

Klassifikation des Architekturstils von Gebäudefassaden

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin der technischen Wissenschaften

eingereicht von

Gayane Shalunts

Matrikelnummer 0929213

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.Ing. Dr.techn. Robert Sablatnig
Betreuung: Ao.Univ.Prof. Dipl.Ing. Dr.techn. Markus Vincze

Diese Dissertation haben begutachtet:

(Ao.Univ.Prof. Dipl.Ing.
Dr.techn. Robert Sablatnig)

(Ao.Univ.Prof. Dipl.Ing.
Dr.techn. Markus Vincze)

Wien, 9.04.2015

(Gayane Shalunts)

Architectural Style Classification of Building Facades

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doctor of technical sciences

by

Gayane Shalunts

Registration Number 0929213

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.Ing. Dr.techn. Robert Sablatnig

Advisor: Ao.Univ.Prof. Dipl.Ing. Dr.techn. Markus Vincze

The dissertation has been reviewed by:

(Ao.Univ.Prof. Dipl.Ing.
Dr.techn. Robert Sablatnig)

(Ao.Univ.Prof. Dipl.Ing.
Dr.techn. Markus Vincze)

Vienna, 9.04.2015

(Gayane Shalunts)

Erklärung zur Verfassung der Arbeit

Gayane Shalunts
Joergerstrasse 24/13, 1170 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasserin)

Acknowledgements

Acknowledgements to the Doctoral College on Computational Perception, Vienna University of Technology.

Abstract

Architectural styles are phases of development that classify architecture in the sense of historic periods, regions and cultural influences. The objective of the dissertation is to build a computer vision software tool, which classifies the architectural style of a Romanesque, Gothic or Baroque building facade, given its image. The work presents the first fundamental algorithm for constructing an image-based architectural style classification software system. The algorithm proposed is general enough to target the classification of a building facade of any visually distinguishable architectural style or displaying a mixture of architectural styles. The universality of the algorithm is also based on its capability to classify images of full facades, partly occluded facades or facade parts. The architectural style of a building is formed by a combination of style typical component parts, called architectural elements. The algorithm for building facade architectural style classification is a style voting scheme of separate architectural elements, such as windows, domes, towers, etc. It is structured as subsequent principle steps of segmentation, classification and voting of architectural elements. The first approaches addressing the segmentation of prominent architectural elements dome and tower are introduced within the bounds of the segmentation step. Each segmentation algorithm is a pipeline unifying bilateral symmetry detection, graph-based segmentation approaches and image analysis and processing techniques, taking into account the visual specificities of the element segmented. The system embeds the knowledge about architectural styles by learning style typical architectural elements in the classification stage. Taking into consideration the grand scale of the work amount, required for the realization of the algorithm for any architectural style, the software implementation is limited to three pan-European architectural styles of major significance, each spanning a few centuries and covering large geographical areas, namely Romanesque, Gothic or Baroque.

For testing and performance evaluation of each module of the project, image databases of building facades belonging to the corresponding architectural styles and featuring the explored architectural elements were gathered. The experiments report high segmentation and classification precision, as well as prove the algorithm of the voting of architectural elements to be effective and promising in regard to the further expansion of the project by new architectural styles.

Kurzfassung

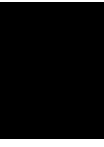
Architekturstile sind Entwicklungsphasen, welche Architektur nach geschichtlichen Epochen, Regionen und kulturellen Einflüssen klassifizieren. Das Ziel der Dissertation liegt im Aufbau eines Computer Vision Software Werkzeugs, welches basierend auf Bildern Fassaden romanischen, gotischen und barocken Stilen klassifizieren kann. Die Arbeit beschreibt den ersten fundamentalen Algorithmus für die Schaffung eines Software Systems zur bildbasierten Klassifizierung von Architekturstilen. Durch seine Allgemeinheit kann der vorgeschlagene Algorithmus eine Klassifizierung jeder Fassade mit optisch erkennbarem architektonischen Stil oder einer Mischung von verschiedenen architektonischen Stilen vornehmen. Die allgemeine Verwendbarkeit dieses Algorithmus basiert auch auf der Fähigkeit, Bilder von ganzen Fassaden, teilweise verdeckten Fassaden oder gar Fassadenteilen vorzunehmen. Der architektonische Stil eines Gebäudes wird durch eine Kombination von typischen Einzelteilen, so genannten architektonischen Elementen, definiert. Der Algorithmus zur Klassifizierung von architektonischen Stilen von Gebäudefassaden ist ein Abstimmungsschema einzelner architektonischer Elemente wie Fenster, Kuppeln, Türmen, usw. Er ist strukturiert als eine Folge von Hauptphasen: Segmentierung, Klassifizierung und Abstimmung von Architekturelementen. Im Rahmen der Segmentierungsphase werden erste Ansätze für die Segmentierung der hervorstechenden Architekturelemente Kuppel und Turm präsentiert. Jeder Segmentierungsalgorithmus ist eine Arbeitsabfolge, welche bilaterale Symmetrierkennung, graphbasierende Segmentierungsansätze sowie Bildanalyse und Bildverarbeitungstechniken vereint, und dabei die visuellen Eigenheiten des zu segmentierenden Elementes berücksichtigt. Das System bindet Wissen um Architekturstile durch Erlernen typischer Architekturelemente im Rahmen der Klassifizierungsphase ein. In Anbetracht des beträchtlichen Arbeitsaufwandes zur Implementierung des Algorithmus für jede Architekturstile beschränkt sich die Software auf drei bedeutende paneuropäische Stile, von denen sich jeder über Jahrhunderte hinweg über große Teile des Kontinents erstreckt: Romanik, Gotik und Barock.

Zum Testen und Beurteilen der Leistungsfähigkeit jedes Projektmoduls wurde eine Bild-datenbank mit zu den unterschiedlichen architektonischen Stilen dazugehörenden und die charakteristischen architektonischen Elemente beinhaltenden Gebäudefassaden geschaffen. Experimente belegen eine hohe Präzision in der Segmentierung und Klassifizierung und beweisen, dass der Algorithmus der Abstimmung der Architekturelemente effektiv und vielversprechend bezüglich der künftigen Erweiterbarkeit des Projekts mit neuen Architekturstilen ist.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement	2
1.3	Aim of the Work	5
1.4	Challenges	5
1.5	Methodological Approach	6
1.6	Contributions of the Work	8
1.7	Structure of the Work	8
2	State of the Art / Analysis of Existing Approaches	9
2.1	Literature Studies	9
2.2	Architectural Style Recognition of Flemish Renaissance, Haussmannian and Neoclassical Building Facades	10
2.3	Literature Studies Related to Building Facade Analysis	15
2.4	Summary	18
3	The Architectural Aspect of the STYLE Project	19
3.1	How to Distinguish Architectural Styles	19
3.2	Romanesque, Gothic and Baroque Architectural Styles	20
3.3	Typical Windows of Romanesque, Gothic and Baroque Architectural Styles	21
3.4	Traceries, Pediments and Balustrades	22
3.5	Renaissance, Russian and Islamic Domes	25
3.6	Romanesque, Gothic and Baroque Towers	27
3.7	Summary	27
4	Methodology: The Voting of Architectural Elements	29
4.1	The Algorithm of the Voting of Architectural Elements	29
4.2	Segmentation of Domes	33
4.3	Segmentation of Towers	37
4.4	Architectural Style Classification of Building Facade Windows	43
4.5	Architectural Style Classification of Traceries, Pediments and Balustrades	45
4.6	Architectural Style Classification of Domes	46
4.7	Architectural Style Classification of Towers	48

4.8	The STYLE GUI and the Scheme of the Voting of Architectural Elements . . .	50
4.9	Summary	51
5	Experiments, Performance Evaluation and Results	55
5.1	The Experimental Setup and Creation of Image Databases	55
5.2	Experiments and Evaluation of Dome Segmentation	56
5.3	Experiments and Evaluation of Tower Segmentation Approach	58
5.4	Experiments and Results of Window Classification Approach	61
5.5	Experiments and Results of Classification of Traceries, Pediments and Balustrades	62
5.6	Experiments and Results of Dome Classification	63
5.7	The Evaluation and Results of Tower Classification	66
5.8	Summary	67
6	Critical Reflection	69
6.1	Comparison with Related Work	69
6.2	Discussion of the Achievements and Open Issues	70
7	Summary and Future Work	73
A	Databases, source code	75
A.1	The List of the Buildings Featuring Domes	75
A.2	The List of the Buildings Featuring Towers	78
A.3	The Logo of the STYLE Project	80
A.4	Source Code	80
	Bibliography	87



Introduction

1.1 Motivation

The history of architecture reflects the history of mankind. The evolution of architectural styles, stretching from prehistoric times to our days, starting from habitations in caves and reaching today's impressive skyscrapers, displays the evolution of the human mind. Each architectural style defines certain forms, design rules, techniques and materials for building construction.

Images of building facades from online image databases usually do not have any labels related to architectural styles. To know the style of an observed building, one can search for the building name. Whereas this is a solution for famous buildings with well-known names, it is not applicable to buildings lacking names. In this case the visual information of the facade holds the only key to its style. Here an alternative to an architectural style visual classification tool could be the visual search engines like Google visual search ¹ or landmark recognition engines like [48]. The results of a Google visual search performance observation for the landmark building St.Stephen's Basilica in Budapest (Figure 1.1a) and an ordinary neo-Baroque building in Vienna (Figure 1.1b) are depicted in Figure 1.2 and Figure 1.3 respectively, demonstrating successful image mining for the landmark building and failure on the non-landmark. This example shows that those visual engines work well for world famous landmarks succeeding in mining not only related images, but also links, from which one can obtain all the information about the building of interest, including its name and architectural style. Nevertheless they fail for ordinary buildings, not even delivering facades belonging to the same architectural style. An automatic tool for image-based classification of architectural styles will solve this problem, not depending on building popularity or its importance as a cultural heritage object. It will also allow indexing of building databases into categories belonging to certain historic periods. This kind of a semantic categorization limits the search of building image databases to certain category portions for the purposes of building recognition [47, 48], Content Based Image Retrieval (CBIR) [21], 3D image-based city-modeling [7, 42] and virtual tourism [41]. An architectural

¹Google Visual Search Engine <http://images.google.com>



a) The landmark building St. Stephen's Basilica



b) An ordinary neo-baroque building

Figure 1.1: Examples of a landmark and an ordinary neo-baroque building

style classification system may also find its application in real tourism if provided with smart phones. Another motivation to automate the recognition of architectural style of building facades, as stated in [27], is that such a recognition is required for building reconstruction by the means of inverse procedural modeling.

The segmentation of building facades, which is one of the three main stages of the methodology proposed by this work, is an active research field in computer vision itself [42]. It is vital for image-based modeling and urban scene understanding [42].

1.2 Problem Statement

Despite the considerable amount of publications related to facade visual analysis, the image-based architectural style classification of facades remains an open problem. The dissertation addresses the problem stated. To accomplish the desired generalization one should begin with exploring the architectural styles and the principles of distinguishing them from one another. This is a challenging task for the following reasons:

- There are decades of architectural styles.
- Historic architectural styles contain principle phases, e.g. Gothic architecture counts historic phases of Early Gothic and High Gothic.
- Each architectural style displays regional signature of building design and materials.
- A mixture of architectural styles in a single building is a common phenomenon, since the sequence of historic periods, reigns and cultural influences of the observed region

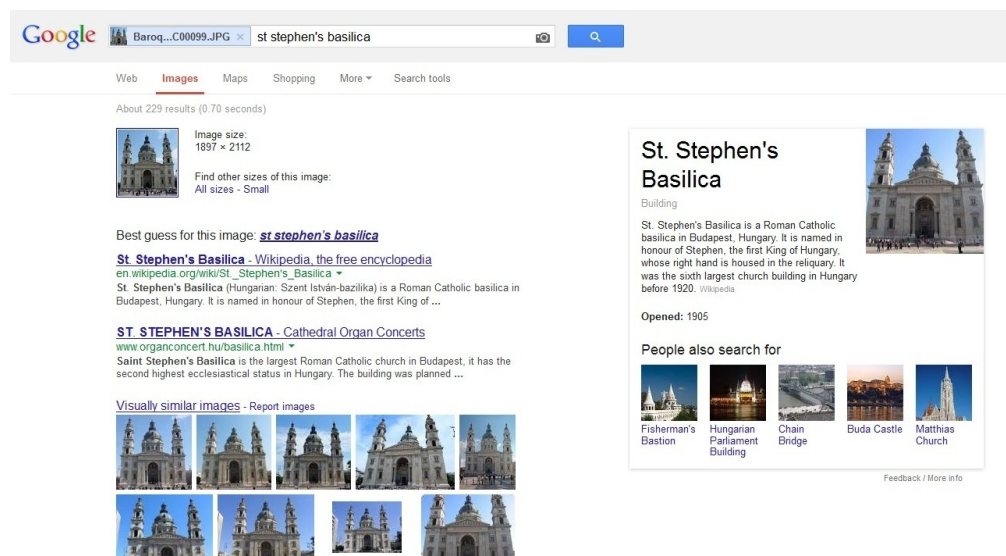


Figure 1.2: Successful Google visual search results for the landmark building St. Stephen's Basilica

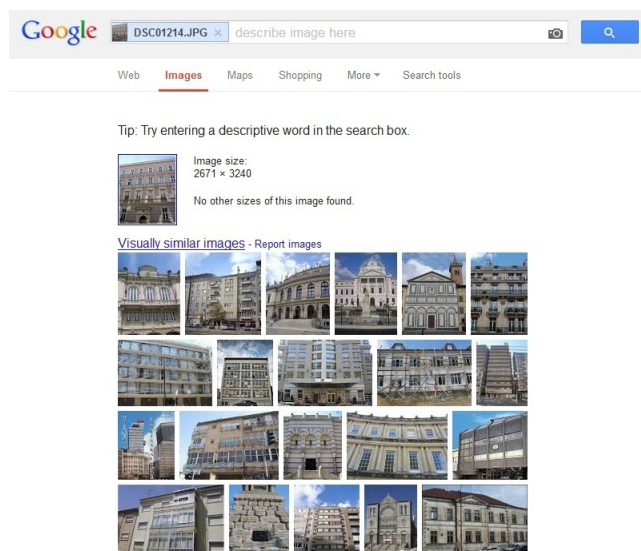


Figure 1.3: Unsuccessful Google visual search results for a non-landmark building

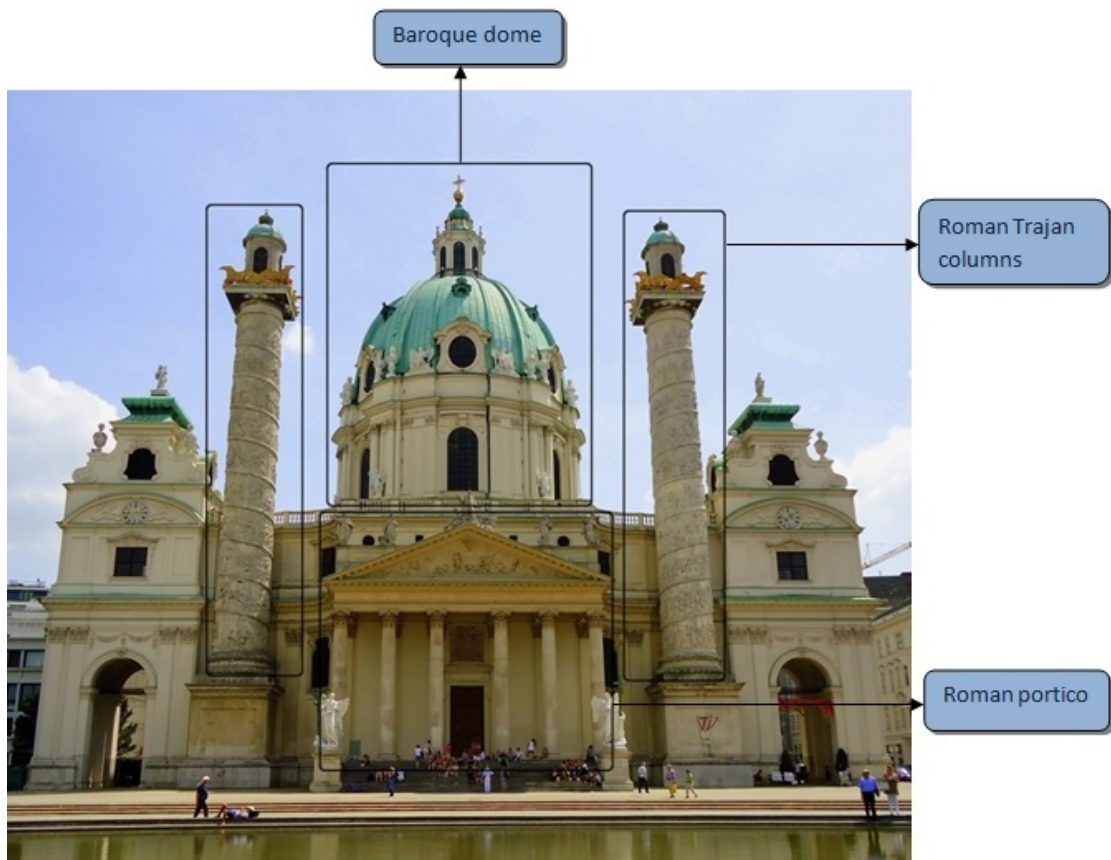


Figure 1.4: St. Charles church - a mixture of architectural styles

led to the evolution of styles from one another. The monumental and unique church of St. Charles Borromeus in Vienna (Karlskirche), as a building instance representing the phenomenon of the mixture of architectural styles, is depicted in Figure 1.4. Though the church is considered Baroque, it also features a Roman portico and double Roman Trajan columns (Figure 1.4). With an altogether unconventional plan, of which an oval forms the central space, and with similar freedom in design of dome and towers, it combines not only a pure Roman portico, but actually a pair of tall Doric columns standing away from the building and recording on spiral basreliefs the various doings of the titular saint of the church after the manner of Trajan's famous column [5].

The image acquisition in its turn introduces additional problems:

- Perspective distortions
- Illumination variations
- Unlimited number of possible 2D projections of 3D facades

The algorithm should succeed in incorporating the definitions of architectural styles in the context of a computer vision system. A 'smart' algorithm, differing from standard learning algorithms solving classification problems, must embed architectural knowledge in its core. The requirements to the algorithm also include the ability to classify:

- Planar and non-planar facades

The analysis of non-planar facades is more challenging, since the detection of dominating horizontal and vertical partitions and rectification are not applicable to them.

- Landmark and ordinary buildings

As observed in Section 1.1, it is possible to find out the architectural style of a landmark building via image mining. The ordinary buildings, which lack names, are not photographed frequently and posted online present the issue.

- Images of full facades, partly occluded facades and facade parts

Facades may be partly occluded in images due to scaffolding, advertisements or objects present in the field of view of the camera. The algorithm should ignore these occlusions as irrelevant visual features. The classification of facade parts must also be possible, if they represent style-typical visual information.

For the evaluation of the algorithm and the designed software tool the challenge of collecting an image database of the examined architectural styles should be overcome, as such a dataset does not exist so far. The image database must contain a sufficient number of buildings from diverse geographical regions to prove the claimed general character of the algorithm.

1.3 Aim of the Work

The goal of the work is to build a software system classifying the architectural style of building facades based on their visual information contained in images. The algorithm for solving the problem stated is intended to be general enough to cover the classification of any architectural style with distinguishable visual features. The software implementation of the algorithm will be realized on selected influential architectural styles. The general character of the algorithm and the modularity of the software permit further expansion of the system with new architectural styles.

1.4 Challenges

Image-based architectural style classification of facades is challenging for the following reasons:

- This is a classification problem of numerous classes, as there are decades of architectural styles.

- Big intra-class diversity is an unwanted factor for any classification problem. The intra-class variety is immense, because of the following factors:
 - 1) Historic architectural styles contain principle phases.
 - 2) Regional differences of historic architectural styles.
 - 3) Almost no buildings repeat each other.
 - 4) The unlimited number of possible 2D projections (images) of 3D facades.
 - 5) The visual intricacy of historic architectural styles.
- Since the mixture of architectural styles is a common phenomenon, the classification output can be not only a single class, but a combination of classes.
- Not everything on a facade presents architectural style relevant visual information. As an example, building first floors, hosting stores, cafes, restaurants, hotel lobbies, etc., are a well-known issue in facade visual processing problems, as stated in [27]. Images of partly occluded facades also carry architectural style irrelevant visual features. Here the challenge lies in the selection of the relevant visual features and ignoring the irrelevant ones.

1.5 Methodological Approach

The presented architectural style classification project is called STYLE. The work is a unique attempt to bridge science and art, combining computer vision technologies and programming skills with architecture and photography.

The method of the STYLE project for architectural style classification consists of three major steps (Figure 1.5):

- Facade segmentation by architectural structural elements
- Classification of the segmented elements by architectural styles
- Style voting of the classified elements

The approach models the human logic for architectural style classification, that is the search of style typical components on a facade. The method, owing its modular character, is possible to extend by adding new architectural elements and architectural styles. It has the advantages of being capable to classify facade parts or partially occluded facades by a single typical architectural element, as well as facades of mixed architectural styles.

Architectural knowledge is embedded in all three stages of the method:

- The developed segmentation approaches in the first stage employ the visual architectural features of the segmented elements. Each segmentation method is a pipeline of symmetry detection and segmentation approaches, image analysis and processing techniques.

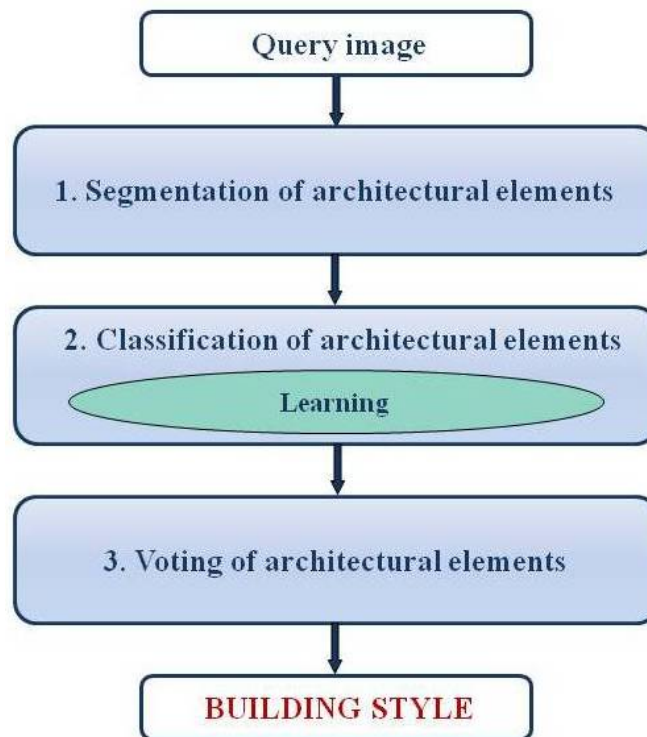


Figure 1.5: The diagram of the STYLE project method

- Learning of the style-typical architectural elements takes place in the classification step. The classification approaches are based on learning of local features, as well as analysis of the dimensions and color of the architectural element when appropriate.
- The architectural style winning in the voting stage by maximum of votes is chosen as the style of the query facade. The voting scheme permits the classification of only those combinations of architectural styles, which are probable in the history of architecture due to multiple examples.

Among different architectural styles there are such that have influenced large regions and dominated a few centuries. The software implementation of the algorithm addresses the classification of building facade images of such influential European architectural styles - Romanesque, Gothic, Baroque, as well as contains some comparative analysis of Renaissance, Russian and Islamic styles. Though the main focus are the buildings, which are representatives of cultural heritage, such as churches, cathedrals, palaces, museums and city halls, the approach is not limited to such buildings. The algorithm's capability to classify buildings of mixed architectural styles is realized by classifying the frequently met mixture of Romanesque and Gothic styles.

1.6 Contributions of the Work

The presented work makes the following contributions:

- The first general algorithm of image-based architectural style classification of building facades and its justification from the architectural analysis (Section 3.1, Section 4.1).
- The first algorithm addressing the segmentation of domes of Renaissance, Baroque, Neo-classical and Islamic architectural styles (Section 4.2, [39]).
- The first algorithm addressing the segmentation of towers of Romanesque, Gothic and Baroque architectural styles (Section 4.3).
- The first approach, classifying Romanesque, Gothic and Baroque windows (Section 4.4, [36]).
- The first approach, classifying Gothic traceries and Baroque pediments and balustrades (Section 4.5, [38]).
- The first approach, classifying Renaissance, Russian and Islamic domes (Section 4.6, [37]).
- The first approach, classifying Romanesque, Gothic and Baroque towers (Section 4.7).
- Integration of all above mentioned segmentation and classification approaches within one framework (Section 4.8).
- The realization of the voting mechanism of architectural elements for the classification of Romanesque, Gothic and Baroque architectural styles, as well as the combination of Romanesque and Gothic styles, demonstrating the algorithm's ability to classify mixture of architectural styles (Section 4.8).

1.7 Structure of the Work

The dissertation is organized as follows: Chapter 2 familiarizes with the state of the art literature related to the image-based architectural style classification. Chapter 3 examines the stated problem and the proposed algorithm for its solution from the architectural aspect, as well as gives an introduction to the classified architectural styles and definitions of the used architectural terms. Chapter 4 presents the innovation of the work - the first general methodology of architectural style classification of facades. The experimental setup, performance evaluation and results are reported in Chapter 5. Chapter 6 contains the comparison of the presented methodology with the related work and discussion. And finally Chapter 7 summarizes and concludes the work. A detailed description of the image databases used for the evaluation of the methodology and source code modules executing significant functions are brought in Appendix A.

