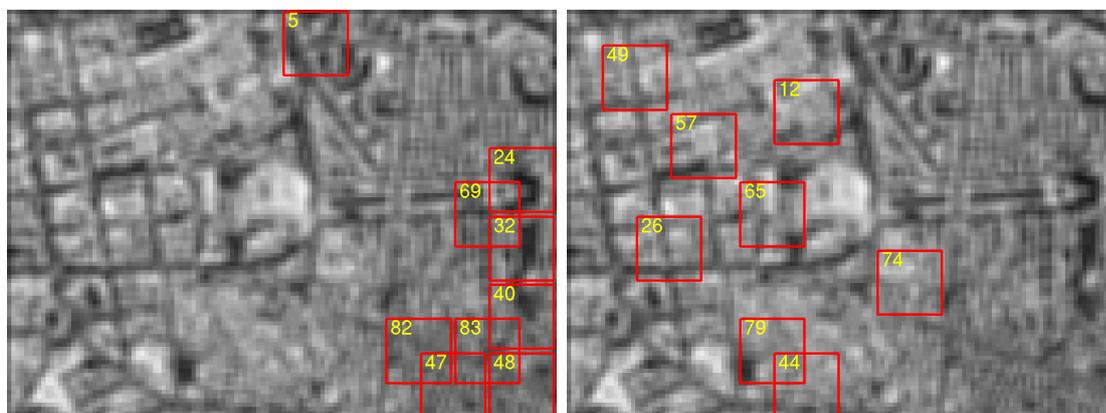
Figure A.43.: Example for landmark selection by hierarchical clustering continued

A.8. Landmark selection by k-means clustering



(a) Cluster 1

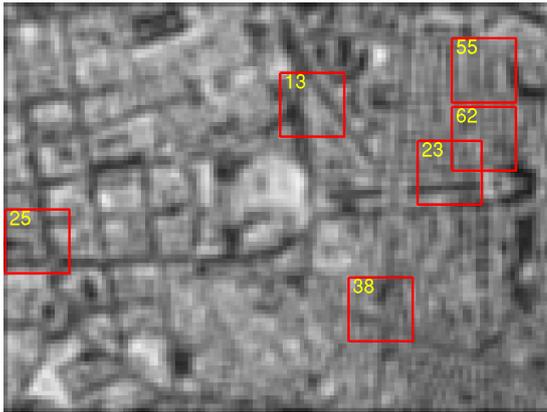
(b) Cluster 2



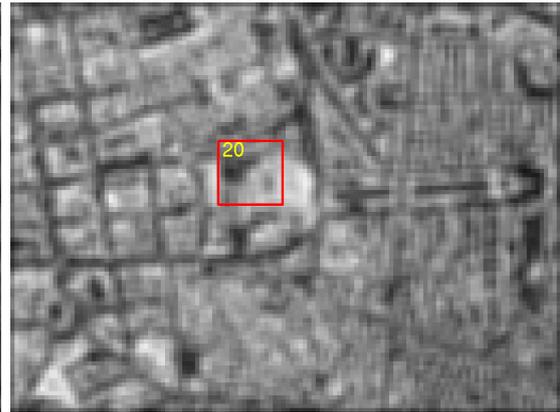
(c) Cluster 3

(d) Cluster 4

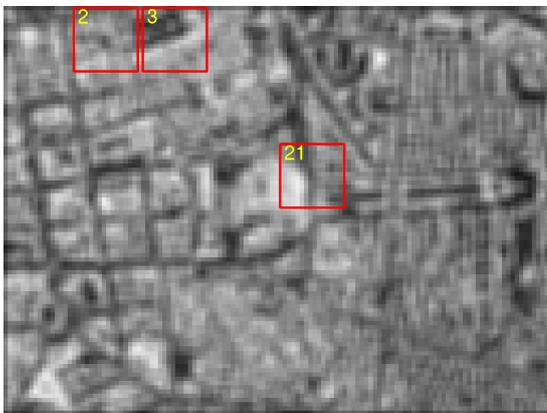
Figure A.44.: Example for landmark selection by k-means clustering; scale factor: 12^{-1} , dimensions: 2, blurring: gauss1, 15 clusters



