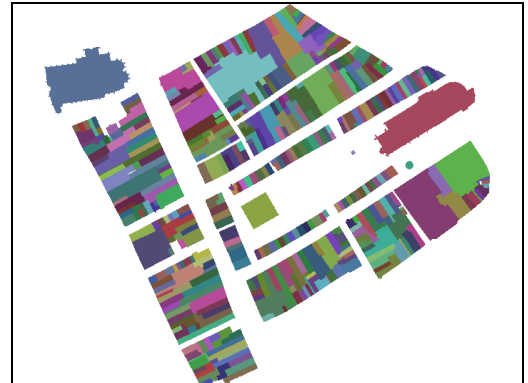
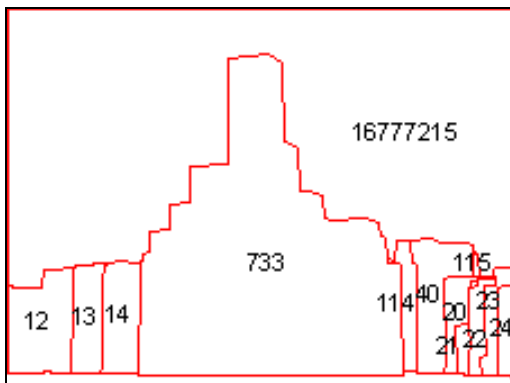
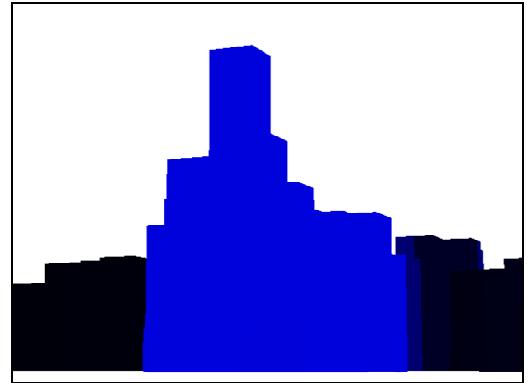
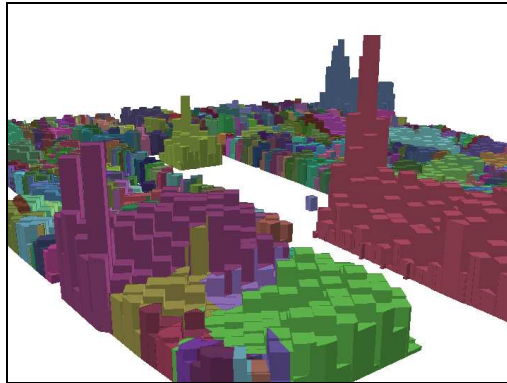


Label placement in 3D georeferenced and oriented digital photographs using GIS technology



MSc. Geomatics thesis by Arnoud de Boer

December 2007



Delft University of Technology
Section GIS Technology
Delft, The Netherlands



European research
project Tripod



Geodan Systems and Research
Amsterdam, The Netherlands

**LABEL PLACEMENT IN 3D GEOREFERENCED AND ORIENTED
DIGITAL PHOTOGRAPHS USING GIS TECHNOLOGY**

by

ARNOUD DE BOER

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

In

Geomatics

Professor: prof.dr.ir. P.J.M. van Oosterom
Supervisor: Ir. E. Verbree

Delft University of Technology
Delft, The Netherlands

December 2007

Abstract

The increasing availability of digital camera devices and camera phones enable users to capture and upload digital photos at any a time and any place. Online image collections face the problem organizing increasing photo collections. Because users experience adding qualitative photo annotations as a time-consuming and tedious task, they omit to do it. However, good photo annotation is very important for fast and reliable photo retrieval. The integration of positioning devices (e.g. GPS) with digital camera enables to capture position along with the photo and this location metadata has been applied in previous research to automatically caption digital photos. The disadvantage of describing a photo using location metadata only is that it only enables to add information about its surrounding and not about the actual objects pictured.

The European research project Tripod takes one step further in assuming in the near future camera devices are available having a GPS and digital compass. As such, it is possible to exactly identify the scene and objects pictured as besides position also the view direction is known. The focus of this research is on how to apply the capture position and view direction information of a photo to identify and localize visible objects inside a digital photo in order to place a label next to it at the best possible location.

The concept is tested by collecting three-dimensional georeferenced and oriented digital photo at the Market Square in the historic city centre of Delft. A collection of low-resolution and high-spatial accuracy photos, and a collection of high-resolution and low-spatial accuracy photos were captured using a Topcon imaging total station and a Nikon digital camera mounted with compass and GPS data logger next to it, respectively.

Using the output of a perspective viewer service (i.e. a virtual scene), the visible object are identified and localized. Therefore, a three-dimensional model is required to be rendered to create and export the virtual scene to an output image. Three different approaches for creating an extrusion model are described using several two-dimensional vector spatial data sources and a raster elevation model. The extrusion model created from a vectorized elevation raster intersected with building footprints is considered to be the best three-dimensional model to be applied for this proof-of-concept.

In order to enable to link the objects inside the virtual scene to the two-dimensional footprints to be able to pick the labels or names of objects from it and to associate it with the corresponding object, the features inside the three-dimensional model are coloured with RGB colour values based on its object identifier.

It is assumed that the best location for label placement inside a digital photo is at the empty areas, defined as the area where no objects inside the virtual scene exist and the pixels of the original photo are above a specific colour value (e.g. from the median of all pixel values to white colour). At these empty areas, overlap of labels with objects is avoided because no object-of-interest is located at that area. Using a two-dimensional label engine of ESRI ArcGIS, the labels are placed outside visible objects using connectors. A depth image is created that contains for each pixel the distance to it from camera positioning and is applied to decrease label font sizes with increasing object distance to maintain the perspective view of the photo. The amount of labels that could be place inside a digital photo depends on the size of empty areas, the number of visible objects and user preferences.

The use of different lenses is explored to evaluate misidentification of objects due to barrel and pincushion distortion; however, the perspective viewer service is correcting for these distortions as the field-of-view angle is changed. GPS and compass inaccuracies do cause misidentification of objects inside a photo, particularly as the inaccuracy increases and the field-of-view angle decrease.

The proof-of-concept of object identification is automated in ESRI ArcGIS using Visual Basic for Applications programming. An implementation proposal is provided, including the key components of a photo labelling service, in order to make the outputs of this research available to the rest of the world via a web or location-based service.

It is concluded that the virtual scenes as output of perspective viewer services are appropriate to apply for object identification and localization. In doing so, the problem of label placement in three-dimensional geographic environments is reduced to a two-dimensional map-labelling problem. The best location of label placement is determined using constraints and rules to be applied to the virtual scene and the reclassified-to-binary image of the input photo, and the depth map enables to vary in label font size depending on the object distance.

Further research is recommended to among others which constraints and rules could be further applied to the label algorithm, how to manage misidentification of object due to compass and GPS inaccuracies, how to automatically get a depth maps along with the perspective view and how manage the large amount of features of the proposed extrusion model from vectorized elevation model to datasets with larger spatial extents then applied for this research.

Samenvatting

Door het stijgende aanbod van digitale fotocamera's en cameratelefoons kunnen gebruikers op elk tijdstip en elke plaats foto's nemen en via het netwerk verzenden. Voor online fotocollecties vormt de organisatie van groeiende collecties een probleem omdat het toekennen van een goede beschrijving aan foto's door veel gebruikers als een tijdrovende en vervelende taak wordt ervaren, en om die reden zij het vaak nalaten. Echter zijn goede fotobeschrijvingen erg belangrijk om een foto snel en betrouwbaar te kunnen terugvinden. Het opnemen van plaatsbepalings-microchips (bijvoorbeeld GPS) in digitale camera's en cameratelefoons maakt het mogelijk om tegelijk met de foto de positie van opname op te slaan en in voorgaande onderzoeken is deze locatie metadata toegepast om automatisch foto's te beschrijven. Nadeel van het beschrijven van foto's op basis van locatie alleen is dat enkel een beschrijving over de omgeving van de foto toegekend kan worden, en niet over de objecten welke werkelijk op de foto zijn vastgelegd.

Het Europese onderzoeksproject Tripod gaat een stap verder door te veronderstellen dat in de nabije toekomst fotocamera's beschikbaar zijn met GPS en digitaal kompas. In dat geval is het mogelijk om precies de bepalen welke omgeving en objecten zijn vastgelegd in de foto omdat naast opnamepositie ook de kijkrichting bekend is. Dit onderzoek richt zich op het toepassen van opnamepositie en kijkrichting om zichtbare objecten in een foto te identificeren en lokaliseren om zodoende bij deze objecten een label op de best mogelijke positie in de foto te kunnen plaatsen.

Daarvoor zijn op de Markt in the historische centrum van Delft, foto's met driedimensionale positie en kijkrichting verzameld. Met behulp van een Topcon imaging total station en een Nikon digitale camera waarop een digitaal kompas is bevestigd en een GPS data logger, worden respectievelijk foto's een lage resolutie en hoge geografische nauwkeurigheid en foto's met een hoge resolutie en lage geografische nauwkeurigheid ingewonnen.

Met behulp van de uitvoer van een instrument dat een virtuele omgeving in perspectief weergeeft (d.i. *perspective viewer service*), worden de zichtbare objecten geïdentificeerd en gelokaliseerd. Daarvoor dient een driedimensionaal model zodanig gevisualiseerd te worden dat deze virtuele omgeving geëxporteerd kan worden naar een uitvoerafbeelding.

Drie verschillende methoden om een dergelijk driedimensionaal model te maken worden beschreven op basis van verschillende tweedimensionaal digitale vectorkaarten en een raster hoogte model. Het model dat gecreëerd is op basis van een gevectoriseerd hoogteraster doorsneden met de grondcontouren van de huizen wordt beschouwd als het beste driedimensionale model en om die reden toegepast voor de toetsing van het onderzoeksconcept.

Om de objecten in de virtuele omgeving te relateren aan de tweedimensionale digitale kaarten om zodoende de labels uit deze tweedimensionale dataset te selecteren en te associëren met de objecten, worden de objecten van het driedimensionale model gekleurd met RGB kleurwaarden welke corresponderen met de unieke objectnummers.

Aangenomen wordt dat de beste locatie voor het plaatsen van een label in een digitale foto in de lege ruimte is, welke definieert is al dat gebied waar geen objecten in de virtuele scene bestaan en de pixels van de digitale foto boven een bepaalde kleurwaarden zijn (bijvoorbeeld de mediaan van alle pixel waarden). Overlap van labels met objecten wordt voorkomen in deze lege ruimte, omdat verondersteld wordt dat geen belangwekkende objecten zich bevinden in dit gebied. Met behulp van een tweedimensionaal labelinstrument van ESRI ArcGIS (Maplex), worden labels buiten de zichtbare objecten geplaatst met verbindinglijnen. Een dieptekaart welke voor elk pixel de afstand tot de camera of waarnemer bevat, is gebruikt om van een label de lettergroottes te variëren in relatie tot de afstand om zo de perspectief in de foto te behouden. Het aantal labels dat geplaatst kan worden hangt af van de grootte van de lege gebieden, het aantal zichtbare objecten en voorkeuren van de gebruiker.

Het gebruik van verschillende lenzen is onderzocht met betrekking tot misidentificatie van objecten als gevolg van geometrische vertekening (nl. *barrel distortion* en *pincushion distortion*); echter de *perspective viewer service* van dit onderzoek (d.i. ESRI ArcScene) corrigeert al voor deze lensvertekening. Onnauwkeurigheden in GPS en kompas waarnemingen veroorzaken wel misidentificatie van objecten in een foto, met name als de onnauwkeurigheid toeneemt en de kijkhoek afneemt.

De toetsing van het onderzoeksconcept is geautomatiseerd in ESRI ArcGIS met behulp van programmeren in Visual Basic for Application. Een voorstel tot implementatie om de foto labelplaatsing via een website of locatiegebonden dienst aan de wereld beschikbaar te stellen, is aangedragen met daarin opgenomen de belangrijkste componenten.

Conclusie is dat de virtuele omgeving welke als uitvoer door een perspective viewer service wordt gegeven geschikt is voor het identificeren en lokaliseren van zichtbare objecten. Zodanig wordt het probleem van labelplaatsing in een driedimensionale ruimte teruggebracht naar een tweedimensionaal probleem. De beste locatie voor een label wordt bepaald aan de hand van voorwaarden en regels welke worden toegepast op de virtuele omgeving en de gebinairiseerde digitale foto, en een dieptekaart maakt het mogelijk om te variëren in labelgrootte op basis van de objectafstand.

Verder onderzoek is aanbevolen naar onder andere welke voorwaarden en regels extra toegevoegd kunnen worden aan het label algoritme, hoe de misidentificatie van objecten op te lossen als gevolg van nauwkeurigheden van GPS en kompas waarnemingen, hoe automatisch een dieptekaart verkregen kan worden uit de perspective viewer service en hoe het grote aantal objecten van het voorgestelde driedimensionale model op basis van het gevectoriseerde hoogtemodel toegepast kan worden voor datasets met grotere ruimtelijke omvang dan gebruikt voor dit onderzoek

Acknowledgements

This thesis is the result of my graduation research which I carried out at the GIS technology department of the OTB Research Institute for Housing, Urban and Mobility Studies (Delft University of Technology) and Geodan Systems and Research from June to December 2007. During the research, lots of people contributed to the research and I would like to thank them for their help.

First of all, I gratefully acknowledge Professor Peter Van Oosterom (Delft University of Technology) and my supervisors Edward Verbree (Delft University of Technology) and Eduardo Dias (Geodan Systems and Research) for their trust, motivation and the interesting discussions to refine ideas.

Lennard Huisman (Delft University of Technology) for helping me to work out the measurement setup for the image collection and Frank van de Heuvel (Cyclomedia) for sharing his time and knowledge about panoramic image processing.

GDMC partner ESRI for providing the ESRI ArcGIS software with three-dimensional visualization tool ArcScene and map labelling engine extension Maplex.

All colleagues of Geodan for their interest, assistance with hardware and software issues and for giving me a free hand to do my research, particularly Evert Meijer, Steven Fruijtier, Alessandra Scotta, Anne Blankert and Barend Gehrels.

My family and friends for their never-ending support and love and to be all ears when I was not able to stop talking about digital photos and my research, particularly my encouraging mother Jelly and my beloved girlfriend Melina.

And finally to all others who contributed, directly or indirectly:

Thank you very much!

Arnoud de Boer
Delft, 18 December 2007

< Remark: This version of the thesis is corrected afterwards for spelling errors and other small mistakes and inaccuracies before a copy is submitted to the University Library and GDMC website >

Contents

ABSTRACT	V
SAMENVATTING.....	VII
ACKNOWLEDGEMENTS	XI
CONTENTS.....	XIII
LIST OF FIGURES	XVI
LIST OF TABLES.....	XVIII
LIST OF ACRONYMS.....	XIX
LIST OF TERMS.....	XXI
1. INTRODUCTION	1
1.1 OUTLINE	1
1.2 OBJECTIVES	3
1.3 STRUCTURE	3
2. DIGITAL IMAGING, 3D COMPUTER GRAPHICS AND CARTOGRAPHIC LABEL PLACEMENT	5
2.1 DIGITAL IMAGING	5
2.1.1 <i>Image capture</i>	5
2.1.2 <i>Image storage and metadata formats</i>	9
2.1.3 <i>Image annotation</i>	14
2.1.4 <i>Image retrieval</i>	17
2.2 3D COMPUTER GRAPHICS	17
2.2.1 <i>3D Computer graphics definition</i>	17
2.2.2 <i>Rendering</i>	18
2.2.3 <i>Depth maps</i>	19
2.2.4 <i>Perspective viewers</i>	19
2.3 LABEL PLACEMENT	20
2.3.1 <i>Cartographic label placement</i>	20
2.3.2 <i>Multimedia cartography</i>	21
2.3.3 <i>Label placement in three-dimensional scenes</i>	21
2.4 RELATED RESEARCH APPLICATIONS.....	23
2.4.1 <i>Cyclomedia panoramic imaging</i>	23
2.4.2 <i>Ordnance Survey pointing device Zapper</i>	24
2.4.3 <i>Nokia Mobile Augmented Reality applications</i>	25
2.4.4 <i>Microsoft PhotoSynth project</i>	26
2.4.5 <i>Quakr project</i>	27

3. RESEARCH METHODOLOGY.....	29
3.1 BACKGROUND.....	29
3.2 PROBLEM DEFINITION AND RELEVANCE OF RESEARCH.....	30
3.3 RESEARCH QUESTION AND OBJECTIVES.....	31
3.4 APPROACH.....	32
3.5 RESEARCH OUTPUTS	33
4. COLLECTION OF 3D GEOREFERENCED AND ORIENTED DIGITAL PHOTOGRAPHS	35
4.1 HIGH-SPATIAL ACCURACY AND LOW-RESOLUTION DIGITAL PHOTOS	35
4.1.1 Topcon imaging total station.....	35
4.1.2 Measurement setup.....	35
4.2.1 Processing of observations.....	38
4.2.2 Results.....	40
4.2 LOW-SPATIAL ACCURACY AND HIGH-RESOLUTION DIGITAL PHOTOS.....	41
4.2.1 Nikon D100 camera with GPS and 3-axis compass	41
4.2.2 Measurement setup.....	41
4.2.3 Processing	42
4.2.4 Results.....	43
4.3 DISCUSSION OF RESULTS.....	43
5. SPATIAL DATA PREPARATION	45
5.1 DATA AVAILABILITY	45
5.2 EXTRUSION MODEL USING ASSUMED HEIGHT VALUES.....	48
5.3 EXTRUSION MODEL USING AHN HEIGHTS.....	49
5.4 EXTRUSION MODEL FROM INTERSECTION WITH RASTERIZED AHN	52
5.5 DISCUSSION OF RESULTS.....	54
6. OBJECT IDENTIFICATION	57
6.1 PROOF-OF-CONCEPT.....	57
6.2 PERSPECTIVE VIEW IN ESRI ARCSCENE	58
6.3 LINKING FEATURES IN ESRI ARCMAP.....	61
6.4 DISCUSSION OF RESULTS.....	62
7. LABEL PLACEMENT	63
7.1 ESRI MAPLEX LABEL ENGINE.....	63
7.2 EXTERNAL ANNOTATION.....	64
7.3 PERSPECTIVE VIEW WITH VARYING LABEL SIZES.....	65
7.4 PRIORITIZATION OF OBJECTS.....	67
7.5 NUMBER OF LABELS TO PLACE INSIDE A PHOTO	68
7.6 DISCUSSION OF RESULTS.....	69
8. EFFECTS OF LENS DISTORTIONS AND POSITIONING AND ORIENTATION INACCURACIES.....	71
8.1 LENS DISTORTIONS.....	71
8.2 POSITIONING AND ORIENTATION INACCURACIES.....	73
8.3 DISCUSSION OF RESULTS.....	78
9. AUTOMATION AND IMPLEMENTATION	79
9.1 AUTOMATION USING ARCOBJECTS AND VBA PROGRAMMING IN ESRI ARCGIS	79
9.2 IMPLEMENTATION OF WEB OR LOCATION-BASED SERVICE	82
9.2.1 Image metadata	82
9.2.2 Web Perspective Viewer Service.....	82
9.2.3 Input images for labeling.....	83
9.2.4 Label algorithm	83
9.2.5 Output image	84
9.3 DISCUSSION OF RESULTS.....	85

10. CONCLUSIONS.....	87
10.1 DISCUSSION	87
10.2 CONCLUSION	88
10.3 RECOMMENDATIONS.....	89
REFERENCES	91
APPENDIX A	95
APPENDIX B.....	99
APPENDIX C.....	101
C1 VBA CODE FOR AUTOMATION IN ESRI ARCSCE.....	101
C2 VBA CODE FOR AUTOMATION IN ESRI ARCMAP.....	105
C3 VBA CODE FOR COLOURING OF OBJECTS WITH OIDS COLOR VALUES	110

List of Figures

Figure 1 Field of View (FOV) and visible extent.

Figure 2 Schematic overview of lens optics.

Figure 3 Examples of raster file formats BMP, GIF, JPEG and PNG and their filesizes.

Figure 4 Geotagging of an image to georeference it on a map (Dias et al., 2007)

Figure 5 Camera direction parameters heading, tilt and roll.

Figure 6 Spatial granularity of image annotation at 5 image libraries (N=100).

Figure 7 The six steps of the rendering process (O'Rourke, 2003)

Figure 8 Example of a digital photo and its depth image (Dofpro.com, 2007).

Figure 9 Growing-border (up) and interval slot (down) view management (Maass, 2006).

Figure 10 Process for the annotation of 3D scenes as proposed by Hagedorn (2007).

Figure 11 *Panoramic imaging of Cyclomedia and one of its applications: projection of cables and pipelines in a cyclorama.*

Figure 12 Ordnance Survey pointing device Zapper.

Figure 13 Augmented Reality research projects by Ordnance Survey.

Figure 14 *A Nokia 6680 mobile device, communicating via a Bluetooth interface with the add-on box containing GPS and sensors, enables to obtain additional information from the WWW of the building/object aimed at.*

Figure 15 Microsoft PhotoSynth 3D model is a point cloud with digital photos connected to it. (Microsoft, 2007).

Figure 16 Distance and direction measurements collected at Markt in Delft.

Figure 17 GBKN control points at the Markt in Delft.

Figure 18 Link Table of similarity transformation of occupation point 1000 data set.

Figure 19 Link Table of similarity transformation of occupation point 2000 data set.

Figure 20 Link Table and RMS-error of occupation point 2000 after neglecting control point 2006.

Figure 21 NAP-pin inside buttress of the Nieuwe Kerk of Delft (left) and its location on the GBKN (right).

Figure 22 Points of camera (POI) and points of photos (POIs).

Figure 23 Nikon D100 camera devices added with digital 3-axis compass and GPS.

Figure 24 Collection of Nikon images at the Markt in Delft.

Figure 25 Datum transformation and clipping of spatial datasets using ESRI ArcGIS ModelBuilder.

Figure 26 Extrusion of 2D features using assumed height values.

Figure 27 Building footprints are extruded to the height values of the centroids.

Figure 28 Using the Interpolate Shape tool to add heights from AHN to the building features.

Figure 29 Floating features as result from AHN height interpolation.

Figure 30 Extrusion to value zero, pushing the floating building features down.

Figure 31 Rasterized features using the AHN cellsize of 5m.

Figure 32 Intersection of vectorized AHN and building footprints.

Figure 33 Detail of Intersection of vectorized AHN and building footprints.

Figure 34 Extrusion model of the intersection of vectorized elevation model with the building footprints.

Figure 35 Visual comparison of the created extrusion models.

Figure 36 From left to right: Examples of a SimpleRenderer, UniqueValueRenderer and ClassBreaksRenderer.

Figure 38 Difference between loss-less compressed PNG (right) and lossy compressed JPEG (left).

Figure 39 Overlay of the virtual scene and its vectorized layer with Topcon images 1010 and 2014.

Figure 40 Object identification results of Topcon images 1010 and 2014.

Figure 41 Digital photo 2014 with external annotations.

Figure 42 Photo 2014 labelled with conflict resolution value 999 to avoid overlap of labels with object as much as possible.

Figure 43 Photo 1010 with labels overlapping visible objects not existing in the three-dimensional model.

Figure 44 Digital photo labelled as such that labels do not overlap objects of the virtual scene and do not overlap pixels above a specific value from binary image.

Figure 45 The photo is labeled with varying font size depending on distance from observer point; the extrusion model is coloured based on the subject distance in ESRI ArcMap and enables to create a depth map using ESRI ArcScene.

Figure 46 Digital photo 1019 labelled with different amount of objects based on the minimum feature size for labelling.

Figure 47 Barrel distortion and pincushion distortion (DPreview.com, 2007).

Figure 48 The virtual scenes are corrected for lens distortions by ESRI ArcScene when changing zoom or field -of-view angle.

Figure 49 Misidentification of objects in terms of percentages as function of the compass inaccuracy and the focal length F .

Figure 50 Misidentification of objects in terms percentages as function of GPS inaccuracy and focal length F at subject distance = 100 meter.

Figure 51 Different virtual abstractions due to GPS inaccuracies.

Figure 52 Screenshot of the demo-application build in ESRI ArcScene using VBA.

Figure 53 Screenshot of the demo-application build in ESRI ArcMap using VBA.

Figure 54 Overall process of a photo labelling service.

List of Tables

Table 1 Typical focal lengths and their 35mm format designations (DPreview.com, 2007).	6
Table 2 Example of EXIF metadata (left) and IPTC headers (right) metadata format for a particular image.	12
Table 3 Images captured at occupation point 1000.	36
Table 4 Images captured at occupation point 2000.	36
Table 5 Height of occupation points.	40
Table 6 Collection of Nikon images captured with different zoom.	43
Table 7 Characteristics of the spatial data sets.	47
Table 8 Misidentification of pictured scene and objects in terms of percentages due to compass inaccuracies.	74
Table 9 Misidentification of pictured scene and objects in terms of percentages due to GPS inaccuracy at subject distance $D=50$ m.	76

List of Acronyms

2D	Two-dimensional
3D	Three-dimensional
AHN	Actueel Hoogtebestand Nederland
AR	Augmented Reality
BMP	Bitmap
CAD	Computer Aided Design
CBIR	Content-Based Image/Information Retrieval
CCD	Charged Coupled Device
CIELAB	Commission Internationale de l'Eclairage LAB (color system).
CMYK	Cyan, Magenta, Yellow , Key (color system)
CRS	Coordinate Reference System
DEM	Digital Elevation Model
DIB	Device Independent Bitmap
DIG35	Digital Information Group 35
DOF	Depth of Field
DTD	Document Type Definition
ESRI	Environmental Systems Research Institute
EXIF	Exchangeable image file
EXTENT	contEXT and contENT
FOV	Field of View
GBKN	Grootschalige Basis Kaart Nederland
GIF	Graphics Interface Format
GIS	Geographic Information System
GML	Geography Mark-up Language
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IMU	Inertial Measurement Unit
INS	International Navigation System
IPTC	International Press Telecommunication Council
ISO	International Organization for Standardization

JPEG	Joint Photographic Experts Group
KML	Key Markup Language
LiDar	Light Detection and Ranging
MARA	(Nokia) Mobile Augmented Reality Applications
MMM	Mobile Multimedia Metadata
MP	Megapixel
NAP	Normaal Amsterdams Peil
NP-hard	Nondeterministic Polynomial-time hard
OGC	Open Geospatial Consortium
OID	Object identifier
OS	Ordnance Survey
PDA	Personal Digital Assistant
PNG	Portable Network Graphics
POC	Point of Camera
POI	Point of Interest
RD	Rijksdriehoekstelsel
RDF	Resource Description Framework
RGB	Red Green Blue (color system)
RMS	Root Mean Square
RTK	Real Time Kinematic
SD	Subject Distance
SLD	Styled Layer Descriptor
SLR	Single Lens Reflex
SVG	Scalable Vector Graphics
TDN	Topografische Dienst Nederland
TIFF	Tagged Image File Format
TIN	Triangular Irregular Network
TOID	Topographic Object Identifier
Tripod	Tri-Partite Multimedia Object Description
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
VA	View Angle
VBA	Visual Basic for Applications
WCS	Web Coverage Service
WFS	Web Feature Service
WGS84	World Geodetic System 1984 (coordinate system)
WLAN	Wireless Local Area Network
WMS	Web Map Service
WPVS	Web Perspective Viewer Service
WTS	Web Terrain Service
WVAS	Web View Annotation Service
WWW	World Wide Web
XML	Extensible Mark-up Language
XMP	Extensible Metadata Platform

List of Terms

Annotation is the act of adding explanatory notes to a photo.

Augmented Reality is an environment that includes both virtual reality and real-world elements.

Captioning is the act of adding textual descriptions to next, above or under a digital photo).

Cartography are the graphic principles supporting the art, science, and techniques used in making maps or charts.

Cyclorama is a panoramic image.

Depth map is an image that contains information about the distance of the surface to the observer on a pixel-by-pixel base.

Depth of field is the distances from the nearest point to the farthest point of acceptably focus.

Digital photo is a collection of pixels (picture) captured using an electronic camera device.

Direction specifies the relative angle between one point with respect to another reference point (a.k.a. the orientation).

Elevation model is a (digital) representation of ground surface topography or terrain, represented as raster or TIN.

Extrusion model is a three-dimensional (visualization) model containing two-dimensional features that are extruded to a certain (height) value.

Feature is a representation of a real-world object on a map represented as point, polygon, multipoint, polyline or multipart.

Field of view is the angular extent of the observable world that is seen at any given moment.

Full spatial metadata refers to three-dimensional position and orientation information.

Geocoding is refers to the cross-referencing between specifically recorded x,y coordinates of a location and non-geographic data such as addresses or post-codes.

Geographical Information System is a defined as a computer-based system to store, manage, analyze and visualize geographic-related information

Georeferencing is the act of aligning geographic data to a known coordinate system.

Geotagging is the act of adding location information to multimedia.

Heading is the orientation around the z-axis with respect to magnetic North.

Hyperfocal distance is the distance from optical center of a lens to the nearest point in acceptably sharp focus of a lens focused at infinity.

Labeling is the act of adding keywords (tags) inside a digital photo.

LIDAR is an optical remote sensing technology that measures properties of scattered light (laser) to find range and/or other information of a distant target.

Location specifies a position in terms of topological relations, e.g. street address, description, postal code.

Metadata is data about data, e.g. name, date, size, and extent of the file.

Multimedia is the combined (interactive) use of several media as sound and full motion video in computer applications.

Object is a digital representation of a spatial or non-spatial entity. Within this research it often denotes an entity that is captured inside a digital photo (e.g. a building, tree, and river).

Object identifier is a unique number or value referring to a specific object.

Pitch is the orientation around the y-axis

Pixel, short for picture element, is a single point in a graphic image

Position specifies a set of coordinates with respect to a well-defined reference system.

Raster is a spatial data model that defines space as an array of equally sized cells arranged in rows and columns, and comprised of single or multiple bands. Each cell contains an attribute value and location coordinates.

Rendering is the process of producing a two-dimensional picture from three-dimensional data.

Roll is the orientation around the x-axis.

Root Mean Square is a statistical measure of the magnitude of a varying quantity.

Subject distance is the distance from the observer point to the focus plane.

Toponym is a place-name or geographical name.

Total Station is an optical instrument combining an electronic theodolite, an electronic distance measuring device and software running on an external computer.