State of the Art / Analysis of Existing Approaches

2.1 Literature Studies

The research related to architectural style classification of building facades is new in computer vision community, since to the best knowledge of the author the only approach in literature targeting the problem of automatic image-based architectural style classification of facades is presented in [27]. The architectural styles classified are Flemish Renaissance, Haussmannian and Neoclassical. The motivation of the work is to automate the recognition of the architectural style of building facades, required for the building reconstruction by the means of inverse procedural modeling.

Procedural modeling has proven to be a very valuable tool in the field of architecture [27]. It describes a building as a series of rules. Starting from a mere footprint or polyhedral approximation, finer detail is added when going from rule to rule. Procedural modeling is quite different from the traditional production of textured meshes. Procedural models are compact, semantically structured and can be easily altered and used to generate photorealistic renderings. Furthermore, they support a wide range of applications, from detailed landmark or building modeling to full-size megalopolis simulations. Whereas meshes or point clouds can be generated through dedicated mobile mapping campaigns, more precise and visually pleasing models were so far made manually. It takes several man years to model accurately an existing city, such as New York or Paris (e.g. 15 man years for the NY model in the King Kong movie) [27]. An alternative comes from inverse procedural modeling. This process aims to reconstruct a detailed procedural model of a building from a set of images, or even from a single image. Buildings are modeled as an instantiation of a more generic grammar. Considering the vast diversity of buildings and their appearances in images, the underlying optimization problem easily becomes intractable, if the search space of all possible building styles had to be explored. Thus, all currently available inverse procedural modeling algorithms narrow down their search by implicitly assuming an architectural style. The authors in [27] propose a four-stage method for automatic building

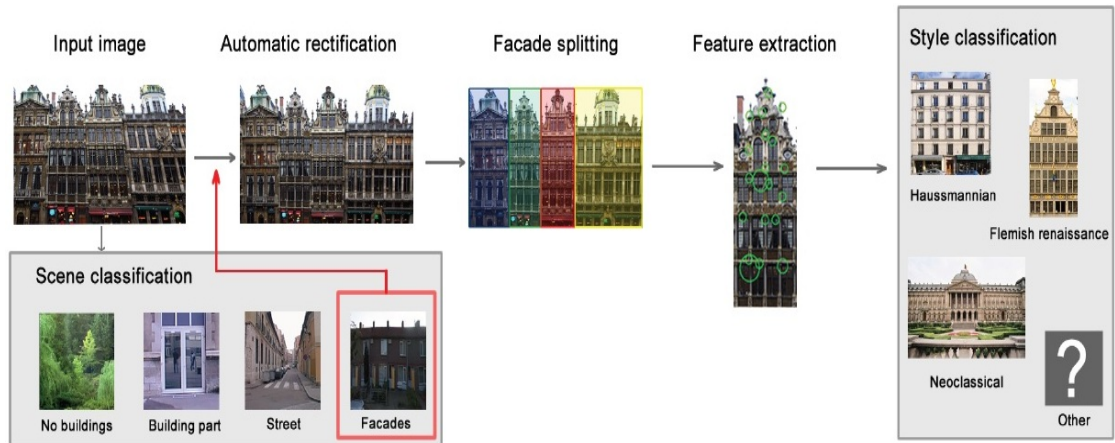


Figure 2.1: System overview of the architectural style classification approach by [27]

classification based on the architectural style. The style information can then be used to select the appropriate procedural grammar for the task of building reconstruction. The clarification of the method follows below.

2.2 Architectural Style Recognition of Flemish Renaissance, Haussmannian and Neoclassical Building Facades

The main goal of the work in [27] is to model cities from images, taken with cameras on a mobile mapping van. Therefore the authors look at the broader problem of selecting images that are useful for the modeling of buildings. It is to be expected that a significant number of photos taken from a mobile mapping van will not even contain buildings, but trees or only building parts. Figure 2.1 gives the system overview of the approach by [27]. The four successive stages of the method are scene classification, image rectification, facade splitting and style classification [27]. The first step in the approach is to determine if the image contains building facades. If this condition is met, the image will pass to the rectification stage, as the images of buildings taken from the street usually contain significant projective distortions. After the image is rectified, identification of individual buildings in the image takes place. In long, unbroken building blocks the architectural styles may vary from facade to facade [27]. Edge information is employed to find individual building separators. Finally features are extracted from the individual facades and a Naive-Bayes Nearest-Neighbor (NBNN) classifier is used to determine the architectural style of the facade.

Scene Classification

Mobile mapping images display different content and quality [27]. There are typically several cameras mounted on a van, with different viewing directions. Therefore, the first step in the process of building classification is the step of selecting from all the collected images a set of

images that contain objects of interest. It is desirable this step to be as fast as possible, due to the fact that it will have to deal with all images taken. On the other hand the algorithm should have good generalization to deal robustly with novel scenes.

It has been shown by [32] that humans have the capability of determine scene type in less than 200ms. This abstract representation of the scene is called gist and has served as a starting point for the development of numerous holistic algorithms for fast scene classification, which attempt to capture the global scene properties through various low-level image features. The suitability of different gist-based approaches for scene categorization is discussed in [40]. Therefore, the authors in [27] opt for a gist-based scene classification.

The classification is performed among the four most common scene types in street-side imagery (Figure 2.1).

- No buildings - images not containing any buildings. Typical examples in urban scenarios are parks, gardens and waterfronts.
- Street - images containing facades captured at a high angle to the facade planes, occurring when camera orientation coincides with street direction.
- Facades - images containing one or more whole facades.
- Building part - images containing only a small part of a facade, not enough for a complete reconstruction.

Among the scene classes listed, only the “Facades” class is suitable for complete facade reconstruction. The appearance of the “No building” class in collected images means there is a gap in the building block, and that no buildings should be reconstructed. Similarly, if the image is classified as “Street”, it displays a street crossing. Finally, the “Building part” class represents the images, in which the building is too large (or the street is too narrow) to be captured in a single image. The approach uses a steerable pyramid of Gabor filters, tuned to 4 scales and 8 orientations. Filter outputs are then averaged on the 4x4 grid. This produces a feature vector comprising of 512 features. Classification is performed using a Support Vector Machine (SVM) with a Gaussian radial basis kernel function. The SVM is trained using a one-versus-all approach.

Image Rectification

Sideways looking cameras in streets have a low chance of capturing most of a facade, as opposed to cameras looking forward, upward or backward. The images taken by these cameras are projectively distorted. The prior rectification of the images to a fronto-parallel view is a prerequisite to not only facade splitting algorithm in [27] but also further processing steps. The rectification is implemented by following the approach from [22]. After the scene classification step it is assumed that the image contains a planar surface with two dominant perpendicular directions.

The relation between points of the image plane x and points in the world plane x' can be expressed by the projective transformation matrix \mathbf{H} as $x' = \mathbf{H}x$, where x and x' are homogeneous 3-vectors. The rectification follows a step-wise process (see Figure 2.2) by estimating the

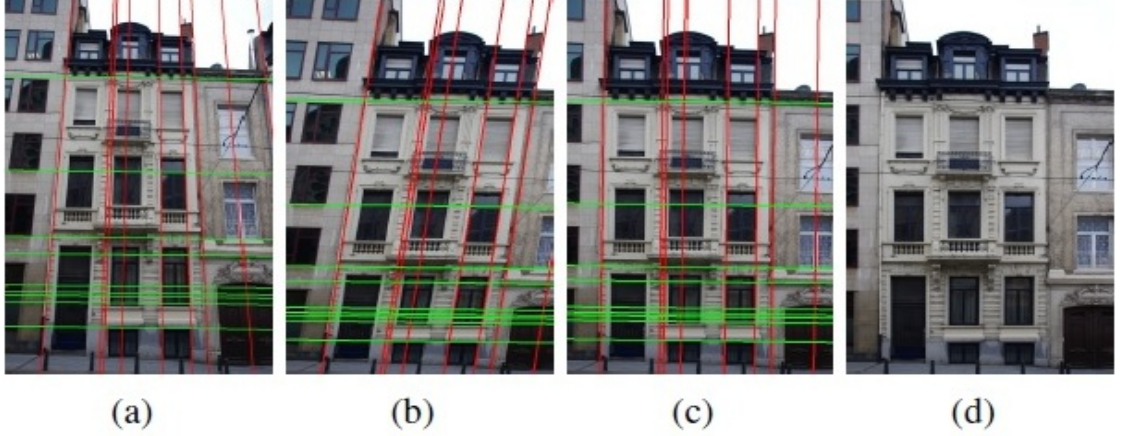


Figure 2.2: Rectification process: (a) input image with dominant lines, (b) projective distortion removal, (c) affine distortion removal, (d) similarity transformation [27]

parameters of the projective \mathbf{P} , affine \mathbf{A} and similarity \mathbf{S} part of the transformation \mathbf{H} , which can be (uniquely) decomposed into:

$$\mathbf{H} = \mathbf{SAP} \quad (2.1)$$

The projective transformation matrix has the form

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix} \quad (2.2)$$

where $l_\infty = (l_1, l_2, l_3)^T$ denotes the vanishing line of the plane. Parallel lines in the world plane intersect in the distorted image at vanishing points. All vanishing points lie on l_∞ . A state of the art line detector is used to find these vanishing points. Then the two vanishing points of the image are detected by RANSAC (RANDOM Sample Consensus) [12].

The affine transformation:

$$\mathbf{A} = \begin{pmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

has two degrees of freedom represented by the parameters α and β . The knowledge of the perpendicular intersection of the dominant lines l_a and l_b is the only constraint to impose, as there is no knowledge about other angles or length ratios in the image. Therefore the affine part of the rectification process can only restore angles, but not length ratios. α and β lie on the circle with center and radius:

$$(c_\alpha, c_\beta) = \left(\frac{a+b}{2}, 0 \right), r = |(a-b)| \quad (2.4)$$

where $a = -l_{a2}/l_{a1}$ and $b = -l_{b2}/l_{b1}$. If the image does not contain any affine distortions, the parameters would have the value $(0; 1)^T$, so the closest point on the circle to that point is chosen

for the correction. Finally the image gets rotated by the rotation matrix \mathbf{R} to align the dominant lines with the axes, scaled by s and translated by t :

$$\mathbf{A} = \begin{pmatrix} s\mathbf{R} & t \\ 0^T & 1 \end{pmatrix} \quad (2.5)$$

Facade Splitting

Continuous building blocks with little or no space between individual buildings are common for urban environments [27]. Each building in the block may have a different architectural style. If the style recognition system takes as an input building blocks, like in [27], it should be able to separate different facades in order to classify them properly. As man-made structures are characterized by strong horizontal and vertical lines, the authors in [27] choose to exploit them as the main cue for building separation. They assume that the individual facades can be separated using vertical lines. The following heuristic is used: horizontal line segments on the building usually span only one facade. Vertical lines, which cross a large number of horizontal lines, have less chance of being a valid facade separator.

After the rectification step, it is known that the vertical lines in the image correspond to the global direction of gravity. A line segment detector is used to find salient edges. Then line segments are grouped in three clusters. The first cluster contains horizontal line segments (with a tolerance of ± 10 degrees in orientation). Similarly, the second cluster contains vertical line segments, while the third one - all other detected line segments. The last cluster will typically have a smaller number of elements, due to the predominance of two perpendicular orientations in urban scenery.

Next, a vertical line is swept over the image. At each position of the line, two values are calculated: support and penalty. Support is defined as the number of vertical line segments that coincide with the sweeping line (or reside in its close vicinity). Every vertical line segment is additionally weighted with its length: longer vertical line segments provide more support for the line. The support from neighboring vertical lines is reduced linearly with the distance to the line. Penalty is calculated through the number of horizontal lines that the sweeping line crosses. Every horizontal line segment is weighted with its length: the longer the crossed segment is, the more penalty it generates. Relative position of the crossing point to the center of the horizontal line segment is also evaluated. Vertical lines that cross horizontal segments near the edges will receive less penalty than those that cut the segments through the middle.

After the line sweeping process there are two vectors of the same size, equal to the image width: support vector and penalty vector. The aim is to find the positions of the vertical line, which correspond to local minima in the penalty vector and local maxima in the support vector. In order to calculate this, the authors in [27] first use the penalty vector to threshold the support vector. All of the line positions which have more than 3% of the maximum penalty value are discarded. After positions having less than 20% of the maximum support value are eliminated as well. The appropriate values in the support vector are set to zero. Finally, local non-maxima suppression is performed on the support vector by a sliding window (9% of the image width). The resulting local maxima then coincide with the desired separator positions. These values

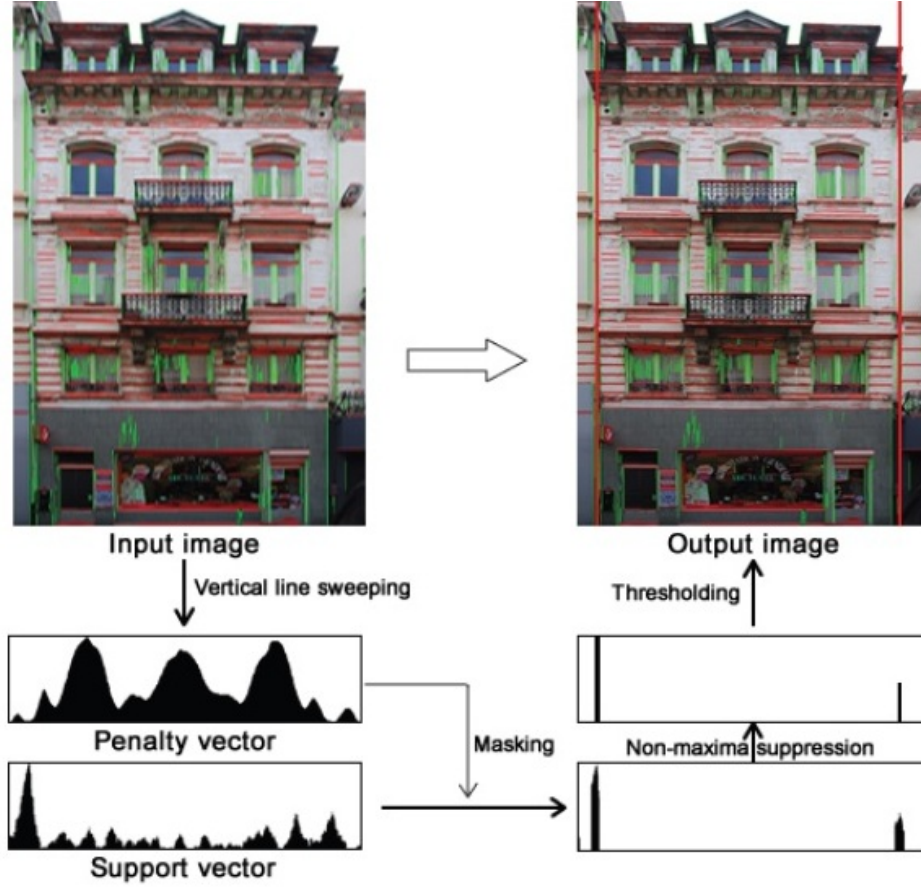


Figure 2.3: The facade splitting algorithm [27]

are used to cut the building block into individual facades. The process of estimating facade separators from line segments is illustrated in Figure 2.3.

Style Classification

The style classification is an important step in order to select an appropriate grammar for the given building. To differentiate between “Flemish renaissance”, “Haussmann”, “Neoclassical” and “Unknown” styles, the NBNN classifier, proposed by [30], is employed. First all descriptors for the training images are calculated and sorted into the different classes. Then for every descriptor d_i of the query image the nearest neighbor distances to each class is approximated using the FLANN library [29]. The sum over the Euclidean distances of each query descriptor d_i denotes the image-to-class distance. The class with the least distance is chosen as the winner class. The mean detection rate of the Scale Invariant Feature Transform (SIFT) [23] features was 84%.

2.3 Literature Studies Related to Building Facade Analysis

Whereas the approach presented to the scientific community in [27] is found to be the only work so far addressing the problem of building facade automatic architectural style classification based on images, extensive research has been carried out in the broader field of building facade analysis. Facade analysis is attractive in the applications of 3D urban environment modeling and facade reconstruction. The extraction and reconstruction of windows, doors and ornaments provides rich information of the buildings and adds visual realism [18].

The previous research, relevant to the current work by addressing the problem of facade segmentation by an architectural element, performs segmentation of windows [1, 4, 17, 18, 20, 33, 34, 42], doors [4, 42], balconies [42], walls [4, 42], roofs [4, 42] and chimneys [42]. Though none of the approaches referred targets the segmentation of the window types classified in the current work, namely Romanesque single, double or triple round arch windows, Gothic style pointed arch and rose windows and Baroque windows with triangular, segmental pediments and balustrades as decorations. The example images display simple rectangular windows, which are visually very different from the ones listed above and featuring complex ornamentation.

The purpose of the building facade segmentation is to discover different regions of a facade image and assign each of them a particular semantic label (walls, windows, roofs, etc) [42]. The main difficulty rests on the significant variations that may exist between buildings, even for ones corresponding to the same architectural style [42]. Furthermore, their visual appearance spans an infinite space, due to either their internal characteristics (walls, windows, roof) or their external ones (occlusions, lighting, reflectance properties, etc.) [42].

Traditionally, segmentation problems are divided into model-based and model-free methods [42]. Model-free approaches make no assumption on the geometry and the appearance of facade components and aim at grouping pixels according to feature similarities. These methods are prone to failure, because pixels belonging to the same facade element do not necessarily share common visual characteristics. Model-based methods provide segmentation, which is a compromise between the available image support and a prior knowledge. The main assumption of these methods is that the space of solutions can be parameterized in a convenient way. This is far from being sufficient when dealing with building facades, simply because no static parameterization is generic enough to encompass different facade layouts. A third class of methods was introduced to cope with the aforementioned limitations. Procedural models like shape grammars offer a flexible and powerful tool to account for such variability while being compact and providing a semantic representation of the obtained results. The idea is to represent buildings using a set of replacement rules and a dictionary of basic shapes. These methods have been extensively used in computer graphics.

The main contribution of [42] is to combine a powerful machine learning approach with procedural modeling as a shape prior so as to fit the purpose of multi-class facade segmentation. Generic shape grammars are constrained to express buildings only. The classes segmented are “window”, “wall”, “balcony”, “door”, “roof”, “chimney”, “sky”, “shop” and “other”. The promising results are demonstrated on an image dataset of Parisian architectural styles - Haussmannian, Restauration and Louis XIV.

The authors in [18] propose an automatic method for segmentation of building facade im-

ages. Firstly individual facades are isolated from general city block images. This step is realized via accumulation of directional color gradients, assuming that facade structures are aligned. Then the sky region is detected based on segmentation approach and color marker extraction. Finally the images are split in floors also employing directional color gradient accumulation. The approach introduced several morphological filters to augment the robustness to problems, like textured balconies, reflections on the bright windows and small obstacles in images [18].

The approach in [4] introduces a double-stage, i.e. course and fine, parsing of architectural scene images. At the first stage the image is parsed into five visual categories - “sky”, “building”, “foliage”, “street” and “sky-mixed”. The “sky-mixed” class represents image parts, which contain pixels of sky, building and foliage, but do not look like either one alone. At the second stage parsing at a finer level is realized, identifying the positions of windows, doors, rooflines, building and roof boundaries, the colors of walls and the spatial extent of particular buildings. Features are computed in a fixed window around each pixel, avoiding segmentation as a pre-processing step. The approach is built as a probabilistic model, where “course” and “detail” labels are assigned to each image pixel. The labeling models are trained using a set of hand labeled samples. The objective in the learning stage is to count the number of times a certain label co-occurs with a certain feature value combination as an empirical estimate $p(l_i|f_m)$. The prior distribution over visual categories is assumed to be uniform and the features are modeled as independent. The features observed are color histogram in L.a.b color space, contour, texturesness, position and their combinations. Experiments prove that the combinations of features perform better than any individual feature.

The approach introduced by [34] performs window detection on facades, located in the Austrian city Graz and its surroundings. It is a modified gradient projection method, robust enough to process complex facades of historic buildings in different weather and lighting conditions, under high perspective distortions. The window detection is realized in three subsequent steps:

- 1) Identification of facade borders in the input image and its rectification.

This first step is achieved by using the method of [19], where the aerial model of the scene with untextured building frames is available and the rectified facades are obtained by projecting a digital image into that model.

- 2) Extraction of facade levels.

Firstly horizontal division of the facade is performed employing the vertical projection. Afterwards the horizontal projection of gradient is computed for each level. The thresholding of the horizontal projection separates the blocks within the levels.

- 3) Detection of windows.

In this step the objective is to find out if the blocks are windows or facade parts. The labeling into “windows” or “facade” categories is determined iteratively by k-means clustering of the color histogram in CIE-Lab color space, the size of the block and the gradient content of the block. Selection of the CIE-Lab color space is advantageous for the following reasons:

- i) The Euclidean distance of two colors in CIE-Lab space is directly proportional to the visual similarity of the colors, which provides a simple metric for clustering.
- ii) The clustering can be performed only in “a” and “b” channels, which represent the color value component. The “L” component represents the luminosity. A single value of “L”, computed as a mean of each color in the cluster can be used as representative for each cluster to cope

with shadows and illumination problems.

The experiments were carried out on an image database of 19 facades with 392 manually labeled windows, 369 of which (94.13%) were detected by the algorithm.

The approach introduced by [33] differs from the algorithm in [34] by exploring the HSV color space instead of the CIE-Lab. In HSV color space the “H” channel represents the hue, the “S” channel - saturation and the “V” channel - value. The approach in [34] considers window detection in a single image scenario, when only a single image of a building is examined. Whereas in [33] also multi-view scenario window detection is performed on multiple images of the same facade by comparing two approaches:

- 1) merging the images into a single, rectified facade and performing the window detection on the merged data,
- 2) applying window detection on each image separately and merging the results in a world coordinate system.

The second method described outperforms the first one while increasing the number of images. The multi-view experiment also shows that at a certain number (around 20) of images the precision of the window detection does not increase for both methods. The experiments are carried out on multiple images of five facades. The average probability of detecting a window is reported to be 91.4%.

The authors in [1] introduce a learning classifier system that provides substantial single window detection and localization and also a Window Region Of Interest (WROI) operator, as a basis for further processing. The main application for this window detection system is considered the facade classification from mobile imagery. For this purpose it suffices to detect only a fraction of all windows on a facade, assuming that either the complete set of windows would belong to a single window class or that the detected windows are sufficiently discriminative for identification of the corresponding facade, so that the remaining undetected windows provide only redundant information. The problem of window detection is viewed as a simple pattern recognition problem, while being aware that the visual content of window patterns may represent considerable variations of appearance and the variation of the viewpoint can lead to projective distortions and scale variance, significantly impacting the quality of the pattern matching as a result. The authors employ the work presented by [44] for detection of the objects of interest. Firstly the evaluation is performed on the detection of single windows, reporting 57% true positive rate as the best result. Since the definition for true positives is less strict for the WROI evaluation case, higher detection rate is achieved in that scenario (60% in the best case). It is observed that window detection worsens for contemporary buildings, buildings with few informative features, as well as for windows exposed to high perspective distortions.

The approach introduced in [45] performs detection of repeated structures in facade images. Since no explicit notion of a window is used in the approach, besides windows other manmade structures are also being identified. Here the methodology of [24] for detection of dominant symmetries is adapted for detection of multiple repeated structures. The methods, addressing the segmentation of architectural elements dome and tower and introduced in the current work in Section 4.2 and Section 4.3 respectively, also employ the methodology of [24] with the purpose of bilateral symmetry detection on building facades.

2.4 Summary

Building facade visual analysis is an active research field. However, to the best knowledge of the author the only approach in the literature, targeting the problem of the image-based architectural style classification of facades is introduced by [27] (Section 2.2). The approach classifies facades of Flemish Renaissance, Haussmannian and Neoclassical architectural styles in four successive steps (Section 2.2) - scene classification, image rectification, facade splitting and style classification. The work motive is the recognition of the architectural style of building facades in order to reconstruct buildings by the means of inverse procedural modeling. Images taken with cameras on a mobile mapping van are the input for the method.

Section 2.3 studies the literature in the broader domain of building facade visual analysis. The problem of facade segmentation by an architectural element is tackled in the approaches segmenting windows [1, 4, 17, 18, 20, 33, 34, 42], doors [4, 42], balconies [42], walls [4, 42], roofs [4, 42] and chimneys [42]. The method proposed in [45] detects repeated structures on building facades.

The Architectural Aspect of the STYLE Project

3.1 How to Distinguish Architectural Styles

Geographical, geological, climate, religion, social, political and historical factors influenced the formation of architectural styles [13]. In fact, a definition of an architectural period cannot be absolute, but it generally applies to ninety examples out of every hundred [5]. Every style of architecture is regarded as the solution of certain fundamental problems, i.e. each building must have all or most of the parts A. to G. (listed below) and consequently there is both interest and instruction to be gained in learning and comparing how each style has solved these points of the problem [13].

- A. Plan or general distribution of the building
- B. Walls, their construction and treatment
- C. Openings, their character and shape
- D. Roofs, their treatment and development
- E. Columns, their position, structure and decoration
- F. Mouldings, their form and decoration
- G. Ornament, as applied in general to any building

The current methodology of building facade architectural style classification is built on the above stated definition of architectural styles, developing it further to fit in the scope of a computer vision system. Architectural elements are the unique details and component parts that together form the architectural style of buildings. While some of the points A. to G., like walls

and columns, are architectural elements, others assume to include multiple architectural elements: i.e. in C. openings refer to doors, windows, etc, in D. roof also includes a convex roof, called dome, whereas in G. ornament refers to a variety of architectural elements. So to recognize the style of the building, typical architectural elements of the style should be looked for. The methodology classifies architectural elements window, dome, tower, as well as tracery, pediment and balustrade.

The work focus is on historic architectural styles, which combine style-typical architectural elements and obey to certain design rules for building construction. Unlike historic architecture, contemporary architecture is not confined by any rules and is difficult to categorize. In addition to that today's construction technologies let bring into life the most extraordinary ideas of architects, thus creating one of a kind buildings, which carry the style signature of the architect rather than of a widely spread general style.