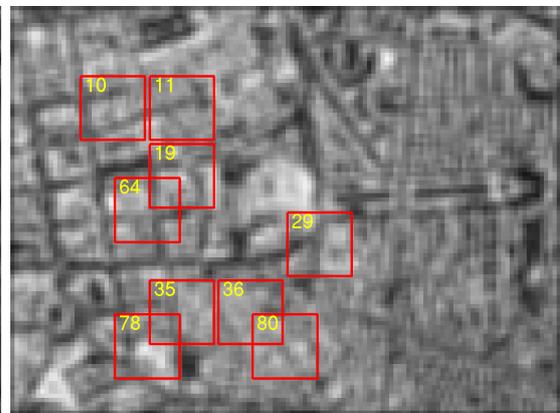
(a) Cluster 5



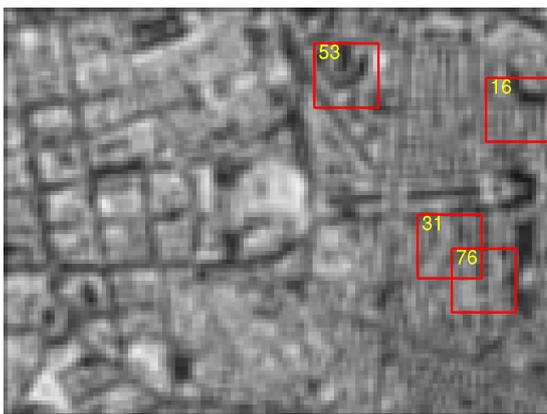
(b) Cluster 6



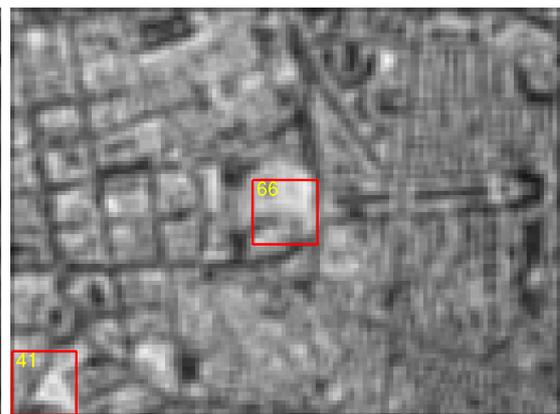
(c) Cluster 7



(d) Cluster 8

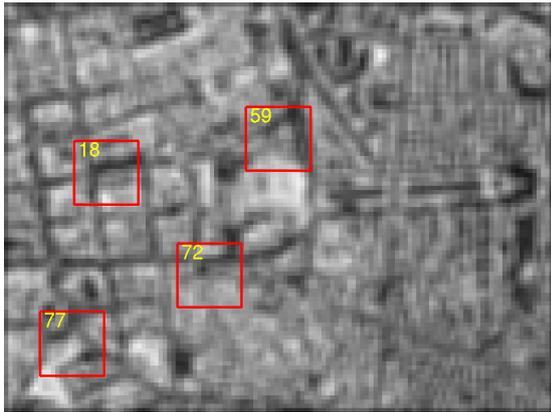


(e) Cluster 9

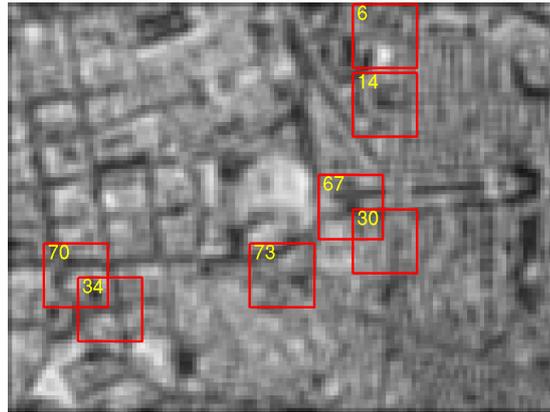


(f) Cluster 10

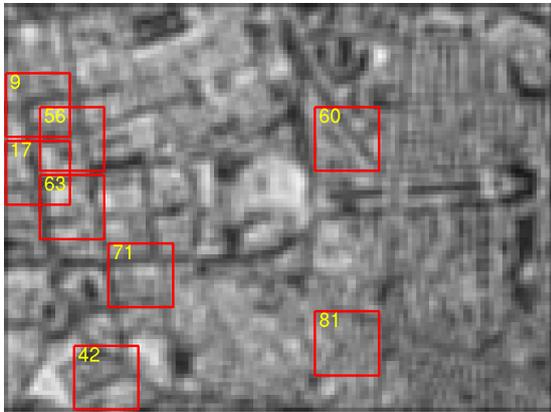
Figure A.45.: Example for landmark selection by k-means clustering continued



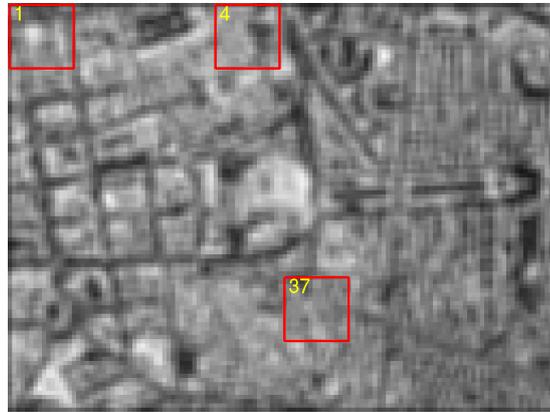
(a) Cluster 11



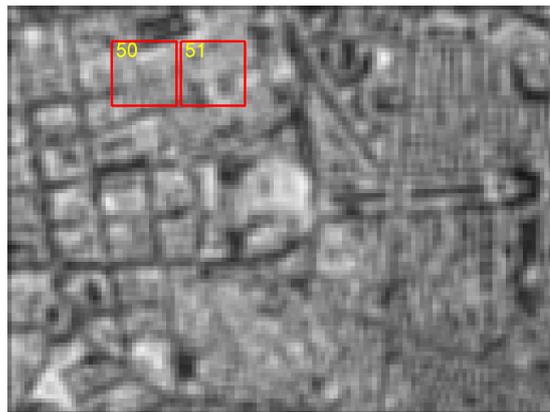
(b) Cluster 12



(c) Cluster 13



(d) Cluster 14



(e) Cluster 15

Figure A.46.: Example for landmark selection by k-means clustering continued

B. Tracking algorithm

Here we are going to present the ideas for a visual path integration algorithm. First we planned to use the algorithm for navigation between the known nodes of the topological map. But since landmark selection was a harder problem than we expected, the algorithm was neither completed nor simulated or real world experiments were done.

For the tracking of points we assume that the blimp only moves forward. Therefore we divide the camera image into several regions: The “incoming-region”, the “center-region” and the “outgoing-region”. The arrangement of the regions is