Chapter 1

Introduction

1.1 Outline

After homecoming from holidays or an event with a fantastic collection of digital photos a tedious task lies ahead, namely giving a captioning to all the digital photos. Who refrains to describe and organize his or her photos in the right way immediately, encounters problems when he or she is looking up a particular photo afterwards. However, numerous people experience describing their photos, by adding titles, tags (keywords) or full textual descriptions, as a time-consuming and annoying task and for that reason numerous people omit to do it (Dias et al., 2007). But as the image collection increases, this disorganization causes photo retrieval to require more effort and it might even fail to succeed.

The availability of digital multimedia (photos, videos and sounds) on the WWW is rapidly increasing. Camera and voice recorders integrated in mobile phones and digital cameras extended with wireless communication enable users to capture and upload multimedia content any time at any place. This enormous expansion of available multimedia on the WWW demands for new approaches for organizing online multimedia collections. Organization of multimedia collections is a major task, but indispensable for quick and easy retrieval. Multimedia collections face the problems of rapidly increasing collection sizes and the difficultness to analyze and index due to the complexity of multimedia content. Management problems occur among others due to cumbersome and time-consuming annotation. Because it is for computers extremely difficult to analyze the multimedia content using vision technology and object recognition techniques only and fast visual scanning cannot be applied to large collections, multimedia content based tools for describing and online search digital photos are not practical for meaning full organizations of large online multimedia collections.

The European research project Tripod tries to find a solution for this organization problem. Its objective is to “improve the access to the enormous body of visual media” by developing new tools to automatically annotate and search for online multimedia; particularly focussing on digital photos in professional online image collections. The Tripod research project anticipates on recent developments of the

integration of camera devices with location-aware chips (based on satellite positioning e.g. GPS). Some examples are camera phones having an integrated GPS (e.g. Nokia N95, HTC P3300) or additional GPS-adapters to be connected to (professional) SLR camera devices (e.g. Nikon GPS adapter for Nikon D-series). This integration enables to automatically save the location of image capture together with the digital photos.

Previous research focussed on how to apply this location information for the organization of digital photos, particularly on how to benefit from the GPS position to describe the image content. This resulted in state-of-the-art photo annotation tools that describe images with information about the capture surroundings, e.g. weather information and geographical names. Also, online image hosting services (e.g. Panoramio or Locr) enable users to upload their digital photo, put it on a map, and thereupon the online service attaches geographical names to the digital photo. Disadvantage of using location information only is that it only provides information about its surroundings and not about the actual subject pictured; therefore also the direction in which the camera is pointed into is required.

The Tripod research project goes one step further in assuming that in the near future also a chip is introduced in these all-in-one camera devices that measures directions in three-dimensions (e.g. using a three-axis digital compass). Actually, this is already happening in camera phones in Asia, where it is been applied for among others (pedestrian) navigation purposes. It is expected that the price of a digital compass-chip the next few years will decrease to cost under USD 10 (WTC, 2007), so as such the assumption of compass-integrated camera devices available to average people is very likely to become true. Together with positioning from GPS, this compass makes it possible to automatically capture camera position and the direction the camera is pointed at, and this enables to exactly identify the extent of the scene that is captured inside the photo. The objective of this research is to use the three-dimensional position and orientation information (a.k.a. full spatial metadata) captured along with a digital photo, called a 3D georeferenced and oriented digital photo, to identify the subject pictured in order to label this photo with the names of the visible subjects. It is part of the Tripod research project and is carried out at Geodan Systems and Research, the research department of the geo-ICT company Geodan, located in Amsterdam, The Netherlands and contributing to the Tripod research project.

In more detail, this research uses the three-dimensional position and view direction captured along with a digital photo to identify and locate visible objects (e.g. buildings, streets, rivers) to be able to place a label next to its object inside the photo to describe the image content. Using geographical analysis tools and digital maps a method is proved to automatically identify visible objects and to determine the best location for a label to be placed inside the digital photo. In doing so, this research proves the use of spatial data sets and geographic information technology to provide a powerful solution to overcome the problem and inability of computers to describe digital photos using content-based tools and vision technology only.

Results of this proof-of-concept are on the one hand innovative tools can be implemented and made available to the world; tools that improve the organization of personal and online image collections due to the automatic way qualitative descriptions can be added to digital photos in order to enable better image retrieval. On the other hand, it can be the basis in developing new tools new tools used to improve accessibility, for example for visually impaired people to sense the digital photo using e.g. large-sized label fonts or sounds on mouse over, as objects inside the photo are well-identified and well-localized.

1.2 Objectives

This research focus is on using GIS technology and spatial data sets for label placement inside 3D georeferenced and oriented digital photos. It applies perspective view generator's output for the identification of object captured inside a scene, and secondly the back-linking of the visible objects inside this virtual abstraction with the original features of the 2D spatial datasets enables to place geographical names next to it.

Therefore, the goal of this master's thesis research is to label objects captured inside digital photos. The main research question is:

How to identify captured objects and where to place labels to annotate pictured objects inside 3D georeferenced and oriented digital photos using GIS technology?

The research exists of two main parts, i.e. the object identification and label placement.

The objectives are to:

- identify and localize visible objects inside digital photos using GIS perspective view generator's output;
- search for the best position for labels to annotate visible objects using 2D GIS label algorithms;
- determine the effects of lens distortions, position and orientation inaccuracies on the (mis)placement of labels;
- propose a system architecture in order to make the photo labelling service available to the rest of the world.

1.3 Structure

This thesis report is organized as follows

- **Chapter 2** provides background information as this research integrates many different fields of knowledge such as digital imaging, geographic multimedia, lens optics, 3D computer graphics, cartographic label placement and GIS.

- **Chapter 3** describes the research methodology including among others the research framework, relevance of the thesis' results, the research questions, objectives, limitations and assumptions.
- **Chapter 4** includes the collection of 3D georeferenced and oriented digital photographs serving as sample data and specifically captured for this research to test and verify the developed algorithms and concepts.
- **Chapter 5** describes the GIS data availability and preparation of extrusion models, which serves as input to create virtual abstractions (corresponding to captured photo scenes) and to pick the pictured objects its names from in order to label them.
- **Chapter 6** provides the proof-of-concept of how to identify and localize pictured objects using GIS perspective view generator's output. It describes the principle of linking the visible objects inside the 3D virtual abstraction to the features of 2D spatial data sets.
- **Chapter 7** continues with the identified objects in order to attach labels to it. It provides some methods and concepts to find the best position for the labels including positioning, prioritization and symbolization issues.
- **Chapter 8** discusses the effects of (mis)placement of labels caused by (mis)identification of objects due to lens distortions, and positioning and orientation inaccuracies.
- **Chapter 9** concerns the implementation and automation of the proof-of-concept. For this research, the process of object identification is automated using VBA programming in ESRI ArcGIS. Also a proposal of system architecture is given for implementation of the proof-of-concept, which describes how to make a photo-labelling tool available to the rest of the world via a web or location-based service.
- **Chapter 10** includes some discussion of the results and some suggestion how to improve and further develop the label placement inside digital photos.
- Finally, **chapter 11** provides the conclusions and some recommendation for further research.

Chapter 2

Digital imaging, 3D computer graphics and cartographic label placement

This chapter provides an overview of related subjects for this research. Section 2.1 gives an introduction to basic concepts of digital imaging with its focus on capture, storage, annotation and retrieval. 3D Computer graphics is described in section 2.2 including rendering principles, depth maps and three-dimensional visualizations. Section 2.3 continues with the principles of GIS label placement and multimedia cartography. Finally, section 2.4 concludes with some related research and application to put this research into perspective.

2.1 Digital imaging

Digital imaging (a.k.a. digital image acquisition) refers to the creation of a digital image from a particular scene or object using a camera or similar device. This section describes the basic concepts of image capture, storage, annotation and retrieval.

2.1.1 Image capture

Electronic imaging instruments use two-dimensional detector arrays of charge-coupled devices (CCDs) for image acquisition (Lillesand en Kiefer, 1994). A charge-coupled device is a silicon-based semiconductor chip bearing a two-dimensional matrix of photo-sensors, or pixels, referred to as the image area. CCD chips detect electromagnetic energy and the magnitudes of charges produced by these electromagnetic strikes are captured proportional to the scene brightness.

The image sensor format (a.k.a. film size) is the shape and size of the image sensor (e.g. CCD) and this image sensor inside the camera body together with the lens record the image data and determines the extent of the scene captured. Larger sensors or films have wider field of views and can capture more of the scene. The field of view (FOV; a.k.a. angle of field, picture or view angle; see Figure 1) determines that portion of the subject appearing in the photographic frame and is given by Eq. 1 as the ratio between the focal length F and the diagonal film size s , i.e.

$$FOV = 2 \arctan \left(\frac{s}{2F} \right) \quad (\text{Eq. 1})$$

The focal length F of a lens is defined as the distance in millimetres from the optical centre of the lens to the focal point, which is located on the sensor or film if the subject (at infinity) is "in focus". Usually the focal length is automatically saved to image metadata at time of photo capture. A change in focal length implies a change in zoom; or equivalent a change in field-of-view angle causes a change in zoom.

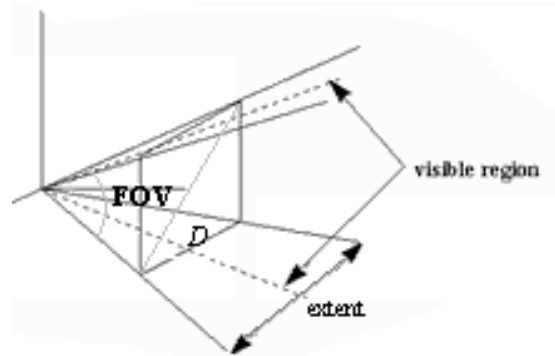


Figure 1 Field of View (FOV) and visible extent.

Different lenses have different focal lengths and consequently different view angles (see Table 1), distinguished are among others

- *Fish-eye lens*; with a view angle equal to or greater than 180 degrees, a field of view beyond the capability of a human eye.
- *Wide-angle lens*; good for shooting wide areas such as landscapes due to the wide view angle and deep depth of field.
- *Standard lens*; natural looking perspective close the view angle of human-eye view;
- *Tele-lens*; is suitable for close-up views of distant subjects.

(Nikon, 2007)

Table 1 Typical focal lengths and their 35mm format designations
(DPreview.com, 2007).

Focal length	Field of view angle
< 20mm	Super Wide Angle
24mm - 35mm	Wide Angle
50mm	Normal Lens
80mm - 300mm	Tele
> 300mm	Super Tele

The FOV associated with a focal length is usually based on the 35mm film photography, given the popularity of this format over other formats. In particular, image sensors in digital SLR cameras tend to be smaller than the 24mmx36mm image area of full-frame 35mm cameras. Lenses produced for 35mm film cameras

may mount well on the digital bodies, but the larger image circle of the 35mm system lens allows unwanted light into the camera body, and the smaller size of the image sensor compared to 35mm format results in cropping of the image compared to the results produced on the film camera. This latter effect is known as field-of-view crop; the format size ratio is known as the crop factor or focal length multiplier.

The aspect ratio is the ratio between the width and height on an image, usually expressed as two integers, e.g. width/height = 1.5 is expressed as width: height = 3:2. Typically, the aspect ratio 35 mm films and most SLR camera is 3:2 (e.g. film size of 24x36mm; computer screens and compact digital camera usually have an aspect ratio of 4:3 or another possible aspect ratio is 16:9 (wide-screen).

Other camera parameters that influence the appearance of the scene are among others:

- *Exposure*: is the amount of light received by the film or sensor and is determined by how wide you open the lens diaphragm (a.k.a. aperture) and by how long you keep the film or sensor exposed (a.k.a. shutter speed). The effect an exposure has depends on the sensitivity of the film or sensor.
- *Shutter speed*: determines how long the film or sensor is exposed to light expressed in fraction of seconds. Normally this is achieved by a mechanical shutter between the lens and the film or sensor, which opens and closes for a time period determined by the shutter speed.
- *Aperture and f-number*: refers to the size of the opening in the lens that determines the amount of light falling onto the film or sensor. Because of basic optical principles, the absolute aperture sizes and diameters depend on the focal length. For instance, a 25mm aperture diameter on a 100mm lens has the same effect as a 50mm aperture diameter on a 200mm lens. If you divide the aperture diameter by the focal length, you will arrive at 1/4 in both cases, independent of the focal length. Expressing apertures as fractions of the focal length is more practical for photographers than using absolute aperture sizes. These "relative apertures" are called f-numbers or f-stops. On the lens barrel, the above 1/4 is written as f/4 or F4 or 1:4.
- *Depth of field*: is the distance from the nearest point of acceptably sharp focus (a.k.a. *near distance*) to the farthest point of acceptably sharp focus (a.k.a. *far distance*) of a scene being photographed. If lens is focussed at infinity, the far distance (and also depth of field) equals infinity; else the dept of field is equal to the difference between the far distance FD and near distance ND as given by Eq. 2. as

$$DoF = FD - ND \quad (\text{Eq. 2})$$

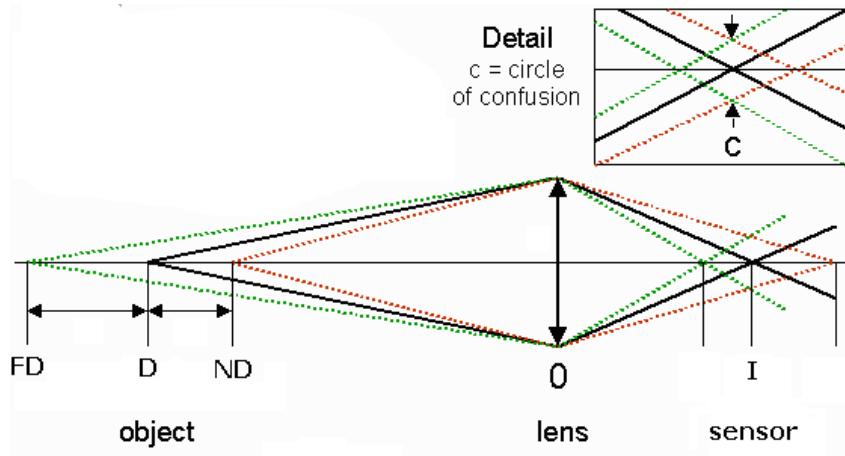


Figure 2 Schematic overview of lens optics.

The far distance FD and near distance ND (see Figure 2) both depend on the hyperfocal distance H and the subject distance D (a.k.a. object distance, shooting distance or view distance), i.e. Eq. 3 and Eq. 4 respectively

$$ND = \frac{H \times D}{H + D} \quad (\text{Eq. 3})$$

and

$$FD = \frac{H \times D}{H - D} \quad (\text{Eq. 4})$$

The hyperfocal distance is defined as distance from the optical centre of the lens to the nearest point in acceptably sharp focus when the lens, at a given f/stop , is focused at infinity. In other words, when a lens is focused at infinity, the distance from the lens beyond which all objects are rendered in acceptably sharp focus is the hyperfocal distance. The hyperfocal distance therefore depends on the f -number (or aperture), focal length F and the circle of confusion C and is given by Eq. 5, i.e.

$$H = \frac{F^2}{f \times C} \quad (\text{Eq. 5})$$

Circle of confusion, i.e. out-of-focus circle on the film with size: $C \geq 25.4 \mu\text{m}$. The permissible circle of confusion is dependent on film size; e.g. a 35mm sensor or film size permits a circle of confusion of $5.1 \mu\text{m}$ the cut-off point where we decide things are no longer sharp and is called the Maximum Permissible Circle of Confusion. Points with the circle of confusion smaller than the resolution of film are “in focus” (DPreview.com, 2007).

The subject distance is obtained from the Gaussian lens law which specifies the relationship between the subject distance D , i.e. distance from lens to pictured

object/subject; the image distance I , i.e. the distance from lens to image sensor; and focal length F ; the lens formula is given by Eq. 6, i.e.

$$\frac{1}{F} = \frac{1}{D} + \frac{1}{I} \quad (\text{Eq. 6})$$

and after re-arranging Eq. 6, the formula (Eq.7) for the subject distance becomes:

$$D = \frac{I * F}{(I - F)} \quad (\text{Eq. 7})$$

Therefore, the distance from lens to sensor I (a.k.a. bellows extension) need to be known very accurately which is the case with the camera manufacturers. Some camera devices automatically store subject distance to the image metadata; however using incompatible lenses cause erroneous distance to be computed using the lens law formulas. Furthermore, f-number, focal length and aperture are automatically saved to the image metadata at the moment of photo capture and are supported by many metadata formats to be automatically saved along with the digital photo.

2.1.2 Image storage and metadata formats

Image file formats provide a standardized method of organizing and storing image data. Image files are made up of either pixel or vector data, which is rasterized to pixels in the display process, with a few exceptions in vector graphic display. The pixels that comprise an image are in the form of a grid of columns and rows. Each of the pixels in an image stores digital numbers representing brightness and colour. Vector graphics (a.k.a. geometric modelling or object-oriented graphics) use geometrical primitives such as points, lines, curves, and polygons, all based upon mathematical equations to represent images in computer graphics. An example of such a vector image format is Scalable Vector Graphics, a special form of XML for describing two-dimensional vector graphics. SVG coding is either declarative or scripted, which results in static or animated graphics. (Cartwright, 2007)

Some commonly used raster image file formats, of relevance for this research, are e.g.

- *BMP file format*
BMP (pronounced: bitmap) sometimes referred to as device-independent bitmap (DIB) is a uncompressed image file format, able store pixels with colour depth of 1, 4, 8, 16, 24, or 32 bits per pixel. Images of 8 bits and fewer can be either greyscale or indexed colour. The main advantage of BMP files is their wide acceptance, simplicity, and use in Windows programs; a disadvantage is their large file sizes due to no compression.
- *Graphics Interchange Format (GIF)*
GIF is a lossless (LZW) data compression format for bitmapped image format, restricted to hold up to 256 colours. It is less suitable for digital photos as it reduces the visual quality if the digital image does not fit

within the palette of 256 colours. One colour could be assigned as transparent colour. (Cartwright, 2007)

- *Joint Photographic Experts Group (JPEG)*
JPEG (pronounced: JAY-peg) is one of the most commonly used methods for compression (lossy) of digital images. Particularly, JPEG is applied for digital photographs as it supports 24-bit colour depth (16,777,215 colours) and for exchanging images over the WWW as JPEG images are compressed at 10:1 to 20:1 (high quality; no noticeable loss), and even greater compression is possible, 30:1 to 100:1 (lower quality).
- *Portable Network Graphics (PNG)*
PNG (pronounced: ping) is an open, extensible bitmapped image format with lossless compression ("deflation"). It supports the storage of up to 16,777,215 colour values and is originally designed to improve and replace the GIF image format and for transferring images on the Internet, not professional graphics, and so does not support other colour spaces (e.g. CYMK, CIELAB). PNG supports 256 levels of variable transparency per pixel referred to as alpha blending. PNG images are therefore considered to have "32-bit colours", i.e. 8-bit red, 8-bit green, 8-bit blue and 8-bit alpha (transparency). Its architecture exists of critical (e.g. header and colour-palette) and ancillary (e.g. histogram and timestamp) information blocks (a.k.a. chunks), which hold information about the image itself. (Crispen.org, 2007).
- *RAW*
Digital cameras commonly produce three kinds of image files (some produce all three, most produce one or two): JPEG, TIFF, and "raw". Raw files represent the image just as it comes off the imaging chip in the camera, and they often contain more than 8 bits of data for each colour channel; some have 12 bits or 14 bits. Since raw files have more bits per channel, and since raw files are not compressed, raw file sizes are larger than e.g. uncompressed TIFF files (an "anti-compression" as big as 1:2).
- *Tagged Image File Format (TIFF)*
TIFF (for Tagged Image File Format) is invented by Aldus Corporation in 1986. It was created for maximum flexibility in exchanging data between machines and applications. It allows compression; however, most commonly it is used in uncompressed 32-bit format. It supports different colour systems as RGB and CMYK.

As the image file stores the image content as a collection of pixels organized in rows and columns, the information about the image file is stored in the image metadata, e.g. information about focal length, time of capture, camera device and so forth. Some relevant image formats are

- *Exchangeable image file (Exif)*
The Exchangeable image file format for digital still cameras specification is developed by Japan Electronics and Information Technology Industries Association (JEITA); current version EXIF 2.2. Numerous cameras and software support the EXIF file format to encapsulate image metadata to JPEG or TIFF images. Also compatible for sound image metadata to RIFF WAVE format; GPS data is stored within an additional GPS IFD. (Exif.org, 2007)
- *Digital Information Group 35 (DIG35)*
An image metadata standard specified by the Digital Information Group 35 (DIG35) Initiative Group of the International Imaging Industry Association (I3A). The DIG35 specification includes a "standard set of metadata for digital images which promotes interoperability and extensibility, as well as a "uniform underlying construct to support interoperability of metadata between various digital imaging devices." It supports capturing location (GPS, address), camera directions and object position inside the metadata file. (I3A.org, 2007)
- *Resource Description Framework / eXtensible Markup Language (RDF/XML)*
RDF/XML is a XML-based general purpose language/standard for representing information in the Web or to describe metadata for documents. In particular, it is stored within XHTML and HTML documents; however, RDF holds potentials to store metadata of any document format (including images). It supports customization, extension, easy transfer and conversion. RDF is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata model but which has come to be used as a general method of modelling information, through a variety of syntax formats (W3.org, 2007).
- *IPTC Metadata for XMP (IPTC4XMP)*
The International Press Telecommunication Council (IPTC) metadata format was employed by Adobe Systems Inc. to describe photos already in the early nineties. A subset of the IPTC "Information Interchange Model - IIM" was adopted as the well-known "IPTC Headers" for Adobe Photoshop, JPEG and TIFF image files which currently describe millions of professional digital photos. Extensible Metadata Platform (XMP) defines a metadata model that can be used with any defined set of metadata items. This XML-based standard is, as its name suggests, designed to be extensible, allowing users to add their own custom types of metadata into the XMP data. XMP supports location metadata as streets and city names; for GPS values the EXIF value is preferred (Adobe.com, 2007).

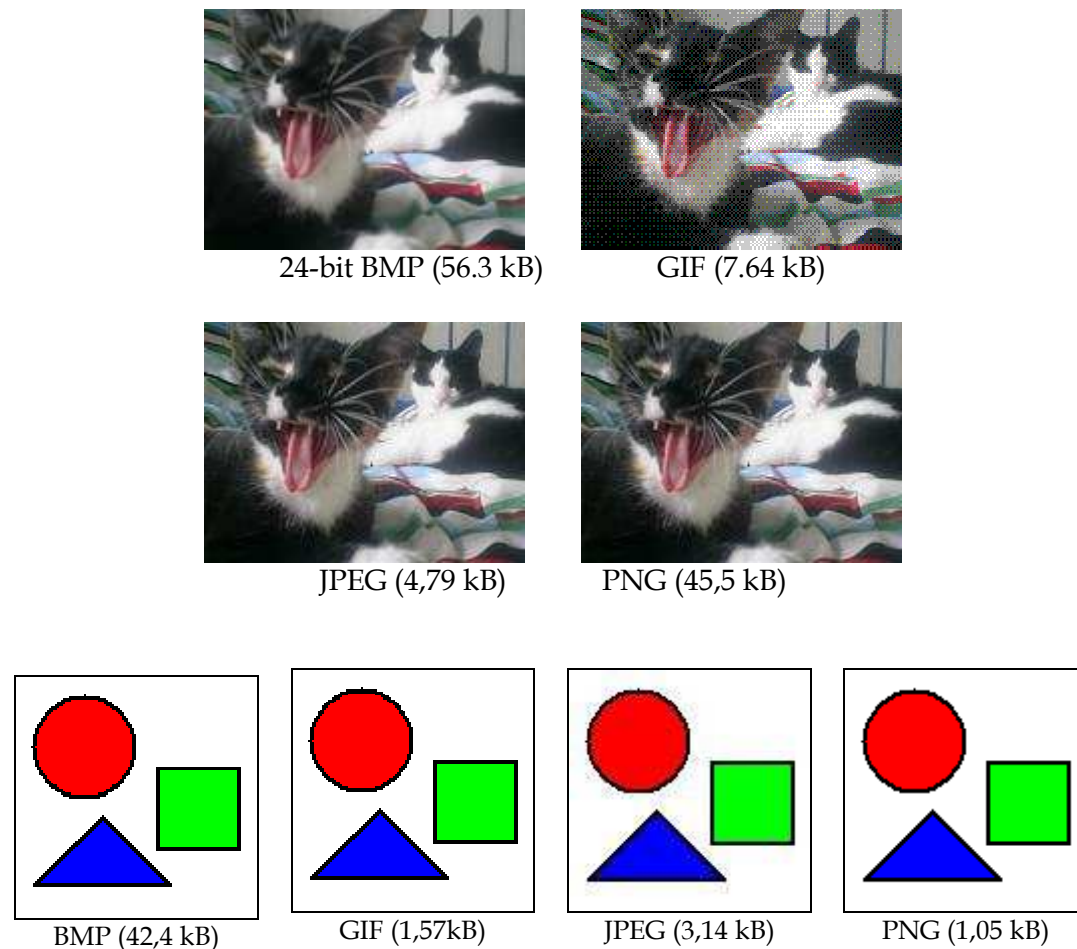


Figure 3 Examples of raster file formats BMP, GIF, JPEG and PNG and their filesizes.

struct		struct	
Aperture Value	F22	By-line	Creator
Color Space	sRGB	By-line Title	Creator: Job Title
Components Configuration	YCbCr	Caption/Abstract	Description
Compression	JPEG compression	City	City
Date/Time	2007:07:15 19:01:58	Copyright Notice	Copyright Notice
Date/Time Digitized	2007:07:15 19:01:58	Country/Primary Location	Country
Date/Time Original	2007:07:15 19:01:58	Credit	Provider
Exif Image Height	2336 pixels	Date Created	Mon Jan 01 00:00:00 PST 1990
Exif Image Width	3504 pixels	Directory Version	2
Exif Version	2.21	Headline	Headline
Exposure Bias Value	-2/3	Keywords	Keywords
Exposure Program	Shutter priority	Object Name	Title
Exposure Time	1/15 sec	Original Transmission Reference	Job Identified
F-Number	F22	Province/State	State/Province
Flash	Unknown (16)	Source	Source
FlashPix Version	1.00	Special Instructions	Instructions
Focal Length	132.0 mm	Unknown tag (0x025c)	Location
		Unknown tag (0x0264)	ISO
		Writer/Editor	Description writer

Table 2 Example of EXIF metadata (left) and IPTC headers (right) metadata format for a particular image.

- *ISO/IEC 15444-2*
The ISO/IEC 15444-2 specifies an XML/GML-based Document Type Definition (DTD). JPEG2000 adopted the ISO/IEC 15444-2 to encapsulate a complete list of metadata information; JPEG2000 further supports IPTC, Geography Markup Language data and sYCC colour space. Currently, the ISO/TC 211 Metadata programme is carried out to further standardization in the field of digital geographic information; including metadata standards for digital imagery and gridded data (ISOTC211.org, 2007).
- *GeoTIFF*
GeoTIFF represents an effort by over 160 different remote sensing, GIS, cartographic, and surveying related companies and organizations to establish a TIFF based interchange format for georeferenced raster imagery. GeoTIFF uses a small set of reserved TIFF tags to store a broad range of georeferencing information, catering to geographic as well as projected coordinate systems needs (GeoTIFF, 2007).

Table 2 shows an example of an Exif metadata format and IPTC metadata format respectively. Several image formats (e.g. Exif, Dig35, GeoTIFF) enable to store location or position of photo capture. Location specifies a position in terms of topological relations, e.g. street address, description, postal code; and the position specifies a set of coordinates with respect to a well-defined reference system acquired from e.g. a GPS device (e.g. WGS84, or local reference system).

Within this document, location refers more general to the place of image capture, e.g. place name in a photo tag, or set of coordinates in image metadata. According to Tayoma et al. (2003), this location is acquired either automatic or manual, i.e.

- *Manual entry*: users can georeference (also referred to as geotag) their images by navigating on and *drag-and-drop* images to a map, or *typing* coordinates or place name into a textbox (a.k.a. reverse-geocoding).
- *Automatic acquisition*: location metadata is acquired automatically from *location-aware devices* such as CellID technology, GPS, A-GPS or WLAN triangulation using timestamp matching, or *other sources*, e.g. synchronization with a calendar, extraction of place names from surrounding text in documents, association with other images and documents, or computer vision technologies.

The process of adding geographical identification (e.g. address or coordinates) metadata to multimedia is also referred to as *geotagging* (see Figure 4). Location metadata is used to geographically index digital photos, but also for photo annotation and retrieval purposes.

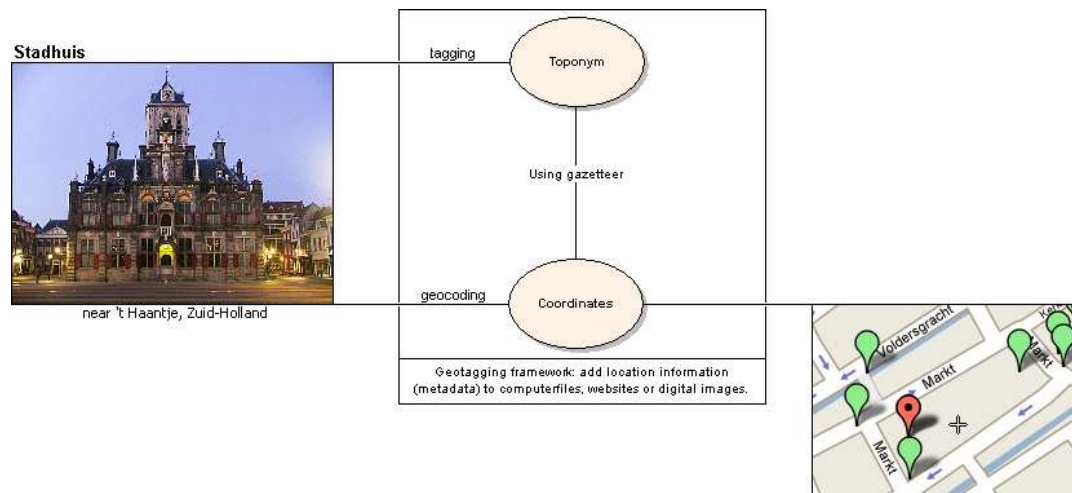


Figure 4 Geotagging of an image to georeference it on a map (Dias et al., 2007)

Some image metadata formats also enable to store *camera direction* (e.g. DIG35). For this research, the assumption is that upcoming camera (phone) technology enables not only to record position (GPS) but also camera direction parameters. *Direction* specifies the relative angle between one point with respect to another reference point (a.k.a. *orientation* e.g. North). Full camera direction refers to pan, tilt and roll (see Figure 5) and can be measured using a gyroscope (INS). *Camera orientation* is often used to denote to angle of image rotation.