3.2 Romanesque, Gothic and Baroque Architectural Styles

The dissertation explores and classifies building facades of the following pan-European architectural styles:

Romanesque (8th - 12th centuries)

Gothic (12th - 16th centuries)

Baroque (17th - mid 18th centuries)

Architectural revivalism, which is a phenomenon of imitation of past architectural styles, should be noted while reviewing the problem of architectural style classification. The singularity of 19th century revivalism, as compared with earlier revivals, was that it revived several kinds of architecture at the same time [6]. These revived styles are also referred to as neo-styles, e.g. Gothic Revival is also referred to as neo-Gothic. The work does not intend to differ between original and revival architectural styles, as only visual information is not enough for such a discrimination. If there be a need to differ original styles from the revived ones, the building date should be provided as an additional feature together with the image. Thus while saying the project addresses the classification of Romanesque, Gothic and Baroque styles, the revivals of those should be also understood by default.

The term Romanesque may be said to include all those phases of Western European architecture, which were based on Roman art, and which were being carried out, in a rough and ready way, in Europe, from the departure of the Romans up to the introduction of the pointed arch in the thirteenth century [13]. The general architectural character is sober and dignified, while picturesqueness is obtained by grouping of the towers [13]. Local influences of taste, climate, geography and geological formations were instrumental in producing different characteristics of each country [13]. The most outstanding feature of the style is the round arch [13]. The current work classifies windows with single, double or triple round arches in Romanesque class (Section 4.4). The architectural element tower, extensively used in Romanesque architecture, is also classified (Section 4.3, Section 4.7).

The principles and characteristics of Gothic architecture were similar throughout Western Europe [13]. The fully developed Gothic art of the thirteenth century was the style which had been slowly developing itself throughout Europe as a necessary sequence of Romanesque art [13]. One style evolved from another so gradually that it is impossible to say exactly where the one ended and the next began [13]. As a consequence of this evolution buildings displaying both Romanesque and Gothic architectural elements are frequently met. Thus the proposed method also performs classification of buildings, which are a mixture of Romanesque and Gothic styles. Gothic style is also referred to as Pointed style [13]. The most essential part of the Pointed style - the part whereon its whole structure and organisation depend, is the pointed arch itself [35]. In Gothic style the current method classifies pointed arch and rose windows (Section 4.4), traceries (Section 4.5) and towers (Section 4.3, Section 4.7).

Baroque is a European style of architecture and decoration, which developed in the 17th century in Italy from late Renaissance and Mannerist forms, and culminated in the churches, monasteries, and palaces of southern Germany and Austria in the early 18th century ¹. This style originated in Rome and is associated with the Catholic Counter-Reformation, its salient characteristics – overt rhetoric and dynamic movement ². Baroque Architecture may be limited to a historical period, varying in date in different countries and cities, but in general beginning as the Renaissance spirit declined to pedantry and ending with the return to pedantry in the eighteenth century [5]. The architectural details of the various buildings vary so much, as to make generalizing almost impossible [5]. Constructional difficulties were so few that the dome - to take one feature - was varied in almost every instance, and was sometimes elliptical instead of circular [5]. The classified elements in Baroque style are windows, featuring as ornamentation balustrades, triangular and segmental pediments (Section 4.4). Balustrades and a bigger variety of pediments, such as triangular, segmental, triangular and segmental broken on the base or top, swans' necks are the classified ornamentations (Section 4.5). The list in Baroque style finalizes the prominent elements tower (Section 4.3, Section 4.7) and dome (Section 4.2, Section 4.6).

3.3 Typical Windows of Romanesque, Gothic and Baroque Architectural Styles

Window is an architectural element, the functionality of which is to let natural light into the building. For architectural style classification of windows typical window examples of Romanesque, Gothic, Baroque architectural periods are chosen. The characteristic feature of Romanesque windows is the single, double or triple round arch (Figure 3.1a, Figure 3.3b and Figure 3.3c respectively), whereas Gothic style is very distinct with pointed arches (Figure 3.2a) and rose windows (Figure 3.2b). The pedimented and decorated window-opening is a favourite motif in the Baroque facade [5]. In Baroque style the examined window types feature as decorations triangular (Figure 3.3a) and segmental (Figure 3.3b) pediments and balustrades (Figure 3.3c).

The window classification approach (Section 4.4) overall categorizes three window classes - Romanesque, Gothic, Baroque and 8 intra-class types - Romanesque single, double and triple

¹ Illustrated Architecture Dictionary <http://buffaloah.com/a/DCTNRY/vocab.html>

² Baroque / Baroque Revival Architecture <http://buffaloah.com/a/DCTNRY/b/baroque.html>



a) Single arch



b) Double arch

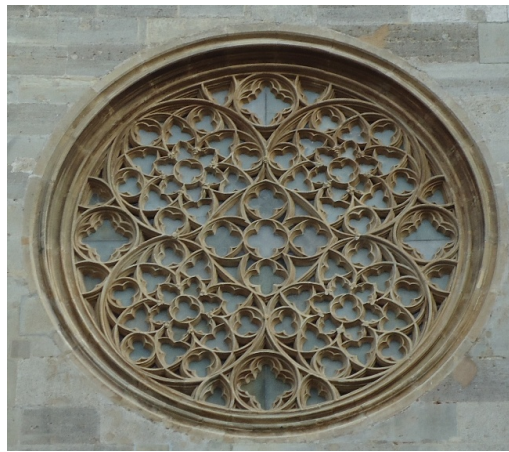


c) Triple arch

Figure 3.1: Romanesque windows.



a) Gothic pointed arch



b) Gothic rose

Figure 3.2: Gothic windows.

round arch windows, Gothic pointed arch and rose windows, Baroque windows with triangular, segmental pediments and balustrades. There is no discrimination among 8 intra-class types, as the objective of the current work is the architectural style classification.

3.4 Tracerics, Pediments and Balustrades

Tracery (Figure 3.4a) is a pattern of interlacing lines. This architectural element is characteristic for Gothic and Gothic Revival architecture. Though its common function is to support stained glass windows, it appears also on walls and other parts of Gothic/Gothic Revival facades. Tracery, adorning a wall or panel but not pierced through, is called blind tracery (Figure 3.4b). It



Figure 3.3: Baroque windows.

is used only as a decorative element, unlike when supporting stained glass windows. Patterns in tracery form ornaments called trefoil (Figure 3.5a), quatrefoil (Figure 3.5b), cinquefoil (Figure 3.5c), sexfoil (Figure 3.5d), multifoil. Foil is the french word for leaf, while the ornament prefixes indicate the number of leaves. Figure 3.5a and Figure 3.5b show pointed foils, whereas Figure 3.5c and Figure 3.5d - round foils.

Pediment is an architectural element, which stands across a portico, door or window. While it covers porticos in Greek, Greek Revival, Neoclassical architecture, here the focus is on pediments, serving as window decorations on Baroque/Baroque Revival facades. The wide variety of pediment types and designs is one of the factors making Baroque buildings so ornate and rich. Here are listed the main types of pediments. Triangular and segmental pediments are shown in Figure 3.6a and Figure 3.6b respectively. The space inside the pediment is called tympanum and

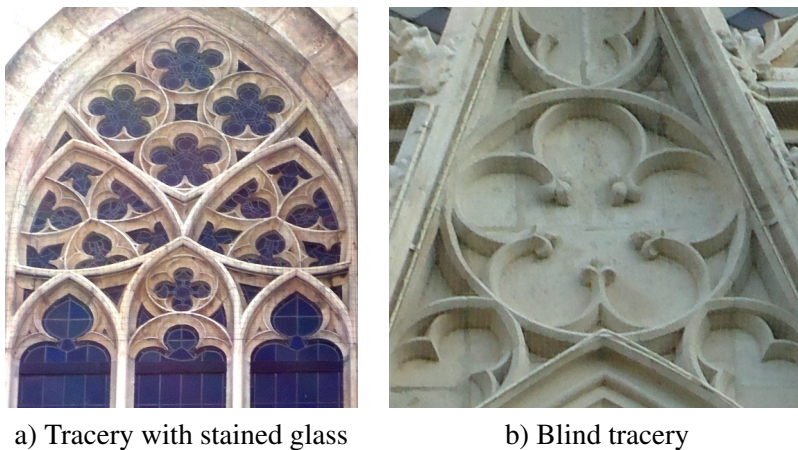


Figure 3.4: Gothic tracery samples.

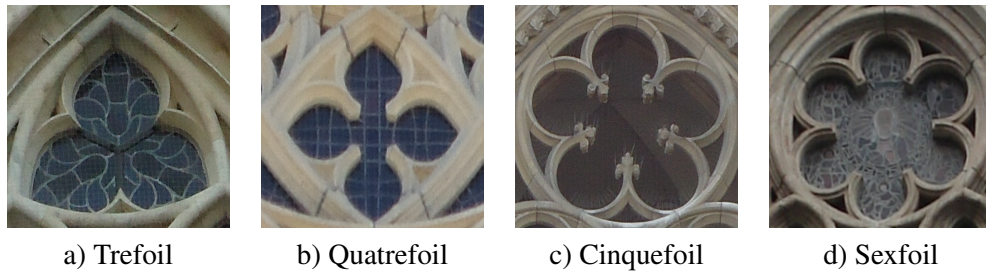


Figure 3.5: Gothic tracery patterns.

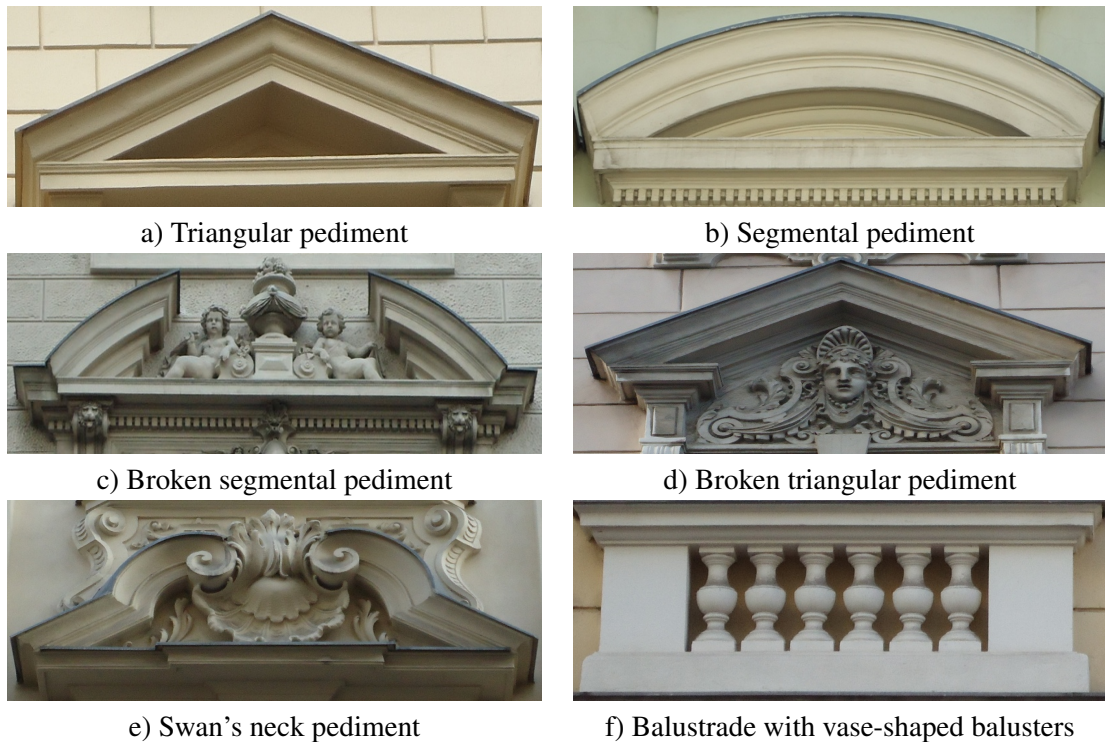


Figure 3.6: Examples of the classified Baroque architectural elements

is ideal for filling with sculptures. A pediment broken at the top, base or both is called broken pediment. Figure 3.6c displays an example of a segmental pediment broken on the top, whereas Figure 3.6d shows a triangular pediment broken on the base. Broken pediments provide more space for complex sculptures typical for Baroque architecture. A type of broken pediment with S-shaped curves is called swan's neck pediment (Figure 3.6e), as it resembles two swans' necks facing each other.

A series of uprights, called balusters and supporting a railing, is called balustrade. They appear as window decorations on Baroque/Baroque Revival facades. A balustrade with vase-shaped balusters is shown in Figure 3.6f. All architectural definitions of the current section are

taken from the Illustrated Architecture Dictionary.

The goal is to classify tracery in Gothic class and pediments, balustrates - in Baroque class (Section 4.5). Though this is a binary classification problem, the classification complexity is high due to unlimited variations of the forms, decorations and ornamentations of the classified elements.

3.5 Renaissance, Russian and Islamic Domes

A dome is a convex roof. Domes are categorized according to the base shape and the section through the dome center. Hemispherical domes have a circular base with a semicircular section. This dome type (Figure 3.7a) is characteristic for grand buildings of Renaissance architecture. The domes of Baroque, Neo-Baroque, Neo-Renaissance and Neoclassical buildings imitate the visual forms of Renaissance domes, thus the dome classification approach classifies them as Renaissance style. The mentioned imitation is displayed on the examples of the domes of St. Charles's Baroque church in Vienna (Figure 3.7b), St. Stephan's Neo-Renaissance Basilica in Budapest (Figure 3.7c) and Neoclassical city hall of San-Francisco (Figure 3.7d).

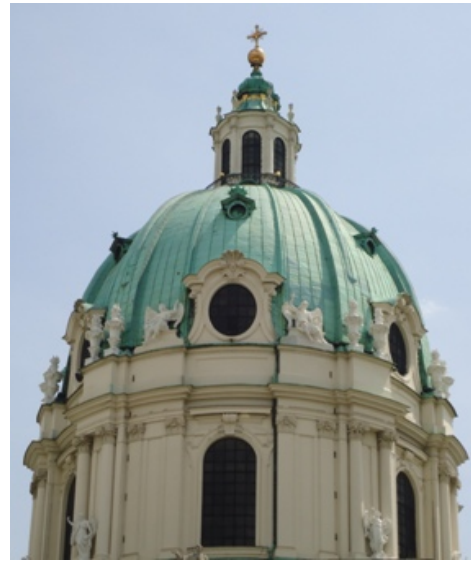
In the design of domes and belfries, but especially the former, the genius of Baroque architects is seen at its best [5]. In fact, with a few exceptions, the art of dome design, initiated in the earlier days of the Renaissance by Brunelleschi, attained its highest level in the seventeenth century [5]. The most copied Renaissance domes belong to St. Peter's Basilica in Vatican (Figure 3.7a) and St. Paul's Cathedral in London.

Saucer domes have a circular base and a segmental, less than a semicircle section. This dome type is typical for Islamic Ottoman mosques, some of which are former Byzantine churches. The saucer dome of Vienna Islamic center is displayed in Figure 3.7e. Onion domes have a circular or polygonal base and an onion-shaped section and are a typical feature of Russian Orthodox churches and Islamic mosques. Russian and Islamic onion domes are shown respectively on the examples of one of the domes of St. Nicholas's Basilica in Vienna (Figure 3.8a) and the Blue Mosque dome in Yerevan (Figure 3.8b). Russian architecture displays its most peculiar feature in the shape and number of the domes [35]. Russian domes are gilded or brightly painted in blue or multiple colors. Architectural definitions are taken from the Illustrated Architecture Dictionary.

The dome classification module (Section 4.6) categorizes three architectural styles - Renaissance, Russian, Islamic, which comprise eight intra-class types - hemispherical domes of Renaissance, Neo-Renaissance, Baroque, Neo-Baroque, Neoclassical buildings in Renaissance class, Russian onion domes, Islamic onion and saucer domes. Here, to avoid misunderstanding, it is worth to remind, that the project at its current state targets the classification of Romanesque, Gothic and Baroque styles. And since domes are not common for Romanesque and Gothic architecture, onion and saucer domes stand as a classification alternative to hemispherical domes, typical for Baroque grand buildings. Though Russian and Islamic domes are introduced only as comparative analysis at this point, the voting algorithm leaves free room for the extension of the project by Russian and Islamic styles by adding other typical architectural elements of these styles. Domes and towers are featured on buildings of religious and secular great significance, as a display of power.



a) St. Peter's Basilica



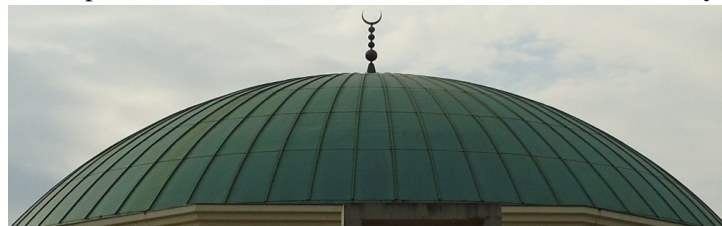
b) St. Charles's church



c) St. Stephan's Basilica



d) San-Francisco city hall



e) The saucer dome of Vienna Islamic center

Figure 3.7: Hemispherical and saucer domes.



a) St. Nicholas's Basilica dome



b) Yerevan Blue mosque dome

Figure 3.8: Onion domes.

3.6 Romanesque, Gothic and Baroque Towers

A tower is a building or part of a building that is exceptionally high in proportion to its width and length ³. The tower segmentation module (Section 4.3) addresses buildings featuring single and double towers. Towers, that are detached from the main building, like the leaning Pisa bell-tower, are out of the scope of the approach, since they are stand-alone buildings themselves. Buildings featuring more than two towers are also not considered, as multiple towers have high chances of occluding one another. The tower segmentation and classification approaches (Section 4.3, Section 4.6) process towers of Romanesque, Gothic and Baroque styles, examples of which are shown in Figure 3.9. Figure 3.9a displays the left tower of Breitenfelder Romanesque church situated in Vienna, Figure 3.9b - the tower of Brussels Gothic city hall and Figure 3.9c - the tower of Loreta Baroque church in Prague.

3.7 Summary

The current chapter illustrated the architectural aspect of the STYLE project. Firstly, Section 3.1 illustrated the human cognitive process of architectural style classification, that is the identification of style typical architectural elements on a building facade. Afterwards it justified the methodology of the STYLE project - the voting of architectural elements, from the architectural point of view, since the method imitates the human logic for solving the problem stated.

Secondly, Section 3.2 introduced the three historic pan-European architectural styles classified within the project - Romanesque, Gothic and Baroque. The styles have general character, being temporally long-lived and geographically widely spread in Europe.

Thirdly, the definitions and examples of the architectural elements classified were delivered in Section 3.3, Section 3.4, Section 3.5 and Section 3.6.

³Tower <http://www.buffaloah.com/a/DCTNRY/t/tower.html>



a) Romanesque Tower

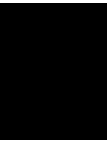


b) Gothic Tower



c) Baroque Tower

Figure 3.9: Examples of Towers.



Methodology: The Voting of Architectural Elements

4.1 The Algorithm of the Voting of Architectural Elements

The ultimate goal of a computer vision tool for architectural style classification is to possess the intelligence to classify any visually distinguishable architectural style automatically. The first milestone of the work is the introduction of an algorithm, which imitates the logic of a human expert during architectural style classification process. Following sufficient research related to architectural styles [5, 6, 13, 35] and based on the definition of architectural styles, brought in Section 3.1, the general algorithm of image-based architectural style classification is constructed as a succession of the three principle phases, listed below:

- 1) The query building facade is segmented by architectural elements.
- 2) Each segmented element is passed to the corresponding module for classification. The classification is accomplished using the corresponding visual codebook, obtained beforehand by training of the same type of elements.
- 3) Each classified element votes for an architectural style. The architectural style, having received the majority of votes, is chosen as the building style.

The algorithm is general enough to classify any visually distinguishable architectural style. The algorithm's ability to classify facade parts, partially occluded facades or facades of mixed architectural styles also speaks well for its universality. Furthermore, the algorithm is efficient, since the votes of the misclassified architectural elements can be suppressed by the votes of numerous elements at the voting phase. Repeated architectural elements (e.g. windows and traceries), presenting redundancy, also promote the suppression of falsely classified votes. The more architectural elements classified, the more robust against misclassifications the algorithm is in the voting phase. The algorithm is innovative, being the first general algorithm for solving the problem stated.

The choice of coming up with a special 'smart' algorithm, imitating the human architectural style classification process, but not trying any of the standard image classification algorithms,

such as the general BoW method, is based on extensive research related to architectural styles [5, 6, 13, 35] and relies on the following strong arguments for the voting algorithm and against general image classification ones:

- Not every visual feature on a facade, speaks about its architectural style. A vivid example of that and a well-known problem, as mentioned in [27], is the first floor phenomenon. Building first floors may host stores, cafes, restaurants, hotel lobbies, etc., the appearance of which does not present architectural style relevant visual information. Advertising boards and signs may additionally stretch beyond the first floor. Figure 4.1 depicts examples of such neo-Baroque facades, which display plenty of architectural style irrelevant visual features, but may well be classified by the proposed algorithm owing multiple windows adorned with triangular pediments.
- Partly occluded facades cannot be classified by a general image classification algorithm, whereas achieve successful classification by the proposed method, analyzing the visible style typical architectural elements.
- The method is designed so that to permit the classification of a mixture of architectural styles, by having the mixture of Romanesque and Gothic implemented so far. Furthermore at this stage it also embeds architectural knowledge in the system by letting only the combinations of styles, which are probable due to evolution of styles from one another, as well as historic periods and reigns, influencing architecture heavily. Classification of a mixture of styles seems quite a challenge for a general classification algorithm.
- The intra-class variety of the classification problem posed is immense, taking into account the unlimited variety of buildings, the infinite number of possible 2D projections (images) and the visual complexity of historic architectural styles. This is another severe negative factor for a general classification algorithm, a major challenge, which the STYLE algorithm aims to overcome by facade decomposition into architectural elements and balancing the complexity of the problem on several modules, each solving a specific simpler task successfully.

While brainstorming about the problem of architectural style classification, analysis of the whole building facades shapes also may occur as a possible solution. Whereas some facade shapes may appear visually distinctive, generally architectural style discrimination is not possible by facade shapes. The arguments supporting this claim are: 1) Historic building facades are not planar, thus they are 3D objects with unlimited number of possible 2D projections (images). 2) The essence of architectural styles is not in the shapes of the whole buildings, but in the shapes of style-typical architectural elements. Facades with the simplest rectangular shape may belong to any architectural style, proving the idea of facade shape analysis pointless.

The second milestone is the implementation of the proposed algorithm, its embodiment from the theoretical idea to a functional software tool. Here the issue is lying in the fact that there are decades of architectural styles and it is obvious, that the realization of the classification of all of them is not achievable due to the requirement of immense resources, as well as technical challenges. Therefore the software implementation of the algorithm is limited to three pan-European



a) A building hosting a hotel



b) A building hosting an ice-cream saloon

Figure 4.1: Neo-Baroque facades featuring architectural style irrelevant visual features.

architectural styles of major importance – Romanesque, Gothic and Baroque. The classified architectural elements are window, dome, tower, tracery, pediment and balustrade, illustrated in detail in Chapter 3. The diagram of the algorithm with its three major steps is presented in Figure 4.2. Whereas the segmentation of windows, traceries, pediments and balustrades is manual, the segmentation of domes and towers is automated at the first step. Section 4.2 and Section 4.3

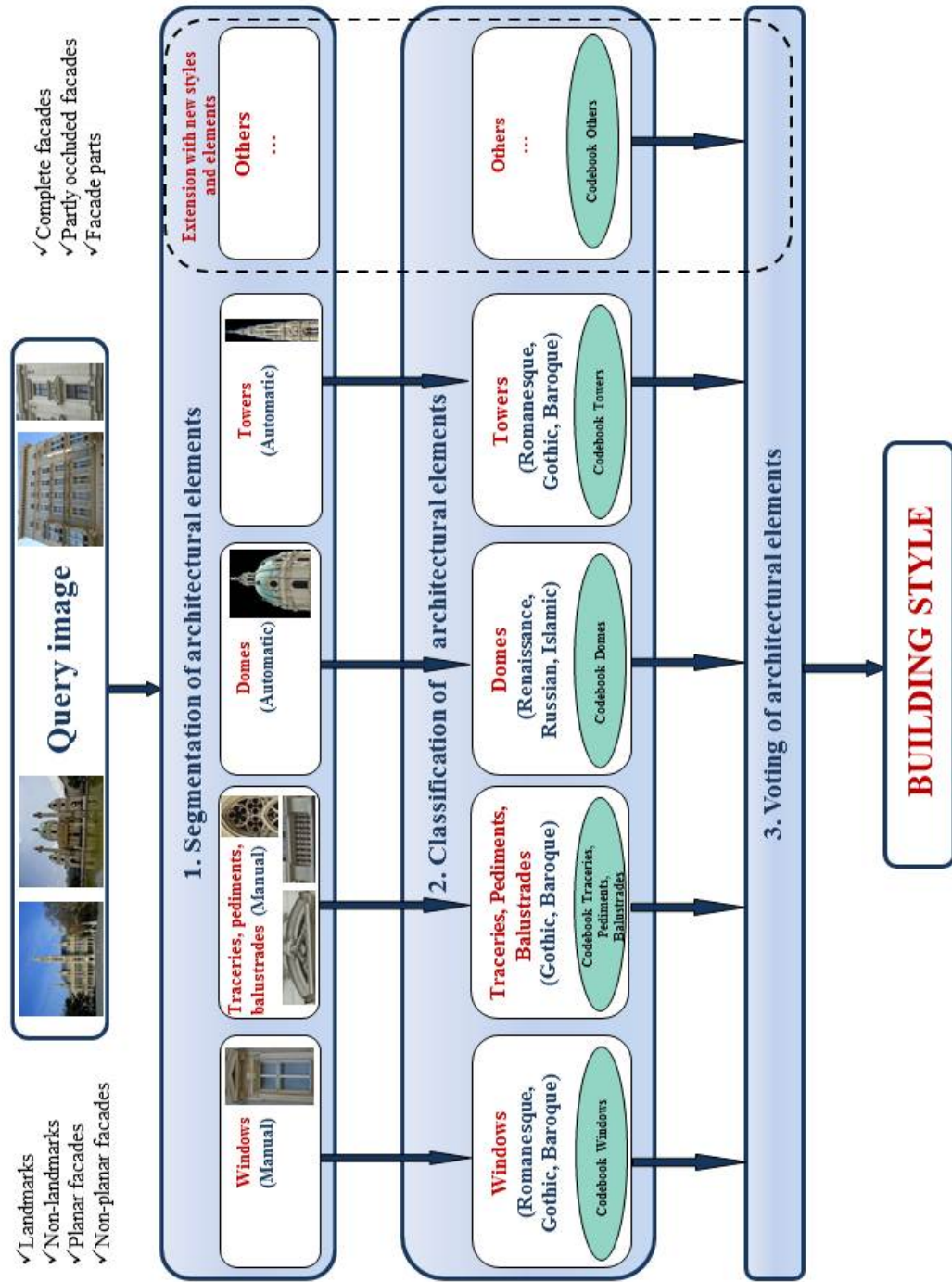


Figure 4.2: The Algorithm of the Voting of Architectural Elements

illustrate the approaches performing segmentation of domes and towers respectively. The classification approaches at the second step categorize windows (Section 4.4), traceries, pediments, balustrades (Section 4.5), domes (Section 4.6) and towers (Section 4.7). The voting mechanism, making the third step of the algorithm, is detailed in (Section 4.8). The dashed module in Figure 4.2 shows that it is possible to extend the project with new architectural elements and styles by taking the same three steps for the newly incorporated elements.