shown in figure B.1. In each step of the algorithm, the corners in the center region are tracked. If the blimp moves forward, new image features will appear there. Therefore we run a corner detector in this image region, detecting new features. These features will be tracked in the next step and need to have a certain distance from points that are already tracked. If the blimp moves some points will move from the center-region to the outgoing-region. Corners in this region of the camera image are discarded, because they will disappear within the next tracking steps. Because the assumption that the blimp only moves forward is sometimes violated because of drifting there can occur situations, where only little corners are in the center-region. Then the corner detector is restarted, detecting corners in the center-region.

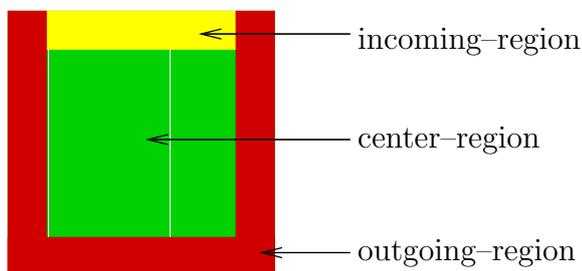


Figure B.1.: Regions for the tracking algorithm

As corner detector, an implementation of the Harris Corner Detector (Harris and Stephens, 1988) and as feature tracker an implementation of the Lucas Kanade Feature Tracker (Shi and Tomasi, 1994) is used¹. Without optimizing the code, real-time tracking is possible for images sized 200×200 pixels. An example sequence is shown in figure B.2.