Depending on the application field, direction parameters are denoted by

- *pan, yaw or heading* for the orientation around the z-axis,
- *pitch or tilt* for the orientation around y-axis.
- *roll* for the orientation around the x-axis.

Within this research, heading, pitch and roll are used to denote the different direction parameters.

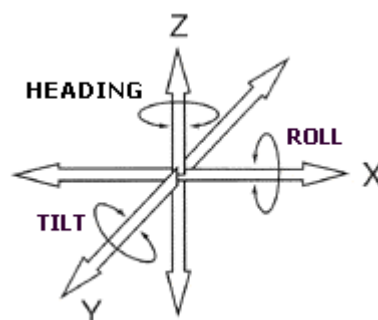


Figure 5 Camera direction parameters heading, tilt and roll.

2.1.3 Image annotation

Image or photo annotation is defined as the act of adding explanatory notes to a photo. Traditional annotation of photos includes describing the photo content using titles (i.e. distinguishing names), captions (i.e. textual descriptions), or

keywords (a.k.a. *tags* or *labels*). These annotation types are stored in the image metadata file and are usually visualized *above*, *under* or *next* to a photo.

Previous annotation research using multimedia content analysis suffered from assumptions as contextual metadata is not available and content analysis must be fully automatic and avoid user involvement (Naaman et al., 2004). Recent research developed image annotation techniques using some level of user interaction and available contextual metadata as user profile, date/time and location. Contextual metadata is among others used to filter a large collection of images or to add associated (geographical) names to an image caption. Some interesting examples of state-of-the-art image annotation tools using contextual (particularly location) metadata are among others:

- ***PhotoCompass*** is an annotation tool developed by the University of Stanford that organizes images in location and events. Using contextual metadata to query databases and search engine, an image is automatically annotated with additional semantic metadata e.g. weather condition, sunset and indoor/outdoor. *Geographical names* are associated with the image using reverse geocoding (gazetteer), and a ranked selection of neighbour cities using frequency of Google results. (Naaman et al., 2004).
- ***Mobile Multimedia Metadata (MMM)*** is a semi-automatic camera phone annotation system. After image capture, the user selects the image subject (Person, Location, Activity, Object) and in return a tag is suggest to the user after association the contextual information (user profile, location) with other photos in a collection. (Davis et al. 2004)
- Dublin City University's ***MediAssist*** uses contextual information to filter a photo collection to perform a content-based analysis on a subset when searching the collection. Includes content-based tools as Colour Layout, Edge Histogram and Texture Descriptor. The annotation approach is a combination of database querying and image association (O'Hare et al., 2005).
- ***EXTENT*** is a contEXT and contENT based annotation tool presented by the University of California at Santa Barbara. Contextual information as time, location and a user's social network first limits the search scope in an image database. SIFT feature-extraction methods are used to recognition of object and landmarks for image annotation (Chang et al., 2005).
- ***PhotoCopain*** is a semi-automatic, context/content-based annotation tool developed by the Universities of Southampton, Sheffield and Aberdeen. It enables to annotate personal photo collections using camera metadata, location, calendar synchronizing and feature extraction techniques. It includes face recognition; captions are suggested to the user (Tuffield et al. 2006).

From the previous, it can be concluded that current state-of-the-art annotation systems can be divided into the three following categories:

1. *Context-based versus content-based*, i.e. extract additional information from contextual metadata or using computer vision technologies for object recognition.
2. *Semi-automatic versus automatic*, i.e. image annotation with user involvement or without.
3. *Database/web search versus association*, i.e. image caption is suggested using database/web search results or associating with other photos in a collection.

However, previous state-of-the-art photo annotation tools annotate photos with information about surroundings and not about the object pictured itself. The lack of camera direction information causes that using GPS or Cell ID positioning only is not sufficient to accurately identify objects inside photos. In order to overcome this problem, O'Hare et al. (2005) use a context/content-based approach, i.e. first use the contextual metadata (particular location) to filter a collection of photos and secondly apply computer vision technology (e.g. edge detection and texture descriptors) to identify objects and associate. This approach was adopted for photo retrieval, but can also be applied to annotate a photo by associating photos based on location metadata in combination with computer vision technology. Similar content/context-based approaches have been applied to EXTENT (Chang et al. 2005) and PhotoCopain (Tuffield et al., 2006); photo association is also applied to Mobile Multimedia Metadata of Davis et al. (2004a).

According to Tayoma et al. (2003), people associate photos with *events*, i.e. something that occurs in a certain *place* during a particular interval of *time*. Naaman et al. (2004a) conclude from a user study that people recall and describe their photos by using categories as outdoor/indoor, people, *location* and events.

From own empirical work of analyzing a collection of 100 photographs at 5 professional image libraries is obtained that 91% of the photo annotations include a *country* name and 81% include a *city* name (see Figure 6). Furthermore, 62% of the photos are annotated with a subject name, i.e. specific name of building (e.g. Big Ben, Eiffel Tower) or other geographical feature (e.g. mountain or lake).

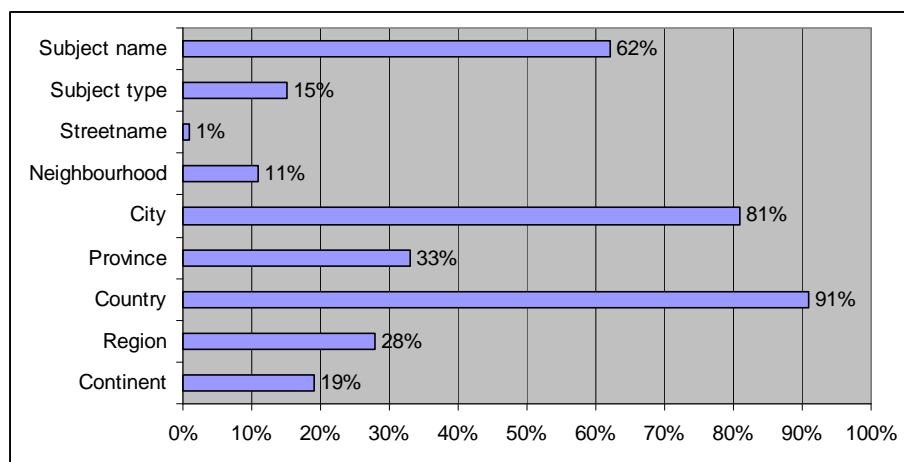


Figure 6 Spatial granularity of image annotation at 5 image libraries (N=100).

Naaman et al. (2004b) used location metadata to annotate photos with geographical names. Based on the photo location, a Google web search is carried out to associate place names with the photo and add geographical names to its caption. Using the photo location together with the timestamp of photos, additional information on the photo content -e.g. weather information, daily information (sunset/sunrise) - is (automatically) added to the photo by Naaman et al. (2004b) and O'Hare (2005). Also online image hosting sites, for example Panoramio.com and Locr.com, enable users to georeference their image (a.k.a. *geotagging*) and automatically add geographical names to the photo caption using the photo position. By doing so, the location of a photo forms the basis for photo annotation and retrieval.

2.1.4 Image retrieval

As observed by Castelli et al. (2002), photographic images are increasingly being acquired, stored and transmitted in digital format. Their applications range from personal use to media and advertising, education, art and even research in the humanities.

The organization and particular annotation and retrieval of photographic content is a true challenge. Retrieval of images suffers from a semantic gap, i.e. the discrepancy between the query a user ideally would and the one that he actually could submit to an information retrieval system. Therefore, visual content annotation should be automatic to enable content-based information retrieval (CBIR). CBIR tools could overcome this problem by applying computer vision technology and object recognition techniques; however, applying this to large image collections is a very computational task.

The goal of object recognition is to determine which objects are present in a scene and where they are located. Process involves the following steps, i.e. the image itself is analyzed, and features extracted from the image using edge operators and pattern recognition. Next, a decision has to be made on what the extracted feature most likely is. An approach to solve this is to match extracted features to a detailed model having for each object detailed geometric constraints (a.k.a. model-based CBIR). Disadvantage of using this approach is that objects could only be classified as the geometric constraints inside the model refer to a *type* of object and not to individual characteristics of objects. Spatial data sets do contain these specific characteristics for all most all real-world features; this research will carry out this object recognition using geographical analysis and spatial data sets.

2.2 3D Computer graphics

This section includes an overview of 3D computer graphics focusing on rendering, depth maps and earth visualization software.

2.2.1 3D Computer graphics definition

Computer graphics is a sub-field of computer science and is concerned with digitally synthesizing and manipulating visual content. Although the term often refers to three-dimensional computer graphics, it also encompasses two-

dimensional graphics and image processing. 3D computer graphics are graphics that utilize a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering 2D images. Apart from the rendered graphic, the 3D model is contained within the graphical data file. However, there are three differences, i.e.

- 3 A 3D model is the mathematical representation of any three-dimensional object. A model is not technically a graphic until it is visually displayed.
- 4 Due to 3D printing, 3D models are not confined to virtual space.
- 5 A model can be displayed visually as a two-dimensional image through a process called 3D rendering, or used in non-graphical computer simulations and calculations.

(Watt, 1990)

2.2.2 Rendering

Rendering is the process of producing two-dimensional pictures from three-dimensional data. Rendering exists of six major components (see Figure 7), i.e.

1. *Geometry*: the three-dimensional shape of the model
2. *Camera*: the point-of-view from which the scene is seen
3. *Lights*: the lightning of the scene
4. *Surface characteristics*: the definition of the surface characteristics of the model
5. *Shading algorithm*: the specific approach the system uses to calculate the shading properties of individual objects
6. *Rendering algorithm*: the algorithm used to render the whole picture

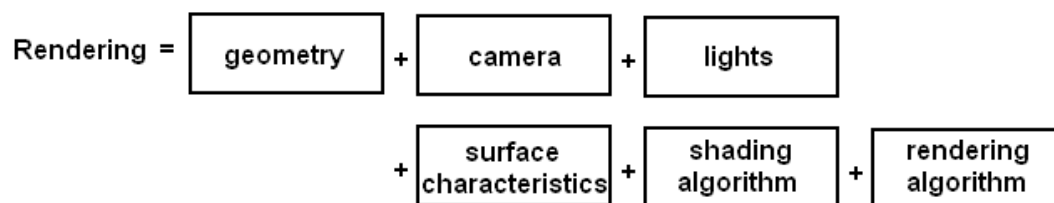


Figure 7 The six steps of the rendering process (O'Rourke, 2003)

O'Rourke (2003) distinguishes three different rendering approaches, i.e.

- The first approach is *see-through wire frame* rendering and is the simplest and fastest computer graphics rendering approach. It represents the object as it has no surfaces, but instead is composed from wire-like edges which enables fast real-time interaction. However; disadvantage is that objects are transparent.
- *Hidden-line rendering* is a rendering approach solves this problem by including the visibility of surfaces. It still represents objects as edges; however, some lines are hidden by the surface in front of them. As hidden-line rendering does not enable to visualize objects using colour and shininess of the surfaces, it provides no information about its characteristics.

- The third approach is the **shaded-surface rendering** (a.k.a. shaded rendering) which overcomes the limitations of both the see-through wire frame and hidden-line rendering; but complex and high-quality shaded-renderings may take a long time to compute.

2.2.3 Depth maps

The pre-sorting of objects by depth can be very time-consuming particularly when the scene involves a lot of objects, as each polygon/ surface per object needs to be sorted by depth. An alternative and very common solution for this problem is known as the Z-buffer algorithm (O'Rourke, 2003). The z-buffer algorithm stores information on a pixel-by-pixel base about the distance of the surface to the camera or observer point, when a computer graphics picture is rendered; similar to frame buffer that stores in a memory block the colour information for a picture.

A z-buffer stores depth information for a scene by storing a number at each pixel. The brightness of the pixel corresponds to the depth of the surface. The z-buffer algorithm fills the z-buffer as follows:

- if a ray cast through pixel one of scan line one hits some object, the depth value of that object is stored in pixel one, scan line one of the z-buffer;
- if the same ray encounters a second object it calculates the depth value again and compares it to the current depth value as stored in the z-buffer;
- if the new value is less (i.e. closer) than the old value, the new value overwrites the older more distant value in that pixel.

As such the renderer is able to determine which object is front of another object; and this information saved to a depth map as shown by Figure 8.



Figure 8 Example of a digital photo and its depth image (Dofpro.com, 2007).

2.2.4 Perspective viewers

Three-dimensional earth viewer (a.k.a. perspective viewer services, scene generators or terrain servers) render three-dimensional geometry models in a perspective view and visualize it as a two-dimensional graphic on e.g. a computer screen. Where the human eye has a cone of vision, computer graphics have a pyramid of vision, and as such the output of perspective viewer services is an rectangular image. Some examples of perspective viewers are:

- **ESRI ArcScene**, a desktop application part of the ESRI ArcGIS 9 for visualizing spatial data sets (vector and raster) in a perspective view. It enables to change the three-dimensional view by defining camera position, roll, angle and field-of-view angles and uses the ArcGIS 3D Analyst extension, which is also capable of performing three-dimensional visibility analyses. (ESRI Desktop Help, 2006)
- **Google Earth** is a three-dimensional earth viewer, enables to navigate through of virtual environment build up from high-resolution satellite and aerial images. It is public accessible and due to its smooth interface one of the most popular earth viewers. Other example of public accessible earth viewers are MSN Virtual Earth 3D and NASA World Wind. (Asperen et al., 2007)
- **OGC Web Perspective Viewer Service (WPVS)** is a Java-based web service which is able to generate perspective terrain views rendered from a requested viewpoint. It is currently waiting for approval to become a true OGC-standard to replaces the former OGC Web Terrain Service (WTS). A WPVS displays vector based and raster data from different storage formats (e.g. WMS and WFS) and returns an image to the client after performing a HTTP GET request (Deegree.org, 2007).

2.3 Label placement

The cartographic label placement is an important task in automated cartography and Geographical Information Systems (GISs). Cartography is described as the graphic principles supporting the art, science, and techniques used in making maps or charts. A Geographical Information System (GIS) is a defined as a computer-based system to store, manage, analyze and visualize geographic-related information (Worboys et al., 2004)

2.3.1 Cartographic label placement

The process of text insertion in maps is referred to as label placement. Label placement is one of the most difficult tasks in automated cartography as according to Yamamoto et al. (2005) positioning text requires

- that overlap among texts be avoided,
- that cartographic conventions and preferences be obeyed,
- that unambiguous association be achieved between each text and its corresponding feature and
- that a high level of harmony and quality be achieved.

Good placement of labels avoids as much as possible overlap of labels with objects and mutual labels; and is applied to provide additional information about a particular feature. *Automatic* label placement is therefore one of the most challenging problems in GIS due to

1. optimal labelling algorithms are very computational expensive for interactive systems, and
2. labels compete with data objects for the same limited space.

(Li et al, 1998)

The problem of optimal label placement is NP-hard (Christensen et al., 1992). NP-hard is defined as the complexity class of decision problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine *in* polynomial time. When a decision version of a combinatorial optimization problem is proved to belong to the class of NP-complete problems, then the optimization version is NP-hard (Nist.gov, 2007). This implies that if the number of possible positions grows exponentially with the number of items to be labelled: having L possible label positions around an object and having n objects then there are L^n possible label positions (for example $L = 8$ and $n = 10$ gives already 1×10^9 possible label positions).

2.3.2 Multimedia cartography

Multimedia Cartography is based on the compelling notion that combining maps with other media (text, pictures, video, etc.) will lead to more realistic representations of the world (Cartwright, 2007). Multimedia is the combined use of several media, as sound full motion video in computer applications. It allows human-computer interaction involving text, graphics, voice and video. Multimedia often also includes concepts from hypertext. Within this document, multimedia denotes photos, videos and web documents (HTML sites).

Augmented reality (AR), considered to be part of the Multimedia Cartography research field, is an environment that includes both virtual reality and real-world elements and forms part of the research. AR is a field of computer research, which deals with the combination of real world, and computer generated data. Its principle is to overlay the real world (captured using a camera device) with supplementary information (e.g. labels) from computer or Internet. It enables users to interact with their environment e.g. by hyperlinking labels inside a AR view. According to Cartwright (2007), interactivity is one of the key-components of multimedia.

Photo labelling forms part of photo annotation and it refers to the act of placing labels that describe the features visible inside the photograph itself. Saving the labels obtained from the virtual scene to a transparent layer enables to put labels associated with an object onto an image. As such, the photo annotation issue is considered to be part of Multimedia Cartography and AR as well; visible tags in images and AR applications enables user interaction with the environment; numerous ubiquitous and/or augmented reality applications are discussed by Kolbe (2007), Toye et al. (2006) and Schmalstieg et al. (2007).

2.3.3 Label placement in three-dimensional scenes

As Li et al. (1998) observe, object and label placement in limited screen spaces is a challenging problem in information visualization systems. Images also have a limited screen space and therefore (particularly automatic) label placement is of concern for this research in order to avoid overlap of labels mutual and labels with objects.

Numerous research already examined the problem of automatic label placement in 2D maps and recent work of Maass et al. (2006), Azuma et al. (2003) and Götzelman et al. (2006) also focused on the placement of labels in 3D landscapes and Augmented Reality views referred to as *view management* (Bell et al., 2001). Azuma et al. (2003) describe four algorithms for AR label placement, i.e.

- 2 *greedy depth first placement*: labels each object in some order e.g. based on priority;
- 3 *discrete gradient descent*: finds the location around an object that is furthest away from all other labels;
- 4 *simulated annealing* tends to find the zero cost arrangement for every placement;
- 5 *cluster-based method* finds clusters of objects that mutual overlap and the algorithms searches for label placement solutions per cluster.

Götzelman et al. (2006) offer complex label layouts which integrates internal and external labels of arbitrary size and shape, and real-time algorithms. Maass and Döllner (2006) describe two point-feature dynamic annotation placement strategies for virtual landscapes including priority aspects.

Labeling is further divided into internal and external (object) annotation (Maass et al., 2006). An internal annotation is drawn inside the visual representation of the referenced object and partially obscures that object. An external annotation is drawn outside the visual representation of the reference object and uses a connecting element such as a line or arc to associate the annotation with the reference object. Maas et al. (2006) propose two algorithms for dynamic annotation placement in virtual environments, i.e. growing-border view management and interval slots view management. The first algorithm places labels stacked which does not allow to place another label on the same line again; the interval slots algorithm determines if there is no collision before placing a label (see Figure 9).

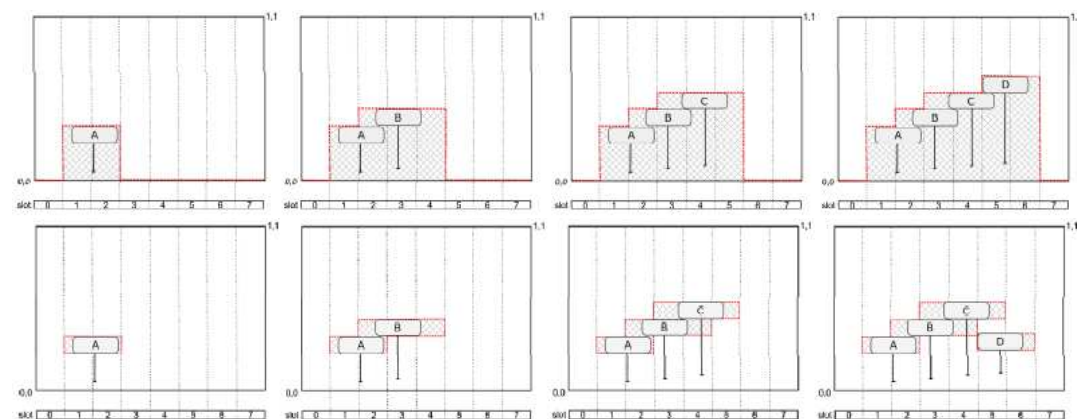


Figure 9 Growing-border (up) and interval slot (down) view management (Maass, 2006).

Maas et al. (2006a) extend their annotation algorithm for virtual environments by using object-integrated billboards. This enables to attach labels on building taken into account the observer position. It uses two generalized variants of the annotated object, i.e.

1. a hull which is a simplification of a three-dimensional
2. and skeleton which represent the internal supporting structure of a three-dimensional object.

For simple objects the annotation algorithm casts a ray to the centre of the hull and the location where the ray intersects the boundary of the hull is chosen as an affixation point of the label; for complex objects, the affixation point is chosen as the minimal distance from the skeleton to the ray in the centre from observer's view.

Hagedorn et al. (2007) describe the use of a web perspective viewer service (WPVS) for the annotation of three-dimensional geographical environments (a.k.a. geo-environments). Furthermore, a three-dimensional Web View Annotation Service (3D WVAS) is proposed as an extension to a WPVS. The perspective view together with a depth image is forwarded to the 3D WVAS together with annotation definitions. This annotation technique calculates the positions of the labels, renders them into a separate image buffer, and combines the resulting image in a depth-sensitive way with the input colour image (see Figure 10).

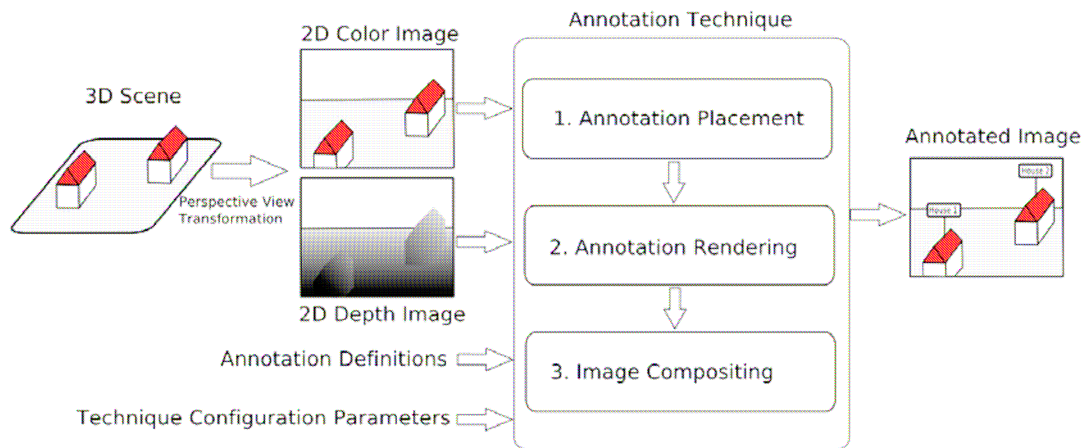


Figure 10 Process for the annotation of 3D scenes as proposed by Hagedorn (2007).

2.4 Related research applications

This section includes some examples of research applications within the framework of this research.

2.4.1 Cyclomedia panoramic imaging

Cyclomedia is a Dutch company that specializes in large-scale and systematic visualization of environments using 360° panoramic images (a.k.a. cycloramas). The company captures 360° panoramic images of all building within the Netherlands using a car equipped with a special camera having a very wide angle view (240°) and that is capable of 360° rotation (Cyclomedia, 2007).

In former days, the positioning of these panoramic images was carried out by manual georeferencing it on a map. An observer indicated its start position on a map and next every few meters a panoramic image was captured and matches

with the car odometer. As such, a positional accuracy was achieved of approx. 1.0 meter.

Current capture system, records GPS position, camera direction and time at each camera observation point. Using NovAtel's SPAN™ GPS/INS Technology, a 0.1 to 0.5 meter positional accuracy is achieved. The camera direction is measured with an accuracy of $\sim 0.1^\circ$ using a very sensitive Inertial Measurement Unit (IMU), i.e. a 3D measurements unit sensing its own rate and direction of motion using a combination of accelerometers and gyroscopes.

For the spatial metadata, Cyclomedia developed its own standard (.CMI) having the advantage to store all information about the capture position and different coordinate reference systems (CRS). For each pixel of a cyclorama, the pixel coordinates are converted into terrain coordinates using camera calibration and collinearity equations from close-range photogrammetry. The cycloramas make many applications possible e.g. application field as valuation of real state, taxation, municipal city management, virtual tourism and other visualization applications (see Figure 11).



Figure 11 Panoramic imaging of Cyclomedia and one of its applications: projection of cables and pipelines in a cyclorama.

Previous research of Scotta (1998) applied cycloramas for object identification using two approach; both using digital maps to identify the object (e.g. name, building, address). The first approach uses two cycloramas captured from different position and loaded in two screen windows. After setting both windows centred at the object-of-interest, first the actual view direction is calculated and secondly, the position of this object is determined using a forward intersection. Second approach uses one cyclorama and after placing the object-of-interest in the middle of screen window, a objects are selected that lie along the line-of-sight of the related view direction; the object that is closest to the capture location is most likely the object-of-interest; however; this turns out not to be for all cases.

2.4.2 Ordnance Survey pointing device Zapper

Pointing device Zapper, developed by the Ordnance Survey (OS; the mapping agency of the UK), allows pointing a personal digital assistant (PDA) at any building in the country and, at the press of a button, receiving information about that building (see Figure 12). This pointing application does not only enable to ask where a particular building is located, it also enables to ask what kind of building it is by adding direction to a conventional location-based query.



Figure 12 Ordnance Survey pointing device Zapper.

The PDA is equipped with a positioning chip, digital compass and a wireless connection. The positioning is carried out using GPS Real Time Kinematics (RTK) satellite positioning, which enables a positional accuracy on a (sub)meter-level. The digital compass enables to measure the direction it is pointed to with an accuracy of approx. 0.5° , which turns out to be insufficient accurate causing misidentification of objects in densely populated areas.

When a big button is pressed on the software's interface, the location and the direction that is aimed the PDA at is sent to a remote server at head office. The server processes the data, first cutting out a 100 m square of OS MasterMap, i.e. high-detailed polygon map of the UK, around the reported location and then 'drawing' a line along the reported bearing from the compass. As it draws this line, it queries each polygon the line intersects along the way, searching for one classified as a BUILDING. When it finds one that meets this criterion, it stops and looks up the topographic object identifier (TOID) associated with the polygon. Once found, the server then runs off to a database to fetch whatever information is attached to that particular TOID and sends it back to the PDA.

Pointing device Zapper is currently being transferred to private companies for further research and development.

Another interesting Ordnance Survey research application is within the field of Augmented Reality (AR). It combines OS computer technology, GPS-RTK, direction sensors and 3D rendering software, enabling to place virtual objects (e.g. names of buildings and streets) in a real-world image with reasonable accuracy; see Figure 13 (OS, 2007).



Figure 13 Augmented Reality research projects by Ordnance Survey.

2.4.3 Nokia Mobile Augmented Reality applications

The Research Centre of Nokia, a Finnish multinational communications cooperation Nokia focussing on wired and wireless telecommunications located in Helsinki, Finland, works on the Mobile Augmented Reality Applications (MARA) project. The research team added a GPS sensor, a compass and accelerometers to a

Nokia smart phone (see Figure 14) to calculate the building's location and uses that information to identify it. Each time the phone changes location, it retrieves the names and geographical coordinates of nearby landmarks from an external database.



Figure 14 A Nokia 6680 mobile device, communicating via a Bluetooth interface with the add-on box containing GPS and sensors, enables to obtain additional information from the WWW of the building/object aimed at.

If the absolute location and orientation of a camera phone is known, along with the properties of the lens, it is possible to determine exactly what parts of the scene are viewed by the camera. MARA uses this information to annotate an image with labels of the object aimed at. The user can then download additional information from the WWW, as URLs are associated with virtual objects, allowing for *hyperlinking* of real world objects; a click on this virtual object about a chosen location returns information about it. Another application is a friend-finder service, that shows the where a second user is located at least as his or her position is also known from GPS positioning and the location uploaded to a server. The prototype of the MARA project also included *automatic switching to map-view* when the user holds the phone horizontally. This displays the users' position on a map of the area, and highlights nearby virtual objects (NRC, 2007).

MARA includes also the use of object-recognition algorithm - i.e. matching objects in an image with a rough model of the object - to improve the object identification. By focussing on *real-time* image-recognition algorithms, the Nokia research team explores the possibility to eliminate the need for location sensors and improve the system's accuracy and reliability

2.4.4 Microsoft PhotoSynth project

Photosynth is a software technology project from Microsoft Live Labs and the University of Washington that takes a large collection of photos of a place or an object, analyzes them for similarities, and then displays the photos in a reconstructed three-dimensional space.

Photosynth works by analysing multiple photographs taken of the same area. Each photo is processed by computer vision algorithms to extract hundreds of distinctive features, like the corner of a window frame or a door handle. Photos that share features are then linked together in a web. Pattern recognition

components compare portions of images to create points; when the same feature is found in multiple images, its 3D position is calculated to convert the image into a model. Photosynth's 3D model is a cloud of points showing where these features are in space (see Figure 15).

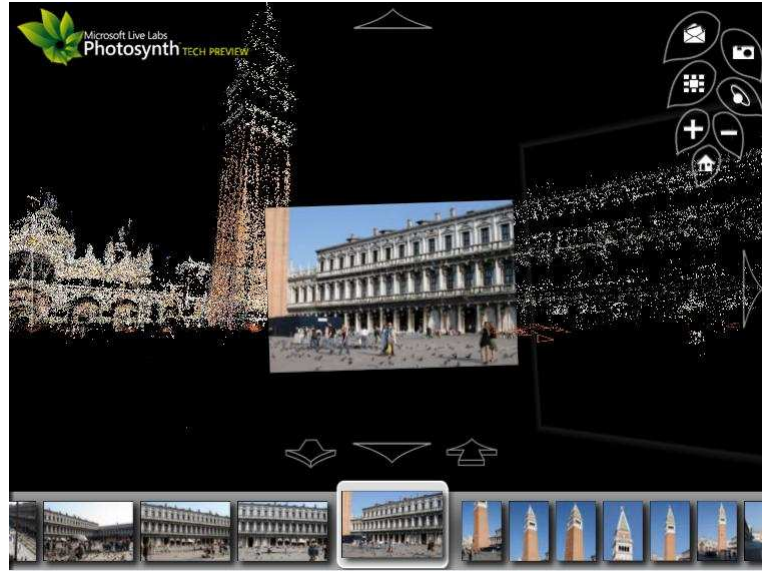


Figure 15 Microsoft PhotoSynth 3D model is a point cloud with digital photos connected to it. (Microsoft, 2007).

This model enables the Photosynth program to show a particular area from various angles, based on the different angles found in the photos. While the process works when only two photographs are used, it is better with more.