The software tool is semiautomatic, requiring minimum user interaction only by indicating the architectural elements present on the query facade and manual segmentation of the elements besides domes and towers. The chapter continues with presenting the segmentation and classification approaches of the architectural elements and is concluded by the illustration of the STYLE project GUI and the voting scheme on a sample facade image.

4.2 Segmentation of Domes

The current approach was introduced to the scientific community in [39].

Facade dome segmentation is a highly complex task, being a high-level semantic segmentation by an architectural element. Color-based segmentation approaches are not applicable, since color is not a distinctive feature and a single dome contains multiple color segments. Though domes have certain geometric forms, defined as hemispherical, onion, etc, shape analysis is also not suitable for segmentation, as these shapes may not be modeled owing unlimited variety.

The block-scheme of the dome segmentation algorithm is depicted in Figure 4.3. The detailed illustration of the algorithm is following below on a sample facade image.

Using symmetry as a feature is logical, since facades, like any artefacts, are highly symmetrical. Facades have symmetry specificities. Firstly, dominant symmetry axes are vertical. Secondly, whereas bilateral symmetry is common for historic facades, translational and rotational symmetries also take place. Bilateral symmetry is of interest for dome segmentation purpose, because domes are 3D objects preserving bilateral symmetry related to the vertical axis passing through their center in 2D projections. Thus at the first step the image bilateral symmetry axes are detected using the method proposed by Loy et al. [24]. In [24] matches of symmetric points are found using modern feature-based methods, such as [23], from which bilateral symmetry axes or centers of rotational symmetry are determined. The method is independent of the feature detector and descriptor used, requiring only robust, rotation-invariant matching and an orientation measure for each feature [24]. This method is successfully used also for detecting repeated structures on facades [45].

Figure 4.4a shows an example image with a tilted dome. Image bilateral symmetry axes and supporting symmetric points are displayed in Figure 4.4b. In case multiple symmetry axes are found, the axis with the strongest symmetry magnitude (supported by the biggest number of symmetry points) is chosen. Here the false positive symmetry descriptors are ignored, as they never succeed in building the strongest symmetry axis. As the dominant symmetry axis passes through the dome center and is vertical, the image is rectified by rotation, making the strongest symmetry axis vertical (Figure 4.4c). Images, whose strongest symmetry axis is vertical, skip this step. Bilateral symmetry detection [24] is performed once more on the rotated image to find the position (column) of the strongest symmetry axis.

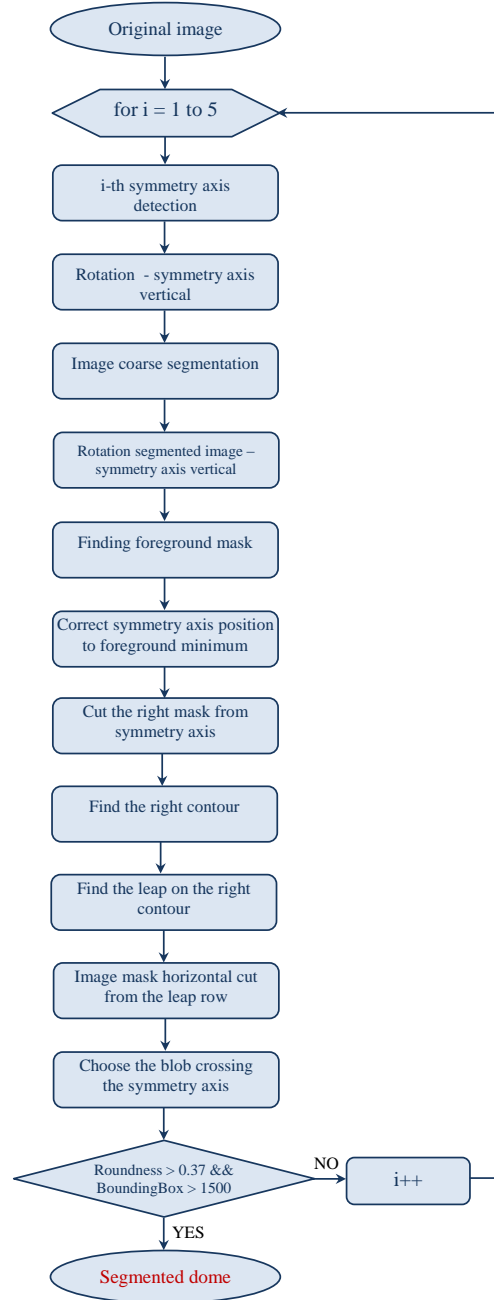


Figure 4.3: The block-scheme of the dome segmentation algorithm.



a) Original image



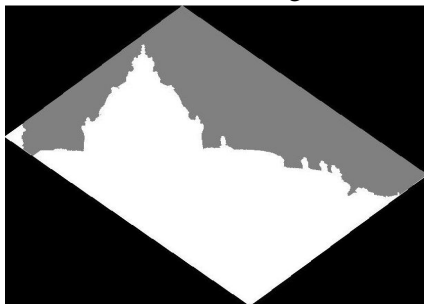
b) Symmetry axes and symmetric points [24]



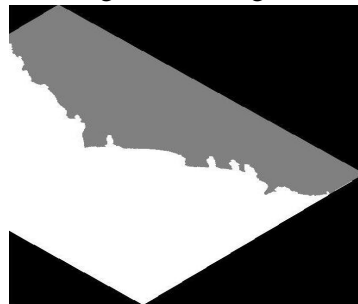
c) Rotated image



d) Segmented image [11]



e) Image mask



f) Figure e) cut from the main symmetry axis



g) (Half) Facade contour



h) Dome: final segment

Figure 4.4: Dome segmentation algorithm steps.

At the second step segmentation of the original image background and foreground is performed. As domes are situated high above the buildings, the sky and clouds form their background. The author segments the original image using the methodology introduced by Felzenszwalb et al. [11] (Figure 4.4d). The pairwise comparison of neighboring vertices, i.e. partitions is used to check for similarities [11, 14]. In [11] a definition of a pairwise group comparison function $Comp(\cdot, \cdot)$ is given, judging if there is evidence for a boundary between two image segments or not. $Comp(\cdot, \cdot)$ contains a scale parameter k , where bigger k prefers larger segmented regions. The function measures the difference along the boundary of two components relative to a measure of differences of components' internal differences. This definition tries to encapsulate the intuitive notion of contrast. Images are preprocessed by Gaussian blurring with σ before the segmentation, as well as postprocessed by merging small regions with the biggest neighboring one [11]. After the segmented image is also rotated to make the strongest symmetry axis vertical.

As the image is already justified so that the sky is on the top of it, the segment (color) of the sky is found by looking for the first non-black pixel starting from the first upper row, which is located on the strongest symmetry axis. Then the image foreground mask is obtained by setting all sky pixels to background color and non-sky pixels - to foreground color (Figure 4.4e).

Having found the image foreground mask, the next task is to analyze its shape in order to crop the dome. As domes are symmetric, for optimization only the image part cropped to the right from the strongest symmetry axis (Figure 4.4f) is analyzed. The strongest symmetry axis is expected to pass through the center of the dome, i.e. through the highest point of the dome. For images, whose symmetry axis position is shifted left or right from the dome center due to perspective distortions, the symmetry axis position is corrected by looking on the image mask for the foreground maxima in the symmetry axis local neighborhood. This step of the methodology means, if the initially detected vertical symmetry axis lies on the dome, it justifies itself to the correct position, making the approach robust to high perspective distortions.

Now the purpose is to find the bottom row of the segmented dome. Here the author uses a visual feature of domes: domes raise out of the main building. This means that the facade contour, formed by the foreground pixel followed by a background pixel in each row (Figure 4.4g), has a leap on the row where the dome meets the main building. The observed leap is found by scanning down row by row the facade contour in Figure 4.4g until the condition in Equation 4.1 is satisfied:

$$Leap(k) / (Row(k) - Row(1)) > 0.15 \ \&\& \ Leap(k) > minLeapThreshold \quad (4.1)$$

where $Leap(k)$ is the column difference of contour pixels on k th and $(k - 1)$ th sequential rows: $Leap(k) = Col(k) - Col(k - 1)$

$Row(k) - Row(1)$ is a normalization factor and is the difference of the k th and first rows of the contour. $minLeapThreshold$ excludes too small leaps between two subsequent rows. It is set to the empirically found value of 18 pixels for images with resolution lower than 1 million pixels and to 26 pixels otherwise. After the image mask is cropped from the found row to discard the image part below, which does not contain the segment of interest. In order to obtain the final dome segment, the blob through which the main symmetry axis passes is picked up and all the other blobs, formed by clouds, trees and any other objects present in the image, are

discarded. Multiplication of the blob mask with the same segment of the original image delivers the segmented dome (Figure 4.4h).

Still a further step is taken by incorporating the feature roundness for domes, to address the following types of complex images:

- 1) the strongest symmetry axis does not pass through the dome center, due to too high perspective distortions or other symmetric objects present in the image,
- 2) the strongest symmetry axis is located on the dome, but is horizontal,
- 3) the facade is reflected in water, thus the strongest symmetry axis is horizontal.

The feature roundness is calculated by Equation 4.2.

$$\text{Roundness} = 4 * \text{Area} / (\pi * \text{MajorAxisLength}^2) \quad (4.2)$$

$$\text{Roundness} > 0.37 \ \&\& \ \text{DomeBoundingBox} > 1500 \quad (4.3)$$

If the dome blob roundness and bounding box resolution pass the experimentally found thresholds in Equation 4.3, the dome segmentation is considered successful. The thresholds in Equation 4.1 and Equation 4.3 were found by experimenting on multiple images. Setting a threshold for dome bounding box resolution excludes too small blobs. In case the condition in Equation 4.3 is not met, the whole segmentation algorithm is rerun by taking the 2nd strongest symmetry axis in the initial step. After the condition in Equation 4.3 is checked again and if not satisfied, the algorithm is rerun by the 3rd strongest symmetry axis and so on. The author limits the dome segment search by the 5th strongest symmetry axis (if such exists), since empirically was found (Section 5.2) that further search was useless.

4.3 Segmentation of Towers

Segmentation of towers, like domes, is a high-level semantic segmentation by an architectural element. Like the dome segmentation problem, in this case also color-based segmentation approaches are not applicable, since color is not a distinctive feature and a single tower may contain multiple color segments. Shape analysis is also not suitable for segmentation, as tower shapes cannot be modeled due to unlimited variety.

The tower segmentation approach is based on the dome segmentation approach, illustrated in Section 4.2 and published in [39], modifying and extending it taking into account the architectural specificities of towers. Such a choice is justified by the following common characteristics of towers and domes:

- Both elements display vertical bilateral symmetry
- Both elements are situated high above the building and have the sky and clouds as background

The tower segmentation approach has two branches, aiming to handle buildings, featuring a single tower and double towers respectively. The block-scheme of the tower segmentation algorithm is depicted in Figure 4.5. The detailed illustration of the algorithm is following below on sample images of single and double tower facades.

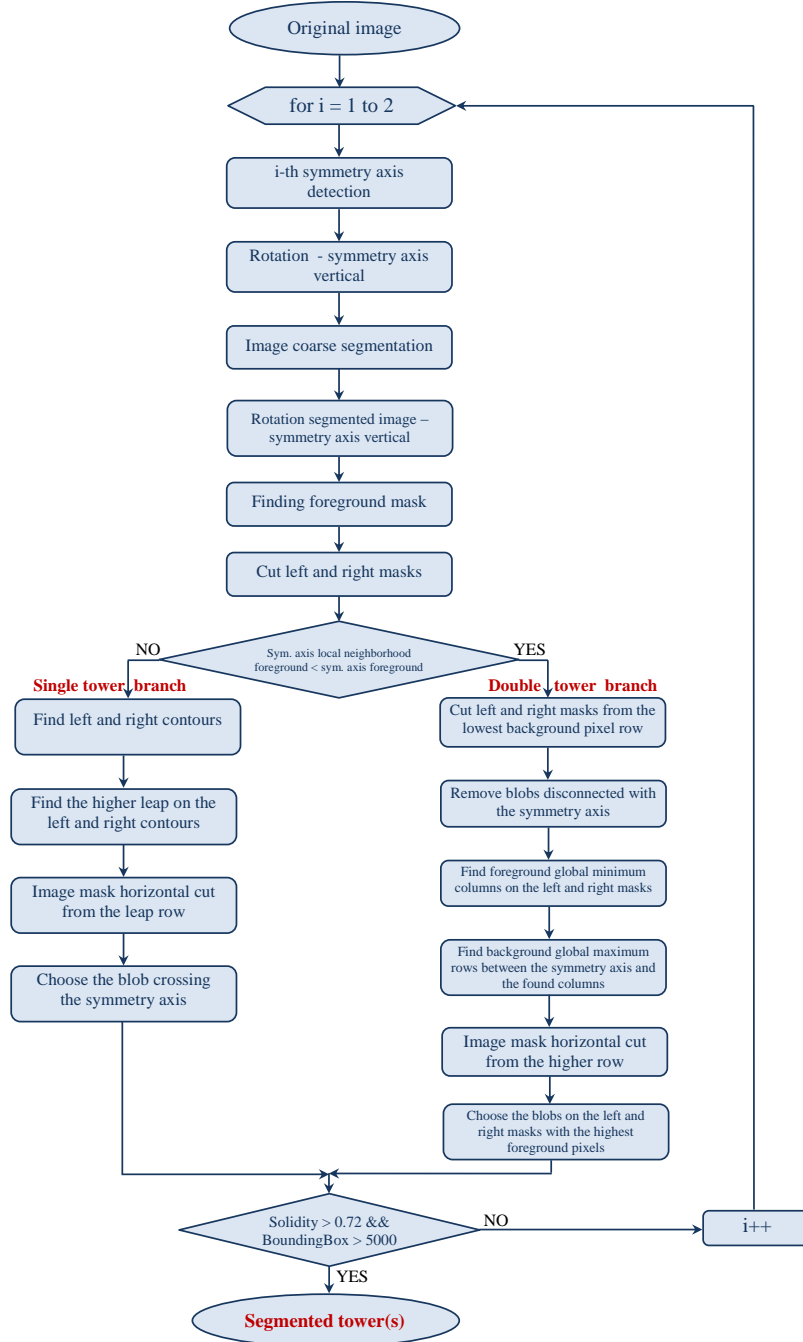


Figure 4.5: The block-scheme of the tower segmentation algorithm with branches for facades featuring a single tower and double towers.

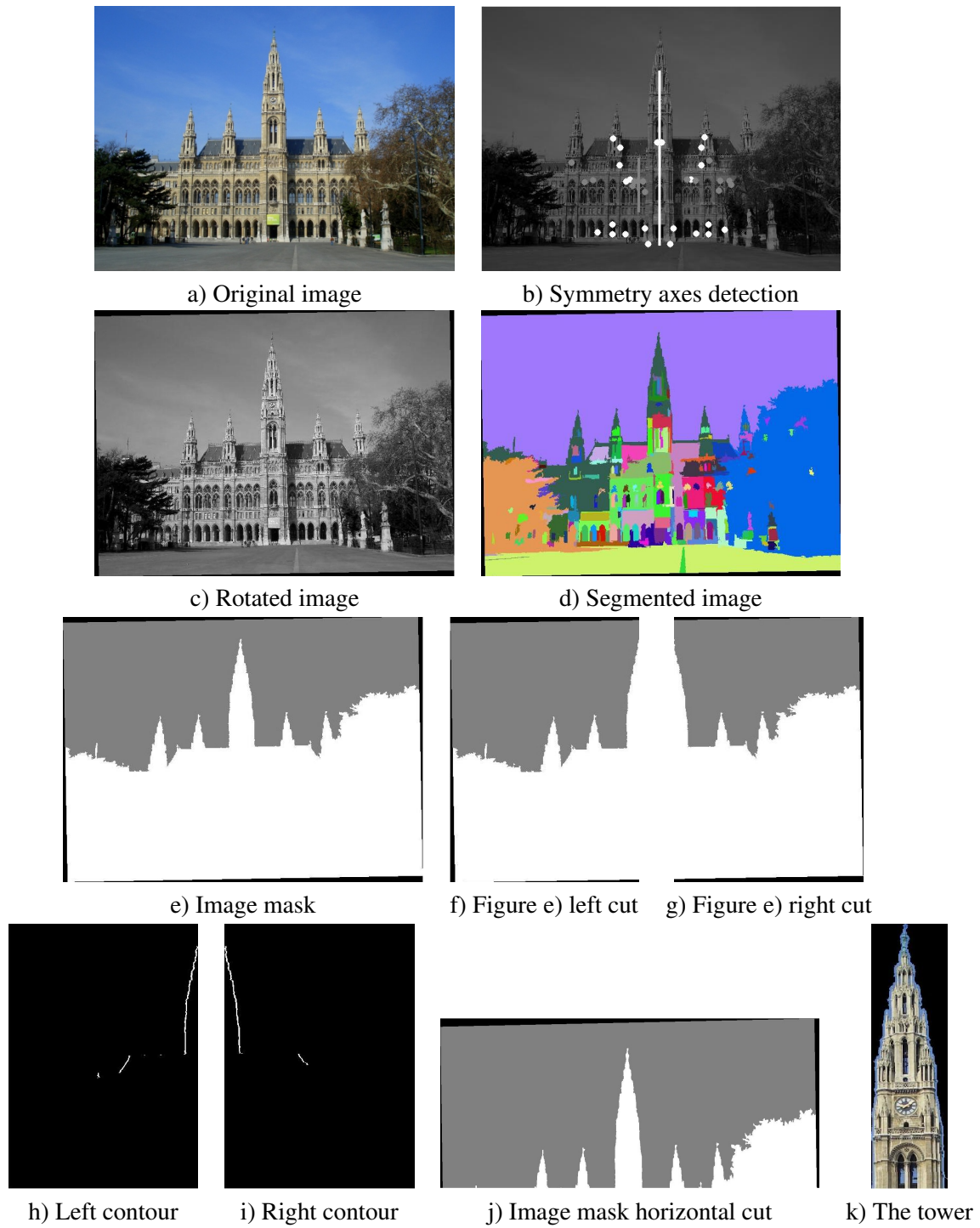


Figure 4.6: Algorithm steps for segmentation of a single tower.

The branch of single tower segmentation is illustrated on the example of Vienna City Hall (Figure 4.6a), of double tower segmentation – Vienna Church of Mariahilf (Figure 4.7a). At the first step the image bilateral symmetry axes are detected, using the method proposed by [24]. The bilateral symmetry axes and the supporting symmetric points of images in Figure 4.6a and Figure 4.7a are displayed in Figure 4.6b and Figure 4.7b respectively. In case multiple symmetry axes are found, the axis with the strongest symmetry magnitude (supported by the biggest number of symmetry points) is chosen. There is priori knowledge that the dominant symmetry axis is vertical for historic facades, so the algorithm proceeds with rectifying the images by rotation, to make the strongest symmetry axis vertical (Figure 4.6c, Figure 4.7c). Images, whose strongest symmetry axis is vertical, skip this step. Bilateral symmetry detection [24] is performed once more on the rotated image to find the position (column) of the strongest symmetry axis.

The purpose of the next step is the segmentation of the original image background and foreground. Since towers like domes are situated high above the buildings, the sky and clouds form their background. The original image is segmented using the methodology introduced by [11] (Figure 4.6d, Figure 4.7d). Afterwards the segmented image is also rotated to make the strongest symmetry axis vertical. As the example images are already justified so that the sky is on the top of them, the segment (color) of the sky is found by tracing down the strongest symmetry axis starting from the first upper row until the first non-black pixel is found. Then the image foreground mask is obtained by setting all sky pixels to background color and non-sky pixels - to foreground color (Figure 4.6e, Figure 4.7e).

In case of domes the strongest symmetry axis is expected to pass through the center of the dome, i.e. through the highest point of the dome. That is why for images, whose symmetry axis position is shifted left or right from the dome center due to perspective distortions, the symmetry axis position was corrected by looking on the image mask for the foreground maxima in the symmetry axis local neighborhood (Section 4.2) [39]. For the tower segmentation purpose the symmetry axis position correction step mentioned is omitted, because there are image instances with double towers, where the towers lie very close to the symmetry axis and the symmetry axis justification shifts the symmetry axis to a false position. Having the image foreground mask found, the next step is to analyze its shape in order to crop the tower(s). Now the purpose is to find the bottom row of the tower(s). The image masks are cut from the main symmetry axis into left (Figure 4.6f, Figure 4.7f) and right (Figure 4.6g, Figure 4.7g) parts.

To this point the author followed the dome segmentation algorithm (Section 4.2) [39], except for the correction of the initial position of the strongest symmetry axis. Here a new step specific for tower segmentation is introduced: a condition is analyzed to find out, if the building features a single or double towers. If the local neighborhood both on the right and left sides of the main symmetry axis have higher situated foreground pixels than that on the main symmetry axis, the building features double towers. More precisely, the neighborhood on the right and left sides from the main symmetry axis is defined as the adjacent 1/8th parts of the right and left images.

The single tower segmentation branch proceeds as follows. As domes are symmetric in all 2D projections, for optimization only the right image mask is analyzed in Section 4.2 [39]. Whereas towers are more sensitive to camera viewpoint, that is why analysis of both left and right image masks is needed. Domes and towers possess a common visual property: they raise

out of the main building. This means that the facade left and right contours, formed by the foreground pixel adjacent to a background pixel in each row (Figure 4.6h and Figure 4.6i), have leaps on the rows, where the tower meets the main building. The observed leaps are found by tracing down row by row the facade left and right contours in Figure 4.6h and Figure 4.6i until the condition in Equation 4.4 is satisfied:

$$\text{Leap}(k) / (\text{Row}(k) - \text{Row}(1)) > \text{LeapThreshold} \ \&\& \ \text{Leap}(k) > \text{minLeapThreshold} \quad (4.4)$$

where $\text{Leap}(k)$ is the column difference of contour pixels on k th and $(k - 1)$ th sequential rows: $\text{Leap}(k) = \text{Col}(k) - \text{Col}(k - 1)$

$\text{Row}(k) - \text{Row}(1)$ is a normalization factor and is the difference of the k th and first rows of the contour. minLeapThreshold excludes too small leaps between two subsequent rows and is set to 18 pixels for images with resolution lower than 1 million pixels and to 26 pixels otherwise. LeapThreshold was found empirically and was equal to 0.15 for dome segmentation (Section 4.2) [39] and to 0.24 for tower segmentation. Note that the facade contours in Figure 4.6h and Figure 4.6i are thickened from 1 pixel to 10 pixel width for visibility purpose only. As the final row, containing the leap, is chosen the one of right and left leap rows, which lies higher. After the image mask is cropped from the found leap row (Figure 4.6j) to discard the image part below, which does not include the segment of interest. In order to obtain the final tower segment, the blob through which the main symmetry axis passes is picked up and all the other blobs formed by clouds, trees and any other objects present in the image are discarded. Multiplication of the blob mask with the same segment of the original image delivers the segmented tower (Figure 4.6k).

The branch, addressing the segmentation of double towers, is also a novel introduction of this chapter and performs the following processing. The image left and right masks (Figure 4.7f and Figure 4.7g) are cut horizontally from the row of the lowest background pixel. Then the blobs, which are not connected with the strongest symmetry axis, are removed (Figure 4.7h and Figure 4.7i). Note that the strongest symmetry axis is located on the last column of the left mask (Figure 4.6f) and on the first one – of the right (Figure 4.6g). The removal of the disconnected blobs aims to eliminate the objects (trees, clouds, etc), which can be situated higher than the towers. Afterwards the columns of the global foreground minimums are located on the left and right image masks. Then the global background maximum rows are found between the strongest symmetry axis and the columns mentioned. The image is cut horizontally from the row situated higher (Figure 4.6j). And finally the left (Figure 4.6k) and right (Figure 4.6l) towers are detected by picking up the blobs, containing the highest foreground pixels on the left and right masks.