¹Both implementations can be downloaded at <http://omni.isr.ist.utl.pt/~alex/resources.html>

Algorithm B.1: Skeleton for visual path integration algorithm

```

1:  $R_c = \text{detectCorners}(\text{center-region})$ 
2: repeat
3:    $P = \text{trackCorners}(R_c)$ 
4:    $R_o = \text{findPointsInRegion}(P, \text{outgoing-region})$ 
5:    $R_c = P \setminus R_o$ 
6:   if  $\|R_c\| \leq \text{cornerTreshold}$  then
7:      $R_c = \text{detectCorners}(\text{incoming-region})$ 
8:   else
9:      $I = \text{detectCorners}(\text{center-region})$ 
10:    for all  $c \in I$  do
11:       $d = \min_{f_i \in R_o \cup I} (\text{dist}(c, f_i))$ 
12:      if  $d \leq d_{\text{treshold}}$  then
13:         $I = I \setminus \{c\}$ 
14:      end if
15:    end for
16:     $R_c = R_c \cup R_I$ 
17:  end if
18: until end of tracking

```

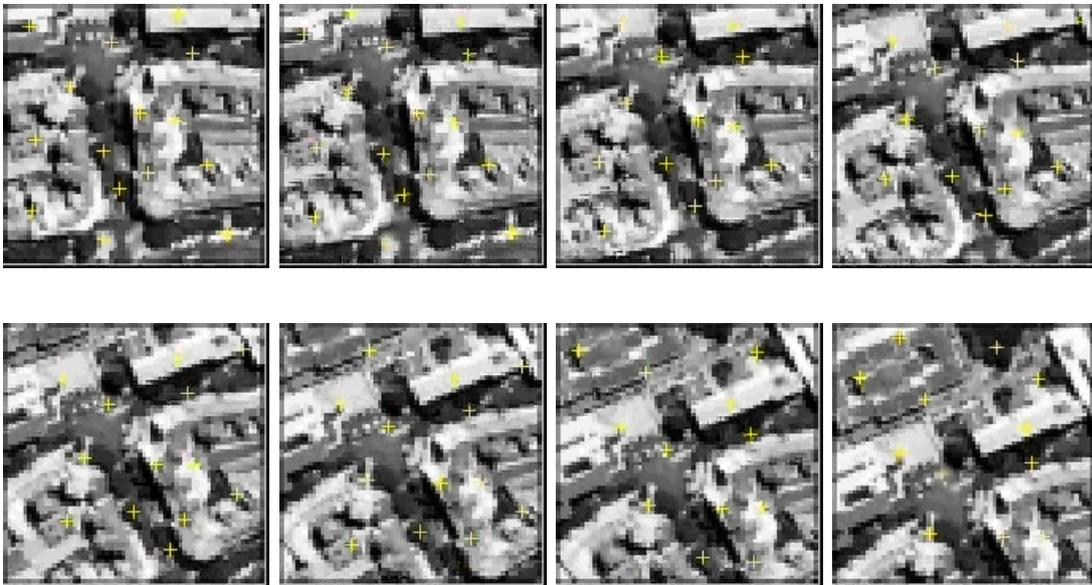


Figure B.2.: Example tracking sequence, frames 1 to 8



Figure B.3.: Example tracking sequence, frames 9 to 28

C. MATLAB files for the experiments

In this chapter we will give a rough overview over the developed MATLAB files and functions. It is supposed to be a help for those people that have to built up on this work, and therefore is not interesting for other readers.

The working directory includes four subdirectories. In the `data`-directory, everything that is used for further computations is kept to save computation time. In the `results`-directory the results of the experiments are kept. Both directories have subdirectories according to the scale factor of the data. The `results` directory has another `results`-subdirectory where results that are not specific for a certain scale factor are stored. The third directory is the seldom used `backup`-directory. The `graveyard`-directory is used to keep all the files that were declared obsolete or that were only used to visualize things for that report.