2.4.5 Quakr project

The Quakr project ideology is to “build the world one photo at a time”. Where the Microsoft Photosynth creates photo-realistic cities, ‘built’ from especially commissioned photos, the Quakr project tries to build a three-dimensional model from digital photos uploaded by average people and amateur photographers. In general, the project has three objectives, i.e.

1. Construct a three-dimensional from digital photos of average users;
2. Develop a tool (Taggr) that support users to tag their photo with spatial identifiers and additional keywords;
3. Develop a service (Viewr) to navigate around photo in a three-dimensional environment.

As it is the Quakr objective to let average users upload images with full spatial metadata, the project contributors build a very simple 7D tiltometer, i.e. a camera with compass and protractors attached to it using glue and stitches. Together with a GPS data logger, it enables to measure 7D spatial metadata parameters: timestamp, latitude, longitude, altitude, heading, pitch and roll. For more information about this project is referred to their web site, i.e. <http://www.quakr.co.uk/>.

Chapter 3

Research methodology

This chapter describes how the research is carried out. First, section 3.1 describes the background and framework of this research, followed by the problem definition in section 3.2. Section 3.3 provides the main question defined for this research; section 3.4 describes how the research is split up in different assignments and some approaches to solve the research assignment. The outputs of this research are summarized in section 3.5.

3.1 Background

This research is carried out at Geodan Systems and Research, the research department of the geo-ICT company Geodan located in The Netherlands (Amsterdam, Den Bosch), Spain (Madrid) and Austria (Salzburg). Geodan's focus is on developing (location-based) services and acting as spatial data broker for the public and private sector. Together with five universities (Sheffield, Zurich, Dublin, Bamberg, Cardiff) and four private partners (Ordnance Survey, Centrica, Alinari, Tilde), it contributes the European Union research project Tripod. The Tripod research objective is to "improve the access to the enormous body of visual media' on the WWW (Tripod, 2006), by developing new tools for annotation and searching online multimedia databases. Visual media includes both videos and photos; however the main focus of the Tripod research project is on (online) digital photos. This research is part of the Tripod research project.

The increasing availability of camera devices, due to integration of cameras in mobile phones and decreasing consumer prices, demands for new tools for managing this enormous amount of digital photos. On the one hand this is achieved by developing new tools for searching online collections, e.g. *geographical search*. According to Tayoma (2003), people associate digital photos with events, where an event is something that occurs at a certain location and time. From the Tripod user requirement scoping, it is obtained that 30% of the users search for images using geographical names (a.k.a. toponyms) in their search string, but only 20% of the online image collections enable users to geographically search (Dias et al., 2007). On the other hand, new tools for annotation result in qualitative and uniform photo descriptions that enable fast and reliable retrieval. From the user requirement scoping, it is obtained that 10% of photographers do not annotate

their photos because they find it a too time-consuming task. 60% of the users do annotate pictures with geographical names and 30 to 44% of the users search for online photos using a toponym in their search string. Automatic annotation tools for enhancing multimedia with qualitative and uniform textual descriptions are expected to become one of the main outputs of the Tripod research project in order to bridge the semantic gap.

The combination of positioning devices (GPS) and camera devices (internal positioning chip or wired/wireless connected) enable to (automatically) capture the observer's position along with the digital photo. Also several online image sites (e.g. Flickr, Panoramio, Locr) allow users to georeference their photos on the map (either via drag-and-drop or typing street name in a textbox). This process of adding geographical identification (e.g. address or coordinates) metadata to multimedia is referred to as geotagging and it is used to geographically index the digital photo and to associate (tag) it with geographical names (a.k.a. toponyms). Disadvantage is that using position only; the digital photo can only be annotated with information about the surroundings of the observer point and with information about the actual pictured scene.

The assumption of the Tripod research project is that in the near future there will be camera devices available having a positioning and direction measurement (e.g. compass or gyro) devices integrated. This allows not only to exactly knowing from where the photo is taken, but it also enables to capture the view direction which does enable to exactly identify the pictured scene and objects (irrespective of inaccuracies in position and view angles). As such it is possible to annotate photos with the objects captured, to add full textual descriptions or to attach labels to the visible objects inside the digital photos.

3.2 Problem definition and relevance of research

As it is very hard for computers to recognize objects inside digital photos using computer vision technology only (Naaman et al., 2004), the combination of full spatial metadata (i.e. position, orientation and camera settings saved to the photo) and GIS spatial datasets enable to identify pictured objects. In order to label objects inside a photo, also its position in the pixel coordinate system needs to be known. One approach to achieve this is to convert pixel coordinates to terrain coordinates using camera calibration and close-range photogrammetry equations. However, this research aims to use GIS data and perspective view generator's output to identify and localize visible objects. Basic idea is that by creating a perspective view from GIS data based on the full spatial metadata, a virtual abstraction is obtained that matches the picture scene inside the digital photo. Linking this virtual abstraction to the original 2D spatial data sets enables to pick the labels from this data set and place the labels inside the digital photos. As the virtual abstraction is actually a 2D representation of a 3D scene, the problem of placing labels inside a 3D Geo-Environment is reduced from a 3D problem to a 2D problem, in general enabling to apply any 2D label algorithm for label placement inside or next to picture objects.

Relevance of this research is that it is expected that research outputs contribute to provide a solution for the following issues:

- *Label placement in 3D Geo-environments:* the labelling of 3D scenes (a.k.a. 3D Geo-environments) is a problem concerning a lot of placement conflicts with respect to the best location to place a label. In reducing the problem from 3D to a 2D problem, this research is expected to contribute to a solution of placing labels inside 3D Geo-environments.
- *Improve retrieval of online photos:* qualitative photo annotation improves the retrieval of online digital photos. As photo annotation is considered to be a time-consuming and annoying task, photographers omit to add qualitative descriptions to their photos. Automatic annotation tools can solve this problem and improve the retrieval and access to online images by adding consistent and qualitative textual description to photos; this research aims to add labels to digital photos using GIS technology and GIS data sets in an automatic way.
- *Improve the understanding of image content:* an image is actually just a collection of pixels with colour values organized in a rectangular grid. For computers, it is very difficult to understand an image as it appears to a computer as just a collection of pixels with values, and as such it is for a computer to identify its (semantic) content. This research aims to identify what is pictured and where it is located; in order to place labels next to visible objects to explain what is actually captured inside the photo. These labels provide information to users to enable them to understand the image content. Besides this approach it is expected to make it possible to develop new services e.g. for visually handicapped people to read the image content (using e.g. sounds or large tags when a user moves its mouse over the photo objects).

3.3 Research question and objectives

This research focus is on using GIS technology and spatial data sets for the label placement inside 3D georeferenced and oriented digital photos. It applies perspective view generator's output for the identification of object captured inside a scene, and secondly the back-linking of the visible objects inside this virtual abstraction with the original features of the 2D spatial datasets enables to place geographical names next to it.

Therefore, the goal of this master's thesis research is to label objects captured inside digital photos. The main research question is therefore

How to identify captured objects and where to place labels to annotate pictured objects inside 3D georeferenced and oriented digital photos using GIS technology?

This research question is split up in two research sub question concerning to be distinct research parts, i.e.

1. How to identify which objects are visible and where these visible objects are located inside a digital photo using GIS data and perspective view generators output?

2. Which rules and constraints should be added to the label engine or algorithm in order to determine the best place for a label to associate it with a visible object?

In addition to the research questions, it is noticed that

- For the identification and localization of pictured objects, no computer vision technology and image recognition operators are applied; this research aims to apply GIS data sets and GIS technology to attain its objectives.
- The best place for a label is considered to be as such that there is no overlap between labels mutual or labels with visible objects; labels should be placed in empty areas, i.e. areas where there exist no objects in the virtual abstraction and where pixels of the digital photo have a particular colour value that is assumed to be an empty area, i.e. light (or pale or white) areas of pixels having a high colour value.
- This research is limited to be applied for outdoor digital photos of buildings, streets and rivers. Reasons for this are that satellite positioning is not possible inside buildings, and the GIS data availability (i.e. no indoor maps).
- Current image metadata formats do not support the storage of view direction attributes along with a particular image; however, it assumed that the integration of camera devices with digital compasses will encourage camera manufacturers and standards organizations to develop a metadata standard that does allow this.

The objectives are to

- identify and localize visible objects inside digital photos using GIS perspective view generator's output;
- search for the best position for labels to annotate visible objects using 2D GIS label algorithms;
- determine the effects of lens distortions, position and orientation inaccuracies on the (mis)placement of labels;
- propose an system architecture in order to make the photo labelling service available to the rest of the world.

3.4 Approach

After a thorough study of requirements, state-of-practice and literature, this research continues with developing a method to place labels inside digital photos. Therefore, first a set of 3D georeferenced and oriented digital photos is collected to be used as sample data. As study area is chosen for the Market Square in the city

centre of Delft, The Netherlands, due to the numerous unique historic buildings of monumental nature and the GIS data availability. After processing the collected photos by adding the full spatial metadata to it, a spatial data set is prepared from the combination of the available GIS data sets, to serve as input of the perspective view generation. This research describes three sorts of extrusion models that created, of which the best is chosen to generate the output scenes.

Next, this process of creating the virtual abstraction is described including a solution to link visible objects in the virtual abstraction to the features of the 2D data set. It is carried out as a proof-of-concept using ESRI ArcGIS 9.2, specifically ESRI ArcScene (for visualizing the extrusion model in a perspective view) and ArcMap (for applying 2D GIS function to link the virtual abstraction with original prepared data set). The following step concerns to find the best location to place a label; constraints and rules are and applied to the digital photos using the ESRI Maplex Label Engine. Pictured objects will be annotated differently depending on among others their size, priority and distance from the observer points. As the result of the object identification and label placement depend on the level of distortions of lenses and inaccuracies of positioning and orientation, the effects of (mis)placement and (mis)identification are worked out.

Finally, some attention is given to how this proof-of-concept should be implemented and made available to the rest of the world. Besides some automation in ESRI ArcGIS using forms and creating functions in Visual Basic for Applications, also system architecture for a photo labelling service is proposed.

3.5 Research outputs

Summarizing the previous sections, the outputs of this research are:

- A unique collection of the 3D georeferenced and oriented digital photos
- An extrusion models, specifically designed for this research; appropriate enough for creating the virtual abstraction matching as good as possible the image content.
- An algorithm on how to identify and localize visible objects inside 3D georeferenced and oriented digital photos using GIS data sets and perspective viewer services.
- A set of constraints and rules that are to be applied to the label algorithm in order to associate labels with visible objects inside photos and place the labels at the best possible place.
- An demo-application in ESRI ArcGIS which demonstrates the process of the object identification and label placement using ESRI ArcScene en ESRI ArcMap.
- A system architecture proposal for implementation of a photo labelling service.

Chapter 4

Collection of 3D georeferenced and oriented digital photographs

This chapter describes the collection of georeferenced and oriented digital photographs. The photos are acquired to serve as sample data to evaluate the object identification and label placement results. Section 4.1 describes the collection of digital photos having high spatial accuracy and low image resolution using a Topcon imaging total station and section 4.2 describes the collection of digital photos having low spatial accuracy and high image resolution. Section 4.3 discusses the results.

4.1 High-spatial accuracy and low-resolution digital photos

4.1.1 Topcon imaging total station

The GPT-7000i series of Topcon imaging total stations (for this research a GPT-7003i is used) enables to capture images along with the distance and direction measurements. The photos have a resolution of 640x480 (0.3 MP) and taken at a focal length of 30mm (wide-angle view) or 248.46 mm (tele-view) with a view angle of 30° and 1° respectively. It measures horizontal and vertical directions with an accuracy of up to 1 mgon (i.e. 400 gon = 360 degrees); non-prism distances are measured with an accuracy of 3~10 mm. The Topcon GPT7000i is able to calculate the coordinates of the observation points using the TopSURV on-board software. It connects directly the observation points to a national system by measuring to known control points, or obtained coordinates are given in a local system. The data can be exported as CAD DGN, ESRI Shapefile, RAW data; the images are saved as JPEG (Topcon, 2007).

4.1.2 Measurement setup

The measurements are collected at the Markt in Delft from two occupation points, i.e. one point in front of the South-corner of the Nieuwe Kerk (point 1000) and one point at the South-East corner of the market square near the historic municipal building of Delft (point 2000). From the two occupation points, several digital photos are collected of e.g. the Municipal Building, Hugo de Groot statue and 'Het Blauwe Hart' artwork, using the imaging capability of the Topcon GPT-7000i. (see

Table 3.). Together with the images, also horizontal and vertical direction (resp. HA and VA) and distances (SD) are measured in a local system (see Figure 16).

Table 3 Images captured at occupation point 1000.



1010 – top00002.jpg



1011 – top00003.jpg



1013 – top00004.jpg



1014 – top00005.jpg



1018 – top00006.jpg



1019 – top00007.jpg

Table 4 Images captured at occupation point 2000.



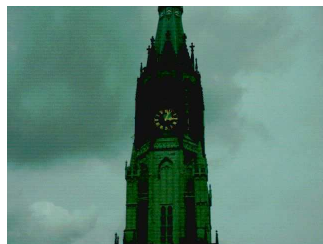
1010 – top00002.jpg



1011 – top00003.jpg



1013 – top00004.jpg



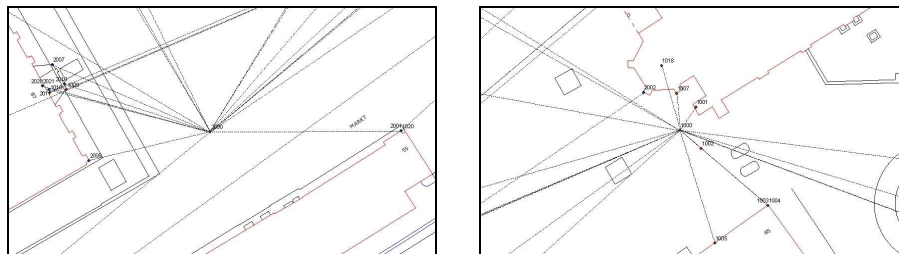
1014 – top00005.jpg



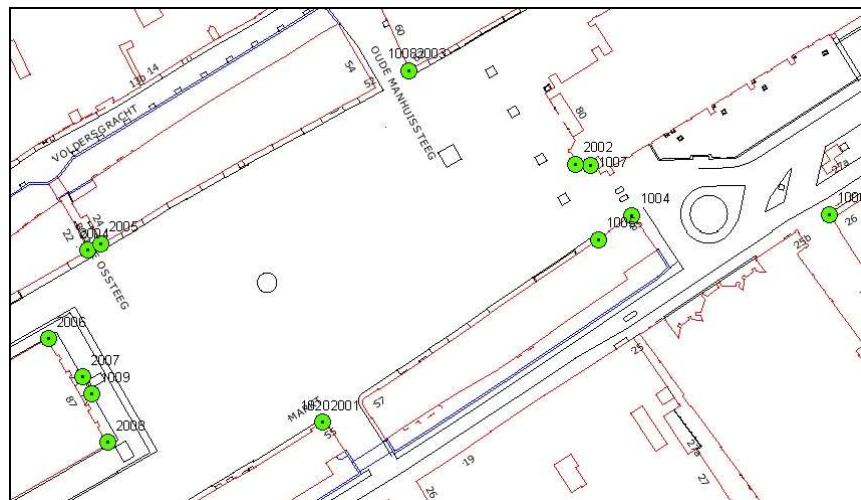
1018 – top00006.jpg



1019 – top00007.jpg



To connect the local point coordinates to the Dutch RD system, corners of buildings are used as control points. Coordinates of control points are extracted from the Large Scale Base Map of the Netherlands (GBKN; see Figure 17). Because GBKN only provides planimetric coordinates, the height information is obtained from a NAP-pin, located in the buttress of the Nieuwe Kerk.



37

4.2.1 Processing of observations

After collecting the measurements and images, the data is adjusted using 'Spatial Adjustment' of ESRI ArcGIS 9.0. Spatial adjustment enables to perform a coordinate transformation (affine, projective or similarity) from one system (*source* e.g. local system) to another system (*target* e.g. RD system). The Topcon GPT-7003i imaging total station calculates the local coordinates from distances and direction and it supports to export these coordinates to an ESRI Shapefile. These local coordinates are linked to the GKBN control points using the Displacement Link function in order to estimate the transformation parameters.

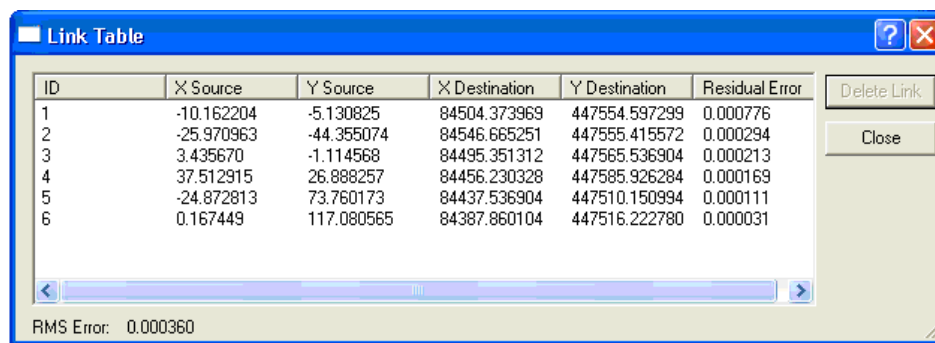
For transforming the local coordinates to the RD system, a similarity transformation of Eq. 8 is required, i.e.

$$\begin{pmatrix} x \\ y \end{pmatrix}_{RD} = \lambda R(\alpha) \begin{pmatrix} x \\ y \end{pmatrix}_{loc} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (\text{Eq. 8})$$

with

$\begin{pmatrix} x & y \end{pmatrix}_{loc}^T$	coordinates in local system
$\begin{pmatrix} x & y \end{pmatrix}_{RD}^T$	coordinates in RD system
λ	scale factor (unknown)
α	rotation angle (unknown)
t_x, t_y	translation parameters (unknown)

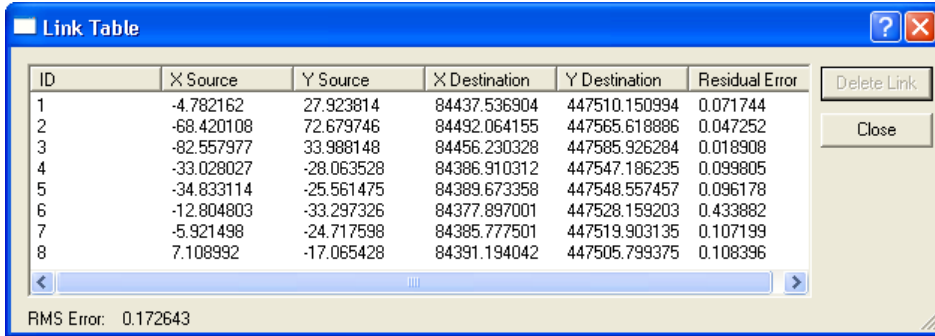
Because the similarity transformation has 4 unknown transformation parameters, 2 control points are the minimum requirement; however more control points are added to improve the reliability of the spatial adjustment. ESRI ArcGIS saves the links to a Link Table and calculates the RMS error. Figure 18 and Figure 19 show the Link Table for occupation point 1000 and 2000 respectively.



ID	X Source	Y Source	X Destination	Y Destination	Residual Error
1	-10.162204	-5.130825	84504.373969	447554.597299	0.000776
2	-25.970963	-44.355074	84546.665251	447555.415572	0.000294
3	3.435670	-1.114568	84495.351312	447565.536904	0.000213
4	37.512915	26.888257	84456.230328	447585.926284	0.000169
5	-24.872813	73.760173	84437.536904	447510.150994	0.000111
6	0.167449	117.080565	84387.860104	447516.222780	0.000031

RMS Error: 0.000360

Figure 18 Link Table of similarity transformation of occupation point 1000 data set.

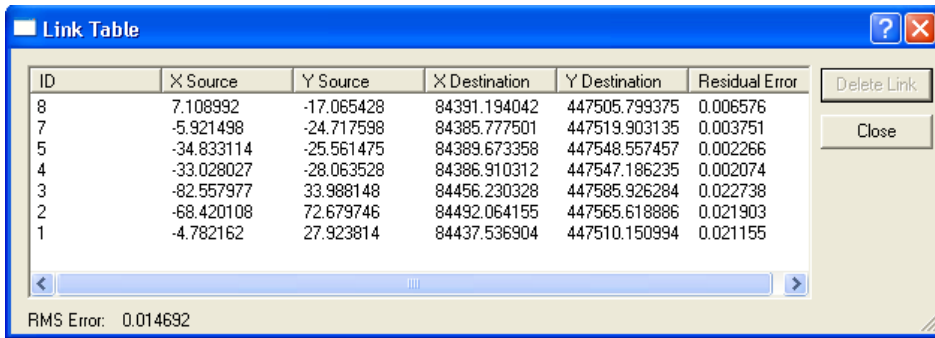


ID	X Source	Y Source	X Destination	Y Destination	Residual Error
1	-4.782162	27.923814	84437.536904	447510.150994	0.071744
2	-68.420108	72.679746	84492.064155	447565.618886	0.047252
3	-82.557977	33.988148	84456.230328	447585.926284	0.018908
4	-33.028027	-28.063528	84386.910312	447547.186235	0.099805
5	-34.833114	-25.561475	84389.673358	447548.557457	0.096178
6	-12.804803	-33.297326	84377.897001	447528.159203	0.433882
7	-5.921498	-24.717598	84385.777501	447519.903135	0.107199
8	7.108992	-17.065428	84391.194042	447505.799375	0.108396

RMS Error: 0.172643

Figure 19 Link Table of similarity transformation of occupation point 2000 data set.

The RMS-error of the Link Table for transforming the local coordinates as observed from occupation point 1000 to GBKN is 0.000360m, which is very accurate. The Link Table for the occupation point 2000 data set shows a high RMS error, i.e. 0.172643 m. This is likely to be caused by an erroneous observation; or linking or selection of an incorrect control points. Removing the control point having the highest residual error (i.e. ID 6 corresponding to control point 2006) gives an RMS-error of only 0.014692m (shown in Figure 20), which value is accepted.



ID	X Source	Y Source	X Destination	Y Destination	Residual Error
8	7.108992	-17.065428	84391.194042	447505.799375	0.006576
7	-5.921498	-24.717598	84385.777501	447519.903135	0.003751
5	-34.833114	-25.561475	84389.673358	447548.557457	0.002266
4	-33.028027	-28.063528	84386.910312	447547.186235	0.002074
3	-82.557977	33.988148	84456.230328	447585.926284	0.022738
2	-68.420108	72.679746	84492.064155	447565.618886	0.021903
1	-4.782162	27.923814	84437.536904	447510.150994	0.021155

RMS Error: 0.014692

Figure 20 Link Table and RMS-error of occupation point 2000 after neglecting control point 2006.

Noteworthy is that the GBKN control points have an accuracy of approx. 20centimeters, so in general the accuracy of observation points can be determined with an accuracy on decimetres-level.

The NAP height of the occupation points is calculated using the vertical direction and distance from the occupation point to the NAP-pin in the Nieuwe Kerk of Delft, given by Eq.9 i.e.

$$H_{\text{occ.point}} = H_{\text{NAP-pin}} - HI + SD \cdot \tan(90^\circ - VA) \quad (\text{Eq. 9})$$

with

$H_{\text{occ.point}}$	height of occupation point (unknown)
$H_{\text{NAP-pin}}$	height of NAP-pin
HI	height of Topcon GPT-7003i imaging total station
SD	subject distance
VA	vertical angle

The height of the NAP-pin, located inside the South-West buttress of the Nieuwe Kerk of Delft and corresponding to point 1001 (see Figure 21) is 1.914m.

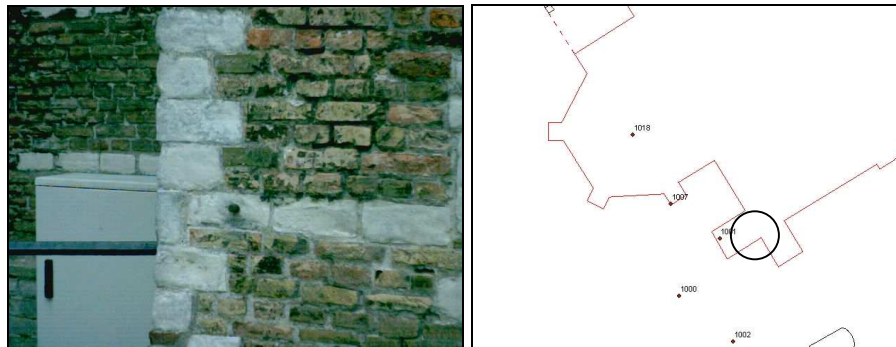


Figure 21 NAP-pin inside buttress of the Nieuwe Kerk of Delft (left) and its location on the GBKN (right).

Table 5 shows the height of the occupation points 1000 and 2000 with respect to ground level, i.e. 0.614 m and 0.677m respectively. The heights of actual observer point are the ground level height added with the height of the instrument.

Table 5 Height of occupation points.

Occ.points	VA	SD	HI	$H_{NAP-pin}$	$H_{occ.point}$
1000	95.2415	2.762	1.553	1.914	0.614379
1000	95.445	2.75	1.563	1.914	0.613131
2000	90.1725	103.363	1.548	1.914	0.677195

4.2.2 Results

After adjustment, the local coordinates are transformed to the RD system. The final output is a collection of 12 digital photographs having *high-accuracy* full spatial metadata, i.e. 3D position and 3D direction parameters (see Appendix A for a full overview). Despite the low camera resolution of the Topcon imaging total station, the high spatial accuracy is expected to serve as valuable input for this research.

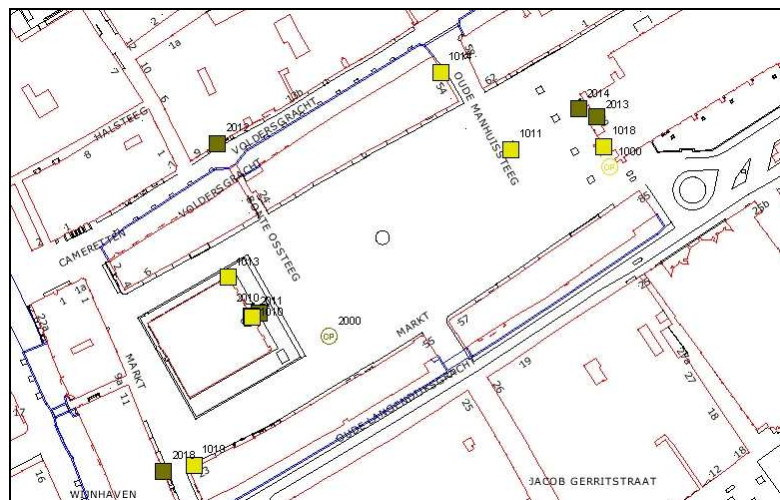


Figure 22 Points of camera (POI) and points of photos (POIs).

4.2 Low-spatial accuracy and high-resolution digital photos.

This section describes the collection of 3D georeferenced and oriented digital photo using the Nikon D100 camera with GPS and 3-axis compass attached to it.

4.2.1 Nikon D100 camera with GPS and 3-axis compass

The Nikon D100 is an D-SLR camera having a 6.0 megapixel low-noise CCD sensor rendering 3,008x2,000-pixel images (aspect ratio 3:2). For this research, it is equipped with its standard Nikon lens and a SIGMA 12-24mm wide-angle lens to capture photos with different field of views (see Figure 23). A calibrated Ocean-server 3-axis digital compass is attached to the hot shoe cover (i.e. *flitsschoentje*). It measures the (changes in) electronic magnetic field to determine the heading and changes in gravity for the pitch and roll; all up to an accuracy of approx. 1 degree. The compass dimensions are only approx. 24mmx24mm and it is supplied with a full-evaluation kit including cables (USB and serial), visualization and logging software. Together with the i.TrekZ1 Bluetooth GPS data logger it is possible to measure both position and view direction of a digital photo. The i.TrekZ1 Bluetooth GPS data logger, having a built-in patch antenna and solar panel, enables to collect position information with a horizontal accuracy up to 3.0 m RMS. Log interval can be set by user based on time, distance or speed and it supports storage of about 50,000 tracking in its internal memory. Output is obtained in CSV, KML or NMEA. The position and view direction are associated to digital photo using time-stamp matching, i.e. the timestamp (and offsets) of compass, GPS and photo are matched to merge the information at moment of capture.

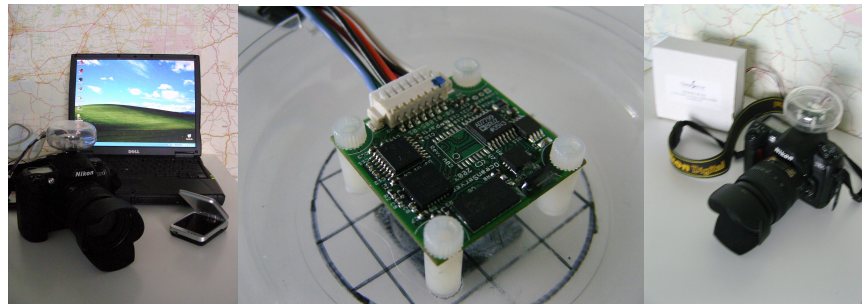


Figure 23 Nikon D100 camera devices added with digital 3-axis compass and GPS.

4.2.2 Measurement setup

At the occupation point 1000, the Nikon camera is mounted with compass on a tripod to collect some digital photos with different lenses, i.e. a standard lens (24-85mm) and wide-angle lens (12-24mm) using different zooms (see Figure 24). The compass is connected to the serial RS232 port of a notebook and compass output together with PC system timestamps are stored in a log file. Using an evaluation-license of Nikon Camera Control Pro, it is possible to control the camera and capture digital photos using the notebook. Advantage of this approach is that photo is immediately saved to the notebook its hard disk and as such the PC system time is added directly which enables timestamp-matching to the compass; no time-offset between PC and camera needs to be determined afterwards. The

GPS data logger is placed on the ground level, i.e. on control point at occupation point 1000.



Figure 24 Collection of Nikon images at the Markt in Delft.

4.2.3 Processing

Using timestamp-matching, the GPS coordinates and compass measurements are associated with the Nikon images. However, during the processing significant errors occurred in the GPS positioning and the compass direction measurements. Although it was expected the observations would show some level of inaccuracy, the errors from GPS positioning are quite large to be further applied for this research.

Due to occupation point 1000 is very close to the Nieuwe Kerk, the GPS data logger could obtain no GPS fix. Because the coordinates of the observation point are already acquired from the Topcon total imaging on a decimetres-level, these coordinates are applied for the perspective view generation in the next chapters.