To address the images, in which the first strongest symmetry axis is not placed on the tower (single tower case) or between towers (double tower case) due to too high perspective distortions or other symmetric objects present in the image, an additional feature, called solidity, is introduced. Solidity is a region property, specifying the proportion of the pixels in the convex hull that are also in the region and is computed by Equation 4.5. If the tower blob solidity and bounding box resolution pass the thresholds in Equation 4.6, the tower segmentation is considered successful.

$$\text{Solidity} = \text{Area} / \text{ConvexArea} \quad (4.5)$$

$$\text{Solidity} > 0.72 \ \&\& \ \text{TowerBoundingBox} > 5000 \quad (4.6)$$

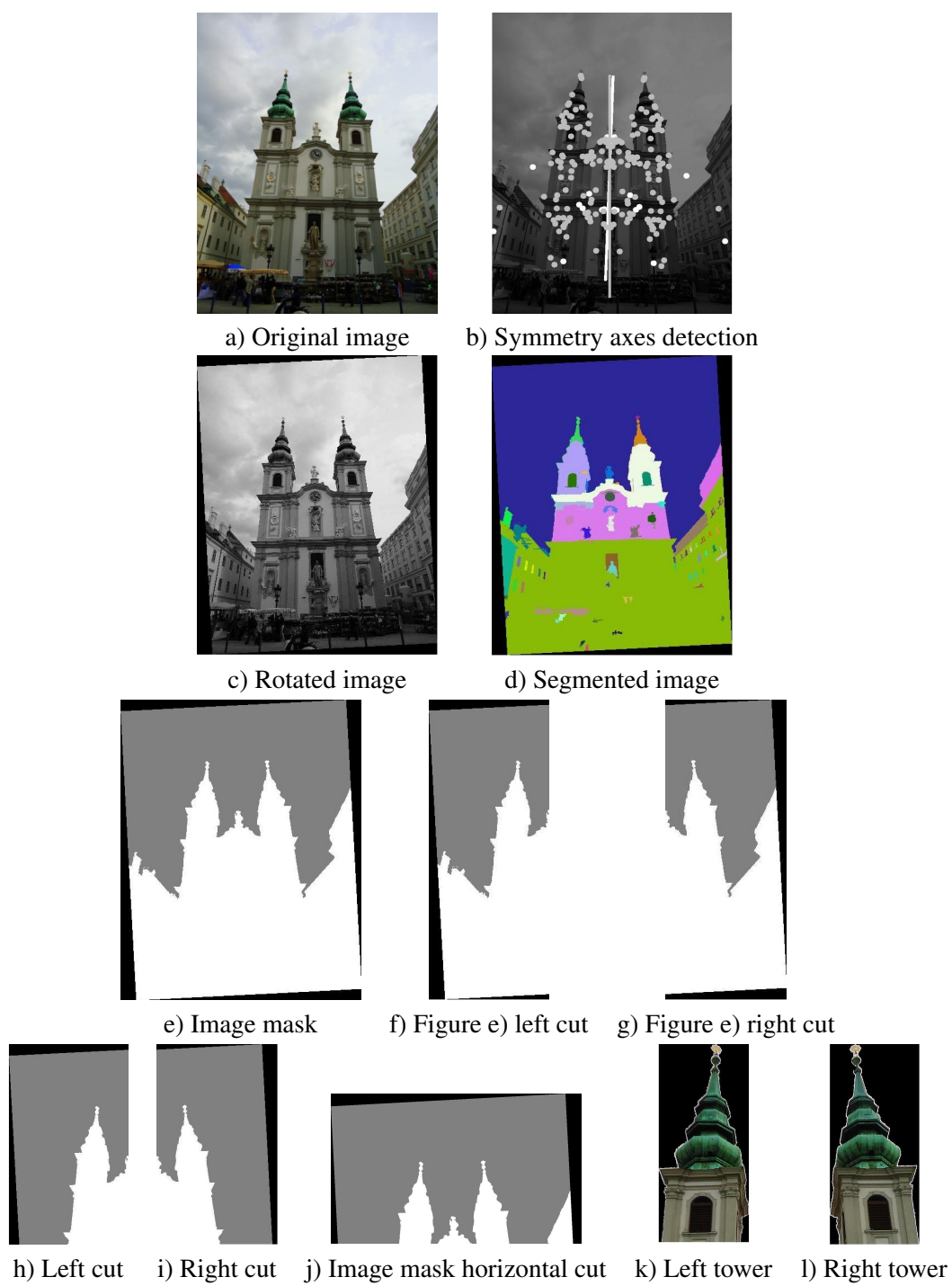


Figure 4.7: Algorithm steps for segmentation of double towers.

The thresholds in Equation 4.4 and Equation 4.6 were found empirically, exercising multiple images. Setting a threshold for tower bounding box resolution excludes too small blobs. In case the condition in Equation 4.6 is not met, the whole segmentation algorithm pipeline is rerun by taking the 2nd strongest symmetry axis in the initial step. The tower segment search is limited by the 2nd strongest symmetry axis (if such exists), since the evaluation database lacks images, for which further search would lead to segmentation success (Section 5.3). Note for comparison, that the dome segmentation approach in Section 4.2 [39] judges the segmentation success by feature roundness, instead of solidity and seeks for the dome segment in the vicinity of up to the 5th strongest symmetry axis, instead of the 2nd, as in the current method.

4.4 Architectural Style Classification of Building Facade Windows

The current approach, addressing the classification of Romanesque (Figure 3.1), Gothic (Figure 3.2) and Baroque (Figure 3.3) windows, was published in [36].

The task of classification of windows by architectural styles is highly complex, because of the high intra-class diversity as well as reflections present in window images. One can use different texture features [15,31] as well as established shape descriptors [3,46]. A local feature-based approach is employed, since it incorporates texture and gradients into an image descriptor. It is shown in [8] that shapes can be represented by local features (peaks and ridges). Since on window shapes of each class certain gradient directions are dominating, local features are used to describe shapes. One can use different local features, like Harris-Laplacian corner detectors [16, 28], difference of Gaussians corner detectors [23] or detectors based on regions [26,43] and local image descriptors [23,26,43]. The goal is to extract characteristic gradient directions, like those describing the pointed arch or triangular pediment and to minimize the influence of non-relevant features, like those from reflections and curtains.

The standard bag of words approach presented by Csurka et al [9] (Figure 4.8) is chosen. In the learning phase the SIFT [23] is used to extract the information of gradient directions. After performing the difference of Gaussians on different octaves and finding minimas/maximas, i.e.

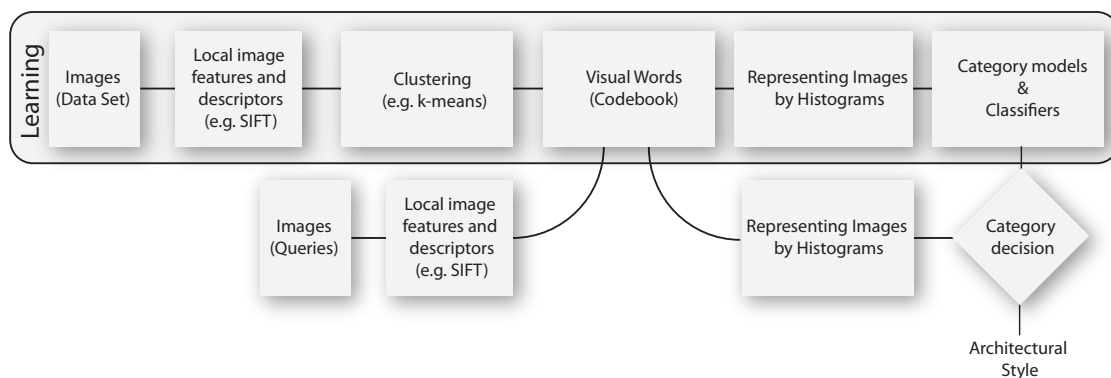


Figure 4.8: Learning visual words and classification scheme.

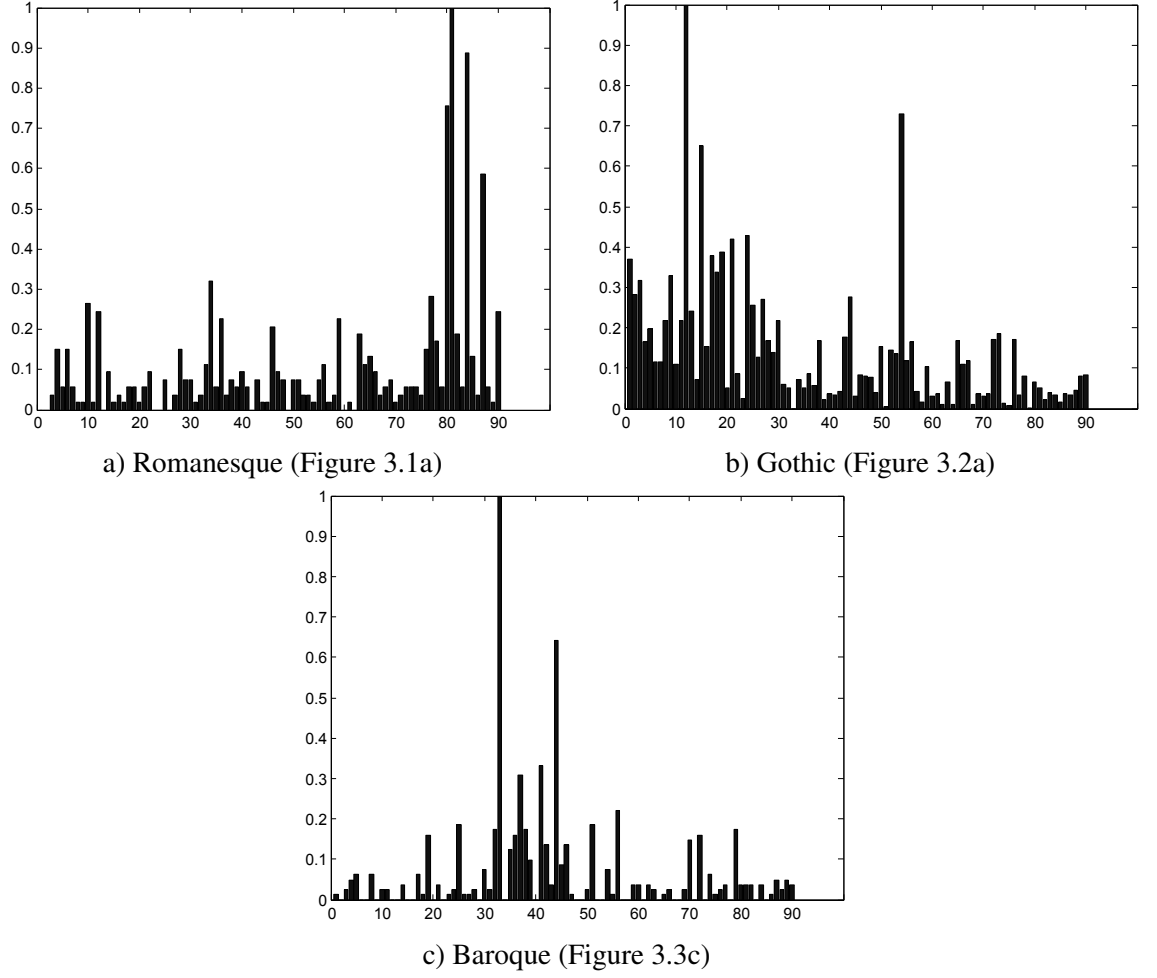


Figure 4.9: Histograms of visual words of the images of different window styles.

finding interest points, rejection of interest points is performed with low contrast by setting a low threshold. All interest points that lie on window edges are kept. Note that the original work [23] is not followed in this step, i.e. the response of the filter along the edges is not suppressed. After finding the interest points the algorithm proceeds with finding local image descriptors (SIFT image descriptors) and normalizing them. The number of local features is large, thus clustering is applied to learn a visual vocabulary (codebook). The codebook of separate classes is made by searching for the visual cluster centers using unsupervised k-means algorithm. The codebook is learnt on a training set. Since the visual codebook plays a central role in all classification approaches, introduced in the current work, the source code responsible for its generation is brought in Appendix A.4.

The classification of a query image begins likewise with the extraction and normalization of SIFT descriptors. The next step is to classify the SIFT descriptors using the codebook of

visual cluster centers, generated during the training stage. For this purpose the author employs the simple, yet efficient nearest neighbor (NN) classifier, which is a special case of the KNN classifier, when $K^1=1$. The Euclidean distances of the query SIFT descriptor to all visual cluster centers are calculated. The nearest cluster center is determined as the class of the SIFT descriptor. After all image descriptors are classified, the image histogram is made, showing the classification distribution of the descriptors among visual cluster centers. The histograms shown in Figure 4.9 are built using a codebook of 30 cluster centers for each class and are normalized in the interval $[0, 1]$. Note that for Romanesque class histogram high responses are located on the bins from 61 to 90, for Gothic class - from 1 to 30 and for Baroque class - from 31 to 60. The class, to which the majority of image descriptors belong, is chosen as the image class, i.e. the image class is decided by finding the maxima of integrated responses of the 3 classes. For example, for the histogram representation shown Figure 4.9a, the sum of all responses of Romanesque class is 5.6038, Gothic class – 1.8868 and Baroque class – 2.3019. Thus the image is correctly classified as Romanesque. Determining the window class by the integrated class responses proves to be effective, as it makes the vote for the right class strong by integration of the high responses and suppresses the false class peaks, which may occur due to irrelevant descriptors located on architectural details, reflections and curtains.

4.5 Architectural Style Classification of Traceries, Pediments and Balustrades

The approach, addressing the problem of architectural style classification of traceries (Figure 3.4, Figure 3.5), pediments (Figure 3.6a-e) and balustrades (Figure 3.6f), was published in [38]. It employs the method, introduced in Section 4.4 [36] for classification of windows,

¹Observe that KNN classifier parameter is noted K (upper case), not to confuse with k parameter of k-means clustering during codebook generation.

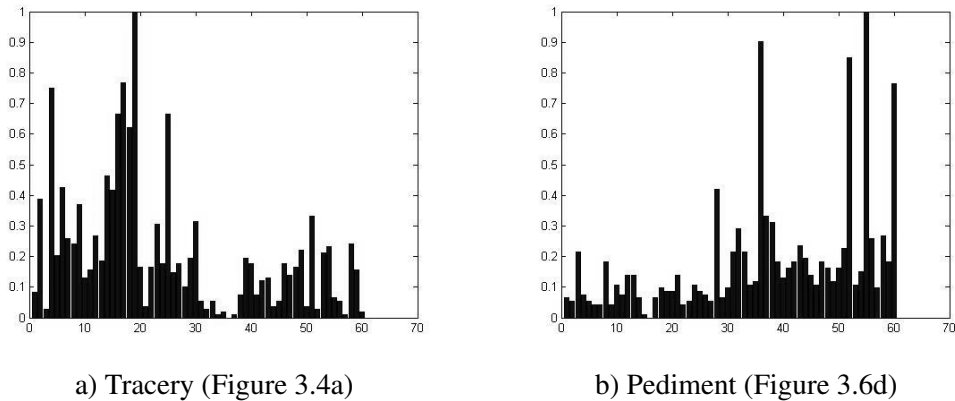


Figure 4.10: Histograms of visual words of Gothic and Baroque elements.

and finds empirically (Section 5.5) the codebook best fitting the database of the architectural elements explored. A histogram instance of the tracery in Figure 3.4a is shown in Figure 4.10a, whereas Figure 4.10b displays an example histogram of the pediment in Figure 3.6d. For the histogram in Figure 4.10b, the sum of all responses of Gothic class makes 2.78, and of Baroque class – 8.35, so the image is correctly classified as Baroque. The histograms shown in Figure 4.10 are built using a codebook of 30 cluster centers for both classes. Observe that for Gothic class histogram high responses are located on the bins from 1 to 30 and for Baroque class - from 31 to 60.

4.6 Architectural Style Classification of Domes

The dome classification module classifies Islamic (Figure 3.7e, Figure 3.8b), Russian (Figure 3.8a) and Renaissance (Figure 3.7a-d) domes and was published in [37]. Classification of dome images by architectural styles is a highly complex task, since dome images are projections of 3D domes in 2D and due to high variability of architectural details and ornaments. The proposed methodology for solving the task is a three-step approach. At the first step, the dimensions of the query image are analyzed. The Islamic saucer dome is the only dome type for which the width is always greater than the height, because of its shallow geometry. Also Islamic onion domes may have their width greater than the height. In either case the query image is classified as Islamic, when the mentioned condition is true.

At the second step color as a feature is used to identify Russian golden onion domes. Though gilded domes are typical for Russian Orthodox churches, they are not the only option. Russian churches also have examples of blue and colorful domes. So by golden color detection at this step the classification rate among Russian gilded onion domes is raised.

For golden color detection the author chooses YCbCr color space, where Y color channel represents luminance, Cb and Cr channels - chrominance. The advantage of YCbCr color space

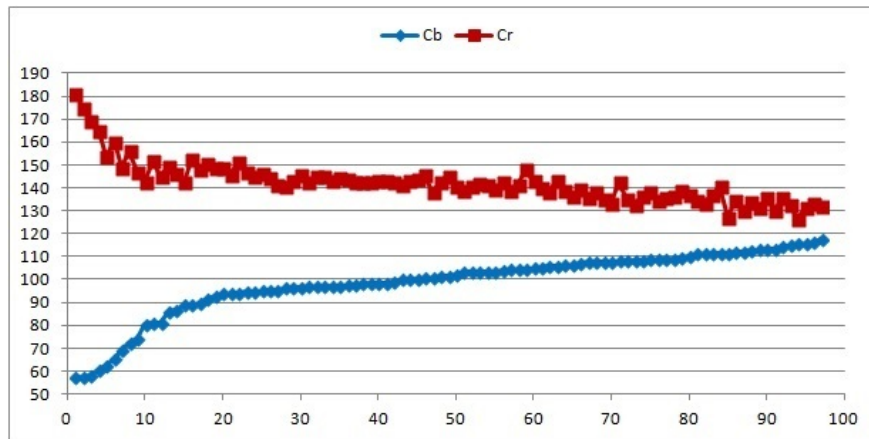


Figure 4.11: Cb and Cr mean values of 97 golden dome patches.

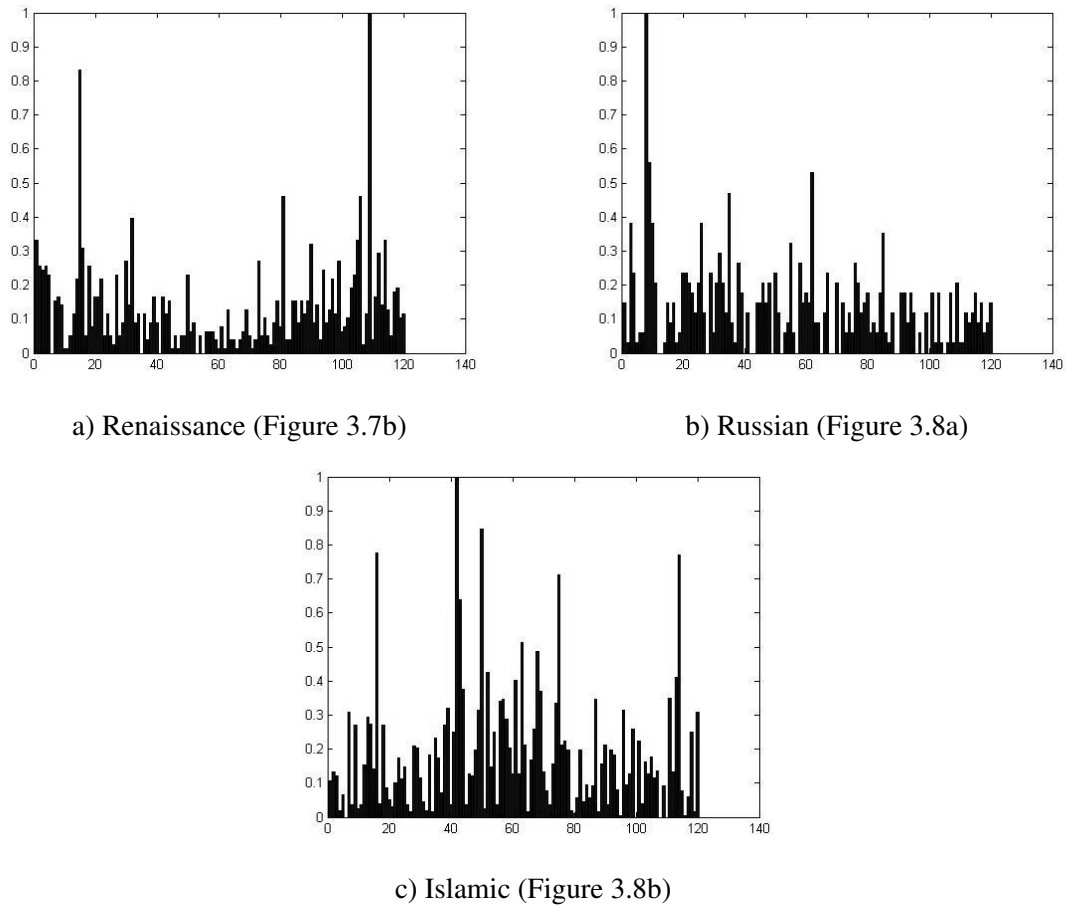


Figure 4.12: Histograms of visual words of the images of different dome styles.

is that it represents luminance (Y) in a single channel and is thus luma-independent. YCbCr color space is widely used for different color detection purposes, including skin detection [2] and facial feature detection [25]. The goal is to find experimentally the ranges in Cb and Cr channels corresponding to golden color, while discarding Y channel. Thus, the mean pixel values of Cb and Cr components of 97 golden patches are counted, which were cut manually from Russian gilded dome images. The reason for considering the mean Cb, Cr values of all image pixels, but not Cb, Cr values of individual pixels is that golden color due to its high reflectivity tends to appear from white to dark brown. So while determining the initial Cb and Cr ranges for golden color the individual pixel values may be false. From each patch a pair of mean Cb and mean Cr components is extracted. Each mean Cb value on Cb line corresponds to a mean Cr value located above it on Cr line (Figure 4.11). The author sorts the mean Cb values in ascending order to find out if there is a relation between Cb and Cr components. As seen in Figure 4.11, with ascending of Cb values the Cr values tend to decrease. Low Cb and high Cr values correspond to highly saturated golden color. The full range of Cb and Cr values is between 16 and 240. In the

mentioned range the initial Cb, Cr ranges found corresponding to golden color are as follows: $58 < Cb < 118$; $126 < Cr < 181$. The final Cb, Cr ranges are determined iteratively, maximizing the true positives and minimizing the false positives while golden color image detection on the training set (Section 5.6).

At the final third step classification of domes is done based on their shapes, employing the approach introduced in Section 4.4 [36] for classification of windows. The codebook, best fitting the database of the dome types explored, is found empirically (Section 5.6). Histograms instances of the 3 classes classified, built using a codebook of 40 cluster centers for each class, are shown in Figure 4.12. Observe that for Russian class histogram high responses are located on the bins from 1 to 40, for Islamic class - from 41 to 80 and for Renaissance class - from 81 to 120. For example, for the histogram representation shown in Figure 4.12c, the sum of all responses of Russian class is 5.72, Islamic class – 10.76 and Renaissance class – 6.03. Thus the image is correctly classified as Islamic.

Islamic saucer domes are not included in the training stage, as query saucer dome images pass accurate classification at the first step and do not need a codebook for classification. The steps of the approach are counted by their priority. If the image is identified as Islamic at the first step, it does not pass to the further steps for analysis. Otherwise, it is passed to the second step for golden color detection. If the response at this step is positive, the dome is classified as Russian, otherwise it is passed to the third step for shape analysis.

4.7 Architectural Style Classification of Towers

The tower classification module classifies Romanesque, Gothic and Baroque towers. It also employs the method, introduced in Section 4.4 [36] for classification of windows and finds empirically (Section 5.7) the codebook best fitting the database of towers. Romanesque, Gothic and Baroque tower samples are displayed respectively in Figure 4.13d, Figure 4.13e and Figure 4.13f. The towers are segmented by the approach illustrated in Section 4.3 [39] from the corresponding facade images located above the towers - Figure 4.13a (St. Anton church in Vienna), Figure 4.13b (Vienna City Hall) and Figure 4.13c (Mariahilf church in Vienna). The histogram representations of the segmented towers in Figure 4.13d are shown in Figure 4.13g and Figure 4.13h. Similarly Figure 4.13i exhibits the histogram of the tower in Figure 4.13e and Figure 4.13j, Figure 4.13k – of the towers in Figure 4.13f. The histograms are built using a codebook of 40 cluster centers for each class and are normalized in the interval [0, 1].

Note that for Baroque class histogram high responses are located on the bins from 1 to 40, for Gothic class - from 41 to 80 and for Romanesque class - from 81 to 120. The class, to which the majority of image descriptors belong, is chosen as the image class, i.e. the image class is decided by finding the maxima of integrated responses of the 3 classes. For example, for the histogram presented in Figure 4.13i, the sum of all responses of Baroque class is 8.51, Gothic class – 19.58 and Romanesque class – 8.86. Thus the image is correctly classified as Gothic.