In the following tables a short description of the files and functions is given. More information is available by using `help filename` in the MATLAB shell. Even more information is available by looking at the source code.

Table C.1.: Special files or functions

File	Description
<code>make.m</code>	Some kind of LINUX-goody in the M\$ world. Manages all the necessary computations.
<code>image_viewer.m</code>	Script that helps to browse the computed data.
<code>README.TXT</code>	The results and overview of the work. Related to section 3. For a complete overview over the work and the data please refer to this report or to Gerstmayr et al. (2004).
<code>Contents.m</code>	List of the filenames. Similar to this section.
<code>ist_map_bw.png</code>	The aerial image used.

Table C.2.: Functions for fundamental computations.

Function	Description
<code>create_training_set</code>	Takes the views out of the floormap.
<code>create_pca</code>	Computes the SVD for a given training set and creates the transformation matrix to PCA-space.
<code>transform_views</code>	Transforms the views to the PCA-space with a certain dimension.
<code>pca_transformation_matrix</code>	Computes the transformation matrix that is necessary to transform views.
<code>visualize_explain</code>	Visualizes the variance covered by the first n principal components.
<code>analyse_dimension</code>	Further visualization of the variance covered by the first principal components.
<code>create_all_subviews</code>	Alternative function to take views out of floormap.

Table C.3.: Functions related to landmark selection.

Function	Description
<code>visualize_distance_matrix</code>	Visualization and computation of distance matrices.
<code>visualize_profiles</code>	Computation of profile matrices and visualization as single line plots.
<code>visualize_profiles_2</code>	Alternative visualization of the profile matrices as surfaces.
<code>profile_rank_max</code>	Performs profile-based ranking to find best landmarks.
<code>profile_rank_min</code>	Performs profile-based ranking to find worst landmarks.
<code>ipca_ranking</code>	Computes the IPCA-based landmark selection.
<code>compare_ranks</code>	Function that compares the different selection methods.
<code>evaluate_ranking</code>	Computes tables of selected landmarks.

Table C.4.: Functions related with reliability evaluation.

Function	Description
<code>evaluate_catchment_area</code>	Evaluation for deviations in robot position.
<code>evaluate_gamma_change</code>	Evaluation of deviations in image brightness.
<code>evaluate_landmark_dissimilarity</code>	Evaluation of average landmark dissimilarity.
<code>evaluate_rotation</code>	Evaluation of deviations in orientation.
<code>evaluate_scaling</code>	Evaluation of deviations in altitude.
<code>analyse_catchment</code>	Visualization of possible deviations in robot position.
<code>analyse_catchment_dim</code>	Same, but varying scale.
<code>analyse_gamma</code>	Visualization of possible deviations in image brightness.
<code>analyse_gamma_dim</code>	Same, but varying scale.
<code>analyse_landmark_similarity</code>	Visualizes average landmark similarity.
<code>analyse_rotation</code>	Visualization of possible deviations in orientation.
<code>analyse_rotation_dim</code>	Same, but varying scale.
<code>analyse_scaling</code>	Visualization of possible deviations in altitude.
<code>analyse_scaling_dim</code>	Same, but varying scale.

Table C.5.: Functions related to localization experiments

Function	Description
<code>visualize_localization</code>	Compute localization matrices.
<code>visualize_localization_matrix</code>	Visualize localization matrices.
<code>analyse_localization_matrix</code>	Numerical analysis of localization matrix.
<code>create_pca_localization_matrix</code>	Compute localization matrices for several dimensions.
<code>visualize_pca_localization_matrix</code>	Visualizes these matrices.
<code>analyse_pca_localization_matrix</code>	Analyzes these matrices numerically.
<code>analyse_peaks_8</code>	Evaluation of peaks in a matrix.
<code>final_evaluation</code>	Combines landmark selection and peak evaluation.
<code>peak_detection</code>	Function to detect minima in a localization matrix.