Another problem that occurred with respect to the compass measurements is that the Nikon camera itself produces electromagnetic fields. While the heading of the compass is observed by measuring the (changes in) the electromagnetic field of the Earth, the compass measurements were affected with two errors from the electromagnetic field of the Earth, i.e.

1. a nearly constant error of approx. -10 degrees caused due to the batteries and metal cover of the Nikon camera;
2. an non-constant error at moment of photo capture of approx. 1-4 dgrs due to the power of lifting the mirror (only appears at SLR- cameras)

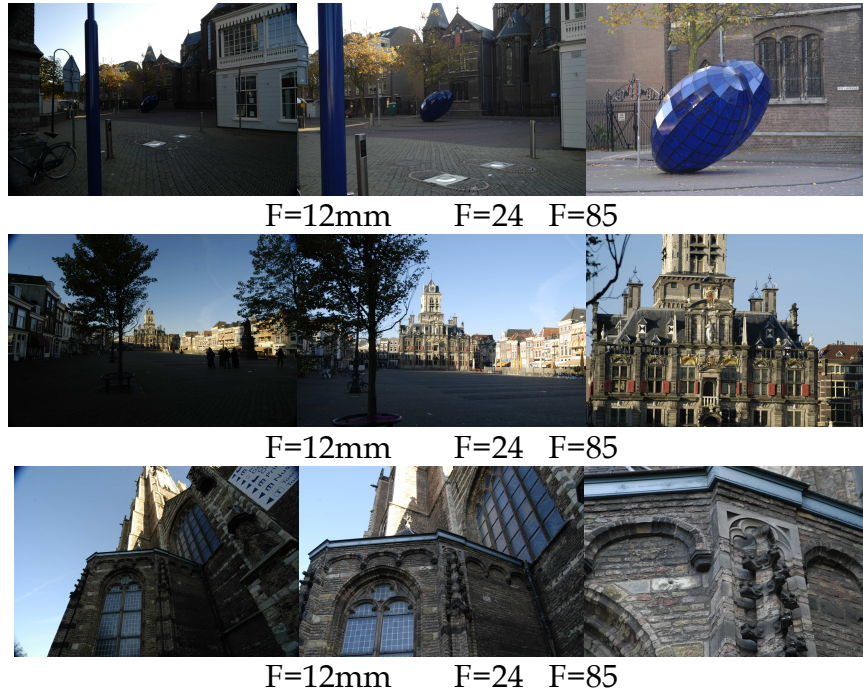
The nearly constant error is obtained by measuring the direction to a control point orientation and as such the compass measurements supposed to be corrected by adding 10 degrees to the observed heading. The non-constant error is corrected by selecting the heading angle just before the photo is taken. Another possibility to solve this problem of electromagnetic fields of the camera body in order to acquire more accurate heading angles is to mount the compass at a certain distance from the camera (as for example is used in the Quakr project).

However, as the high-resolution Nikon images are particularly captured to be used to identify effects of different zooms and (distortions of) lenses, the observed headings are evaluated and corrected manually; further research is recommended to best possible solution to mount the compass to the Nikon camera.

4.2.4 Results

The final result is a collection of Nikon images captured with different lenses and zoom (see Table 1).

Table 6 Collection of Nikon images captured with different zoom.



4.3 Discussion of results

The digital photos collected for this research are captured using a Topcon imaging total station and a Nikon digital camera mounted with digital compass and GPS data logger next to it. The Topcon digital photos have a very low-resolution (0.3 MP) compared to the current available digital camera (>3.0MP). However, the high spatial-accuracy makes this collection of digital photos very appropriate to apply for this proof-of-concept.

The Nikon digital photos have a high-resolution but due to compass errors the spatial accuracy is very poor. The electromagnetic field of the digital camera causes a significant error in the heading measurements and needs to be corrected before it can be applied to the object (mis)identification process. Further research is recommended to camera manufacturers on how to mount a three-axis digital compass to a camera to avoid as much as possible errors of electromagnetic fields. The Ordnance Survey, the mapping agency of the United Kingdom, is concerned with the capture hardware of the Tripod research project and it is proposed to advise them how to correct for electromagnetic field distortion.

For this research, the image metadata is stored in a separate text file; however, to be able to automatically record the view direction and automatically extract it for the photo labelling, an image metadata standard should be developed and accepted to store full spatial metadata parameters.

Chapter 5

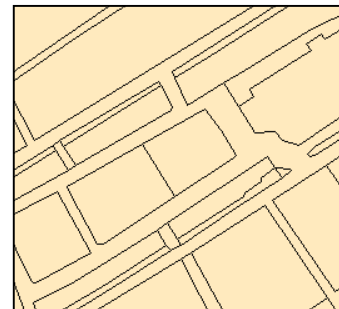
Spatial data preparation

In data analyses and GIS, the research results depend strongly on the spatial data sources. This chapter includes an overview of the preparation of the spatial data for this research. First, section 5.1 describes the available data sets. Section 5.2 describes a two-dimensional approach in which 2D building features are extruded based upon assumed height values. In section 5.3, the building features geometries are updated with height values from an elevation model. Section 5.4 describes the creation of an extrusion model by vectorizing the elevation model. The results are discussed in section 5.5 .

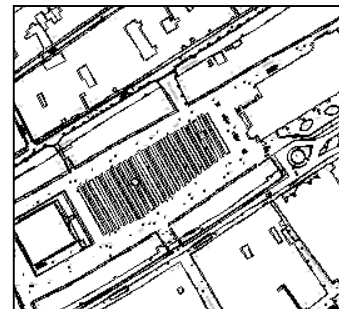
5.1 Data availability

For this research, the following spatial data sets are available (more details in Table 7) :

- *Buildings from TOP10NL*
2D feature data set extracted from the Top10nl 1:10,000 scale vector map, containing footprints of buildings and building blocks of the province Zuid-Holland, The Netherlands. It is clipped to the study area of Delft and updated with more detailed features from the GBKN.



- *Grootchalige Basis Kaart Nederland.*
1:500 to 1:5,000 large-scale basemap of the Netherlands containing terrain line features as building footprints, roads, lamp posts, sewery (putten) and so forth. This data set is converted from linestrings to polygons and several historic and cultural points-of-interest are selected to update the TOP10NL buildings data set.



- *3D building model:*
Three-dimensional extrusion model, containing the buildings of the city centre of Delft. It is created from the TOP10NL building block footprints, which are split up in separate buildings manually from an aerial photo; buildings are extruded to AHN elevation heights.
- *Actueel Hoogtebestand Nederland (AHN):*
5m-resolution Digital Elevation Model, derived from an interpolation of selected height points of the original base file measured with laser altimetry (a.k.a. LiDar). The base file has a point density of 1~16 points per 16 km² and the accuracy of the points is approx. 5 centimetres with a standard deviation of 15 cm. Heights points are measurements on ground level as peaks due to buildings and vegetation are filtered out of the selected points; except build-up areas, i.e. rural and city centres having an area >1km². The AHN is clipped to the Delft study area for this research.
- *Roads network, water and POIs from TeleAtlas:*
2D feature data sets, containing the roads network, hydrology and Points-of-Interests of Zuid-Holland respectively. It is clipped to the extent of the Delft study area. All data sets contain names of objects serving as possible input for photo labelling.

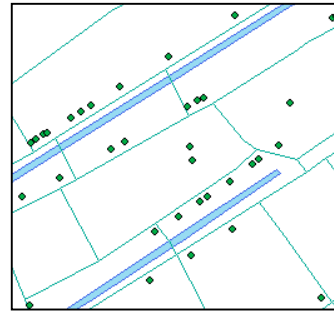
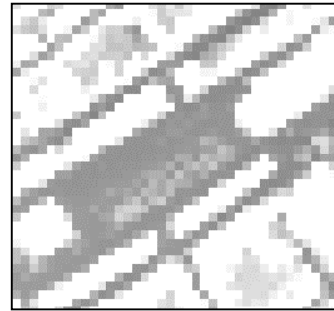
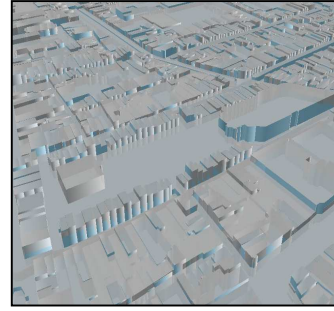


Table 7 Characteristics of the spatial data sets.

Name	Format	Geometry	Dim	CRS	Extent			
					Left	Top	Right	Bottom
3D building model	KML	PolygonZ	3D	WGS84	4°20'43"	52°01'21"	4°22'37"	52°00'13"
AHN	GeoTIFF	Raster	2D	RD	80,000	450,000	89,999	443,752
Buildings Top10nl	Shape	Polygon	2D	WGS84	4°19'05"	52°25'49"	5°03'43"	51°58'03"
GBKN	CAD	Linestring	2D	RD	83,959	84,873	448,228	447,230
POIS TeleAtlas	Shape	Point	2D	WGS84	3°50'36"	52°18'51"	5°07'21"	51°39'37"
Roads TeleAtlas	Shape	Linestring	2D	WGS84	3°49'25"	52°19'05"	5°9'14"	51°39'30"
Water TeleAtlas	Shape	Polygon	2D	WGS84	3°17'51"	53°32'21"	6°21'11"	51°22'05"

The buildings of TOP10NL and selected features of GBKN are updated with names. One layer is created with historic and cultural object names (among others taken from the TeleAtlas POIs), and house numbers from GBKN labels are added to end houses. A second layer is created by adding shop names to buildings at the Delft Market square using own field observations (*pen and paper*).

The 3D model is not applied for this research to serve as true 3D building model as it is created by extruding 2D features to average or modal heights from AHN

(giving flattened cuboids). This chapter describes how to create a better 3D model showing a more factual picture of Delft; however, because the TOP10NL and GBKN do not include separate buildings (and the 3D model does because of the manual mapping from aerial images) the building footprints are updated with the footprints of the 3D model (after converting KML to 2D ESRI Shapefile).

Because numerous 2D map coordinate system are in the WGS84 coordinate system and the AHN is provided in the Netherlands Rijksdriehoekstelsel coordinate system, a datum transformation (projection) is required in order to prevent datum conflicts at a later stage. Due to the coordinates of the Topcon sample photos are computed in the RD system, the WGS84 features of the 2D spatial data sets are projected to RD system using ESRI ArcGIS Project Tool (see Figure 25). For the Nikon test data, the WGS84 can be converted to RD using e.g. the RDNAP.nl coordinate calculator web service (RDNAP).

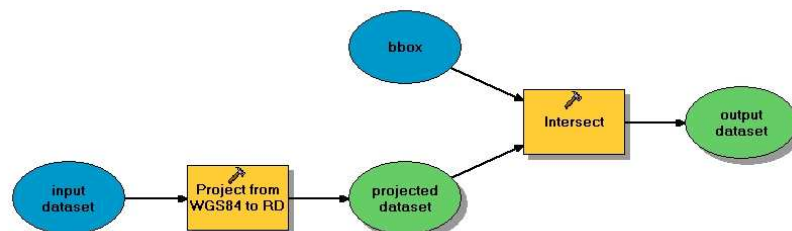


Figure 25 Datum transformation and clipping of spatial datasets using ESRI ArcGIS ModelBuilder.

Also, the spatial data sets have different spatial extents. Because the selected study area for this research is the city centre of Delft, the spatial data sets are clipped using a bounding box (bbox) of (83615, 448730) and (85213,446490) to obtain a smaller extent using ESRI ArcGIS Clip Tool (see Figure 25). Reason for changing the extent is that it is expected computations and analyses can be carried out much faster because of less data and features.

5.2 Extrusion model using assumed height values

The easiest and simplest way to construct a '3D model' from 2D maps is to extrude 2D features based on assumptions. Imagine that for the city centre of Delft, the following assumptions apply, i.e.

- An average building in the centre of Delft has a height of about 12 meters, i.e. assuming 4 floors x 3 meters of height per floor.
- Higher buildings (e.g. apartments) have a height of about 20 meters, i.e. assuming 6-7 floors.
- Due to the flatness of The Netherlands, assume street and water are at mean sea-level, i.e. assume height is equal to zero (default value in ESRI ArcScene).

Using ESRI ArcScene, the features are extruded to its assumed value. Therefore using ESRI ModelBuilder, the data sets are extended with a new attribute field 'Z_value' containing the assumed height values for each building type. From the TOP10nl buildings, the average buildings (TDN-code 1000 and 1013) and (assumed) higher buildings (TDN-code 1023, 1030 and 1073) are selected and height values are added.

Secondly using ESRI ArcScene, the TOP10nl building features are extruded (at the Properties » Extrusion Tab Window) with extrusion value equal to the values of the 'Z_value'-field. This result is visualized in Figure 26, showing the model with at the bottom left, the extruded footprint of the Nieuwe Kerk of Delft and to its opposite the extruded historic municipal building.

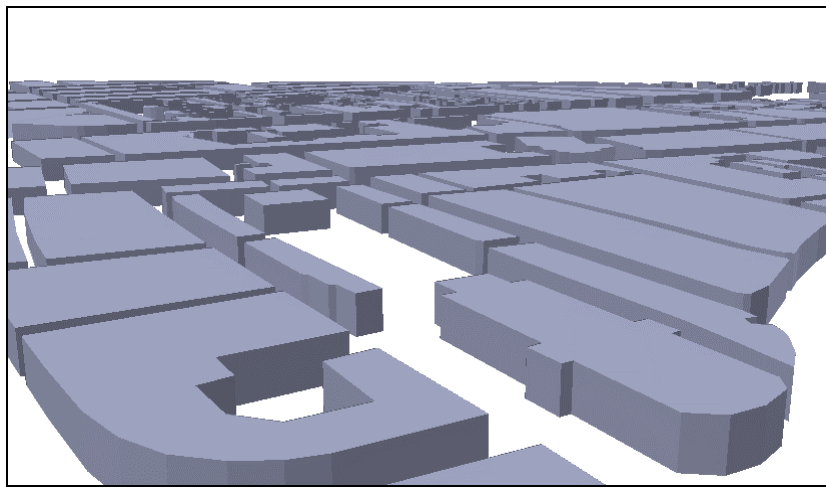


Figure 26 Extrusion of 2D features using assumed height values.

5.3 Extrusion model using AHN heights

This section describes two approaches on creating an extrusion model using the elevation model, i.e. add heights to centroids of building footprints and use the elevation model as base heights for the building footprints.

The first approach to create an extrusion model is to extract heights values from the elevation raster and add these values to the centroids of the building footprints. Using a spatial join the two-dimensional footprints are extruded to the AHN height values. Because the pixel values of the AHN are given in centimetres and the linear unit of RD coordinates is 'meters', it is required to specify a Z factor that converts from centimetres to meters. So, a z factor of 0.01 is inserted to obtain heights in meters. The obtained extrusion model is shown in see Figure 27.

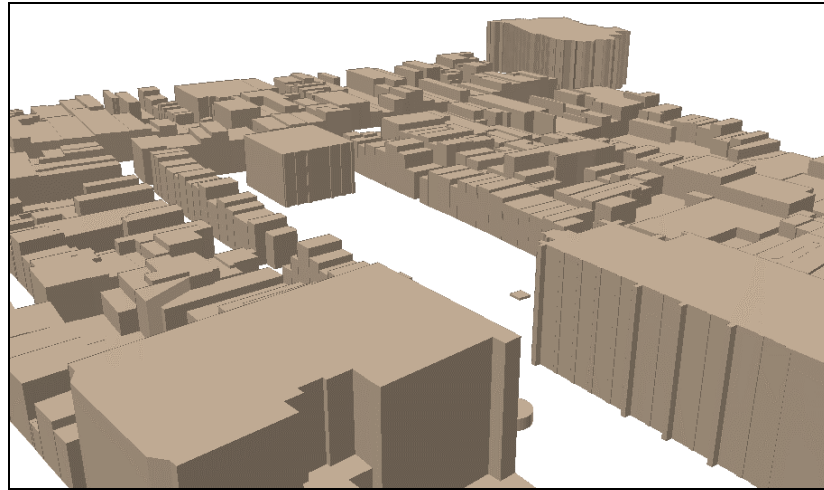


Figure 27 Building footprints are extruded to the height values of the centroids.

Disadvantage is that each building is visualized with a flat roof; which in the real world is not the case, but for numerous visualization applications this model is appropriate enough. However; in order to localize pictured objects as good as possible, a high-detailed three-dimensional is required.

The second approach uses the height values from the elevation as base heights for the building footprints. Using the ESRI 3D Analyst Interpolate Shape Tool together with ESRI ArcGIS ModelBuilder, each feature of the 2D map is updated with the underlying height value from AHN, i.e. pixel value at the same position. The interpolation is carried out BILINEAR, i.e. it uses the value of the four nearest input cell centers to determine the weighted average of these four values on the output raster (other possible options are LINEAR and NATURAL_NEIGHBORS which are only available for Triangular Irregular Networks and not for raster).

The output is a feature class containing building features of geometry type 'Polygon ZM' (the Z refers to 3D (height) information; M-values are used to measure the distance along a line feature from a vertex (a known location) to an event.). Figure 28 shows the model.

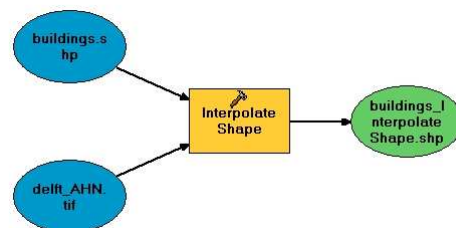


Figure 28 Using the Interpolate Shape tool to add heights from AHN to the building features.

Visualizing this in ESRI ArcScene gives a scene with floating objects as shown in see Figure 29. (Note: the Interpolate Shape Tool gives same results as using the Properties » Base Height Tab Window and then select the 'Obtain height for layer from surface' and refer to the elevation model).

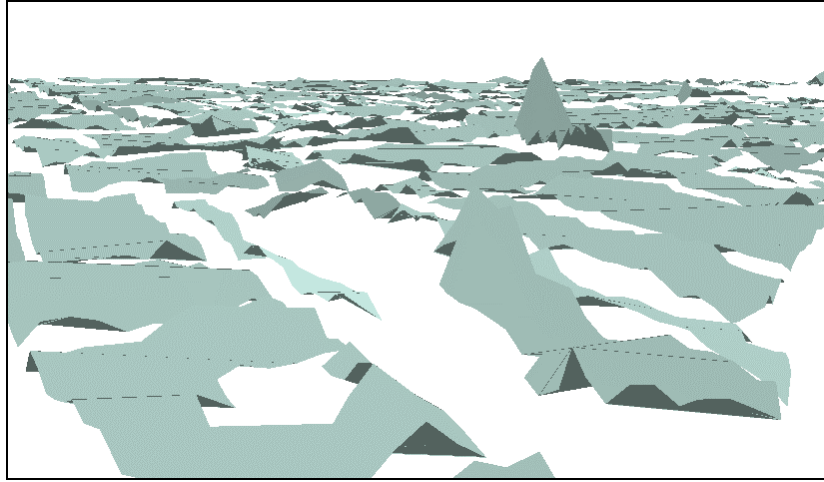


Figure 29 Floating features as result from AHN height interpolation.

Instead of an extrusion with a value as such that features are lifted, it is also possible to extrude features by pushing them down. For this 2.5D approach, the features are extruded to mean sea-level, i.e. a height equal to zero (corresponding to the value in Section 5.2 about the extrusion of 2D features using assumed height values). Other possible values are e.g. the minimal value of the elevation model (i.e. AHN of Delft -8.41m NAP) or the ground level of the area of interest (i.e. Markt in Delft $\sim +0.6$ m NAP). So, the final step is to extrude the features (using the Properties » Extrusion Tab Window) to a value of zero. This result is shown in Figure 30.

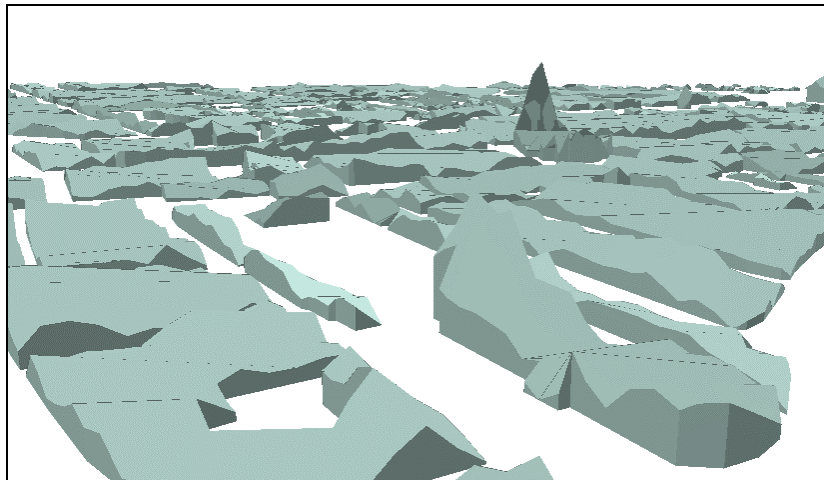


Figure 30 Extrusion to value zero, pushing the floating building features down.

The three-dimensional model Figure 30 of can be considered as some sort of rough building model. The output strongly depends on the resolution of the elevation model, because the height value assigned to the features is the weighted average of four neighbouring pixels of the elevation raster file (a.k.a. *bilinear* interpolation). Disadvantage is that the heights are only added to the nodes of the building footprints which results no reliable height information is available inside the buildings. This mainly causes differences of real-world objects with extruded features inside the model for historic and irregular shaped buildings.

5.4 Extrusion model from intersection with rasterized AHN

The assumed heights model of section 5.2 only works for standard buildings (e.g. row houses and apartment building when number of floors is known), but for other objects as churches and historic buildings this model will not be sufficient for demarcation of the pictured objects. The extrusion model using the AHN to set the features' based heights (section 5.3) has the disadvantage that only the nodes of the building footprint are updated with height values which causes that no knowledge is available of the height 'inside' the building footprint polygons.

To solve this problem, the idea was to update the AHN with the OIDs of the building footprint features by first converting the building footprints to raster cells (see Figure 31).



Figure 31 Rasterized features using the AHN cellsize of 5m.

However, an extrusion of a raster data set was not available in the ESRI 3D visualization software (ESRI ArcScene), so the solution is to go the other way around, i.e. converting AHN raster to vector features.

First, the AHN raster is converted to vector format disabling feature simplification in order to maintain square (pixel-related) features. Output of this raster-vector conversion is a feature data set having >10,000 features with attribute GRIDCODE, which corresponds to the AHN elevations (i.e. height values in centimetres).

Next, the AHN features are intersected with the building footprints resulting in feature class showing some sort of rasterized building footprints (as shown in Figure 32 and Figure 33). This is due to an intersection of AHN features with the building footprints is a set of features that contains all features of AHN that also belong to the building footprints. Where there exists only a partial overlap of AHN features with the building footprints, only that part is selected that is contained in both data sets and it is added as separate feature. Result is a feature class of 10,381 features having the attributes height values and object identifiers of the corresponding building footprints assigned to it.

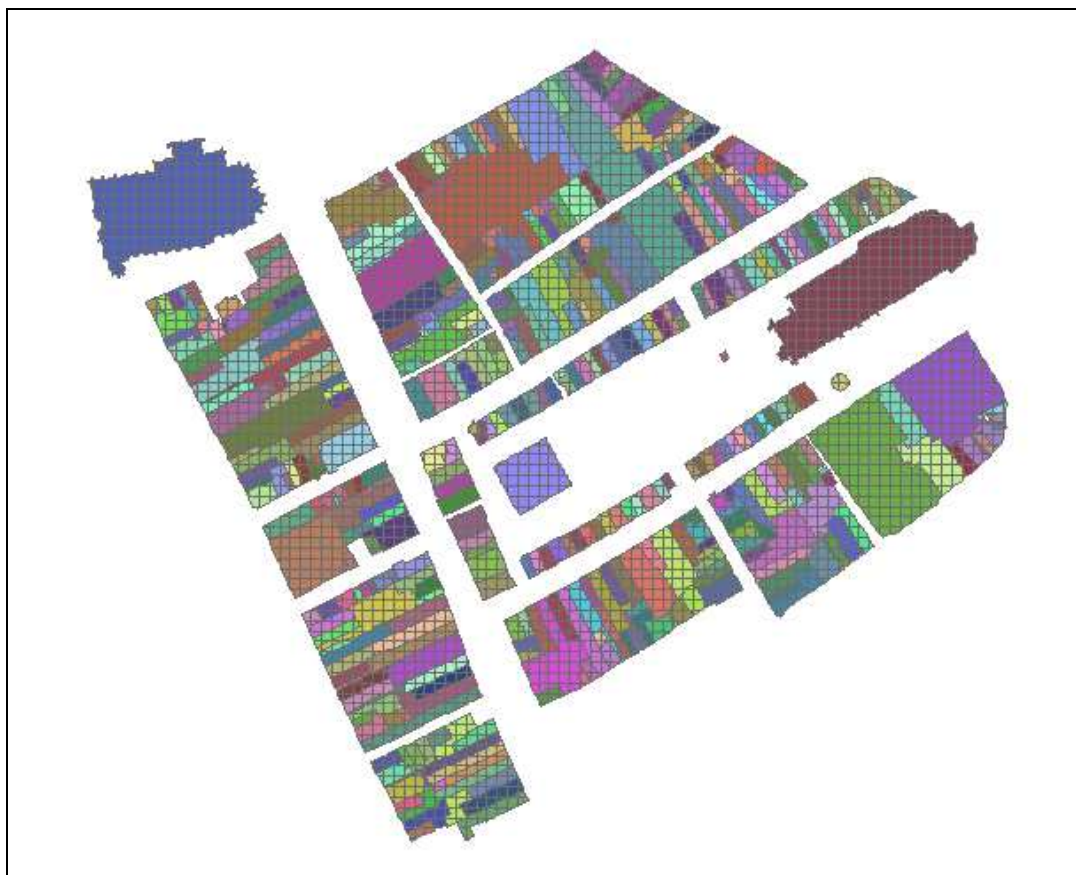


Figure 32 Intersection of vectorized AHN and building footprints.

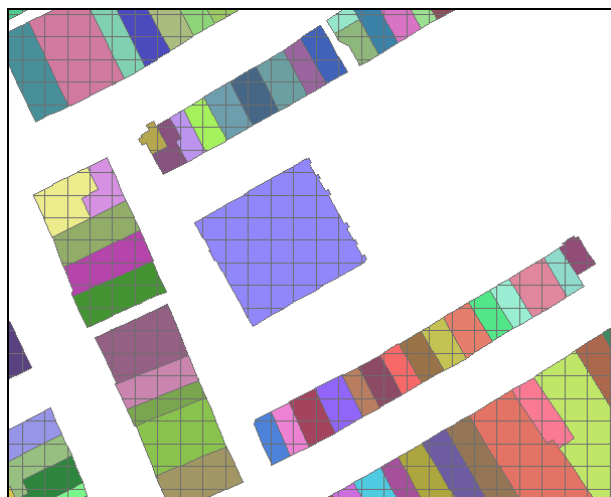


Figure 33 Detail of Intersection of vectorized AHN and building footprints.

Two advantages of using a vectorized elevation model and the intersection approach are that the building footprint its geometry is kept, but also that (differences in) heights are known inside of the building features (and not only at the nodes/vertices). Extrusion of this feature class gives the resulted extrusion model as shown in Figure 34.

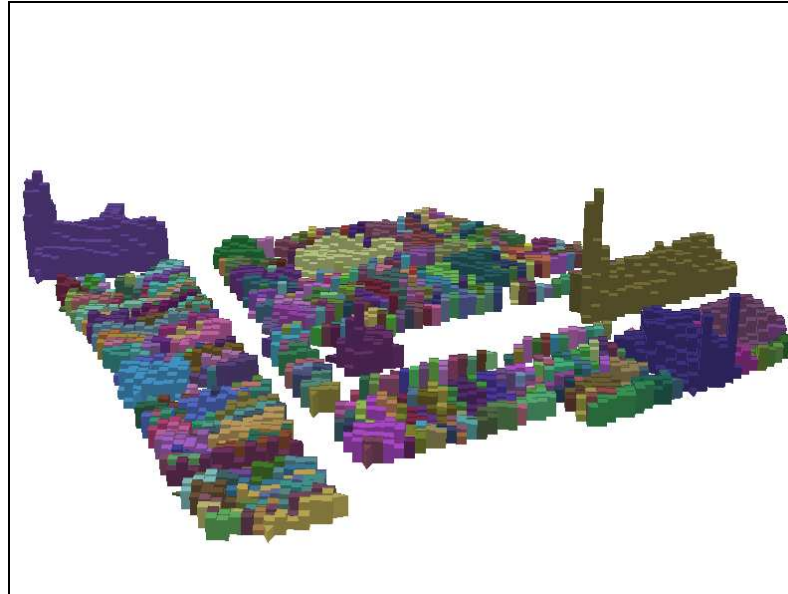


Figure 34 Extrusion model of the intersection of vectorized elevation model with the building footprints.

5.5 Discussion of results

This chapter proposed four approaches to create a three-dimensional model. The differences between the extrusion models are visualized in Figure 35.

The extrusion model created from a vectorized elevation model intersected with the two-dimensional footprints of the buildings is considered to be the best model, because it represents the buildings as much as possible in conformity with the real world. It is expected that for a photo labelling service the objects-of-interest are particularly historic and irregular-shaped buildings, and this extrusion model visualizes these buildings very well. Advantage of this three-dimensional model is that it provides the heights inside the buildings and not only at the nodes of the footprints, which is the case for the extrusion model with base heights set from the elevation model) and that the footprint its geometry is kept, which is not the case the footprints would have been rasterized (see Figure 31).

Disadvantage of this extrusion model is the enormous amount of features that are created from the raster-vector conversion of the elevation model. For the extent of the study at the Market square in the historic city centre of Delft, already over 10,000 features were obtained. For larger extents, this causes serious problems for analysis and fast rendering of the three-dimensional model. As the number of features increases, the processing and rendering of the data slows down; and applying this extrusion model to a larger extent (e.g. country) is expected to cause serious problems.