Besides using the KNN classifier with $K=1$, as in Section 4.4 [36], an additional experiment is conducted to see, if the classification accuracy may increase on the training dataset by applying $K>1$ values. But simply using the KNN classifier with equal weights to all K nearest neighbors in this 3-class classification scheme may lead to classification ambiguity, when



a) Romanesque



b) Gothic



c) Baroque



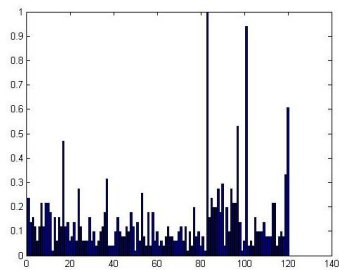
d) The towers of Figure 4.13a



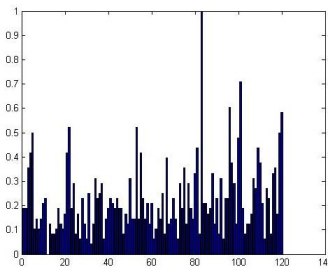
e) The tower of Figure 4.13b



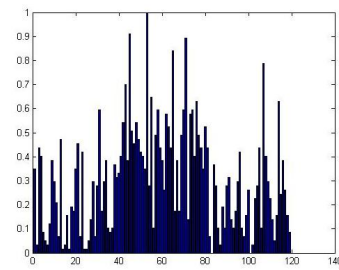
f) The towers of Figure 4.13c



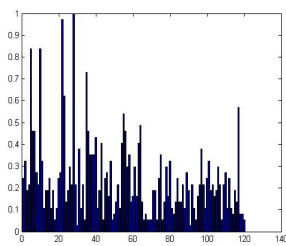
g) Figure 4.13d left



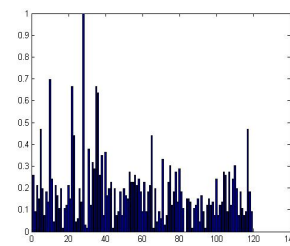
h) Figure 4.13d right



i) Figure 4.13e



j) Figure 4.13f left



k) Figure 4.13f right

Figure 4.13: Facades of Romanesque, Gothic, Baroque styles, the segmented towers and the histograms of visual words.

the classes have equal number of nearest neighbors. To exclude the ambiguous cases and to weigh the closer neighbors more heavily than the farther ones, the author chooses the distance-weighted KNN rule (WKNN) proposed by [10]. According to the rule the votes of the different members among K nearest neighbors are weighed by a function of their distances to the query object (Equation 4.7).

$$w_i = \begin{cases} (d_K^{NN} - d_i^{NN}) / (d_K^{NN} - d_1^{NN}) & ; d_K^{NN} \neq d_1^{NN} \\ 1 & ; d_K^{NN} = d_1^{NN} \end{cases} \quad (4.7)$$

where d_i^{NN} is the distance of the i-th nearest neighbor to the query object, d_1^{NN} is the distance of the nearest neighbor, d_K^{NN} is the distance of the K-farthest neighbor. In Equation 4.7 the nearest neighbor's weight is equal to 1, the farthest neighbor's – to 0, the rest of the neighbors get their weights in the interval (0,1), so that a neighbor with smaller distance scores a heavier weight than one with greater distance.

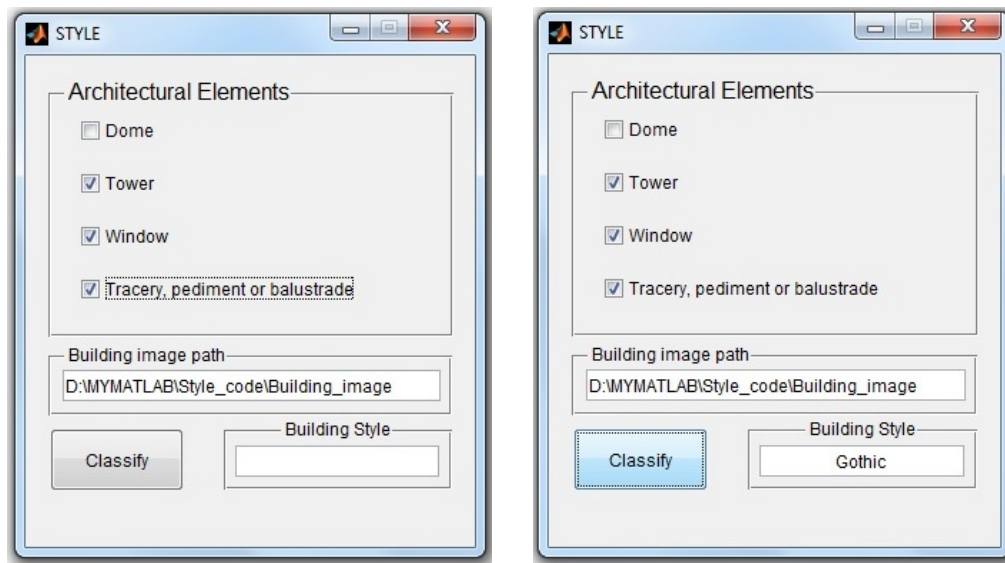
Nevertheless the experiment (Section 5.7) shows that with the increase of K, the classification precision on the training set decreases, thus the classification of the testing dataset will be led by NN classifier, i.e. K=1 value.

4.8 The STYLE GUI and the Scheme of the Voting of Architectural Elements

Running the STYLE program initializes a simple Graphical User Interface (GUI) (Figure 4.14a), where the user should check the architectural elements, present on the query facade image. The path of the query image to be classified can be provided in the field topped with “Building path” label, if other than by default displayed path. Whereas towers and domes will be segmented automatically, user interaction is required to deliver the bounding boxes of windows, tracteries, pediments and balustrades, if such exist. The manually segmented windows should be placed in “SegmentedWindows” subdirectory of building path directory, tracteries, pediments, balustrades – in “SegmentedElements” subdirectory .

The program is demonstrated on the example of a query image, displaying the neo-Gothic facade of Votiv church in Vienna (Figure 4.15). The facade contains typical Gothic elements – pointed arch windows (Figure 4.16a, Figure 4.16b, Figure 4.16c), towers (Figure 4.16d, Figure 4.16e) and tracteries (Figure 4.16f, Figure 4.16g, Figure 4.16h, Figure 4.16i). After the checkboxes “Tower”, “Window”, “Tracery, pediment or balustrade” are checked and the bounding boxes of windows and tracteries are cropped and put in the corresponding folders, “Classify” button should be pressed to classify the architectural style of the building. The style of the building will be displayed in the field, labeled “Building style” (Figure 4.14b). The style of the query image is classified as Gothic, since all 9 architectural elements, depicted in Figure 4.16, have voted for the Gothic style. Figure 4.15 is a good example, as it shows that though the facade is partly occluded, it is correctly classified by the visible typical elements of Gothic style.

As declared before, the architectural style of the building is determined as the class, having received maximum votes. In the following cases the building style will be classified as a mixture of Romanesque and Gothic, which is more probable:



a) The STYLE GUI with checked elements b) Building style classified as Gothic

Figure 4.14: GUI demonstration of the STYLE project.

1. All the 3 classes have equal number of votes.
 2. The Romanesque and Gothic classes have equal number of votes, which is greater than the votes for Baroque class.
 3. The Romanesque class has maximum votes and there are 2 and more Gothic votes. Here the Gothic votes are thresholded by 2, since 1 vote could be the outcome of misclassification.
 4. The Gothic class has maximum votes and there are 2 and more Romanesque votes.
- The source code of the function, executing the algorithm of the voting of architectural elements, is presented in Section A.4.

4.9 Summary

The current chapter presented the main contribution of the work - the methodology of the STYLE project, addressing the problem of image-based classification of building facades by architectural styles. The imitation of the human cognitive process of architectural style classification makes the basis of the method proposed. The architectural style of a facade is formed by style-typical architectural elements. The method comprises three major steps - segmentation, classification and voting of architectural elements (Section 4.1). Whereas the algorithm is general enough to classify any visually recognizable architectural style, its software implementation is carried out on pan-European architectural styles Romanesque, Gothic and Baroque.

In the scope of the segmentation step are presented the first approaches, segmenting the outstanding architectural elements dome and tower. The dome segmentation approach (Section 4.2) is a pipeline employing the visual features of this prominent architectural element - vertical bilateral symmetry, raising out of the main building and roundness. The algorithm successfully

incorporates bilateral symmetry detection and segmentation approaches with image analysis and processing technics to achieve very high segmentation rate (Section 5.2). The tower segmentation approach (Section 4.3) is based on the dome segmentation method, since these architectural elements share common visual features - vertical bilateral symmetry and raising out of the main building. The approach is divided into two branches targeting the segmentation of single tower and double tower facades respectively. The validation of the segmented tower is checked by the feature solidity, instead of roundness as for the dome case.

The visual information about architectural styles is learned in the classification stage. The first approach classifies the architectural style of windows (Section 4.4) by the means of clustering and learning of local features. The same approach is employed for the classification of architectural elements tracery, pediment and balustrade (Section 4.5) and tower (Section 4.7), tuning the parameters of the clustering and feature extraction to fit best to the experimental image dataset. The classification of domes (Section 4.6) is a three step approach, which at the first step analyzes the width and height proportion of the dome for the identification of the Islamic domes, at the second step performs golden color detection in YCbCr color space to detect Russian gilded domes and at the third step, likewise the previous classification approaches, clusters and learns local features. Here as an alternative to Renaissance/Baroque domes are classified Russian and Islamic domes, since the dome is not typical for Romanesque and Gothic architectural styles.

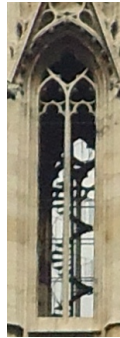
In the third stage the voting of the classified architectural elements takes place (Section 4.8). The architectural style, having received the maximum of votes, is chosen as the style of the building. The voting mechanism also permits the classification of mixtures of architectural styles which are probable. In the current state only the mixture of Romanesque and Gothic styles is allowed.



Figure 4.15: Query image for classification – Votiv church in Vienna



a) Pointed window



b) Pointed window



c) Pointed window



d) The left tower



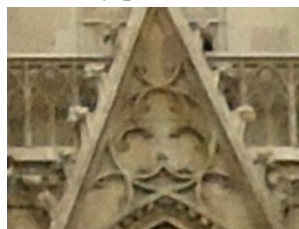
e) The right tower



f) Tracery pattern



g) Tracery pattern



h) Tracery pattern



i) Tracery pattern

Figure 4.16: Segmented architectural elements from Figure 4.15

Experiments, Performance Evaluation and Results

5.1 The Experimental Setup and Creation of Image Databases

An extensive study was conducted to test and evaluate each module of the STYLE classification system. One of the challenges for testing the system is the lack of databases labeled by architectural styles. To test each module the author collected an image database of the corresponding architectural styles and featuring the corresponding architectural elements. The database includes the author's own photographs, as well as images from Flickr ¹ image database, overall 2315 images. Though the segmentation modules are not sensitive to the resolution of the input images, the modules classifying architectural elements – are. The reason for this is that the lower image resolution, the less descriptors are extracted and the sparser and less discriminative image histograms become. The classification modules receive as an input segments of the original image, picturing the architectural elements, which are a few times lower in resolution than the original image. Thus the original input image of the complete building facade should have high resolution. The author's share of images in the database was taken by SONY DSC-W320 camera with 14.1 megapixel resolution.

There are building facade image databases publicly available to the scientific community, such as:

1. TSG-20: Tourist Sights Graz Image Database ²,
2. TSG-20: Tourist Sights Graz Image Database ³,
3. Zurich Building Image Database ⁴,

¹<http://www.flickr.com>

²<http://dib.joanneum.at/cape/TSG-20/>

³<http://dib.joanneum.at/cape/TSG-60/>

⁴<http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html>

However they are not suitable for the experimental evaluation of the presented work for the following reasons:

1) The number of buildings and the number of images in the image datasets is not sufficient to achieve the generalization addressed in the current work. The first image dataset comprises only 60 images of 20 buildings, the second database - 80 images of 20 buildings, whereas the third one contains 1005 images of 201 buildings.

2) Only a tiny fraction of the images, namely only 2 – 3 images in each of above mentioned datasets, contains the types of building facades, examined in the current work.

3) Again for the purpose of the desired sufficient generalization, the current work needs evaluation on image datasets, containing building facades, located in a vast geographical area, i.e. in numerous European countries. The first and second image datasets display buildings located in the Austrian city Graz, whereas the third one - in Zurich.

4) The classification modules of the architectural elements of the current project require high-resolution images as an input. All the three image databases observed do not meet this condition: in the first database the resolution of the images equals $768 * 1024$ pixels, in the second one - $240 * 320$ pixels and in the third one - $480 * 640$ pixels.

The software of the STYLE project is implemented by Matlab.

5.2 Experiments and Evaluation of Dome Segmentation

To test for robustness and evaluate the dome segmentation approach (Section 4.2, [39]) an image database of buildings featuring domes was created. The database exhibits Renaissance, Neo-Renaissance, Baroque, Neo-Baroque, Neoclassical and Islamic buildings. It includes 550 images of 77 buildings, among those the most famous world landmarks, like St. Peter's Basilica in Vatican, Florence Cathedral, St. Paul's Cathedral in London, Pantheon in Paris, United States Capitol in Washington, capitol buildings of 24 US states and Taj Mahal in India. The full list of the buildings, used in the experiments, can be found in Appendix A.1. The resolution of the images ranges from 108×82 to 3681×5522 pixels, proving the algorithm to be resolution-independent. The approach can handle both day and night images due to being color-independent. The only limitations of the method are:

1) segmentation of occluded domes is not supported,

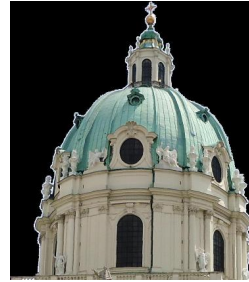
2) the rare cases when the dome background is formed by cityscape, not the sky, as a result of shooting from a level higher than the ground (building roofs, helicopter, etc) are also not handled.

The allowed tilt of the dome is (-90 to 90) degrees related to the vertical axis. This is not considered a limitation, since the search showed that building images taken upside down or tilted more than 90 degrees related to vertical axis are very rare.

The value by default for both parameters of graph-based segmentation algorithm [11] σ and k is 2000. The chosen big value is explained by the fact that the objective is a coarse segmentation of sky and non-sky segments. For images taken by night illumination, foggy weather condition or having low resolution the values of σ and k should be tuned down to obtain the non-sky segment with the precise dome edge. Whereas for images with strong cloud edges in the dome vicinity the values of σ and k should be tuned up to blur the cloud edges. Clouds



a) St. Charles's Church



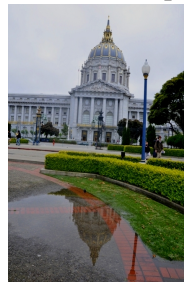
b) The dome of Figure 5.1a



c) California Capitol



d) The dome of Figure 5.1c



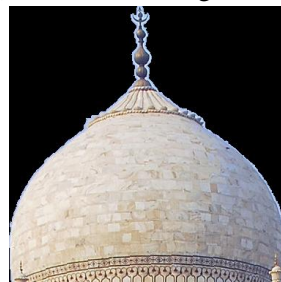
e) San-Francisco City Hall



f) The dome of Figure 5.1e



g) Taj Mahal



h) The dome of Figure 5.1g

Figure 5.1: Examples of buildings with domes and the segmentation of domes.

Table 5.1: Dome segmentation rate vs. symmetry axis magnitude.

	1st	2nd	3rd	4th	5th	Segmented domes	All
Seg. rate	504	11	7	5	1	528 (96% of All)	550
% of segmented	95.45%	2.08%	1.33%	0.95%	0.19%	100%	

not touching the dome do not affect the segmentation output, as they are discarded segments. For the current evaluation image database both σ and k are in the range from 50 to 16000.

The segmentation results are demonstrated on building examples in complex scenes - St. Charles's Church (Figure 5.1a), California Capitol (Figure 5.1b), San-Francisco City Hall (Figure 5.1c) and Taj Mahal (Figure 5.1d) and the respective segmented domes located next to each image in Figure 5.1. The accurately segmented dome was obtained on 528 out of the 550 tested images, which yields an average 96% rate for the approach (Table 5.1). On 504 out of 528 images, i.e. on 95.45% of the correctly segmented images, the segmentation was accomplished by looking for the dome in the vicinity of the 1st strongest symmetry axis (Table 5.1). And it is only on 4.55% of the correctly segmented images that the dome is segmented by trying symmetry axis weaker in magnitude than the 1st strongest symmetry axis, i.e. on 11 images the dome segment was found in the neighbourhood of the 2nd strongest symmetry axis, on 7 images - 3rd, on 5 images - 4th and on 1 image - 5th (Table 5.1).

The author also analyzed the reasons of unsuccessful segmentation for all 22 images. On 18 images the symmetry axis passing through the dome was not detected or was weakly detected, due to too high perspective distortions, other dominant symmetries present in the image vast panorama or building reflection in water. Images, where the dome top touches the image top fail, since the segmentation algorithm [11] fails to deliver 1 sky segment, but ends up with 2 sky segments on each side of the dome. This leads to attachment of one of sky segments to the foreground mask, further resulting in either delivering the dome segment with having one sky segment on the background or failing to obtain the dome due to not overcoming the roundness threshold. The current database happened to contain two such images. One image failed because of clouds touching the dome, the strong edges of which segmentation method [11] failed to ignore. As a result they appeared as a part of the foreground mask, leading to analysis of a false facade contour. And on one image finding the leap row between the dome and the main building was unsuccessful owing high perspective distortions.

5.3 Experiments and Evaluation of Tower Segmentation Approach

The portion of the image database, collected to evaluate the performance of the tower segmentation and classification modules (Section 4.3, Section 4.7), consists of images of cathedrals, churches and city halls. Since tower is a prominent architectural element, constructed as a display of power and might, its presence is logical among buildings of religious and secular great importance. The database includes 325 images of 35 buildings, among those famous world landmarks, like Notre Dame Cathedral in Paris, city halls of Vienna and Brussels. Original and revived Romanesque, Gothic and Baroque architectural styles are represented in the database.

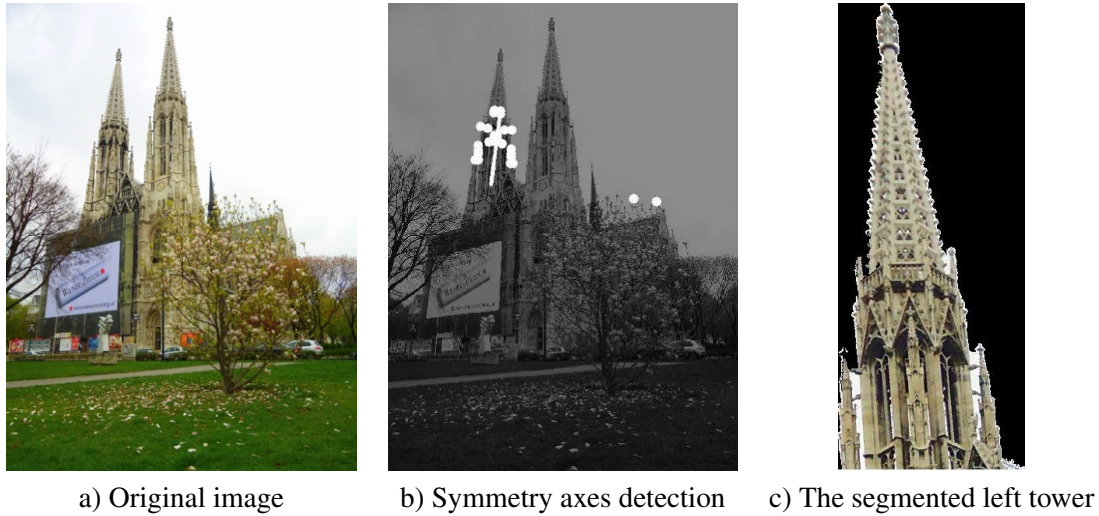


Figure 5.2: Segmentation of one of double towers

The complete list of the buildings, used in the experiments, can be found in Appendix A.2. The share of the author's tower image database is available for free for scientific research ⁵.

The tower segmentation method has the same advantages and limitations as the dome segmentation method (Section 5.2, [39]). The approach is capable to handle both day and night images due to being color-independent. It also achieves success on buildings in complex scenes and exposed to high perspective distortions. The only limitations of the method are:

- 1) the segmentation of occluded towers is not supported,
- 2) the rare cases when the tower background is formed by cityscape, not the sky, as a result of shooting from a level higher than the ground (building roofs, helicopter, etc) are also not handled.
- 3) the images, in which the tower(s) touch the top of the image, are also excluded, since on such images the background segmentation by [11] algorithm fails to deliver a single background (sky) segment.

The allowed tilt of the towers is (-90 to 90) degrees related to the vertical axis. This should not be considered a limitation, as the search showed that building images taken upside down or tilted more than 90 degrees related to the vertical axis rarely meet.

The values by default for both parameters of graph-based segmentation algorithm [11] σ and k are 1000. Here also, likewise the dome segmentation, the goal of obtaining coarse segmentation of sky and non-sky segments is the ground for the chosen big values. For images taken by night illumination, foggy weather condition or having low resolution the values of σ and k should be tuned down to obtain the non-sky segment with the precise tower edge. Whereas for images with strong cloud edges in the tower vicinity the values of σ and k should be tuned up to blur the cloud edges. Only the rare cases, when cloud-sky edges are stronger than cloud-tower edges fail. Here the cloud gets attached to the tower(s) in the foreground segment, leading to

⁵https://www.flickr.com/photos/lady_photographer/sets/72157636149550844/

Table 5.2: Tower segmentation rate.

	Image seg. rate (%)	Tower seg. rate (%)
1st symmetry axis	303 (93.23%)	539 (91.20%)
2nd symmetry axis	5 (1.54%)	5 (0.85%)
1 tower of 2	18 (5.54%)	18 (3.05%)
1 tower as if 1 of 2	6 (1.85%)	6 (1.02%)
Success	308 (94.77%)	544 (92.05%)
Fail	17 (5.23%)	47 (7.95%)
Total	325	591

failure in the final step of solidity thresholding. Clouds not touching the tower do not affect the segmentation output, as they are discarded segments. For the current evaluation image database the empirically found values of σ and k are in the range from 200 to 5000.

Interesting cases to study are images, where facades are exposed to high perspective distortions. Here the challenge is in the very first step of symmetry detection. The experiments revealed 2 curious ways of behaviour of symmetry detection, deciding the segmentation output:

1. On facade images, featuring double towers (Figure 5.2a), the strongest symmetry axis is located on one of the towers (Figure 5.2b), instead of passing through them. As a consequence the algorithm follows the single tower segmentation branch, ending up with the delivery of one of double towers (Figure 5.2c). Nevertheless, we assume the segmentation successful, since the single tower will pass to further classification.

2. On facade images, featuring single towers, the strongest symmetry axis is located on another vertical axis, than the one passing through the tower. This makes the segmentation algorithm follow the double tower branch. However the segmentation output is successful, since the objects that could be falsely segmented as the second tower fail to pass the solidity and bounding box size thresholds at the final step.

Table 5.2 summarizes the results of the tower segmentation approach performance. The second column in Table 5.2 presents the segmentation rate in terms of images. Here the segmentation is assumed successful in the case, when one of double towers is segmented (case 1. in the paragraph above), since the single tower will pass to further classification and voting. 308 out of 325 images achieved successful segmentation, indicating an average 94.77% segmentation rate. On 303 out of 308 successfully segmented images the position of tower(s) was found by the 1st strongest symmetry axis and only on 5 images - by the 2nd strongest symmetry axis. On 18 images one of the double towers was segmented. 6 images passed successful segmentation, though the single tower was segmented as if it were one of the double towers. A closer observation revealed that in 5 out of these 6 images was depicted Brussels city hall, which is not an absolutely symmetrical building and thus symmetry detection is more challenging in images exposed to high perspective distortions. The segmentation was unsuccessful only on 17 images (5.23%). 15 images owe the failure to the semmetry detector and the remaining 2 images failed, because the cloud-sky edges were stronger than the cloud-tower edges. The third column in Table 5.2 evaluates the segmentation rate in terms of towers. 47 out of 591 towers failed the segmentation,

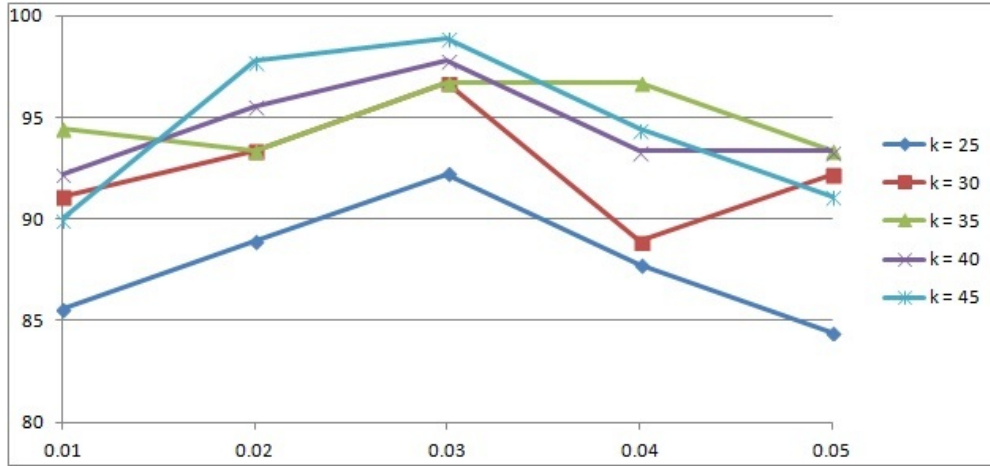


Figure 5.3: Window classification accuracy. Finding the best size of codebook (k) and SIFT peak threshold (p – horizontal axes).

18 of them being one of the towers on a double tower facade. The average segmentation rate in terms of towers yielded 92.05%.

5.4 Experiments and Results of Window Classification Approach

For testing and evaluation of the window classification approach (Section 4.4, [36]) a database of 400 images was assembled, 351 of which belong to the own and the rest - to Flickr image datasets. 90 images of the database make the training set (1/3 of each class). The resolution range of the images is from 138×93 to 4320×3240 pixels.

To evaluate the issue of the codebook size (vocabulary size) an experiment with different codebook sizes (k) is performed (Table 5.3 and Figure 5.3). The value of peak threshold for SIFT feature extraction and the value of k for k -means clustering algorithm are searched so that the final classification rate is maximised on the training set. As it is obvious from Figure 5.3, SIFT peak threshold values bigger than 0.03 decrease the classification rate. The reason for this is that

Table 5.3: Window classification accuracy on the training set with different codebook sizes.