Table C.6.: Functions related with landmark clustering

Function	Description
<code>visualize_scatterplots</code>	Visualizes 2D or 3D scatter-plots.
<code>visualize_cophenet</code>	Visualizes and computes cophenetic matrices.
<code>analyse_cophenet_matrices</code>	Searches for optimal clustering parameters.
<code>visualize_clustered_scatterplot</code>	Visualization of clustering results.
<code>visualize_dendrogram</code>	Visualizes dendrograms.
<code>visualize_kmeans_clustering</code>	Experiments and visualization for k-means clustering.
<code>centroids2viewnumbers</code>	Searches best view for given image coordinates.

Table C.7.: Auxiliary functions

Function	Description
<code>ssd</code>	Computes the SSD between two vectors.
<code>mssd</code>	Computes the SSD between two matrices.
<code>find_row_index</code>	Function that searches for a row vector in each column of a matrix.
<code>int2framenummer</code>	Formatting function.
<code>pick_given_element</code>	Function that picks an element of a vector.
<code>mark_matrix_elements</code>	Function that marks matrix elements that are smaller then a given threshold.
<code>mark_matrix_tresholds</code>	Marks matrix elements according to a given threshold.
<code>pcacov</code>	Computes PCA out of a given covariance matrix.
<code>randint</code>	Computes a matrix of random integers.
<code>randn_bound</code>	Computes a limited Gaussian random distribution.
<code>draw_rectangle</code>	Draws a rectangle.
<code>auto_crop</code>	Crops the boarder around an image.
<code>auto_crop_current_dir</code>	Crops all the images in the current directory.

Bibliography

- C.S. Andersen, S.D. Jones, and J.L. Crowley. Appearance based processes for visual navigation. In *Proceedings of the SIRS '97*, pages 227–236, 1997.
- M. Artac, M. Jogan, and A. Leonardis. Incremental pca for on-line visual learning and recognition. In *Proceedings of the ICPR 2002*, pages 781–784, 2002a.
- M. Artac, M. Jogan, and A. Leonardis. Mobile robot localization using an incremental eigenspace model. In *Proceedings of the ICRA 2002*, pages 1025–1030, 2002b.
- T. Bailey and E. Nebot. Localization in large-scale environments. *Robotics and Autonomous Systems*, 37:261–281, 2001.
- A. Bernardino, L. Custódio, J. Frazão, T. Krause, P. Lima, M. I. Ribeiro, and A. Vale J. Santos-Victor and. 2nd year technical report. Technical report, Instituto de Sistemas e Robotica, Instituto Superior Tecnico, Lissabon, 2003.
- M. Betke and Leonid Gurvits. Mobile robot localization using landmarks. *IEEE Transaction on Robotics and Automation*, 13:251–263, 1997.
- N. Bolshakova and F. Azuaje. Improving expression data mining through cluster validation. In *Proceedings of the 4th Annual IEEE Conference on Information Technology Applications in Biomedicine*, pages 19–22, 2003.
- E. Bourque, G. Doudek, and P. Ciaravola. Robotic sightseeing – a method for automatically creating virtual environments. In *Proceedings of the ICRA 1998*, pages 3186–3191, 1998.
- E. Bourque and G. Dudek. On-line construction of iconic maps. In *Proceedings of the ICRA 2000*, pages 2310–2315, 2000.
- D. Burschka, J. Geiman, and G. Hager. Optimal landmark configuration for vision-based control of mobile robots. In *Proceedings of the ICRA 2003*, pages 3917–3922, 2003.
- V.C. de Verdiere and J.L. Crowley. Local appearance space for recognition of navigation landmarks. In *Proceedings of the SIRS 1998*, pages 261–269, 1998.
- X. Deng, E. Milios, and A. Mirzaian. Landmark selection strategies for path execution. *Robotics and Autonomous Systems*, 17:171–185, 1996.