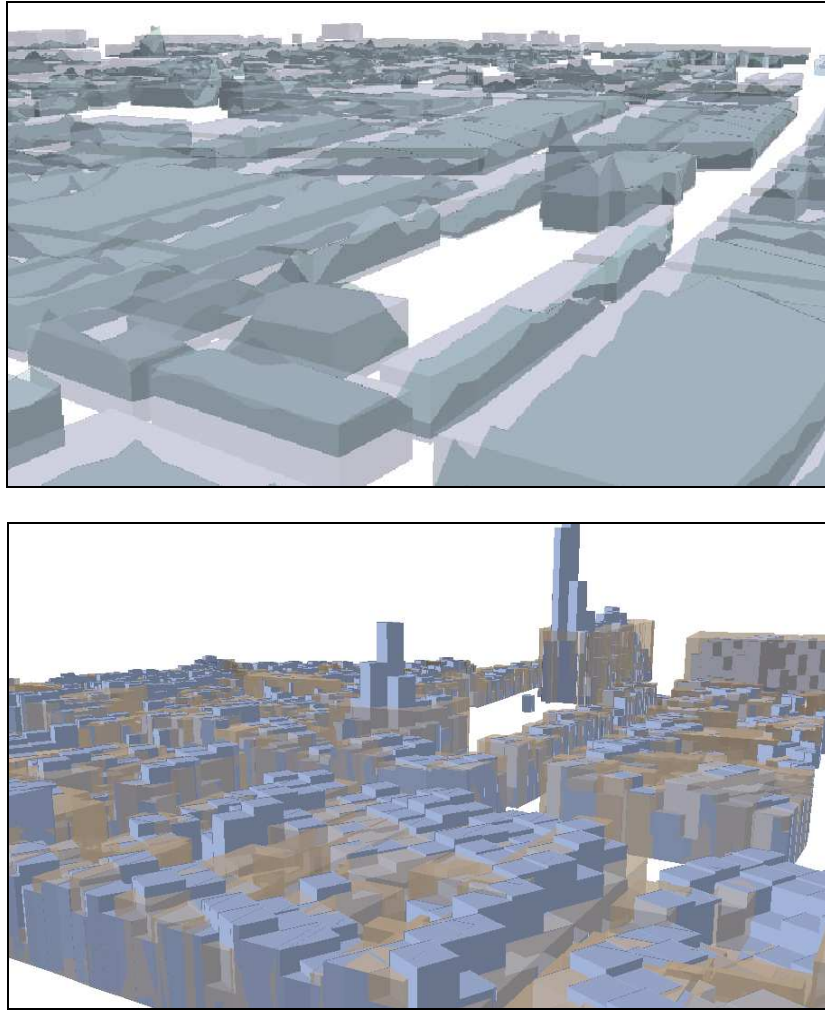


Figure 35 Visual comparison of the created extrusion models.

Further research is therefore recommended to how to apply the extrusion model from a vectorized digital elevation model intersected with two-dimensional footprints for larger spatial dataset extents. Some ideas are merge features that share the same height on a decimeter-level (the elevation model used for this research was on a centimeter-level and it is questionable whether such a high-accuracy is required for this application), or to render only that part of the extrusion model that is visible inside the photo. The two-dimensional footprint (a.k.a. region polygon) that is visible inside the digital photos is obtained using trigonometry equations or using a GIS visibility analysis (e.g. viewshed analysis). In doing so, the number of features that should be extruded and rendered is significantly decreased.

Chapter 6

Object identification

This chapter describes the elaboration of the object identification (and localization) part of this research by means of a proof-of-concept. The concept is described in section 6.1 and it is proofed using GIS software and data as described in the following sections, i.e. section 6.2 explains how to create and export a perspective view using the extrusion model and a perspective viewer service; and section 6.3 describes how to link the perspective view to the 2D spatial data sets in order to pick the labels from it and place inside the digital photo. Section 6.4 discusses the results.

6.1 Proof-of-concept

After the collection of 3D georeferenced and oriented digital photographs and the spatial data preparation, the next phase of this research is identify and localize visible objects inside the digital photo. The goal is to link the objects inside the photo to a 2D spatial data set, so it is possible to pick the object names from this dataset and place it as labels inside the digital photo. Therefore two steps are required, i.e.

1. localize visible objects inside the photo so it is known *where* captured objects are positioned
2. and identify *what* is visible inside the photo by linking the visible features to a 2D spatial dataset.

The first step is carried out using a perspective view (a.k.a. virtual abstraction or scene) that matches the digital photo. This is achieved using a scene generator (e.g. ESRI ArcScene) that visualizes spatial datasets in a 3D perspective view. After this perspective view is exported to a (loss-less compressed) image format (e.g. PNG), the objects are converted back into vector features using an average raster-vector conversion. The final step is to link the features to the 2D dataset, which is the essential step. The linking of the features with the 2D spatial dataset is for example carried out by converting for each pixel or features its photo coordinates to coordinates of a geographic-related coordinate system using camera calibration and (close-range) photogrammetry equations.

This research proposes another method using the scene generator's output. The virtual abstractions (i.e. the exported scene) provides information about which pixel is occupied because its value differs from the non-occupied background pixels (by default this would be a white colour). But whether or not a pixel is occupied or not does not identify the objects. Therefore it is considered to colour each object in the 3D perspective view based on its object identifier, as every object in the 2D spatial data set has a unique object identifier (OID). After exporting the scene, the perspective view is linked to the 2D features by joining the datasets (virtual abstraction raster and 2D vectors features) on colour values (3D view) and OIDs (2D dataset). The renderer typically carries out colouring the objects. By converting decimal OIDs to RGB colour values, the renderer assigns a particular colour value to a corresponding feature.

Summarizing the previous and to carry out this process the following steps are required:

1. Colour features of the extrusion model (created from 2D features) to its object identifier
2. Create and export the 3D scene using the view parameters to an image format (e.g. PNG)
3. Link the colour values of the scene to the 2D features OIDs.
4. Overlay the photo with labels of the virtual abstraction.

Previous concept results in a photo containing names of object pictured; labels are placed inside or next to the visible object (depending on the object its size). This concept is applied to the Topcon digital photos in the next sections using ESRI ArcScene and ArcMap.

6.2 Perspective view in ESRI ArcScene

The first two steps of the proof-of-concept is carried out using ESRI ArcScene, the 3D visualization software of ESRI ArcGIS. First, the extrusion model of rasterized AHN intersected with building footprints is imported into the project. Next, the extrusion model features are coloured to its object identifier (OID). This requires to configure the renderer, which is a mechanism that defines how data appears when displayed (ESRI Glossary). By changing or configuring the renderer, the spatial data will be displayed differently, among others it converts the geometry when the view changes by zooming or panning, it colours the objects and includes information about texturing, shading and lightning.

In ESRI ArcGIS, different types of renders are distinguished among others (see Figure 36):

- *SimpleRenderer*: Each feature in the layer is symbolized with the same symbol
- *UniqueValueRenderer*: Features are drawn with different symbols depending on the unique values of one or more attribute fields

- *ClassBreaksRenderer*: Features are drawn with graduated colour (choropleth) or graduated size symbolization based on values in an attribute field.
- *ProportionalSymbolRenderer*: Features are drawn with proportionally sized symbols based on values in an attribute field.

(EDN, 2007)



Figure 36 From left to right: Examples of a SimpleRenderer, UniqueValueRenderer and ClassBreaksRenderer.
(source: ESRI ArcGIS Desktop Help)

For this application, a UniqueValueRenderer is required as each object needs to be coloured uniquely. Normally, a colour ramp (e.g. algorithmic or random) would be defined which defines the range of colours that fills the renderer with colours that will be assigned to features. However, this is not possible for this application as

- Objects should be coloured with a specific colour excluding a random colourramp and
- Gaps in OIDs' sequence of 2D spatial datasets causes mistakes in applying a algorithmic colour ramp.

Therefore, the renderer is filled on a one-by-one base using a loop through all OIDs; implemented using some VBA example code available at the ESRI Developer Network site (EDN, 2007). For that, decimal OIDs are converted into RGB colour values. The following relationships exist, i.e.

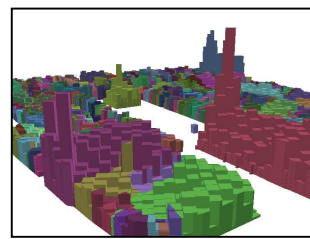
$$RGB_{decimal} = (65536 * RED) + (256 * GREEN) + BLUE \quad (\text{Eq. 10})$$

In ESRI ArcScene, the following VBA code is added to the loop for filling the renderer with colours:

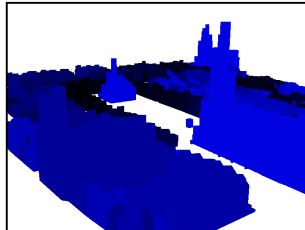
```
pNextColourRGB.Blue = pOID Mod &H100
pNextColourLong = pOID \ &H100
pNextColourRGB.Green = pOID Mod &H100
pNextColourLong = pOID \ &H100
pNextColourRGB.Red = pOID Mod &H100
```

The result of colouring the extrusion model is shown in Figure 37. Note that the renderer should not include shading effects because as such one object has several colours, which is very useful for average visualization, but for this application it would result in additional objects inside the scene.

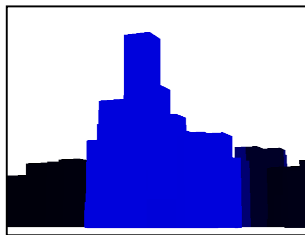
Activities in ESRI ArcScene



1. Load the extrusion model in ESRI Arc Scene



2. Color the extrusion model with RGB color values based on the OIDs



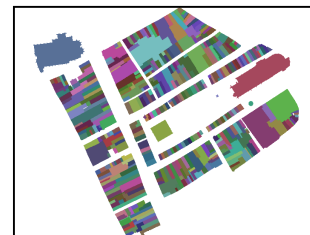
3. Set the view settings and export the virtual scene to ESRI ArcMap.

POI (Target) (84385.39994 447515.7996 18.237352)
POC (Observer) (84495.671 447561.9386 0.614378942)
Heading 247.29°
Pitch 7.60°
Roll 0°
SD 120.495 m
FOV 30°



I

Activities in ESRI ArcMap

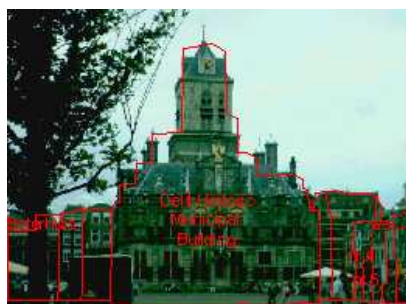


4. Convert RGB color values to OIDs using a weighted sum of rasterbands.

5. Convert virtual scene from raster to features.

6. Link OIDs of virtual scene features with OIDs of 2D spatial dataset.

7. Pick object names from 2D dataset and label objects of virtual scene.



F

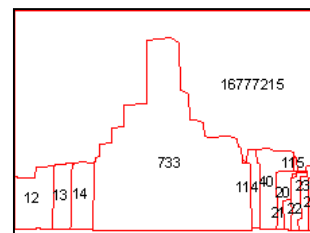


Figure 37 Process flow of object identification using ESRI ArcGIS.

Secondly, inserting position, orientation and camera parameters in the View Settings User Interface sets the view. As this UI only enables to type in observer point, target point, pitch, roll and view angle; the conversion from heading and subject distance to target point is later implemented in VBA.

Next, the scene is exported to PNG file format. PNG is a loss-less compression format maintaining the visible objects its crisp boundaries, which is required for the raster-vector conversion. Also, saving the scene as JPEG causes more non-existing objects to appear because of the fuzzy boundary as shown in Figure 38.



Figure 38 Difference between loss-less compressed PNG (right) and lossy compressed JPEG (left).

6.3 Linking features in ESRI ArcMap

Two Topcon sample photo are selected, i.e. 1010 and 2014, for further processing in ESRI ArcMap. In ESRI ArcMap, the exported scene is loaded into the workspace. Next, a raster-vector conversion is applied to this scene. Therefore all three colour bands (Red, Green and Blue) are summed up using the weights that convert the colour values back to OIDs again, i.e.

$$OID = (65536 * REDcolorband) + (256 * GREENcolorband) + BLUEcolorband \quad (\text{Eq. 11})$$

Reason for this weighted sum is on the one hand to avoid only the first colour band (red) to be converted to vector features, but on the other hand these values (saved as GRIDCODE) are in the end required to link the scene features to the 2D features. The raster-conversion is carried out based on the pixel values (GRIDCODE field) without simplification of the objects geometry. Figure 39 shows an overlay of the original photos with the exported scenes and their vectorized layers.

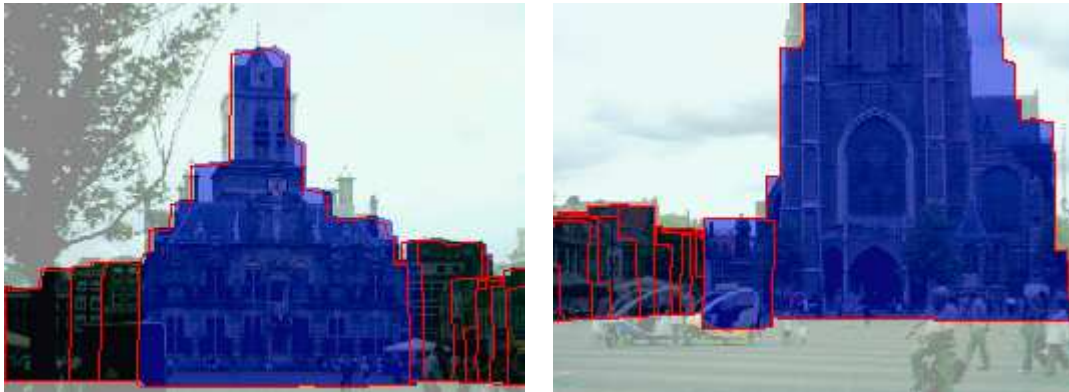


Figure 39 Overlay of the virtual scene and its vectorized layer with Topcon images 1010 and 2014.

Final step is to link the scene output to the 2D features. Using a layer's Join based on the GRIDCODE field of the vectorized scene and the OID of the 2D spatial data set. Now, labels can be place inside the photo as all visible object captured inside the photo -that also exist in the 2D GIS data set- are identified and localized; see Figure 40 for the two selected example photos and Appendix A for all Topcon images.



Figure 40 Object identification results of Topcon images 1010 and 2014.

As such the object identification part of this research is finished and carried out; not using computer vision technology to identify edges and patterns in digital photos or by converting pixel coordinates to terrain coordinates, but by benefiting the output of a perspective view generator.

6.4 Discussion of results

This chapter described the object identification using OID-based colour values to link perspective viewer service output to the three-dimensional model. This approach provides good results as is shown in Appendix B which shows the Topcon images overlaid with vectorized virtual scenes and object names.

It is expected that the more detailed the spatial data sets and the elevation model are, the better the object identification can be carried out. The increasing availability of three-dimensional building models (captured using e.g. LiDar) and increasing resolution of elevation models will in the future enable very accurate delineation of visible objects inside 3D georeferenced and oriented digital photos.

Chapter 7

Label placement

This chapter proposes three approaches to place a label at the best position. Because cartographic label placement is an optimization problem, the best position for label depends on its purpose (user requirements). Using constraints defining whether or not a label may or may not be placed, the algorithm determines the best location for the label. As the exported virtual scene is actually a two-dimensional abstraction, the problem of placing labels in a 3D environment is reduced to a 2D problem which allows to apply 2D GIS label algorithms. For this proof-of-concept, the ESRI Maplex Label Engine extension is used to visualize the results of different configuration settings of the label algorithm; some basics of ESRI Maplex are described briefly in section 7.1. Section 7.2 describes the use of external, i.e. placing labels outside the visible objects. Section 7.3 continues with how to maintain a perspective view by varying in label size depending on the distance between observer point and visible object. Furthermore, section 7.4 describes a prioritization of visible objects as a user is expected not only to be interested in names of historic or touristy buildings, but also e.g. in labels of streets, rivers and shops. How many labels should be placed inside a photo is included in section 7.5. Section 7.6 discusses the results.

7.1 ESRI Maplex Label Engine

Maplex for ArcGIS is a set of tools that improve the quality of label placement on a map. For example different levels of importance to features could be assigned to ensure the more important features are labeled before less important ones. It allows several functionalities, among others,

- *Placement properties*: internal or external annotation using a specified offset, repetition of labels at a specified distance along a line, search tolerance to remove duplicate labels.
- *Conflict resolution*: conflicts between labels are resolved when the available space is limited by removing, or weighting of features to determine label placement.

7.2 External annotation

One of the objectives of this research is to place labels outside visible objects avoiding overlap between labels mutually and labels with visible objects. Outside placed labels, also referred to as external annotation, are related to the corresponding object using connectors (a.k.a. *leader*), such as a line or text balloon. It should be noticed that external annotation is not in all cases the most proper solution; e.g. using curved labels to annotate rivers. Figure 41 shows digital photo 2014 with labels placed outside the object, i.e. *placement property outside* at the ESRI Label Manager.



Figure 41 Digital photo 2014 with external annotations.

However, this does not avoid labels to overlap other features. The vectorized exported scene also contains features having GRIDCODE value of 16,777,215; which refers to the maximum RGB colour, i.e. white (255,255,255). As these features represent empty areas, i.e. areas where no objects are existing in the extrusion model when visualized in a specific perspective view; this would be an ideal area to place labels as it is assumed it will not include objects-of-interest. So, a constraint is defined that labels must be placed outside without overlapping the interior of other features.

Using weights, it is controlled whether labels will be placed when there are potential conflicts. In the ESRI Label Manager, the non-overlap of labels and features is achieved setting the interior feature weight; a weight of 0 allows and a weight of 1000 disallows to overlap labels and features of a specific label class. To prevent labels to be removed if there exists a conflict, a weight value of 999 is applied. After the features having GRIDCODE value of 16,777,215 are deleted from the vectorized exported scene, the labelling results for digital photo 2014 is shown in Figure 42.



Figure 42 Photo 2014 labelled with conflict resolution value 999 to avoid overlap of labels with object as much as possible.

Nevertheless, it could happen that digital photo contains subjects pictured that do not exist in the 2D spatial data set, e.g. people, flora and fauna, and so forth. To avoid these subjects to be covered with labels, an additional analysis is carried out to identify whether or not a pixel is occupied or not. This is achieved by assuming a threshold value to specify whether or not a pixel should be considered as occupied.

Using ESRI ArcMap, the digital photo 1010 is reclassified after summing all colour bands with threshold value equal to the median pixel value; subsequently a union with the vectorized virtual abstraction merges this layer with additional non-identified but localized features. In Figure 43, the tree on the left is overlapped with label 'Boterhuis', next the binary image is reclassified with threshold value 347 (i.e. median) and after raster-vector conversion and union with the vectorized virtual scene, Figure 44 shows the final output, i.e. the digital photo with labels placed avoiding overlap with any identified and non-identified visible object.

7.3 Perspective view with varying label sizes

In a perspective view, the objects extend to the back causing objects to decrease in size from the observer's viewpoint. It is expected that in average digital photos, the subject of interest is usually in front of the pictured scene and that an increasing distance from observer's view point to other visible objects decreases the importance. Therefore, varying in label sizes depending on distance from view point results in labels to nearest objects to stand out and it also maintains the perspective of the pictured view.

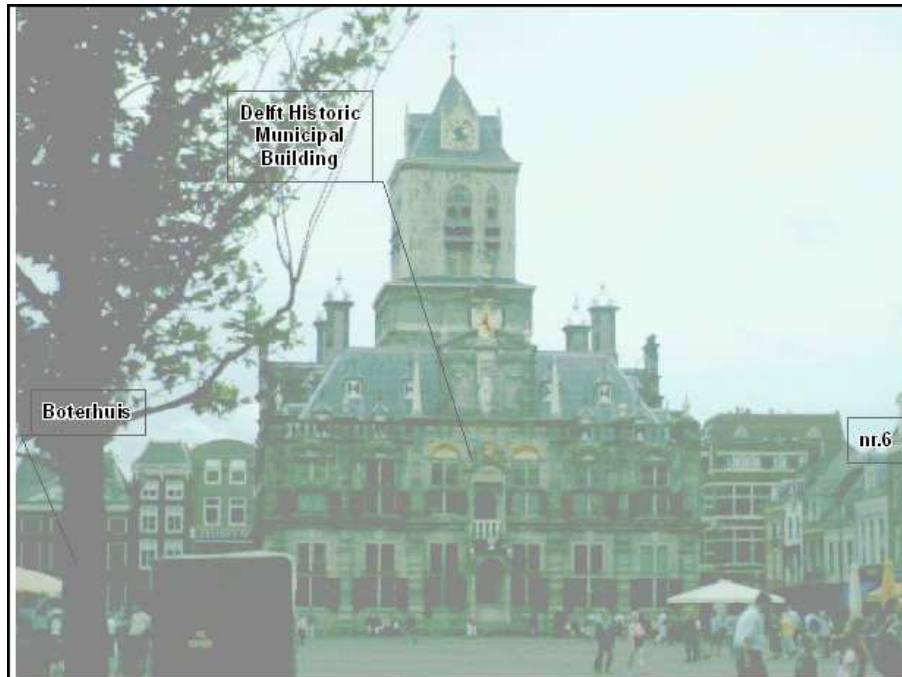


Figure 43 Photo 1010 with labels overlapping visible objects not existing in the three-dimensional model.

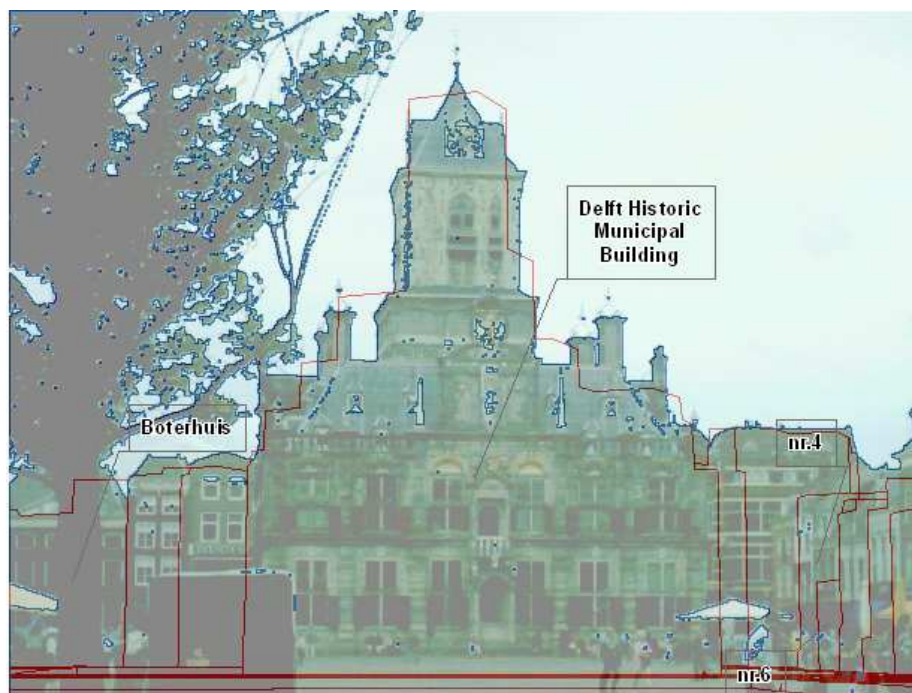


Figure 44 Digital photo labelled as such that labels do not overlap objects of the virtual scene and do not overlap pixels above a specific value from binary image.

To achieve varying label sizes, a depth image is applied. A depth image stores the depth of a generated pixel (i.e. z-coordinate) in a buffer (i.e. the z-buffer or depth buffer), when an object is rendered by a 3D graphics card (or 3D software). The larger the depth (distance), the higher the z-coordinate/pixel value. In ESRI ArcGIS, this depth image is created by colouring the extrusion model based on the distance from observer point to each feature of the extrusion model (each element

as result from the intersection with the rasterized elevation model and building footprints); this distance is added as temporary field to the extrusion model. The extrusion model is coloured using graduated colours based on the distance field; Figure 45 shows the result for digital photo 1010. Next, the scene is exported and reclassified to three distance object classes, i.e. near distance objects, mid distance objects, and far distance objects. After raster vector conversion is applied on the depth image, per distance class an intersection is carried out on the vectorized virtual abstraction resulting in three layers that allow to be labelled with different text size (for this example the photo is labelled with object identifiers) and symbology as shown in Figure 45.

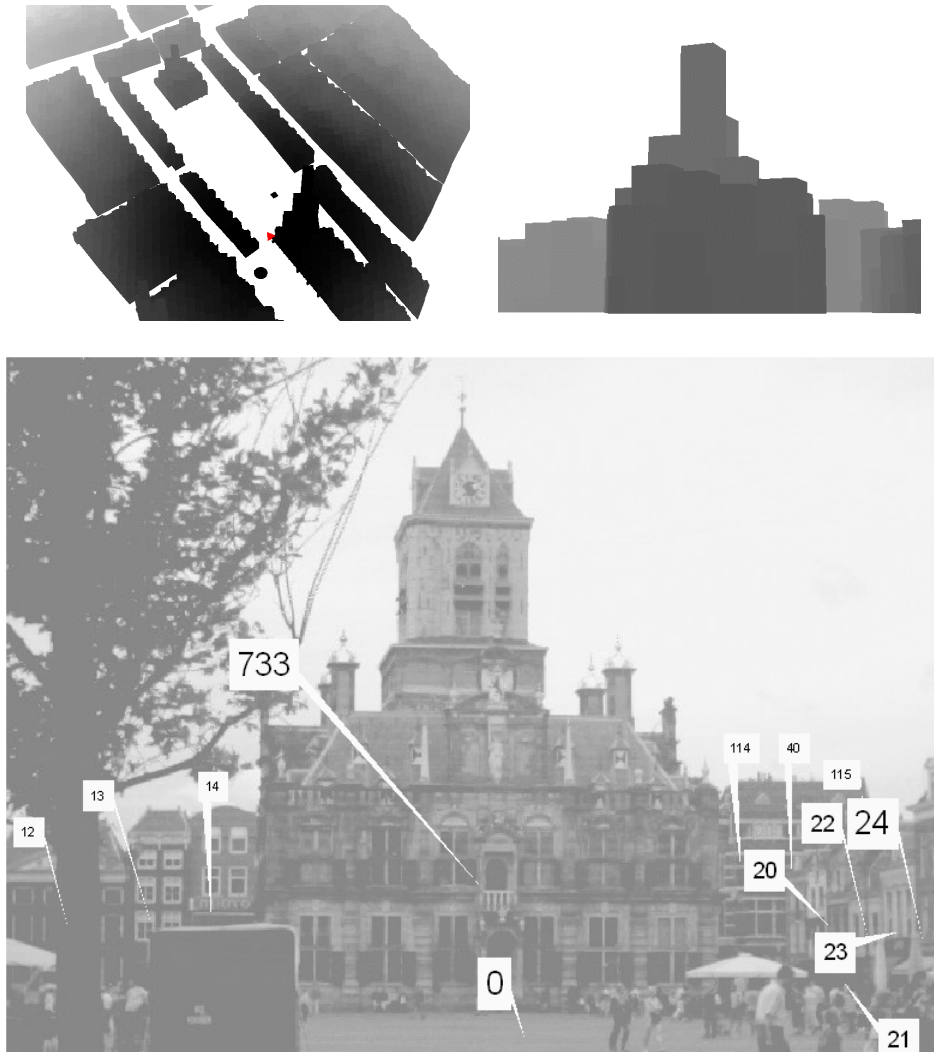


Figure 45 The photo is labelled with varying font size depending on distance from observer point; the extrusion model is coloured based on the subject distance in ESRI ArcMap and enables to create a depth map using ESRI ArcScene.

7.4 Prioritization of objects

From user requirements scoping (Tripod), it is known which sort of labels users and photographers prefer to annotate or to get with a photo. From this user studies it is among others obtained that people particularly prefer to annotate photos with tags as subject names, city names and country names. It is obvious that these preferences depend on user profiles, e.g. tourist are interested in historic and

cultural buildings, building inspection authorities prefer house or parcel numbers, biologists are interested in fauna (e.g. tree) names. For labelling in ESRI ArcMap, using the Maplex label engine a ranking could be applied on the available datasets. This enables to place first the labels; as this proof-of-concept mainly included the annotation of building features, this option is not further examined.

7.5 Number of labels to place inside a photo

The number of labels that good be placed inside a photo depends on several factors, i.e.

1. The size of empty areas, i.e. areas where no objects-of-interest are visible inside the photo.
2. The number of objects that are visible inside a digital photos
3. The font size for labelling
4. The size of the features captured inside the photo.
5. The preferences of a user

For this proof-of-concept, digital photo 1019 is labelled with shop names. By varying the minimum feature size (based on polygon perimeter) in order to place a label next to it, the amount of labels varies as is shown in Figure 46.

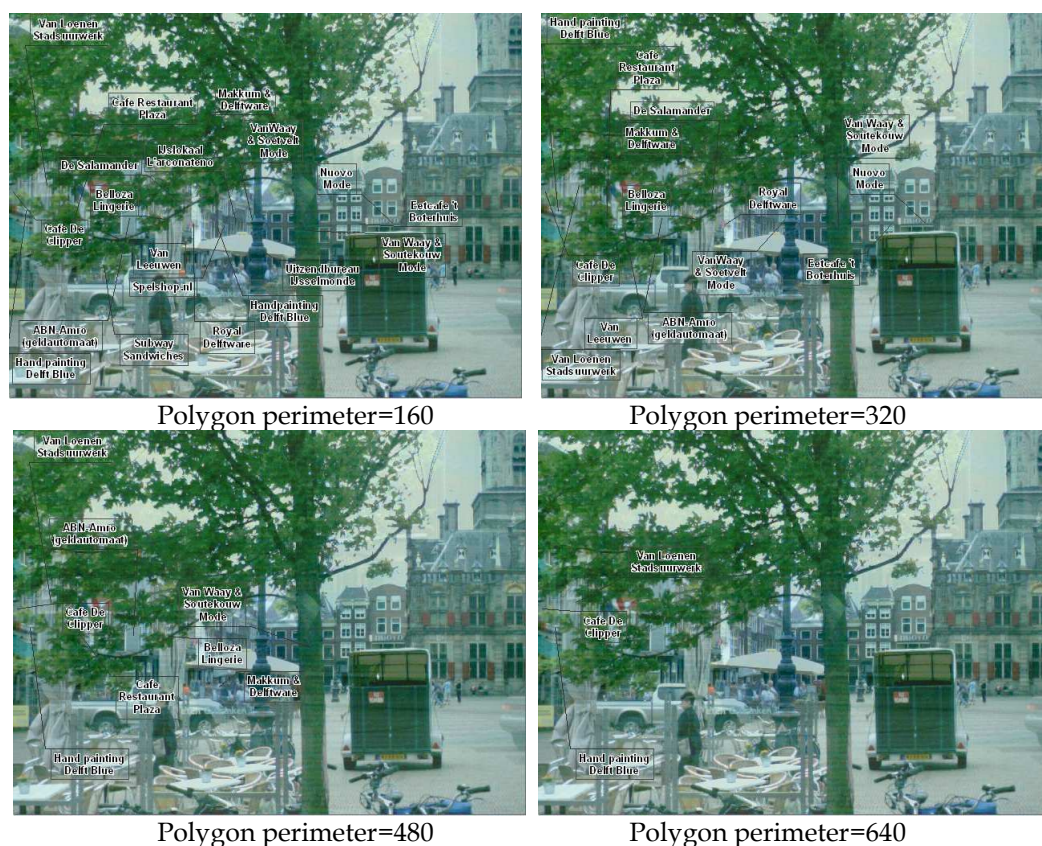


Figure 46 Digital photo 1019 labelled with different amount of objects based on the minimum feature size for labelling.