Peak Threshold (p)	$k = 25$	$k = 30$	$k = 35$	$k = 40$	$k = 45$
0,01	85,56	91,11	94,44	92,22	90,00
0,02	88,89	93,33	93,33	95,56	97,78
0,03	92,22	96,67	96,67	97,78	98,89
0,04	87,78	88,89	96,67	93,33	94,44
0,05	84,44	92,22	93,33	93,33	91,11

Table 5.4: Confusion matrix and the accuracy rate in parenthesis.

	Gothic	Baroque	Romanesque	Sum
Gothic	100 (98.1%)	1	1	102
Baroque	3	111 (92.5%)	6	120
Romanesque	1	3	84 (95.4%)	88
Sum	104	115	91	310

the extraction of a smaller number of SIFT descriptors than that with peak threshold value equal to 0.03 decrease the number of extracted SIFT descriptors describing the dominating gradients characteristic for each window class. Whereas peak threshold values smaller than 0.03 tend to extract descriptors located on window reflections and background construction material textures, i.e. overfitting takes place. Figure 5.3 also shows that the best choice for k -means algorithm k parameter is in the range 25 – 45. The k parameter values smaller than 25 decrease the classification rate, as the number of cluster centers is not enough for the discrimination of visual words of different classes. Whereas values higher than 45 make the image histograms sparser, i.e. non-representative visual words are got. The final codebook choice for testing the approach is the one with $k = 30$ and peak threshold equal to 0.03. Running the classification with the mentioned codebook on a testing dataset of 310 images results in 15 false classified images, which yields an average classification rate of **95.16%**. A confusion matrix, with true positives, is given in Table 5.4. As the approach uses SIFT features for classification, it is rotation and scale invariant [23]. The experiments also prove the approach camera viewpoint invariant, as the classification of windows is accurate under high perspective distortions.

5.5 Experiments and Results of Classification of Tracerics, Pediments and Balustrades

For testing and evaluation of the module, classifying tracerics, pediments and balustrades (Section 4.5, [38]), a database of 520 images was created, consisting of images from own and Flickr datasets. 100 images of the database make the training set (50 for each class).

Here also a similar experiment, illustrated in Section 5.4 for window classification, is carried out to find the SIFT peak threshold (p), which allows to extract enough SIFT descriptors describing image dominant gradient directions. On the other hand, the number of the extracted SIFT descriptors should not be so big, as to include descriptors located on construction material textures, i.e. not to overfit. The optimal value found for SIFT peak threshold is equal to 0.03. Figure 5.4 and Table 5.3 show that the best choice for k -means algorithm k parameter is in the range 25 – 50. The k parameter values smaller than 25 decrease the classification rate, as the number of cluster centers is not enough for the discrimination of visual words of different classes. Whereas values higher than 50 make the image histograms sparser, i.e. one gets non-representative visual words. The parameters of the final codebook for testing the system are $k = 30$ and peak threshold equal to 0.03. Running the classification with the mentioned codebook on a testing dataset of 420 images results in 14 false classified images, which yields

Table 5.5: Classification accuracy on the training set with different codebook sizes.

(p)	$k = 25$	$k = 30$	$k = 35$	$k = 40$	$k = 45$	$k = 50$
0,02	94	95	95	98	98	97
0,03	95	97	97	97	98	99
0,04	93	98	95	97	96	97

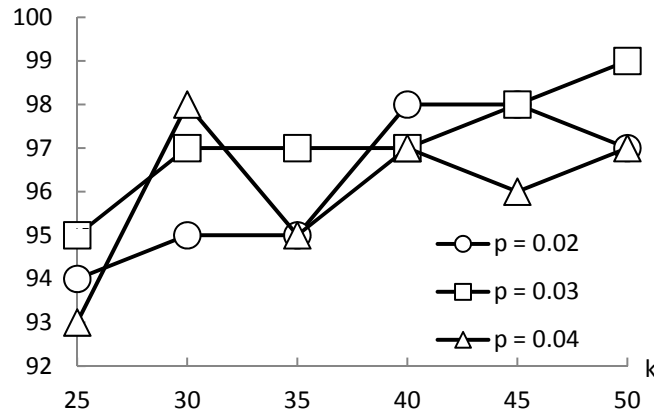
Table 5.6: Confusion matrix and the accuracy rate in parenthesis.

	Gothic	Baroque	Sum
Gothic	190 (97.44%)	5	195
Baroque	9	216 (96%)	225
Sum	199	221	420

an average classification rate of **96.67%**. A confusion matrix, with true positives, is given in the Table 5.6.

5.6 Experiments and Results of Dome Classification

To test and evaluate the module of dome classification (Section 4.6, [37]) a database of 520 images was created, also consisting of images from own and Flickr image datasets. 100 images of the database make the training dataset, annotated with labels of golden color and architectural styles. Golden labels are needed for identification of Cb and Cr ranges corresponding to golden color, whereas architectural style labels are used for codebook generation. 20 images out of 100 are labeled as golden. 30 images belong to Russian, 30 - to Islamic and 40 - to Renaissance styles. The resolution range of the images is from 127×191 to 3753×4314 pixels. The

**Figure 5.4:** Classification accuracy vs. k and p for finding the optimal k and p .

database includes the most famous domes of the world, like those of St. Peter's Basilica in Vatican, Florence Cathedral, St. Paul's Cathedral in London, Pantheon in Paris, United States capitol in Washington, capitol buildings of 24 US states, Taj Mahal in Agra, St. Basil's Cathedral in Moscow, Hagia Sophia in Istanbul.

In order to find the final Cb and Cr ranges corresponding to golden color, the author uses the fact that with the ascending of Cb values Cr values tend to decrease (Figure 4.11). Thus instead of having one Cb and Cr range pair, the final Cb range is divided into Cb subranges and corresponding Cr ranges, so that the number of golden true positives is maximized, while keeping the number of golden false positives minimum. The final Cb subranges and their corresponding Cr ranges are found experimentally and are as follows:

Cb (31-100]; Cr (134-202]	Cb (100-110]; Cr (131-142]
Cb (110-115]; Cr (126-137]	Cb (115-120]; Cr (124-130]

Golden color detection with the above mentioned subranges on the training dataset of 100 images results in 14 true positives out of 20 golden images and only 1 false positive.

To determine the codebook size (vocabulary size) best fitting to the database of domes here also an experiment with different codebook sizes (k) was performed (Figure 5.5 and Table 5.7). The value of peak threshold for SIFT feature extraction and the value of k for k -means clustering algorithm are searched so that the final classification rate is maximized on the training dataset.

Raising the SIFT peak threshold leads to the decrease of the extracted SIFT descriptors. As shown in Figure 5.5, experiments with SIFT peak threshold value 0.04 have lower classification rate, since the number of extracted SIFT descriptors, describing the dominating gradients of each dome class, is decreased. Extraction of a bigger number of SIFT descriptors than that with peak threshold value equal to 0.03 tends to extract descriptors located on construction material textures leading to overfitting. Figure 5.5 also shows that the best choice for k -means algorithm k parameter is in the range 25–45. k values smaller than 25 decrease the classification rate, as the number of cluster centers is not enough for the discrimination of visual words of different classes. Whereas values greater than 45 make the image histograms sparser, i.e. non-representative visual words take place. The final codebook choice for testing the system is the one generated with the values of $k = 40$ and peak threshold equal to 0.03.

At first the classification is performed using only the first and the third steps, i.e. without golden color detection. The result is 54 false classified images out of 420 testing images, which yields an average classification rate of 87.14%. The confusion matrix, with true positives, is given in Table 5.8. After the classification is run switching on the second step of the approach for Russian golden onion dome detection. To detect Russian golden domes the lower 1/3 of the

Table 5.7: Dome classification accuracy on the training set with different codebook sizes.

Peak Threshold (p)	$k = 25$	$k = 30$	$k = 35$	$k = 40$	$k = 45$
0,02	88	92	92	92	90
0,03	82	86	91	91	93
0,04	80	82	81	87	86

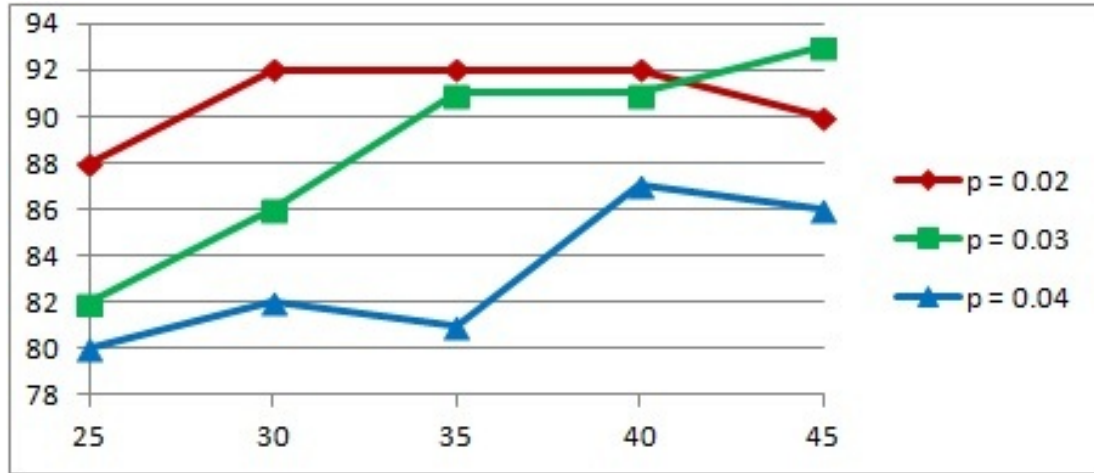


Figure 5.5: Dome classification accuracy. Finding the best size of codebook (k – horizontal axis) and SIFT peak threshold (p).

Table 5.8: Confusion matrix and the accuracy rate excluding the classification step 2.

	Russian	Islamic	Renaissance	Σ
Russian	84 (73.04%)	6	25	115
Islamic	7	115 (92.74%)	2	124
Renaissance	5	9	167 (92.27%)	181
Σ	96	130	194	420

Table 5.9: Confusion matrix and the accuracy rate including the classification step 2.

	Russian	Islamic	Renaissance	Σ
Russian	98 (85.22%)	4	13	115
Islamic	7	115 (92.74%)	2	124
Renaissance	6	9	166 (91.71%)	181
Σ	111	128	181	420

query image is cropped, since it is assumed that dome basement is on the bottom of the image (Figure 3.8a). The image patch to be analyzed is further trimmed from right, left and bottom by 1/8 size of the initial image. The cropping of the query image is done to avoid the segmentation of the sky and clouds and to exclude false positives among Renaissance domes, which have examples with golden decorations on the dome upper part. Recall that the segmentation of russian domes is not automatic, because the approach presented in Section 4.2 [36] targets the segmentation of hemispherical and islamic onion domes. The approach is not applicable to russian domes, since from one hand multiple russian domes have high probability of occluding one another, from another hand symmetry detection is not robust to perspective distortions due

to small dome sizes. The query image is considered golden, if more than 70% of pixels of the analyzed patch passes the golden condition, i.e. the pixel Cb and Cr values fall within the defined range of golden color in Cb and Cr channels. The confusion matrix in Table 5.9 shows that the classification rate of Russian domes is raised by 14 true positives, while resulting in 1 false positive. 60 out of 80 golden domes were detected, which yields 75% rate for the golden color detection module and raises the rate of classification for Russian class from 73.04% to 85.22%. Golden color detection raises the final classification rate of the whole testing database from 87.14% to **90.24%**.

5.7 The Evaluation and Results of Tower Classification

Running tower segmentation algorithm (Section 4.3) on 325 input images delivered 544 segmented towers (Section 5.3). As a training set for codebook generation are used 102 of the segmented towers and the testing set is formed from the remaining 442 images. Both training and testing image datasets are labelled by architectural styles, the training dataset – for building the visual codebook, the testing set – as ground truth for the evaluation of the classification rate. An experiment, exercising the values of k for k -means clustering algorithm and SIFT peak threshold (p), is carried out to choose the codebook best fitting the tower image database. Table 5.10 and Figure 5.6a display the results of the experiment, pointing how NN classifier classification rate on the training set depends on k and p . On one hand the value of k should be big enough, so that the number of clusters is sufficient to discriminate the visual words of the 3 classes. On the other hand raising the value of k too much tends to make image histograms sparse, i.e. non-representative visual words take place. The experiment shows that the optimal value of k lies in the range 25 – 45 (Table 5.10 and Figure 5.6a). Now coming to the analysis of the p parameter: the bigger p value, the less SIFT descriptors are extracted. So while choosing the optimal value of p it should be kept in mind, that a value too big will not extract enough discriminative descriptors, whereas a value too small will overdo by extracting descriptors located on tower construction material textures. The sought value of p is in the range 0.01 – 0.03 (Table 5.10 and Figure 5.6a). The chosen values to generate the final codebook are $k=40$ and $p=0.02$. Here the preference not in the favor of the pair $k=45$ and $p=0.03$, yielding the highest classification rate, is justified by the fact, that the codebook best performing on the training set may be not general enough to succeed the classification on the testing set.

The goal of the second experiment is to find out, whether the classification precision on the training set may increase with application of $K>1$ values on KNN classifier. $K=2$ and $K=3$ values will apparently repeat the result of $K=1$, since according to Equation 4.7 in Section 4.7

Table 5.10: Tower classification accuracy on the training set with different codebook sizes.

Peak Threshold (p)	$k = 25$	$k = 30$	$k = 35$	$k = 40$	$k = 45$
0,01	88,24	92,16	85,29	82,35	88,24
0,02	86,27	84,31	88,24	91,18	88,24
0,03	88,24	82,35	88,24	92,16	94,12

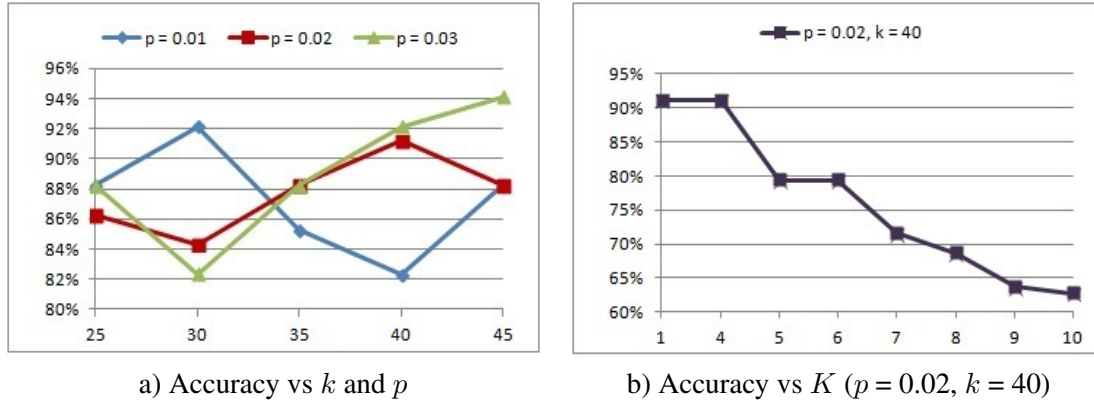


Figure 5.6: Classification accuracy on the training set. Finding the best codebook size (k), SIFT peak threshold (p) and KNN classifier K .

Table 5.11: Tower classification. Confusion matrix and the accuracy rate.

	Baroque	Gothic	Romanesque	Sum
Baroque	84 (73.0%)	24	7	115
Gothic	8	171 (87.69%)	16	195
Romanesque	0	31	101 (76.52%)	132
Sum	92	226	124	442

$w_1=1$ and $w_K=0$. Figure 5.6b shows that whereas $K=4$ value provides the same classification rate as $K=1$, with further raise of K the classification rate descends. Thus the classification on the testing database will be performed by NN classifier ($K=1$), like in (Section 5.4) [36].

Running the classification with the mentioned codebook on the testing dataset of 442 images results in 86 false classified images, which yields an average classification rate of **80.54%**. A confusion matrix, with true positives, is given in the Table 5.11. As the approach uses SIFT features for classification, it is rotation and scale invariant [23].

5.8 Summary

The chapter presented the experimental setup, evaluation and results of the STYLE project methodology. The first challenge, that is the absence of image datasets labeled by architectural styles, was overcome by creation and ground truth labeling of an image database, consisting of 2315 images belonging to the author and Flickr users. The image dataset includes buildings geographically widely spread, proving that the method proposed is general (Section 5.1).

The segmentation approaches have reported extremely high segmentation rates, that is 96% in the case of the dome segmentation (Section 5.2) and 94.77% for the tower segmentation (Section 5.3). The window classification approach (Section 5.4) reached 95.16% average clas-

sification precision, whereas the classification of traceries, pediments and balustrades - 96.67% (Section 5.5). The approaches, classifying the architectural elements dome (Section 5.6) and tower (Section 5.7), reported 90.24% and 80.54% average classification rates respectively.

Critical Reflection

6.1 Comparison with Related Work

As remarked before, to the best knowledge of the author the only image-based building facade architectural style classification approach so far is introduced by [27]. The approach is clarified in detail in Chapter 2. While comparing the architectural style classification approach represented in the current work to the one, proposed by [27], the following attributes and differences should be pointed out:

- The current algorithm encapsulates the definition of an architectural style as a combination of style typical elements and is thus general enough to target the classification of any distinguishable style.
- Due to its generality the current method possesses the capability to be expanded with new architectural styles by addition of typical elements of the new styles.
- The current method is not assumed to be used only in facade reconstruction context, like in [27], and is also able to classify partially occluded facades or facade parts by style typical elements.
- The algorithm presented in this work and in [27] classify different architectural styles, Romanesque, Baroque, Gothic and Flemish Renaissance, Haussmannian and Neoclassical respectively.
- By the problem statement the current system receives images of individual facades as input, whereas in [27] mobile mapping images make the image database. Thus the scene classification and facade splitting tasks, making the first and third steps of [27] algorithm, are out of the scope of the current work.
- In the second step authors in [27] perform facade rectification, assuming the facades to be planar surfaces, containing two dominant perpendicular directions. Whereas facade

rectification is a good idea for the type of buildings considered in [27], it is not applicable for the image database of the current method, since most of the buildings do not have planar surfaces. Vivid examples of such are Baroque facades with dramatic 3D sculptures or hemispherical domes and Gothic facades with emphasized vertical accents instead of two dominant perpendicular directions.

- In [27] NBNN classifier is used in the classification step, whereas the current approach uses the NN classifier.
- Both approaches use SIFT descriptors as features in the classification stage, though in one of the modules of the current work, that is in the approach of classification of domes [37], additionally dome dimensions and color are used.
- Besides classification approaches published in [36], [38], [37], the current work is the first to propose segmentation approaches of prominent architectural elements dome [39] and tower.

6.2 Discussion of the Achievements and Open Issues

Development of a computer vision tool for architectural style classification of building facades is a broad problem. Its first milestone is the invention of a general algorithm, solving the problem, which was successfully achieved by introducing the algorithm of the voting of architectural elements. The algorithm was implemented for three influential pan-European architectural styles - Romanesque, Gothic and Baroque. The classified architectural elements were window, dome, tower, tracery, pediment and balustrade. Surely the list of the typical architectural elements of Romanesque, Gothic and Baroque styles is not completed by the elements mentioned, but the proposed algorithm left free room for the future expansion of the project with new architectural elements and architectural styles.

Whereas the ultimate goal is the achievement of complete automation, the developed tool is semiautomatic, requiring minimum user interaction for the indication of the presence of architectural elements on the query facade and the manual segmentation of the architectural elements besides domes and towers. The segmentation of domes and towers was possible to automate by employing their common visual features - bilateral symmetry and raising out of the building. The latter visual property allowed whereabouts detection and segmentation of the dome/tower background - the sky. Though windows, traceries, pediments and balustrades also possess bilateral symmetry, the detection of their whereabouts on facade and background segmentation remain open issues. The segmentation automation of the windows in the context of the current project is a broad problem of its own, since it leads to the segmentation of windows of multiple shapes together with their visually intricate ornamentation. The complexity of the issue becomes obvious by recalling the window types classified within the project - Romanesque single, double, triple round arch windows, Gothic pointed arch and rose windows, Baroque windows with triangular, segmental pediments and balustrades. Though the approaches in [1,4,17,18,20,33,34,42] address the detection of windows, the experimental datasets display facades with simple rectangular windows, which are visually very diverse from the ones listed above and featuring intricate

ornaments. If pursuing the aim of the automation of window segmentation in the future, one should start with performing experiments to see how the available window detection algorithms perform on the visually complex window types mentioned.

The work also gave the clue for widening the project with new architectural styles by the following steps, identical to those performed for Romanesque, Gothic and Baroque styles so far:

1. Research of the new architectural style to find out its typical architectural elements.
2. Segmentation of the chosen typical architectural elements (manual or automatic).
3. Classification of the segmented architectural elements by an approach employing the visual features of the elements.
4. Integration of the votes of the new architectural elements and the new architectural style in the style voting mechanism, also by taking into account if the newly added architectural style could possibly be mixed with the existing architectural styles on a single facade.

Summary and Future Work

The current work presented the first general methodology, addressing the problem of image-based architectural style classification of building facades. On the basis of the method lies the concept of discriminating architectural styles by style typical component parts - architectural elements. Thus the method imitates the human cognitive process of architectural style classification. As the architectural style of the building is claimed the style, to which the majority of architectural elements belong. The proposed algorithm is general enough to cover the classification of any architectural style with the precondition that the style to be classified has clearly defined visual characteristics. It is also advantaged by bearing the capability to classify facade portions, as well as facades, which are partially occluded or display a mixture of architectural styles. The algorithm was realized for three influential pan-European architectural styles – Romanesque, Gothic and Baroque.

The methodology was constructed as successive phases of segmentation, classification and voting of style typical architectural elements. At the segmentation phase two novel approaches, targeting the segmentation of outstanding architectural elements dome and tower respectively, were introduced. The approach for segmentation of domes was a pipeline uniting bilateral symmetry detection, graph-based segmentation approaches and image analysis and processing techniques. The approach of tower segmentation modified and advanced the approach of dome segmentation to two branches, targeting the tower segmentation of facades featuring single and double towers individually. The first approach, introduced at the classification stage, was aiming at the classification of windows. It was based on clustering and learning of local features. The next classification module categorized the architectural elements tracery, pediment and balustrade. It employed the approach of window classification, by tuning its parameters to suit best to the image database of the examined elements. The algorithm, classifying the prominent element dome, was an extension of the window classification approach. It comprised three steps, detecting Islamic saucer domes at the first step by the proportion of the dome width and height, identifying Russian golden domes at the second step by golden color detection in YCbCr color space and performing classification by shapes only at the third step. The list of classification modules finalized the tower classification approach, also originating from window classification approach.

The third and final step of the architectural style classification system was the integration of all segmentation and classification approaches into a single frame and the implementation of the voting mechanism of architectural elements.

To test each module of the system image databases displaying buildings of the explored architectural styles and featuring the corresponding architectural elements were created. Extensive experiments, carried out to evaluate each module separately, reported high segmentation and classification precision. The list of the typical architectural elements of Romanesque, Gothic and Baroque architectural styles is not completed by the elements examined so far. But the proposed algorithm left free room for the future expansion of the project with new elements. By increasing the number of architectural elements, the system will be able to classify a broader variety of facades.

Future work will be in the direction of extending the system with new architectural elements and styles. In addition to that future research will pursue the aim of automating the segmentation of all architectural elements.