- R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley–Interscience, 2001.
- G. Dudek and D. Jugessur. Robust place recognition using local appearance based methods. In *Proceedings of the ICRA 2000*, pages 1030–1035, 2000.
- B.S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Edward Arnold, London, 2001.
- M. O. Franz and H. A. Mallot. Biomimetic robot navigation. *Robotics and autonomous Systems*, 30:133–153, 2000.
- R. Freitas, J.A. Santos-Victor, M. Sarcinelli-Filho, and T. Bastos-Filho. Performance evaluation of incremental eigenspace models for mobile robot localization. In *Proceedings of the ICAR 2003*, pages 417–422, 2003.
- U. Frese and G. Hirzinger. Simultaneous localization and mapping - a discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics, Seattle*, pages 17 – 26, 2001.
- J. Gaspar, N. Winters, and J.A. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions on Robotics and Automation*, 16:890–898, 2000.
- L. Gerstmayr, A. Bernardino, and J. Santos-Victor. Appearance based landmark selection and reliability evaluation for topological navigation. In *Proceedings of the IFAC04, accepted*, 2004.
- N. Gracias. *Mosaic-based Visual Navigation for Autonomous Underwater Vehicles*. PhD thesis, Instituto Superior Técnico, Lisbon, Portugal, 2002.
- R. Greiner and R. Isukapalli. Learning to select useful landmarks. In *Proceedings of the AAAI 1994*, pages 1251–1256, 1994.
- C. Grigorescu, N. Petkov, and M.A. Westenberg. Improved contour detection by non-classical receptive field inhibition. In *Proceedings of the BMCV 2002*, pages 50–59, 2002.
- P. M. Hall, A. D. Marshall, and R. R. Martin. Incremental eigenanalysis for classification. In *Proceedings of the BMCV 1998*, pages 286–295, 1998.
- C.J. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.
- E. Hugues, F. Guilleux, and O. Rochel. Contour detection by synchronization of integrate-and-fire neurons. In *Proceedings of the BMCV 2002*, pages 60–69, 2002.
- F. Iida. Biologically inspired visual odometer for navigation of a flying robot. *Robotics and Autonomous Systems*, 44:201–208, 2003.

- M. Jogan and A. Leonardis. Robust localization using eigenspace of spinning-images. In *In Proceedings of IEEE Workshop on Omnidirectional Vision*, pages 37–44, 2000.
- A.E. Johnson. Surface landmark selection and matching in natural terrain. In *Proceedings of the CVPR 2000*, pages 413–420, 2000.
- D. Jugessur and G. Dudek. Local appearance for robust object recognition. In *Proceedings of the CVPR 2000*, pages 834–839, 2000.
- A. Kelly. Mobile robot localization from large scale appearance mosaics. *International Journal of Robotics Research*, 19:1104–1125, 2000.
- M. Knapek, R.S. Oropeza, and D.J. Kriegmann. Selecting promising landmarks. In *Proceedings of the ICRA 2000*, pages 3771–3777, 2000.
- M. Kolesnik, A. Barlit, and E. Zubkov. Iterative tuning of simple cells for contrast invariant edge enhancement. In *Proceedings of the BMCV 2002*, pages 27–37, 2002.
- E. Krotkov. Mobile robot localization using a single image. In *Proceedings of the ICRA '89*, pages 978–983, 1989.
- J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7:376–382, 1991.
- A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78:99–118, 2000.
- Z. Li. Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex. *Computation in Neural Systems*, 10:187–212, 1999.
- P. Lima, M. I. Ribeiro, L. Custodio, and J. Santos-Victor. The rescue project – cooperative navigation for rescue robots. In *Proceedings of the ASER'03*, 2003.
- J.J. Little, J. Lu, and D.R. Murray. Selecting stable image features for robot localization using stereo. In *Proceedings of the IROS 98*, pages 1072–1077, 1998.
- L. Lucchese and S.K. Mitra. Color image segmentation: A state-of-the-art survey. *Image Processing, Vision and Pattern Recognition*, 2003. to appear.
- M. Mata, J.M. Armingol, A. de la Escalera, and M.A. Salichs. A visual landmark recognition system for topological navigation of mobile robots. In *Proceedings of the ICRA 2001*, pages 1124–1129, 2001.
- MathWorks. *Statistics Toolbox for use with MATLAB*. The MathWorks, 2002.
- H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal on Computer Vision*, 14:5–24, 1995.