7.6 Discussion of results.

This proof-of-concept only provided preliminary constraints and rules to insert labels inside three-dimensional georeferenced and oriented digital photos. The issue with label placement is that it is an optimization problem and that it can be improved continuously. That implies that further research to which constraints and rules could be further applied to the label algorithm for label placement inside digital photos could deliver better results than obtained at this proof-of-concept. Particularly the inclusion of user profiles and multiple object types (as water, railways and so forth) with respect to prioritization can extend and improve the label placement algorithm e.g. placing curved labels inside rivers and roads, use perspective or sloped labels and so forth.

Chapter 8

Effects of lens distortions and positioning and orientation inaccuracies

This chapter describes (mis)identification of objects and (mis)placement of labels inside 3D georeferenced and oriented digital photos caused by lens distortions and inaccuracies in positioning and orientation. Section 8.1 describes the effects of lens distortions and section 8.2 includes the consequences of inaccuracies in position and orientation. Section 8.3 discusses the results.

8.1 Lens distortions

It is expected that the use of different lenses cause misidentification of objects and misplacement of labels due to lens distortions. Lens (or image) distortion is a deviation from rectilinear projection, i.e. the projection in which straight lines in a scene remain straight in a photo, and it depends on the zoom (or related; the view angle).. Distortions can be irregular and regular; due to the symmetry of the photographic lens, usually a regular radially symmetric distortion is encountered for digital photographs. Distortions are most visible in images with perfectly straight lines, especially when these lines are close to the edge of the image frame. Two types of radial distortions are distinguished i.e.

- *Pincushion distortion* is a lens effect, which causes images to be pinched at their centre. Pincushion distortion is associated with tele lenses and typically occurs at the tele end of a zoom lens. Consumer cameras have a typical pincushion distortion value of 0.6%.
- *Barrel distortion* is a lens effect, which causes images to be spherised or "inflated". Barrel distortion is associated with wide-angle lenses and typically occurs at the wide end of a zoom lens. Consumer cameras have a typical barrel distortion value of 1.0% (dpreview.com, 2007).

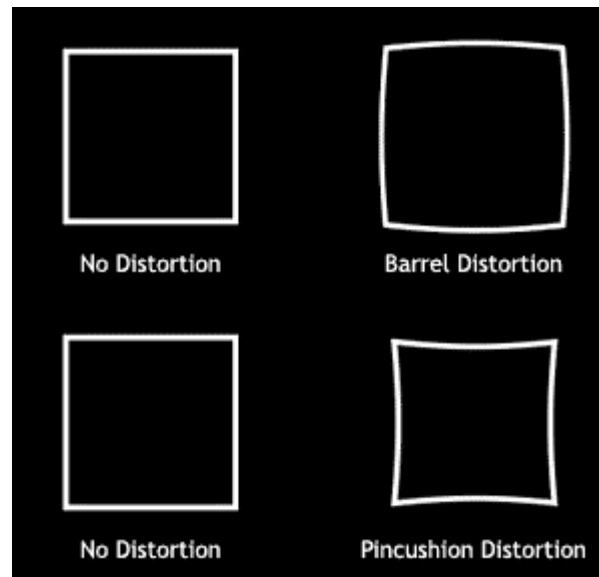


Figure 47 Barrel distortion and pincushion distortion (DPreview.com, 2007).

Barrel and pincushion distorted image from nearly all camera can be corrected using a image transformation function. The correcting function is a third order polynomial (Desch, 1999). It relates the distance of a pixel from the centre of the source image (r_{src}) to the corresponding distance in the corrected image (r_{dest}):

$$r_{src} = (a * r_{dest}^3 + b * r_{dest}^2 + c * r_{dest} + d) * r_{dest} \quad (\text{Eq. 12})$$

The parameter d describes the linear scaling of the image. For example using $d=1$, and $a=b=c=0$ leaves the image as it is. Choosing other d -values scales the image by that amount. a, b and c distort the image. Barrel distortion is counteracted using negative values for a, b and c , i.e. shifts distant points away from the centre. Using positive values shifts distant points towards the centre, which counteracts pincushion distortions. Correcting using $a \neq 0$ affects only the outermost pixels of the image, while using a value $b \neq 0$, the correction is more uniform. Finally, pincushion and barrel distortions might be corrected in the same image, i.e. if the outer regions exhibit barrel distortions, and the inner parts pincushion, a negative value for a and positive b values should be used. If no scaling of the image is advisable, the d -value should be chosen as such that $a + b + c + d = 1$. These transformations are implemented in numerous image processing software (e.g. Adobe Photoshop CS2).

To evaluate the effects of virtual abstractions matching the collected Nikon images are created and overlaid with the image. Because of the inaccuracies in heading, the view settings for the virtual abstractions are adapted as such they fit the pictured scene (see Figure 48). However, ESRI ArcScene renders the image taking into account lens distortion at different view angles:

As the angle increases, so does distortion until it appears as if you are looking through a fish-eye, or wide angle, lens.

(ESRI Desktop Help, 2007)

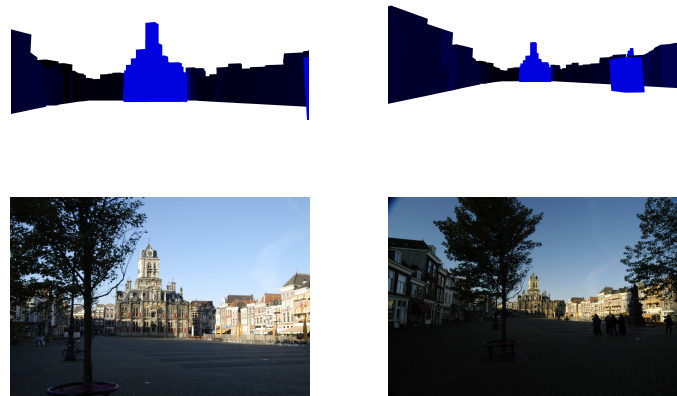


Figure 48 The virtual scenes are corrected for lens distortions by ESRI ArcScene when changing zoom or field -of-view angle.

This functionality is not included in all three-dimensional perspective viewer services. For example, the deegree WPVS does not correct for this lens distortion, so the question remains whether in that case the digital photo or the virtual abstractions should be corrected to enable object identification as good as possible and how this correction should be achieved.

What does cause serious misidentification of pictured objects (or at least should be taken into consideration) is the use of lenses that are from a different camera or lens manufacturers than the camera body its manufacturer. For example, using the focal length (e.g. 24mm) of the SIGMA lens to calculate the field of view causes a mismatch with actual pictured scene; however using a different focal length (e.g. 20mm) the exported scene fits with actual image (see Figure 48). This effect of differences in view angle, in this case connecting a SIGMA lens to a NIKON camera body, is well-known but it should certainly be taken into account when calculating the view angle from focal length and film size before creating the virtual abstraction of the pictured scene.

8.2 Positioning and orientation inaccuracies

Inaccuracies in positioning and orientation result in virtual abstractions to be created from a different point-of-view or directed towards a different object. Satellite positioning devices show particularly larger errors in build-up areas due to the multi-path effects (i.e. reflection of GPS signal by buildings (Husti, 2000)) and signal-blocking (i.e. obstacles within the line-of-sight from GPS satellites and receiver); causing accuracy-loss and loss-of-fix respectively. The compass inaccuracies are caused due to unexpected electromagnetic fields (i.e. not being the electromagnetic field of the Earth) and deviation in mounting the compass to the camera.

The effects of inaccuracies of compass and GPS of misidentification of objects inside a photo also depend on the subject distance D , focal length F and related view angle. The horizontal extent of the captured scene increases along with the

distance and depends on the horizontal view angle VA_{hor} and subject distance D , i.e.

$$Extent_{hor} = 2D \tan(\frac{1}{2}VA_{hor}) \quad (\text{Eq. 13})$$

The misidentification due to horizontal compass angle (heading) deviations $\sigma_{compass}$ also depends on the subject distance D when considered in absolute values, i.e.

$$M_{abs} = D \tan(\sigma_{compass}) \quad (\text{Eq. 14})$$

However, in terms of percentages of the total pictured extent, the misidentification due to heading deviations does not depend on the subject distance D anymore, i.e.

$$M_{proc} = \frac{D \tan(\sigma_{compass})}{D \tan(\frac{1}{2}VA_{hor})} = \frac{\tan(\sigma_{compass})}{\tan(\frac{1}{2}VA_{hor})} \quad (\text{Eq. 15})$$

This implies for example that a compass deviation of $\sigma_{compass} = 2^\circ$ at a focal length of $f = 50.0\text{mm}$ ($VA = 39.6^\circ$) results in a mismatch of the virtual abstraction with the actual pictured scene (and objects) of 10 %. So, if an object is smaller then the misplacement in terms of percentages, there exist no overlap between the actual image and the virtual abstraction and full misidentification is the result.

Table 8 Misidentification of pictured scene and objects in terms of percentages due to compass inaccuracies.

Focal length F	$\sigma_{compass} = 1^\circ$	$\sigma_{compass} = 2^\circ$	$\sigma_{compass} = 5^\circ$
F=24.0mm	2.3%	4.7%	11.8%
F=50.0mm	5.0%	10.0%	25.0%
F=80.0mm	7.8%	15.5%	38.8%

From the percentages of Table 8 it is concluded that the larger the focal length (i.e. more zoom in) the higher accuracy of compass direction is required for good identification and localization of pictured objects. shows the course of misplacement in terms of percentages as function of the compass inaccuracies and the focal length.

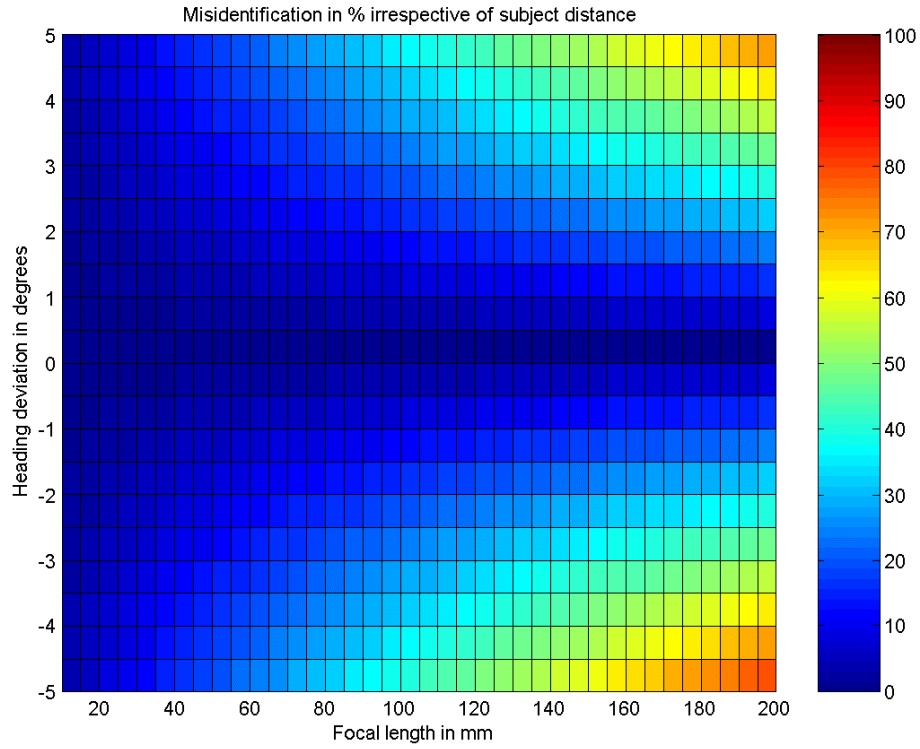


Figure 49 Misidentification of objects in terms of percentages as function of the compass inaccuracy and the focal length F .

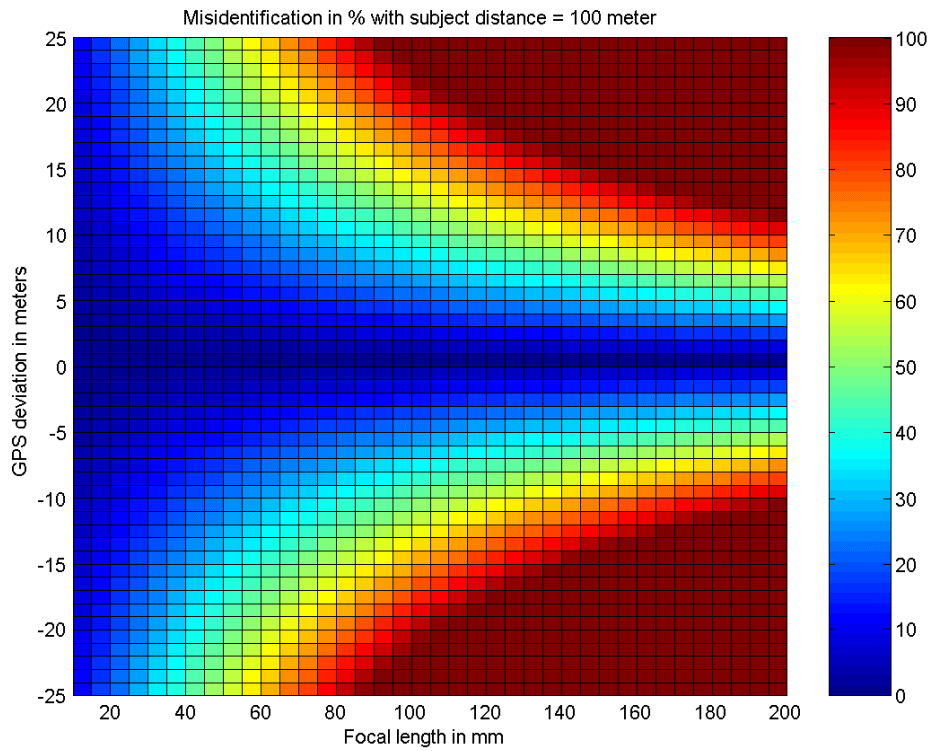


Figure 50 Misidentification of objects in terms percentages as function of GPS inaccuracy and focal length F at subject distance = 100 meter.

Analogously, the misidentification M due to GPS inaccuracy σ_{GPS} is given in terms of percentages of the total pictured scene as function of the subject distance D and horizontal view angle VA_{hor} , i.e.

$$M_{proc} = \frac{\sigma_{GPS}}{D \tan\left(\frac{1}{2}VA_{hor}\right)} \quad (\text{Eq. 16})$$

This implies that an inaccuracy of the GPS position of for example $\sigma_{GPS} = 10$ meters at focal length $f = 50.0\text{mm}$ results in the mismatch with the actual pictured scene of 55.5% at subject distance $D = 50$ m; and a mismatch of 27.8% at a distance $D = 100\text{m}$. Figure 50 represents the misidentification in terms of percentages as function of the GPS inaccuracy and the focal length at a subject distance of 100 meter.

Table 9 Misidentification of pictured scene and objects in terms of percentages due to GPS inaccuracy at subject distance $D = 50$ m.

Focal length F	$\sigma_{GPS} = 1\text{m}$	$\sigma_{GPS} = 5\text{m}$	$\sigma_{GPS} = 10\text{m}$
24.0mm	2.6%	13.3%	26.7%
50.0mm	5.5%	27.8%	55.5%
80.0mm	8.8%	44.4%	88.8%

From Eq 16 is obtained that the misidentification decreases with the distance from observer point to object. As the previous calculations assumed a GPS inaccuracy parallel to the focus plane, the effects of misidentification of objects due to errors in GPS position are evaluated using a random point feature class, created within a buffer of 10 meters around occupation point 1000 (see Figure 51). Next, for each random point the virtual abstraction for photo 1000 and 1012 is created using the erroneous capture locations.

From these virtual abstractions is concluded that the inaccuracies of GPS positioning are a major issue for the labelling service. Only capture location 4 and 5 show comparisons with the actual digital photo; camera position that due to positioning inaccuracies lie within a building cause no objects could be identified. Further research is required to how to manage these problems, e.g. a combination with vision technology abstraction to identify distinctive features and to match these with the virtual abstraction.

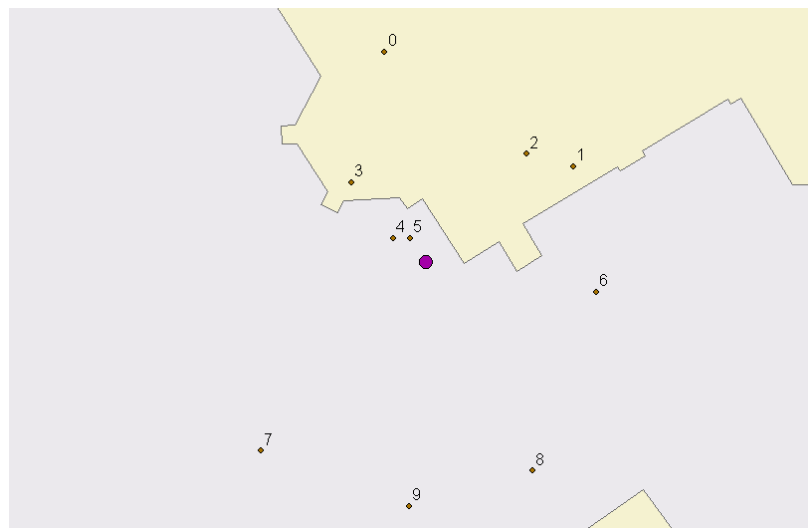
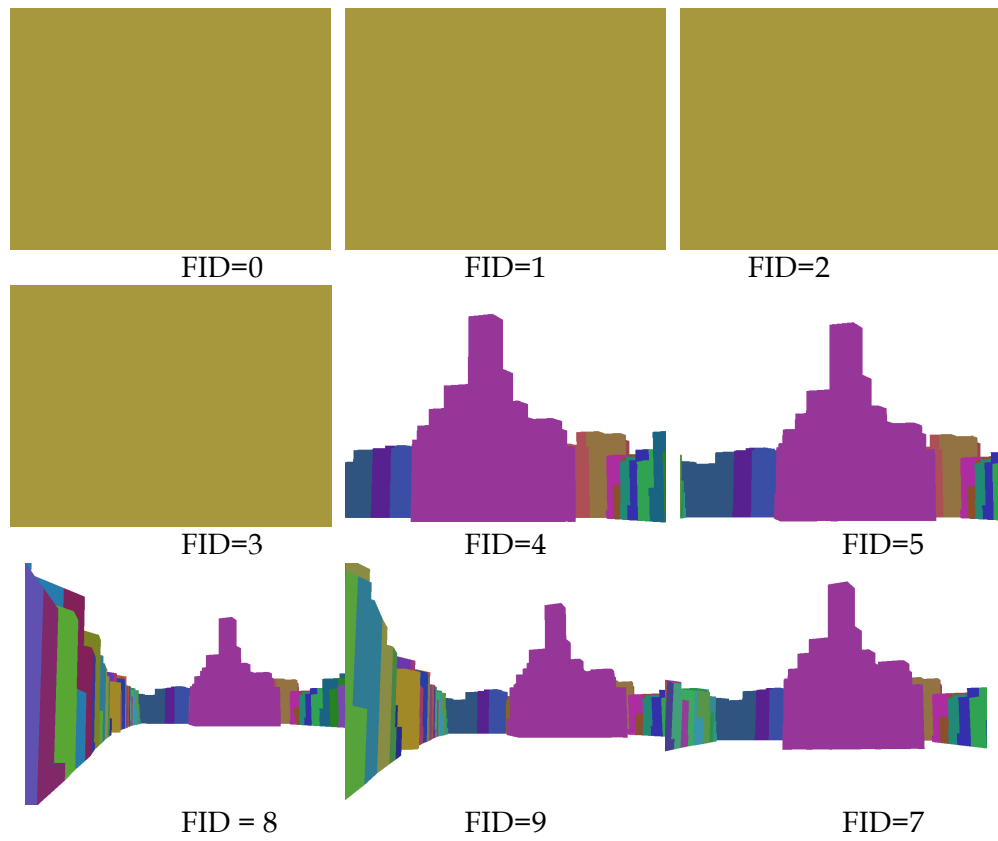


Figure 51 Different virtual abstractions due to GPS inaccuracies.

8.3 Discussion of results

This chapter described the effects of misidentification and it showed particularly erroneous positioning of GPS causes large deviations with respect to the actual digital photos. Therefore further research is required to how to manage the misidentification of objects due to compass and GPS inaccuracies. Due to these inaccuracies, virtual scenes are shifted or facing the wrong object. Using image-processing techniques, features from the input photo could be extracted and matched with the objects of the virtual scene. Pattern recognition and edge detection operators should be applied to transform (scale, rotate or shift) the virtual scene as such that in the end it matches the input photo.

Another effect that is expected to cause problems is the technology of image stabilization. Higher-end binoculars and zoom or telephoto lenses for SLR cameras often come with image stabilization which helps to steady the image projected back into the camera by the use of a "floating" optical element. It is often connected to a fast spinning gyroscope which helps to compensate for high frequency vibration (hand shake for example) at these long focal lengths. However; it is expected it causes errors with respect to the compass measurement and the scene pictured. Also this might be subject of further research.

Currently the degree WPVS does not allow the correction for barrel and distortion and pincushion distortion. It is recommended to further research whether it is possible to implement this 'distortion' using the field-of-view parameters; or that it is more advisable to correct the input digital photo.

Chapter 9

Automation and implementation

This chapter describes how the proof-of-concept of label placement inside digital photos is further worked out. Section 9.1 includes how the process of object identification and label placement is automated in ESRI ArcGIS using ArcObjects and Visual Basic for Applications. Section 9.2 includes how such a photo labelling service could be made available to the rest of world by proposing an implementation design. Finally, the results are discussed in section 9.3.

9.1 Automation using ArcObjects and VBA programming in ESRI ArcGIS

In ESRI ArcGIS, the proof-of-concept is automated using ArcObjects and Visual Basic for Applications (VBA) programming and forms to be able to create the virtual abstraction in a fast and easy way. ArcObjects is the development platform for ArcGIS, built using the Microsoft Component Object Model technology (ESRI, 2007). VBA is a COM-compliant programming language and is already embedded in ArcGIS. Using the Customization tool, buttons are added to a new-created toolbar in ESRI ArcScene and ESRI ArcMap. These buttons open a form to fill in among others the photo file name, folder name, observation point, view direction and angles and the objects-of-interest to label.

Figure 52 shows a screenshot of ESRI ArcScene and the form, designed for this research. After opening the project and loading the extrusion model, the form is launched and after it is filled in the VBA-script is run. In sequence it

1. *Loads the photo using the `OpenRasterDataSet()`-function:* this is done for visualization purpose and to check the aspect ratio of the digital photo. Depending on the aspect ratio, i.e. 4:3 or 3:2 implemented, the VBA-script decides which hidden viewer, i.e. 640x480 or 640x427 respectively, to create and export the virtual abstraction.
2. *Sets the view using the `ICamera Interface`:* the parameters from the form are used to set the camera properties. Because the renderer creates the view from the target point to the observation point, the target point is computed from the observer point and the view directions.

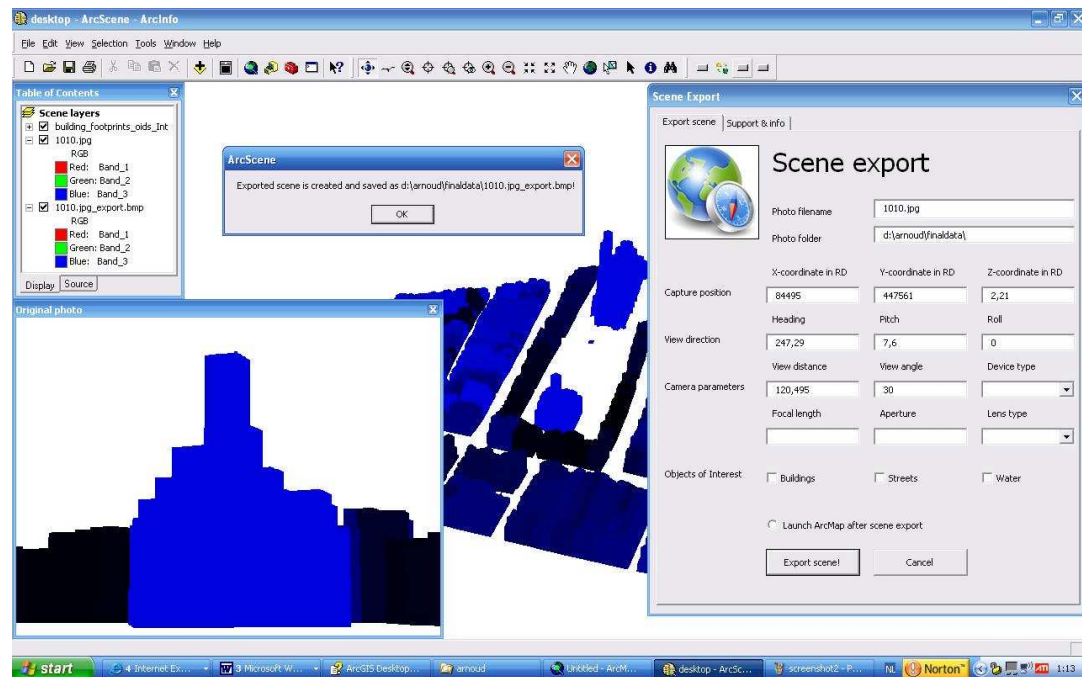


Figure 52 Screenshot of the demo-application build in ESRI ArcScene using VBA.

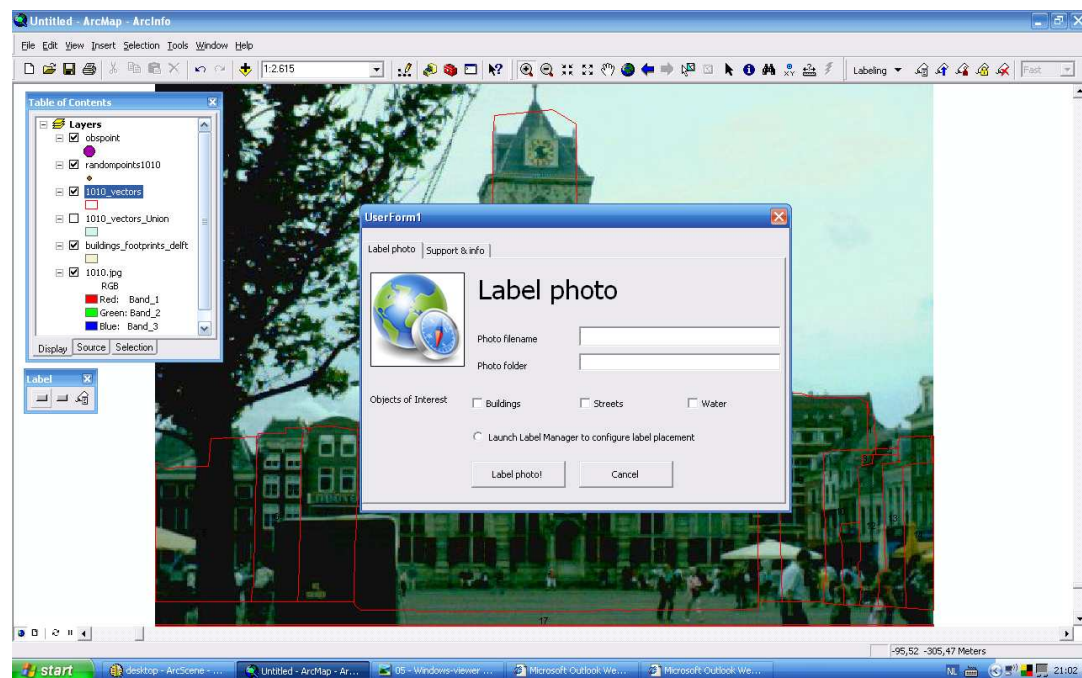


Figure 53 Screenshot of the demo-application build in ESRI ArcMap using VBA.

3. *Exports the scene using the GetSnapshot() function:* GetSnapshot is used to export as the IExport Interface only allows to export the main viewer. Because the size of the scene to be exported depends on the aspect ration, the GetSnapshot() function is applied to the active viewer.

Disadvantage of using the GetSnapshot()-function is that it only supports the image to be exported in file formats BMP and JPEG; as JPEG is a lossy compression and the crisp boundaries should be maintained, a scaled (related to photo dimensions) snapshot is saved as BMP.

Figure 53 shows the form created in ESRI ArcMap for object delineation and linking the exported scene to the 2D GIS data. After opening the project and loading the 2D data sets, the form is launched and after it is filled with the required parameters, the VBA-script is run. In sequence it

1. *Opens the digital photo and the exported scene using the OpenRasterDataSet()-function:* the digital photo is used for visualization, the exported scene is opened for further processing.
2. *Calculates the weighted sum of the RGB colourbands:* using the IRasterBand Interface ces the RGB exported scene is split up in a Red, Green and Red colourband. Next a weighted sum is applied: two constant rasters with values 256 and 65536 are created using the RasterMaker()-function and using IMathOp en RasterMathOp the rasters and colourbands are multiplied and summed to obtain one colourband having pixel values that equal the object identifiers.
3. *Converts the summed colourbands to polygon features using the IConversionOp Interface:* the colourband obtained from the previous step is converted to polygon features without simplification. The pixel values are added to new field GRIDCODE and the features are saved to a new Feature Class. Afterwards, the feature class is added as new layer to the project.
4. *Links the vectorized exported scene to the 2D spatial data sets using JoinToLayer()-function:* this function joins the dataset on GRIDCODE and object identifier field respectively. Next, the Label()-function enables to place labels inside the identified objects.

Afterwards, the ESRI Label Manager together with ESRI Maplex Label Engine enables to configure the label algorithm to place the labels in the best-preferred place; however, this is not implemented further. The VBA-script can be found in Appendix C.