Databases, source code

A.1 The List of the Buildings Featuring Domes

The database of buildings collected for the evaluation of the dome segmentation approach [39] has a wide geography, including buildings situated in Europe, North and South America and Asia. Since dome is an outstanding architectural element, it is featured on buildings of religious and secular great importance. The database comprises the author's own photographs and images from Flickr image database. Below are listed the buildings of the database, among those the capitols of 24 US states:

1. Arkansas Capitol, USA
2. California Capitol, Sacramento, USA
3. Colorado Capitol, USA
4. Idaho Capitol, USA
5. Illinois Capitol, USA
6. Indiana Capitol, USA
7. Iowa Capitol, USA
8. Georgia Capitol, USA
9. Kansas Capitol, USA
10. Kentucky Capitol, USA
11. Maine Capitol, USA
12. Michigan Capitol, USA

13. Minnesota Capitol, USA
14. Missouri Capitol, USA
15. New Hampshire Capitol, USA
16. Oklahoma Capitol, USA
17. Pennsylvania Capitol, USA
18. Rhode Island Capitol, USA
19. Texas Capitol, USA
20. US Capitol, USA
21. Utah Capitol, USA
22. Washington Capitol, USA
23. West Virginia Capitol, USA
24. Wisconsin Capitol, USA
25. Wyoming Capitol, USA
26. San-Francisco City Hall, USA
27. Pasadena City Hall, USA
28. Baltimore City Hall, USA
29. St. Paul Cathedral, Minnesota, USA
30. St Charles's Church Vienna, Austria
31. Hofburg Palace, Vienna, Austria
32. Art History Museum, Vienna, Austria
33. Maria von Siege, Vienna, Austria
34. Vienna central cemetery church, Austria
35. Salzburg Cathedral, Austria
36. Berlin Cathedral, Germany
37. Charlottenburg Palace, Berlin, Germany
38. St. Peter's Cathedral, Vatican
39. Florence Cathedral, Italy

40. Basilica di Superga, Turin, Italy
41. Blue Mosque, Yerevan, Armenia
42. St. Stefan's Cathedral, Budapest, Hungary
43. Budapest Parliament, Hungary
44. Budapest Royal Palace, Hungary
45. Marble church, Denmark
46. Dublin Custom House, Ireland
47. Helsinki Cathedral, Finland
48. Dome Church, Paris, France
49. Paris Pantheon, France
50. Les Invalides church, Paris, France
51. Old College Edinburgh, Scotland
52. St. Georges Church Edinburgh, Scotland
53. The Bank Of Scotland
54. Mitchell Library, Glasgow, Scotland
55. St. Isaac Cathedral, Saint Petersburg, Russia
56. St. Nicolas church, Prague, Czech Republic
57. St. Paul Cathedral, London, England
58. Argentina Congress Building, Buenos Aires
59. Capitolio Havana, Cuba
60. Taj Mahal, India
61. Sheikh Lotfollah Mosque, Isfahan, Iran
62. Imam Mosque, Isfahan, Iran
63. Bibi Khanoum Mosque, Samarkand, Uzbekistan
64. Khast Imam Mosque, Uzbekistan
65. Kalon Mosque, Bukhara, Uzbekistan
66. A church in Athens, Greece (The name not annotated with the image)
67. Some other mosques in Iran, etc (The names not annotated with the images)

A.2 The List of the Buildings Featuring Towers

The section of the image database, gathered to evaluate the performance of the tower segmentation and classification approaches, includes images of 35 cathedrals, churches, basilicas and city halls spread in Austria, Germany, Sweden, Czech Republic, Hungary, France, Spain, Luxembourg, England, Belgium, Switzerland and China. This database was also gathered from the author's own photographs and images from Flickr image database. The buildings are listed below:

1. Vienna City Hall, Austria
2. Votiv church, Vienna, Austria
3. Maria Treu church, Vienna, Austria
4. Jesuit Church, Vienna, Austria
5. Mariahilf church, Vienna, Austria
6. Maria of Siege church, Vienna, Austria
7. St. Anton church, Vienna, Austria
8. Breitenfelder church, Vienna, Austria
9. Franz of Assisi church, Vienna, Austria
10. Altlerchenfelder church, Vienna, Austria
11. Salzburg Cathedral, Austria
12. Cologne Cathedral, Germany
13. St. Pantaleon Church, Cologne, Germany
14. Bremen Cathedral, Germany
15. Fulda Cathedral, Germany
16. Basilica St. Castor, Koblenz, Germany
17. St. Gall Abbey, Switzerland
18. Notre Dame Cathedral, Paris, France
19. Reims Cathedral, France
20. Abbaye Aux Hommes Caen, France
21. York Minster, England

22. Westminster Abbey, London, England
23. Barcelona Cathedral, Spain
24. Burgos Cathedral, Spain
25. St. Michael and St. Gudula Cathedral, Brussels, Belgium
26. Brussels City Hall, Belgium
27. Church of Our Lady of Laeken, Brussels, Belgium
28. St. Petrus And St. Paulus Church, Ostend, Belgium
29. Loreta church, Prague, Czech Republic
30. St. Peter and St. Paul Church Vysehrad, Prague, Czech Republic
31. St. Mary Magdalene church, Karlovy Vary, Czech Republic
32. St. Stefan's Cathedral, Budapest, Hungary
33. Church Saints Cosmas and Damian, Luxemburg
34. Lund Cathedral, Sweden
35. St. Michael's Cathedral, Qingdao, China



Figure A.1: The logo of the STYLE project.

A.3 The Logo of the STYLE Project

The logo of the STYLE project (Figure A.1) depicts one of the monumental buildings of the famous contemporary architect Santiago Calatrava - Reina Sofia Palace of the Arts, located in Valencia, Spain. The choice of the building conveys the idea, that contemporary buildings are unique examples, carrying the style signature of the architect and, unlike historic buildings, are not possible to categorize into certain classes of architectural styles.

A.4 Source Code

The Function Performing the Voting of Architectural Elements

The function, carrying out the main idea of the work, that is the architectural style classification of the building facade by the voting of architectural elements, is the backbone of the presented software tool. The function `pushbuttonOK_Callback` is executed on pressing the “Classify” button of the GUI shown in Figure A.2, after the user marks the architectural elements present on the observed facade. The function first checks for the presence of the architectural elements and after proceeds to the processing of the existing ones. Here is also, where the modular structure of the software lies, i.e. the segmentation/classification of each architectural elements is implemented in a separate module, which makes the software easy to extend by new architectural elements. The extension by new architectural styles may be likewise accomplished by adding new styles to the `voteVector` vector. After the segmentation/classification of all the architectural elements, voting of the architectural elements is achieved by the algorithm, explained in detail in Section 4.8. The code of the `pushbuttonOK_Callback` is presented below.

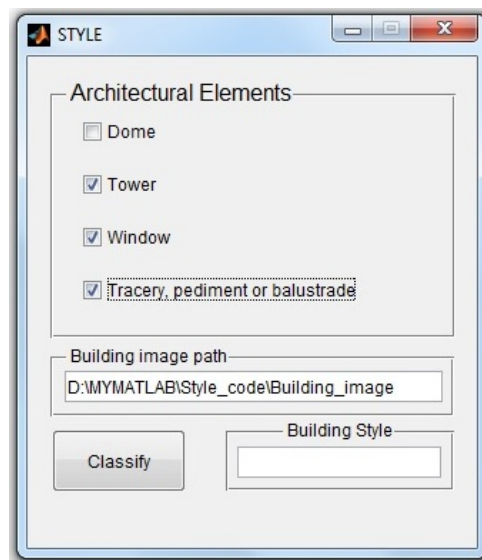


Figure A.2: GUI demonstration of STYLE project.

```

% --- Executes on button press in pushbuttonOK.
function pushbuttonOK_Callback(hObject, eventdata, handles)
% hObject      handle to pushbuttonOK (see GCBO)
% handles      structure with handles and user data (see GUIDATA)

global voteVector
global imagePath
voteVector = struct( 'Romanesque', {0}, 'Gothic', {0}, ...
    'Baroque', {0}, 'Russian', {0}, 'Islamic', {0} );

isTower = get(handles.checkboxTower, 'Value');
isDome = get(handles.checkboxDome, 'Value');
isWindow = get(handles.checkboxWindow, 'Value');
isElement = get(handles.checkboxElement, 'Value');
imagePath = get(handles.path, 'String');

if (isTower)
    SegmentTowers();
    ClassifyTowers();
end

if (isDome)
    SegmentDome();
    ClassifyDome();
end

if (isWindow)
    ClassifyWindows();
end

if (isElement)
    ClassifyElements();
end

voteVector

% THE VOTING OF ARCHITECTURAL STYLES !!!

% Romanesque, Gothic and Baroque votes are equal or
% Romanesque and Gothic votes are >= than Baroque vote

```

```

if ( voteVector.Romanesque == voteVector.Gothic )
    if ( voteVector.Romanesque >= voteVector.Baroque )
        fprintf('THE FACADE STYLE IS Romanesque and Gothic mix!');
        set(handles.arcStyle, 'String', 'Romanesque and Gothic');
        return;
    end
end

maxVote = 0;
styles = fieldnames(voteVector);
for i = 1:length(fieldnames(voteVector))
    currentVote = voteVector.(char(styles(i)));
    if (currentVote > maxVote)
        maxVote = currentVote;
        winnerIndex = i;
    end
end

% Gothic(Romanesque) has the maximum vote and the votes of
% Romanesque(Gothic) >= 2
if ( (strcmp(char(styles(winnerIndex)), 'Romanesque') && ...
voteVector.Gothic >= 2) || ...
(strcmp(char(styles(winnerIndex)), 'Gothic') && ...
voteVector.Romanesque >= 2) )
    fprintf('THE FACADE STYLE IS Romanesque and Gothic mix!');
    set(handles.arcStyle, 'String', 'Romanesque and Gothic');
    return;
end

set(handles.arcStyle, 'String', char(styles(winnerIndex)));
fprintf('THE FACADE STYLE IS %s !', char(styles(winnerIndex)));

```

The Generation and Saving of SIFT descriptors from Training Datasets

SIFT descriptors are extracted from the training dataset of each classification module, namely the modules for classification of windows (Section 4.4), traceries, pediments, balustrades (Section 4.5), domes (Section 4.6) and towers (Section 4.7). Multiple sets of SIFT descriptors are generated and saved by exercising the peak threshold parameter. The saved SIFT descriptor sets will be loaded in the next step with the purpose of generating the corresponding codebooks. For each SIFT descriptor set, corresponding to one value of the peak threshold, multiple codebooks will be generated by exercising the k parameter of the k-means algorithm further. The optimal codebook, obtained as a result of tuning the parameters peak threshold and k, will be chosen for style prediction on the testing dataset. The source code below extracts and saves the SIFT

descriptor set for the tower training dataset with the peak threshold value set to 0.02. The extraction and saving of the SIFT descriptors for any other classification module is done easily by changing the vector of the architectural styles to be classified, the training dataset directory path and the value of SIFT peak threshold.

```
styleVector = struct('style',{ 'Baroque','Gothic','Romanesque'});

fprintf('\n GENERATION OF TOWER SIFT DESCRIPTORS ! \n');

% Training set directory
fileDir = fullfile('D:\MYMATLAB\Tower_code', 'TOWER_TRAINING_SET');
files = dir(fileDir);

for i = 1:size(files,1)
    filename = files(i).name;

    if (files(i).bytes < 1 || files(i).isdir)
        continue;
    end

    % To skip non-image files
    [filenameraw ext] = delimgext(filename);
    if (isempty(ext))
        continue;
    end

    info = imfinfo([fileDir,'/',filename]);
    disp(['Loading Image: ', filename]);
    imgRGB = imread([fileDir,'/',filename], info.Format);

    % Labeling the training set images from their filenames
    for ind = 1:length(styleVector)
        if (strfind(filename, styleVector(ind).style))
            label = ind;
        end
    end

    imgGray = myRgb2Gray(imgRGB);

    % Peak threshold parameter value
    peakTh = 0.02;
```

```

[ftmp,dtmp] = vl_sift(imgGray, 'PeakThresh', peakTh);

if exist('f','var')
    f = [f'; ftmp']';
    d = [d'; dtmp']';
    labels(end:end+size(ftmp,2)) = label;
else
    f = ftmp;
    d = dtmp;
    labels(1:size(ftmp,2)) = label;
end

end

% Saving SIFT frames, descriptors and labels for peakTh value
siftFileName = sprintf('TowerSifts_peakTh%f.mat', peakTh);
save(siftFileName, 'f', 'd', 'labels');

```

The Generation of a Codebook from SIFT descriptors

Visual codebook generation has a principal role in all classification approaches of the current work. Its aim is to obtain via k-means clustering a visual codebook from the SIFT descriptors, which have been extracted from the training image dataset of the observed architectural element(s) (Section A.4). A codebook of the corresponding element(s) is generated for each classification module. The source code, realizing the codebook generation with the parameters adjusted for the optimal tower codebook, follows below.

```

fprintf('\n\n\n CODEBOOK GENERATION START ! \n\n');
numberOfStyles = 3;
peakTh = 0.02;

% Loading SIFT frames, descriptors and labels
siftFileName = sprintf('TowerSifts_peakTh%f.mat', peakTh);
sifts = load(siftFileName);
d = sifts.d;
labels = sifts.labels;

% k-means clustering
k = 40;

[m1 p1] = size(d);
[m2 p2] = size(labels);

```

```

C = [];
% Finding dictionary words - codebook - cluster centers
for lIndex = 1:numberOfStyles
    % To use m-binary labels for m-category labels
    %(e.g. 3 labels - 100, 010, 001)
    labelsBinary = (labels == lIndex);
    indFirst = find(labelsBinary, 1, 'first');
    indLast = find(labelsBinary, 1, 'last');

    currentClassDescriptors = d(:, indFirst:indLast);
    currentClassDescriptorsScaled=ScaleMatrix(currentClassDescriptors);

    [Ctmp, A] = vl_kmeans(currentClassDescriptorsScaled', k);
    C = [C; Ctmp'];
end

% Saving the codebook
save('Codebook_Tower.mat', 'C');

```


Bibliography

- [1] Haider Ali, Christin Seifert, Nitin Jindal, Lucas Paletta, and Gerhard Paar. Window detection in facades. In *Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP 2007)*, pages 837–842, Modena, Italy, 2007. Springer-Verlag.
- [2] Jorge Alberto Marcial Basilio, Gualberto Aguilar Torres, Gabriel Sanchez Perez, L. Karina Toscano Medina, and Hèctor M. Perez Meana. Explicit image detection using YCbCr space color model as skin detection. In *Proceedings of the 2011 American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications*, pages 123–128, 2011.
- [3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.
- [4] Alexander C. Berg, Floraine Grabler, and Jitendra Malik. Parsing images of architectural scenes. In *Proceedings of the 11th International Conference on Computer Vision (ICCV '07)*, pages 1–8, Rio de Janeiro, Brazil, 2007. IEEE.
- [5] Martin Shaw Briggs. *Baroque Architecture*. University of California Libraries, 2 edition, 1913.
- [6] Peter Collins. *Changing Ideals in Modern Architecture, 1750-1950*. McGill-Queen’s University Press, 1998.
- [7] Nico Cornelis, Bastian Leibe, Kurt Cornelis, and Luc Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 78:121–141, 2008.
- [8] James L. Crowley and Alice C. Parker. A representation for shape based on peaks and ridges in the difference of lowpass transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(2):156–170, 1984.
- [9] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In *Proceedings of International Workshop on Statistical Learning in Computer Vision (ECCV 2004)*, pages 1–22, Prague, Czech Republic, 2004.

- [10] S. A. Dudani. The distance-weighted k-nearest neighbor rule. *IEEE Transactions on System, Man, and Cybernetics*, SMC-6:325–327, 1976.
- [11] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.
- [12] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [13] Banister Fletcher. *A history of architecture on the comparative method*. Scribner’s, Batsford, London, 5 edition, 1920.
- [14] Laurent Guigues, Le Men Herve, and Jean-Pierre Cocquerez. The hierarchy of the cocoons of a graph and its application to image segmentation. *PRL*, 24(8):1059–1066, 2003.
- [15] Robert M. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, 67:786–804, 1979.
- [16] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The 4th Alvey Vision Conference*, pages 147–151, 1998.
- [17] J.-E. Haugeard, S. Philipp-Foliguet, and F. Precioso. Windows and facade retrieval using similarity on graphs of contours. In *IEEE International Conference on Image Processing (ICIP 09)*, pages 269–272, Cairo, Egypt, 2009. IEEE.
- [18] Jorge Hernandez and Beatriz Marcotegui. Morphological segmentation of building facade images. In *Proceedings of the 16th International Conference on Image Processing (ICIP 2009)*, pages 3977–3980, Cairo, Egypt, 2009. IEEE.
- [19] Sung Chun Lee, Soon Ki Jung, and Ramakant Nevatia. Automatic integration of facade textures into 3d building models with a projective geometry based line clustering. *Computer Graphics Forum*, 21(3):511–519, 2002.
- [20] Sung Chun Lee and Ramakant Nevatia. Extraction and integration of window in a 3d building model from ground view image. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pages 113–120, Washington, DC, USA, 2004. IEEE.
- [21] Yunpeng Li, D.J. Crandall, and D.P.; Huttenlocher. Landmark classification in large-scale image collections. In *Proceedings of 12th International Conference on Computer Vision (ICCV 2009)*, pages 1957–1964, Kyoto, Japan, 2009. IEEE.
- [22] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1998)*, pages 482–488. IEEE, 1998.

- [23] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [24] G. Loy and J. Eklundh. Detecting symmetry and symmetric constellations of features. In *Proceedings of 9th European Conference on Computer Vision (ECCV 2006)*, volume 3952 of *LNCS*, pages 508–521, Graz, Austria, 2006. Springer-Verlag.
- [25] Ilias Maglogiannis, Demosthenes Vouyioukas, and Chris Aggelopoulos. Face detection and recognition of natural human emotion using markov random fields. *Personal and Ubiquitous Computing*, 13(1):95–101, 2009.
- [26] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the 13th British Machine Vision Conference (BMVC 2002)*, pages 384–393, Cardiff, UK, 2002. BMVA Press.
- [27] M. Mathias, A. Martinovic, J. Weissenberg, S. Haegler, and L. Van Gool. Automatic architectural style recognition. In *Proceedings of the 4th International Workshop on 3D Virtual Reconstruction and Visualization of Complex Architectures*, International Society for Photogrammetry and Remote Sensing, Trento, Italy, 2011.
- [28] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference in Computer Vision (ICCV 2001)*, pages 525–531, Vancouver, Canada, 2001. IEEE.
- [29] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proceedings of VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.
- [30] M. Irani O. Boiman, E. Shechtman. In defense of nearest-neighbor based image classification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8. IEEE, 2008.
- [31] T. Ojala, M. Pietikainen, and T. Mäenpää. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.
- [32] Aude Oliva and Antonio Torralba. Building the gist of a scene: the role of global image features in recognition. In *Progress in Brain Research*, pages 23–36, 2006.
- [33] Michal Recky and Franz Leberl. Window detection in complex facades. In *Proceedings of the 2nd European Workshop on Visual Information Processing (EUVIP 2010)*, pages 220 – 225, Paris, France, 2010. IEEE.
- [34] Michal Recky and Franz Leberl. Windows detection using k-means in cie-lab color space. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR 2010)*, pages 356–359, Istanbul, Turkey, 2010.
- [35] Albert Rosengarten. *A handbook of architectural styles*. Chatto & Windus, London, 1912.

- [36] Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig. Architectural style classification of building facade windows. In *Proceedings of the 7th International Symposium on Visual Computing (ISVC '11)*, volume 6939 of *LNCS*, pages 280–289, Las Vegas, Nevada, USA, 2011. Springer-Verlag.
- [37] Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig. Architectural style classification of domes. In *Proceedings of the 8th International Symposium on Visual Computing (ISVC '12)*, volume 7432(2) of *LNCS*, pages 420–429, Rethymnon, Crete, Greece, 2012. Springer-Verlag.
- [38] Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig. Classification of gothic and baroque architectural elements. In *Proceedings of the 19th International Conference on Systems, Signals and Image Processing (IWSSIP '12)*, pages 330–333, Vienna, Austria, 2012. IEEE.
- [39] Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig. Segmentaion of building facade domes. In *Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP '12)*, volume 7441 of *LNCS*, pages 324–331, Buenos Aires, Argentina, 2012. Springer-Verlag.
- [40] C. Siagian and L. Itti. Comparison of gist models in rapid scene categorization tasks. In *Proceedings of the Vision Science Society Annual Meeting (VSS08)*, 2008.
- [41] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transaction on Graphics*, 25:835–846, 2006.
- [42] Olivier Teboul, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Segmentation of building facades using procedural shape priors. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pages 3105–3112, San Francisco, California, USA, 2010.
- [43] T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference (BMVC 2000)*, pages 412–425, Bristol, UK, 2000. BMVA Press.
- [44] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pages 511–518, Kauai, Hawaii, USA, 2001. IEEE Computer Society.
- [45] S. Wenzel, M. Drauschke, and W. Föstner. Detection of repeated structures in facade images. *Pattern Recognition and Image Analysis*, 18(3):406–411, 2008.
- [46] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.
- [47] Wei Zhang and Jana Kosecka. Hierarchical building recognition. *Image and Vision Computing*, 25(5):704–716, 2004.

- [48] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: building a web-scale landmark recognition engine. In *Proceedings of the 20th International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1085–1092, Miami, Florida, USA, 2009. IEEE.

GAYANE SHALUNTS



E-mail: shal@caa.tuwien.ac.at
Address: Joergerstrasse 24/13, 1170 Vienna, Austria
Mobile phone: (+43) 699 14034917

BIOGRAPHICAL INFORMATION

- **Date of Birth:** 9 November 1980
- **Marital Status:** Single

EDUCATION

Vienna University of Technology (www.tuwien.ac.at)

- **2010 February – Till today**

Doctoral College on Computational Perception

Institute of Computer Aided Automation (<http://www.caa.tuwien.ac.at/>)

Doctorate Degree Program

Dissertation topic: Architectural Style Classification of Building Facades

Advisors: Robert Sablatnig, Markus Vincze

The State Engineering University of Armenia (www.seua.am)

- **2001/2002 – 2002/2003**

Field: Informatics and Computer Systems

Master Degree Program (with honours)

- **1997/98 – 2000/2001**

Field: Software for Computing Technics and Automated Systems

Bachelor Degree Program (with honours)

TECHNICAL SKILLS AND COMPETENCES

LANGUAGES:

- C++, Visual C++
- Matlab
- Perl
- TCL Script Language
- Verilog Hardware Description Language
- XML

OS:

- Windows
- UNIX

SOFTWARE PACKAGES:

- MS Visual C++
 - LaTeX
 - CVS
 - SVN
 - MS Office
 - MS Visio
 - ModelSim
-
- Strong mathematical background
 - Algorithmic programming
 - Object oriented programming
 - Strong problem-solving skills
 - Good team player
 - Good communication skills
 - Creative

PROFESSIONAL EXPERIENCE

➤ **SAIL LABS Technology GmbH** (www.sail-labs.com/)

Date: May 2012 – Till today

Department: Research Group

SAIL LABS - Speech, Artificial Intelligence and Language Labs is one of the world's leading innovators in the field of Speech Technology. By creating high-end software for speech and multimedia analysis, SAIL LABS addresses the markets of media and communication mining.

Responsibilities and duties:

- ❑ **Software tool SentiSAIL**
 - Objective: Multilingual (English, German, Russian) sentiment analysis in the domains of general and natural disaster related news
 - Both traditional and social media
 - Developed in Perl
- ❑ **Regression testing tool for SentiSAIL**
 - Developed in Perl
- ❑ **Software tool for clustering of documents**
 - Clustering into not predefined clusters
 - Unilingual and multilingual clustering
 - Developed in Perl
- ❑ **Software tool CAVA (Continuous Automatic Vocabulary and language model Adaptation)**
 - Objective: Automatically keeping the vocabulary and language model of a speech-recognizer up-to-date using parallel sources from the Web
- ❑ **Project QuOIMA (Quelloffene Integrierte Multimedia Analyse – Open Source Integrated Multimedia Analysis)**
 - Objective: Analysis of open source information of both traditional and social media for decision support and risk estimation in crisis and disaster management
 - Partners:
 - Joanneum Research
 - SAIL LABS Technology GmbH
 - Universität Wien
 - AEI (Agentur für europäische Integration und wirtschaftliche Entwicklung)
 - ZentDok/LVAk/BMLVS
 - My responsibilities: Sentiment analysis and clustering of traditional and social media data in the period of the European floods 2013
- ❑ **Updating of language models used for automatic speech recognition in Russian**
 - Redesign of a vocabulary to mirror current terminology
 - Streamlining of pronunciations
 - Rebuilding of a statistical language model reflecting the current use of the newly devised vocabulary in contemporary Russian sources

➤ **Virage Logic Corporation Yerevan Branch** (www.viragelogic.com)
(now part of Synopsys Corporation <http://www.synopsys.com>)

Date: January 2002 – February 2010

Department: Embedded Test & Repair

Virage Logic was a world market leader in developing Semiconductor IP (Intellectual Property) Platforms. Virage Logic's rich portfolio of highly differentiated IP included embedded memory, logic and I/Os. Its various forms of IP are used in the design and manufacture of system-on-a-chip integrated circuits for communications and Internet infrastructure, handheld and portable devices, computer and graphics applications.

Responsibilities and duties:

❑ **Self Test and Repair (STAR) Multi-memory System Compiler Development**

(Developed by a team of 3 members)

- Customers: LSI Logic, AMD, NEC, Cisco Systems, ATI, Mentor Graphics, etc.
- Developed in TCL
- Development of TCL templates and automation procedures of the main modules, the running of which with customer specific parameters generates the embedded STAR memory system required by the customer
- Preparation of documentation templates for automatic documentation generation for each customer project

This compiler is an automation tool, allowing to integrate different memory systems into one chip and to provide JTAG (Joint Test Access Group) – IEEE Standard 1149.1 interface for testing the entire memory system network.

❑ **STAR Silicon Debugger Tool** (Developed by a team of 10 members)

- Customers: LSI Logic, Freescale, Avago Technologies, etc.
- Developed in C++
- Module developed by me: STAR Silicon Debugger Verilog Plug-in

STAR Silicon Debugger, developed in C++, is a multi-platform tool providing means to Virage Logic customers to test and debug their chips. Verilog Plug-in output is a verilog code, a module called "testbench" for testing customer memory systems, which may be simulated in environments such as NC Verilog, ModelSim, Verilog XL, etc.

PUBLICATIONS

1. Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig, "Architectural style classification of building facade windows," in *Proceedings of the 7th International Symposium on Visual Computing (ISVC '11)*, ser. LNCS, vol. 6939, Las Vegas, Nevada, USA, 2011, pp. 280–289.

2. Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig, "Classification of Gothic and Baroque architectural elements," in *Proceedings of the 19th International Conference on*

Systems, Signals and Image Processing (IWSSIP '12), Vienna, Austria, 2012, pp. 330–333.

3. Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig, “Architectural Style Classification of Domes,” in *Proceedings of the 8th International Symposium on Visual Computing (ISVC '12)*, ser. LNCS, vol. 7432, Part 2, Rethymnon, Crete, Greece, 2012, pp. 420-429.

4. Gayane Shalunts, Yll Haxhimusa, and Robert Sablatnig, “Segmentation of Building Facade Domes,” in *Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP '12)*, ser. LNCS, vol. 7441, Buenos Aires, Argentina, 2012, pp. 324-331.

5. Gayane Shalunts, Gerhard Backfried and Katja Prinz, “Sentiment Analysis of German Social Media Data for Natural Disasters”, in *Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM '14)*, University Park, Pennsylvania, USA, 2014, pp. 752-756.

6. Gayane Shalunts and Gerhard Backfried, “SentiSAIL: Sentiment Analysis in English, German and Russian”, in *Proceedings of the 11th International Conference on Machine Learning and Data Mining (MLDM '15)*, Hamburg, Germany, 2015.

LANGUAGES

- **English (fluent)**
- **Armenian (native)**
- **Russian (good)**
- **German (intermediate, level B1)**

PERSONAL INTERESTS

Travelling, photography, architecture, art, music, playing the piano, dancing, cooking.