- R.R. Murphy, D. Hershberger, and G.R. Blauvelt. Learning landmark triples by experimentation. *Robotics and Autonomous Systems*, 22:377–392, 1997.
- H. Neumann and W. Sepp. Recurrent v1–v2 interaction for early vision information processing. In *Proceedings of the ESANN'1999*, pages 165–170, 1999.
- K. Ohba and K. Ikeuchi. Recognition of the multi specularity objects using the eigen window. In *Proceedings of the ICPR 1996*, 1996.
- K. Ohba and K. Ikeuchi. Detectability, uniqueness and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1043–1048, 1997.
- C. Olson. Selecting landmarks for localization in natural terrain. *Autonomous Robots*, 12:201–210, 2002.
- C.F. Olson. Mobile robot self-localization by iconic matching of range maps. In *Proceedings of the ICAR 1997*, pages 447–452, 1997.
- A. Rencher. *Methods of Multivariate Analysis*. Wiley–Interscience, 2002.
- A. Rupa, A. Leonardis, and N. Mramor Kosta. Robust recognition using the tensor-rank principle. In *Proceedings of the 26th Workshop of the Austrian Association for Pattern Recognition*, pages 45–52, 2002.
- C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535, 1997.
- C. Schmid, R. Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal on Computer Vision*, 37:151–172, 2000.
- S. Se, D. Lowe, and J. Little. Vision–based mobile robot localization and mapping using scale–invariant features. In *Proceedings of the ICRA 2001*, pages 2051–2058, 2001.
- A. Shaw and D. Barnes. Landmark recognition for localisation and navigation of aerial vehicles. In *Proceedings of the ASTRA 2002*, pages 427–434, 2002.
- J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR'94*, pages 593–600, 1994.
- R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proceedings of the ICRA 1998*, pages 1060–1065, 1998.
- R. Sim, S. Polifroni, and G. Dudek. Comparing attention operators for learning landmarks. Technical Report TR–CIM–03–03, McGill University, Quebec, Canada, 2003.
- S. Simhon and G. Dudek. Selecting targets for local reference frames. In *Proceedings of the ICRA 1998*, pages 2840–2845, 1998.

- Wolfgang Stürzl. *Sensorik und Bildverarbeitung für landmarkenbasierte Navigation*. PhD thesis, Universität Tübingen, 2003.
- K.T. Sutherland and W.B. Thompson. Localizing in unstructured environments: Dealing with errors. *IEEE Transactions on Robotics and Automation*, 10:740–755, 1994.
- Y. Takeuchi, Patrick Gros, M. Hebert, and K. Ikeuchi. Visual learning for landmark recognition. In *Proceedings of DARPA Image Understanding Workshop*, pages 1467–1474, 1997.
- A. Thielscher, A. Schubue, and H. Neumann. A neural model of human texture processing: Texture segmentation vs. visual search. In *Proceedings of the BMCV 2002*, pages 99–108, 2002.
- S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33:41–76, 1998.
- M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.
- I. Ulrich and Illah Nourbakhsh. Appearance-based place recognition for topological navigation. In *Proceedings of the ICRA 2000*, pages 1023–1029, 2000.
- A. Vale and M.I. Ribeiro. Environment mapping as a topological representation. In *Proceedings of the ICAR '03*, pages 29–34, 2003.
- R.F. Vasallo, J.A. Santos-Victor, and H.J. Schneebeli. Using motor commands for topological mapping and navigation. In *Proceedings of the 2002 ICIRS 2002*, pages 478–483, 2002.
- L. Vincent and P. Soille. Watersheds in digital images: An efficient algorithm based on immersion simulations. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13:583–598, 1991.
- N. Winters and J.A. Santos-Victor. Information sampling for vision-based robot navigation. *Robotics and Autonomous Systems*, 41:145–159, 2002.
- E. Yeh. Toward selecting and recognizing natural landmarks. Technical Report Technical Report 9503, Departement of Electrical Engeneering, Yale University, 1995.
- K.J. Yoon and I.S. Kweon. Color image segmentation considering of human sensitivity for color pattern variations. In *Proceedings of the SPIE2001*, pages 269–278, 2001.
- K.J. Yoon, I.S. Kweon, C.Y. Kim, and Y.S. Seo. Moving object segmentation based on human visual sensitivity. In *Proceedings of the BMCV 2000*, pages 62–72, 2000.