It is noteworthy that a Web Perspective Viewer Service (WPVS), which generates a scene image based on specified view parameters, could replace the steps carried out in ESRI ArcScene. If ESRI ArcMap allows to import an virtual abstraction as output from WPVS-request, the photo labelling service could be fully executed in ESRI ArcMap, ESRI ArcMap already allows to import data from WMS and WFS sources, so compatibility with WPVS is expected as a WPVS-request returns

actually the same as an WMS, namely a raster image (e.g. PNG) and a WMS is actually a WPVS with fixed pitch angle of 90°.

9.2 Implementation of web or location-based service

Eventually, the photo labelling proof-of-concept could be made available to the rest of the world via a web service or location-based service. For example, a web service would enable users to upload a digital photo to an online image hosting service and after the user requests to label the photo; the photo is shown labelled on his/her computer screen. A location-based service should enable users to send their photo in the field via a GSM/GPRS or UMTS connection and a photo labelling service would return a (compressed) labelled photo. This section proposes an overall process flow of such a web or location-based service as shown in Figure 54.

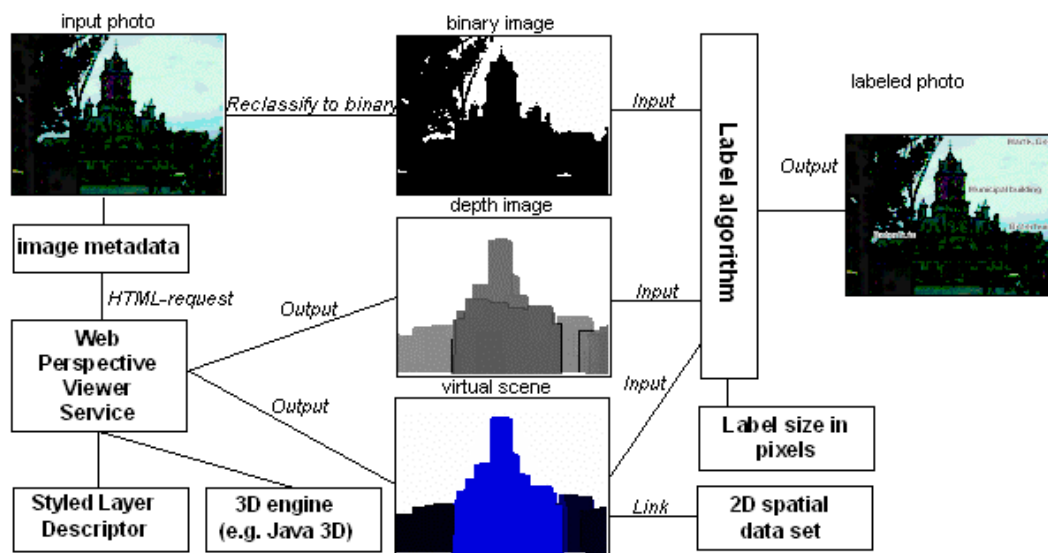


Figure 54 Overall process of a photo labelling service.

9.2.1 Image metadata

For an automatic photo labelling service, the view parameters should be automatically extracted from the image metadata after the digital photo or metadata is uploaded to a web service (e.g. an image hosting site). As current image metadata formats applied by camera manufacturers do not allow storing view direction parameters along with the digital photo, the Tripod research project is exploring the possibilities of developing a new direction-compatible metadata standard (probably a combination of IPTC and Exif, under development by the Dublin University). The image metadata parameters are used as input for the web request for the WPVS.

9.2.2 Web Perspective Viewer Service

Core of this service is a web perspective viewer service (WPVS) for creating the virtual abstraction and to link it to the 2D data set. An implementation of a WPVS is available at the degree.org web service project. A WPVS visualizes data from online web services in a perspective view, i.e.

1. feature data (e.g. shapefiles) from a Web Feature Service (WFS),
2. raster maps as aerial images from a Web Map Service (WMS) and
3. other raster coverages e.g. DEM from a Web Coverage Service (WCS).

The deegree.org WPVS supports as input feature data in CityGML format, which is open data model and XML-based format for the storage, representation and exchange of three-dimensional virtual urban models. The extrusion model of the vectorized elevation model intersected with the two-dimensional footprints should be converted into this CityGML to serve as data source for WPVS. The appearance of spatial datasets could be applied using a *Styled Layer Descriptor (SLD)*, which enables to colour each object to each object identifier. The requested virtual scene is returned in a particular image file format (e.g. PNG or JPEG) after the user send a HTML-request e.g.

```
http://localhost:8080/deegree-wpvs/services?BACKGROUND=cirrus&
ELEVATIONMODEL=saltlakedem&CRS=EPSG:26912&YAW=0&REQUEST=GetView&OU
TPUTFORMAT=image/png&DATETIME=2007-03-21T12:00:00&PITCH=20&DISTANCE=
1550&VERSION=1.0.0&AOV=60&ROLL=0&POI=424994.4,4513359.9,1550&BOUNDINGBO
X=424585.3,4512973.8,425403.5,4513746.0&SCALE=1.0&SPLITTER=BBOX&HEIGHT=600&
DATASETS=Utah_Overview,satelite_images&STYLES=default&FARCLIPPINGPLANE=11
000&WIDTH=800&EXCEPTIONFORMAT=INIMAGE
```

(degree.org, 2007)

The creation of a depth image using the WPVS is currently not been implemented. However; the deegree WPVS renderer uses the Java3D which should enable to deliver this depth map as it uses this to detect what is visible and what is not. It is a typical software developer's issue to solve this problem in order to get the depth image along with the virtual scene from the WPVS. The deegree WPVS does not include the correction for the barrel and pincushion distortion when changing the field-of-view angle; also this typical a problem that needs to be solved by software engineers.

9.2.3 Input images for labelling

In general, three output images are required to continue applying the label algorithm, i.e.

1. *object image*: output from the WPVS based on a SLD that colours each object based on its object identifier
2. *depth image*: output either obtained from the renderer or created using a SLD that colours each object based on its distance from observer to object.
3. *binary image*: output from a reclassification of the original photo

9.2.4 Label algorithm

Next, a GRID-based label algorithm could be applied. A GRID-based label algorithm is easier to implement then feature-based labelling; the proof-of-concept applied a feature-based labelling due to inability of ESRI ArcGIS to label raster cells. Such GRID-based label algorithm could do the following process, i.e.

1. *draw a buffer around the object image*: this buffer around the objects captured inside the image specifies an offset for label placement to enable external annotation using connectors.
2. *identify which and how many objects are visible in the object image*: loop through the array of pixels of the object image and save all distinct pixel values to a temporary file.
3. *localize the objects by calculating e.g. the object bounding boxes*: loop through the array of pixels of the object image and save for each identified object its maximum and minimum row and column values.
4. *suggest a label position for the visible objects*: loop through all the visible objects starting with the most important or largest object; suggest a position (e.g. upper right) and calculate the label size (using the Java for this position (or object position). The font size of the label depends on the corresponding pixel value of the depth image.
5. *verify if this label position causes conflicts*: check if the pixels that the label would intersect are empty areas, i.e. areas where the binary image and where no labels are placed yet.
 - a. if yes, hold this location as temporary label position and continue with next object from 4.;
 - b. if no, suggest another location (e.g. upper left) and restart for this object from 4.
6. *place labels inside the image*: after all -or as much as possible- labels have suggested places inside the image, continue to change temporary locations to definitive locations and as such finish the labelling.

This label algorithm is just an example how all three output images could be applied to identify the best location to place a label inside a digital photo. Normally, the starting location for a label algorithm would be the bottom-right to label an object.

For digital photos, it is expected the empty areas (i.e. sky) are most likely to be located above the object interest. Therefore, it is more likely to suggest a label place in the following e.g. upper-right, upper-left, upper-centre, bottom-right, bottom-left). But of course also additional analysis could be applied to identify where with respect to the object the empty areas located before a label suggestion algorithm is chosen.

9.2.5 Output image

Finally, the labelled photo is returned to the user, either in PNG or SVG, depending whether or not the photo labelling service should allow dynamic annotation, i.e. varying label size when zooming on the image.

9.3 Discussion of results

For this proof-of-concept, the object identification and label placement is automated in ESRI ArcGIS using Visual Basic for Applications. The steps involved for ESRI ArcScene could be replaced by a OGC web perspective viewer service (WPVS). Currently, ESRI ArcGIS enables to import output from Web Map Servers (WMSs) and Web Feature Servers (WFSs) so it is expected in the end ESRI ArcGIS will also support the WPVS output to import immediately in the project environment.

In order to apply the proposed process flow, a depth map is required. Question is how to automatically get a depth map along with the virtual scene from the perspective viewer service as the deegree WPVS does not supports his. A depth map show the distance from observer (or camera) to object and is generated by the renderer to detect visibility of surfaces and objects. For this proof-of-concept, the depth map is created by adding a distance field to the three-dimensional (extrusion) model from one particular observation point and the model is coloured with graduated colours based on this distance field. However, because the depth map is generated at the rendering stage (by hardware or software e.g. Java3D), it is expected it is possible to get this depth map along with the perspective view in an automatic way.

Chapter 10

Conclusions

This research discussed the label placement inside 3D georeferenced and oriented digital photos using GIS technology and this chapter provides the conclusions of this study. Section 10.1 discusses the research outputs and conclusions are drawn in section 10.2. Finally, recommendations for further research are given in section 10.3.

10.1 Discussion

This research proposed to use GIS technology as an alternative for computer vision technology and photogrammetric coordinate transformation to identify and localize visible objects inside 3D georeferenced and oriented digital photos to be able to associate labels with pictured objects. Using three-dimensional spatial models, a virtual scene matching the digital photo is created and linked to the objects inside the model using RGB colour values based on object identifiers.

Object identification (or recognition) using image-processing tools (e.g. pattern recognition and edge operators) is a complex matter as extracted features need to be match with a detailed model having for each object detailed geometric constraints. These models only contain geometric constraints referring to a type of object and not to individual object characteristics. As such, the object recognition becomes actually an object classification. Spatial datasets and three-dimensional (building) models do contain individual characteristics of real world features, but as these models contain a lot of objects, the application of computer vision technology to match the objects with this three-dimensional model to recognize pictured objects becomes a very computational task.

Disadvantage of the conversion of pixel coordinates to terrain coordinates is that it not only requires information about capture position and view direction; also the camera calibration needs to be carried out and known very well. Thereby, camera calibration of average camera devices needs to be known very accurately in order to carry out this conversion in order to link the pixel-coordinates to terrain coordinates, and thereupon digital maps.

Using the full spatial image metadata, i.e. 3D position and orientation information (automatically) captured along with the digital photo, to perform a request on a perspective viewer service to create a virtual scene, a perspective view of a three-dimensional model is obtained that matches the pictured scene. As (three-dimensional) spatial data becomes increasingly available from different sources, this approach is considered to be very appropriate to apply for image object identification. Advantage is that it can be applied for average camera devices extended with a positioning and direction measurement device (without accurately knowing the camera calibration parameters). Also it is expected to provide better, faster and more reliable results for object identification than computer vision technology. Because implementation of this proof-of-concept and proposed system architecture using a web perspective viewer service reduces the computational load at the client side, allowing average users to easily label or caption their digital photos resulting in improving the accessibility of online image content.

10.2 Conclusion

The goal of this research is to place labels inside three-dimensional georeferenced and oriented digital photographs using geographic information technology and spatial data sets. The main research question is:

How to identify captured objects and where to place labels to annotate pictured objects inside 3D georeferenced and oriented digital photos using GIS technology?

With respect to this research question it is concluded that

- The output from a perspective viewer service, which renders a three-dimensional geographic model in a perspective view, is very appropriate for identifying and localizing visible objects inside a digital photo.
- The virtual scene is successfully linked to the spatial features from the source data set when colouring the objects of the three-dimensional model with RGB colour values related to their object identifiers (OIDs).
- An average two-dimensional label algorithm could be applied to label the virtual scene and as such the problem of label placement in a three-dimensional environment is reduced to a two-dimensional map labelling solution.
- That the best location for a label inside a digital photo is at the empty areas, avoiding overlap of labels with objects and labels mutual. This empty area is detected from the virtual scene and the reclassified-to-binary image of the input photo.
- A depth image contains is an appropriate rendering output to be applied to decrease label font sizes with increasing object distance to maintain the perspective view of the photo.

- The amount of labels that should be placed inside a digital photo depends on the size of empty areas, the number and size of visible objects captured inside the photo and user preferences.
- Barrel and pincushion lens distortions do not cause misidentification of objects as the perspective viewer service corrects for this distortions if the field-of-view angles is changed.
- Compass and GPS inaccuracies are the main cause of misidentification of objects related as such that a smaller field-of-view angle implies a larger misidentification in terms of percentages at a constant compass or GPS inaccuracy.
- The proof-of-concept of this photo labelling research should be implemented using an OGC web perspective viewer service (WPVS) and a GRID-based label algorithm applied to the virtual scene, depth image and binary image.

10.3 Recommendations

Further research is recommended to the following issues, i.e.

- *How to apply the extrusion model from a vectorized digital elevation model intersected with two-dimensional footprints for larger spatial datasets.* Vectorization of an elevation raster results in a large amount of features because for almost each pixel of the elevation model, a feature is created with a height attribute. For this research, a spatial extent of only several 100 square meters is applied resulting already in an extrusion model having over 10,000 features. As the number of features increases, the processing and rendering of the data slows down; and applying this extrusion model to a larger extent (e.g. country) is expected to cause serious problems. So, the central issue is how to manage this large number of features or how to limit the amount of features to be rendered in the perspective viewer e.g. using a sub-selection of the whole spatial dataset based on a region polygon (i.e. two-dimensional footprint of the visible extent) before colouring the extrusion model and creating the virtual scene.
- *How to mount a three-axis digital compass to a camera to avoid as much as possible errors of electromagnetic fields and how to save the view direction parameters along with the captured photo.* The electromagnetic field of camera as it is a typical issue for (camera) manufacturers to deal with to limit compass errors as much as possible. For this research, the image metadata is stored in a separate text file; however, to be able to automatically record the view direction, an image metadata standard should be developed and accepted to store full spatial metadata parameters.
- *How to automatically get a depth map along with the virtual scene from the perspective viewer service.* A depth map shows the distance from observer (or camera) to object and is generated by the renderer to detect visibility of surfaces and objects. For this proof-of-concept, the depth map is created by

adding a distance field to the three-dimensional (extrusion) model from one particular observation point and the model is coloured with graduated colours based on this distance field. However, because the depth map is generated at the rendering stage (by hardware or software), it is expected it is possible to get this depth map along with the perspective view in an automatic way.

- *Which constraints and rules could be further applied to the label algorithm for label placement inside digital photos.* This research configured the label algorithm as such it places. It also explored to vary in font sizes of labels depending on the distance from the corresponding object to the camera or observer position. But more constraints and rules could be added that improve the result of the photo label placement, e.g. curved labels inside rivers and roads, use perspective or sloped labels and so forth. This proof-of-concept provided some preliminary constraints and rules for a photo label algorithm; further development and implementation of label constraints is expected to improve the appearance of labels inside a digital photo.
- *How to manage the misidentification of objects due to compass and GPS inaccuracies.* Due to these inaccuracies, virtual scenes are shifted or facing the wrong object. Using image-processing techniques, features from the input photo could be extracted and matched with the objects of the virtual scene. Pattern recognition and edge detection operators should be applied to transform (scale, rotate or shift) the virtual scene as such that in the end it matches the input photo.

References

Adobe.com, 2007. Homepage of Adobe, <http://www.adobe.com/>. Last visited: March, 2007.

Asperen, P. and Kibria, M.S., 2007, Comparing 3D Earth Viewers. GIM International, November 2007, Volume 21, Issue 11.

Bell, B. S. Feiner, and T. Höllerer, 2001, View Management for Virtual and Augmented Reality. In: Proceedings of the 14th annual ACM symposium on User interface software and technology, Orlando, Florida, 2001 pp. 101 – 110.

Cartwright, W., G. Gartner and M.P. Peterson, 2007, Multimedia Cartography Second Edition. Berlin: Springerlink-Verlag Berlin Heidelberg.

Castelli, V. and D. Bergman, 2002, Image Databases: Search and Retrieval of Digital Imagery. New York: John Wiley & Sons, Inc.

Chang, E.Y., 2005, EXTENT: Fusing Context, Content, and Semantic Ontology for Photo Annotation. In: ACM SIGMOD CVDB Workshop, June 2005.

Chang, K., 2004. Programming ArcObjects with VBA: Using a Task-oriented approach. UK: CRC Press.

Christensen, J., J. Marks, and S. Shieber. 1992. Labeling point features on maps and diagrams. Center for Research in Computing Technology, Harvard University, TR-25-92, December. <http://citeseer.ist.psu.edu/christensen92labeling.html>

Crispen.org, 2007. Homepage of GIF, JPEG or PNG?, <http://web-building.crispen.org/formats/>. Last visited: November 2007.

Cyclomedia.org, 2007. Homepage of Cyclomedia, <http://www.cyclomedia.nl>. Last visited: August 2007.

Davis, M. S. King, N. Good and R. Sarvas, 2004a, From Context to Content: Leveraging context to infer media metadata. In: Proceedings of the 12th International Conference on Multimedia (MM2004), pages 188-195. ACM Press, 2004

Deegree.org, 2007. Homepage of Deegree.org, <http://www.deegree.org/>. Last visited: December 2007.

Desch, 1999. Homepage of Helmut Desch, Technical University Furtwangen, <http://www.all-in-one.ee/~dersch/barrel/barrel.html>. Last visited: November 2007.

Dias, E., A. de Boer, S. Fruijtier, J.P. Oddoye, J. Harding, C. Matyas and S. Minelli, 2007, Requirements and business case study report (Tripod deliverable).

Dofpro.com, 2007. Homepage of Depth of Field Pro, <http://www.dofpro.com/>. Last visited: December 2007.

DPreview.com, 2007. Homepage of Digital Photography review, <http://www.dpreview.com/>. Last visited: December 2007.

ESRI Desktop Help, 2006.

Exif.org, 2007. Homepage of Exif.org, <http://www.exif.org>. Last visited: November 2007.

GeoTIFF, 2007. Homepage of GeoTIFF, <http://remotesensing.org/geotiff/>. Last visited: March, 2007.

Götzelman, T., K. Hartman, and T. Strothotte, 2006, Agent-based annotation of Interactive 3D Visualizations. In: 6th International Symposium on Smart Graphics, Vancouver, Canada, July 2006 pp. 24-35.

Hagedorn, B., S. Maass, and J. Döllner, 2007, Chaining Geoinformation Services for the Visualizarion and Annotation of 3D Geovirtual Environments.

I3A.org, 2007. Homepage of I3A, <http://www.i3a.org/>. Last visited: December 2007.

ISOTC211.org, 2007. Homepage of ISO/TC211 Geographic Information/ Geomatics, <http://www.isotc211.org>. Last visited: December 2007.

Kolbe, T.H., 2007, Augmented Videos and Panoramas for Pedestrian Navigation. In: Proceedings of the international conference on Advances in computer entertainment technology.

Li, J., C. Plaisant, and B. Schneiderman, 1998, Data object and label placement for information abundant visualizations In: Proceedings of the 1998 workshop on New paradigms in information visualization and manipulation, Washington D.C. United States, 1998 pp. 41-48.

Lillesand, T.M. and R.W. Kiefer, 1994, Remote sensing and Image Interpretation Third Edition. New York: John Wiley & Sons, Inc.

Maass S. and Jürgen Döllner, 2006, Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes. In: 6th Int. Symposium on Smart Graphics, Vancouver, Canada, July, 2006, 1-12.

Maass, S. and J. Döllner, 2006a, Dynamic Annotation of Interactive Environments using Object-Integrated Billboards. In: Proceedings 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'2006, Plzen, Czech Republic, Jan/Feb 2006, 327-334.

Microsoft, 2007. Homepage of Microsoft Live Labs PhotoSynth, <http://labs.live.com/photosynth>. Last visited: December 2007.

Naaman, M., S. Harada, Q.Y. Wang, H. Garcia-Molina, and A. Paepcke, 2004, Context Data in Geo-referenced digital photo collections, In: Proceedings of the 12th ACM international conference on Multimedia (MM '04).

Naaman, M., S. Harada, QY, Wang, H. Garcia-Molina, and A. Paepcke, 2004b, Automatic Organization for Digital Photographs with Geographic Coordinates. In: Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries.

Nikon, 2007. Homepage of Nikon Japan, <http://www.nikon.jp/>. Last visited: November 2007

Nist.gov, 2007. Homepage of National Institute of Standards and Technology, <http://www.nist.gov/>. Last visited: November 2007.

NRI, 2007. Homepage of Nokia Research Center, <http://research.nokia.com/>. Last visited: August 2007.

O'Hare, N., C. Gurrin, G.J.F. Jones, and A.F. Smeaton, 2005, Combination of content analysis and context Features For Digital Photography Retrieval. In: 2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies.

O'Rourke, 2003, M., Principles of Three-Dimensional Computer Animation: Modelling, Rendering and Animating with 3D Computer Graphics. New York: W.W. Norton & Company, pp.83-145.

OS, 2007. Homepage of Ordnance Survey, Britain's national mapping agency, <http://www.ordnancesurvey.co.uk/>. Last visited: July 2007.

Panoramio.com, 2007. Homepage of Panoramio, <http://www.panoramio.com/>. Last visited: July 2007.

Schmalstieg, D. and G. Reitmayr, 2007, The World as a User Interface: Augmented Reality for Ubiquitous Computing. In: Location Based Service and TeleCartography, 2007, Springer, pp. 369-391.

Scotta, A., 1998, Visual Reality and Object identification on Spatially Referenced Panoramic Images. Master thesis.

- Topcon, 2007. Homepage of Topcon Europe, <http://www.topcon.eu>
- Toyama, K., R. Logan, A. Roseway, and P. Anandan, 2003 Geographic Location Tags on Digital Images. ACM Multimedia 2003, New York, October 2003.
- Toye, E., R. Sharp, A. Madhavapeddy, D. Scott, E. Upton and A. Blackwell, 2006, Interacting with mobile service: an evaluation of camera-phones and visual tags. In: Personal and Ubiquitous Computing, Jan 2006, pp. 1 - 10.
- Tripod, 2006, Description of Work of the TRI-Partite multimedia Object Description, November 2006.
- Tuffield, M.M., S. Harris, D.P. Dupplaw, A. Chakravarthy, C. Brewster, N. Gibbins, K. O'Hara, F. Ciravegna, D. Sleeman, N.R. Shadbolt, and Y. Wilks, 2006, Image annotation using Photocopain. In: Proceedings of the fifteenth world wide web conference (www06).
- W3.org, 2007. Homepage of W3C Semantic Web Activity, <http://www.w3.org/RDF/>. Last visited: August 2007.
- Watt, A., 1990, Fundamentals of Three-dimensional Computer Graphics (Second edition), Wokingham: Addison-Wesley
- Worboys M. and M. Duckham, 2004. GIS: a computing perspective. Second Edition. London: Taylor & Francis Ltd.
- WTC, 2007. Homepage of Wight Technology Consulting, http://www.wtc-consult.de/english/fr_e2.htm/. Last visited: November 2007.
- Yamamoto M. and L.A.N. Lorena, 2005. A Constructive genetic approach to point-feature cartographic label placement. Metaheuristics: Progress as Real Problem Solvers, New York: Springerlink.

Appendix A

Collection of Topcon images



ID	1010
POI_coord	(84385.39994 447515.7996 18.237352)
POC_coord	(84495.671 447561.9386 0.614378942)
Heading	247.29°
Pitch	7.60°
Roll	0°
Subject distance	120.495 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 14:04
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	1011
POI_coord	(84465.23614 447567.4608 7.6559733)
POC_coord	(84495.671 447561.9386 0.614378942)
Heading	280.28°
Pitch	9.92°
Roll	0°
Subject distance	31.39 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 14:06
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	1013
POI_coord	(84377.897 447528.1592 10.51891247)
POC_coord	(84495.671 447561.9386 0.614378942)
Heading	299.19°
Pitch	7.90°
Roll	0°
Subject distance	60.221 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 14:09
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	1018
POI_coord	(84493.8542 447568.2418 17.1027121)
POC_coord	(84495.671 447561.9386 0.614378942)
Heading	343.92°
Pitch	53.64°
Roll	0°
Subject distance	10.997 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 14:18
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	1014
POI_coord	(84443.57744 447591.042 2.48645391)
POC_coord	(84495.671 447561.9386 0.614378942)
Heading	106.50°
Pitch	0.73°
Roll	0°
Subject distance	25.042 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 14:11
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	1019
POI_coord	(84367.58406 447470.4546 9.810260)
POC_coord	(84495.671 447561.9386 0.61437894)
Heading	54.46°
Pitch	2.78°
Roll	0°
Subject distance	157.538 m
View angle	30°
Focal length	8 mm
Timestamp	8/8/2007 14:21
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	2010
POI_coord	(84387.56363 447517.0145 6.113621)
POC_coord	(84409.194 447510.0159 0.67719519)
Heading	287.93°
Pitch	9.53°
Roll	0°
Subject distance	23.032 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 14:57
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor





ID	2012
POI_coord	(84374.73525 447569.1629 3.929984)
POC_coord	(84409.194 447510.0159 0.67719519)
Heading	329.78°
Pitch	1.41°
Roll	0°
Subject distance	68.464 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 15:00
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor



ID	2011
POI_coord	(84385.23103 447516.2318 17.47391)
POC_coord	(84409.194 447510.0159 0.67719519)
Heading	284.54°
Pitch	28.48°
Roll	0°
Subject distance	28.071 m
View angle	30 °
Focal Length	8 mm
Timestamp	8/8/2007 14:58
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor

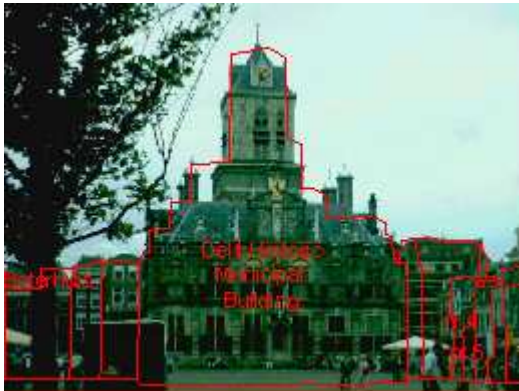


ID	2013
POI_coord	(84491.59859 447577.3564 67.93485)
POC_coord	(84409.194 447510.0159 0.67719519)
Heading	50.74°
Pitch	28.54°
Roll	0°
Subject distance	120.788 m
View angle	30°
Focal Length	8 mm
Timestamp	8/8/2007 15:03
Resolution	640x480
Device	Topcon GPT-7003i
Sensor	0.3M pixels (VGA) CMOS Sensor

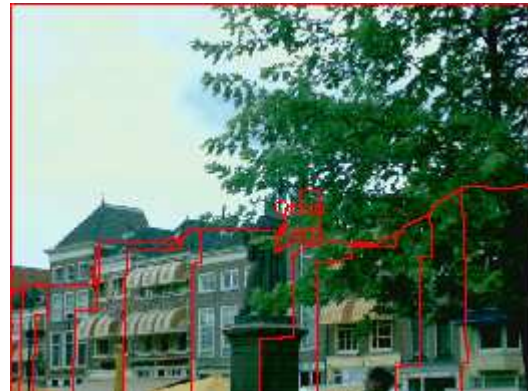
					
ID	2014		ID	2018	
POI_coord	(84485.89177 447579.9073 10.3164)		POI_coord	(84358.08686 447468.665 13.42763)	
POC_coord	(84409.194 447510.0159 0.6771951)		POC_coord	(84409.194 447510.0159 0.6771951)	
Heading	47.66°		Heading	51.02°	
Pitch	4.44°		Pitch	7.90°	
Roll	0°		Roll	0°	
Subject distance	104.027 m		Subject distance	66.601m	
View angle	30°		View angle	30°	
Focal Length	8 mm		Focal Length	8 mm	
Timestamp	8/8/2007 15:04		Timestamp	8/8/2007 15:10	
Resolution	640x480		Resolution	640x480	
Device	Topcon GPT-7003i		Device	Topcon GPT-7003i	
Sensor	0.3M pixels (VGA) CMOS Sensor		Sensor	0.3M pixels (VGA) CMOS Sensor	

Appendix B

Object identification of Topcon images



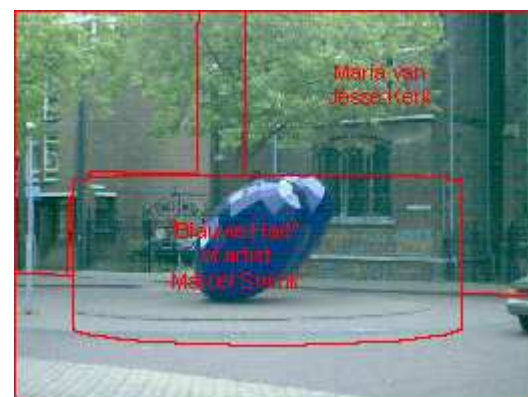
1010



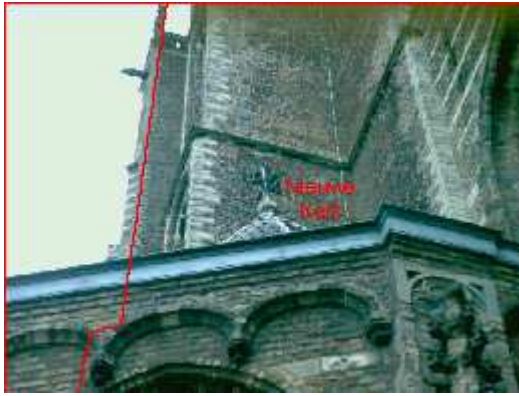
1011



1013



1014



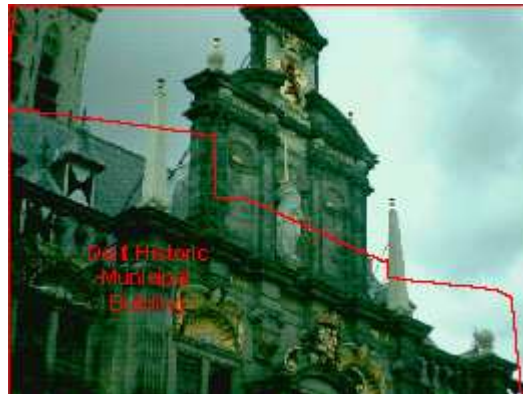
1018



1019



2010



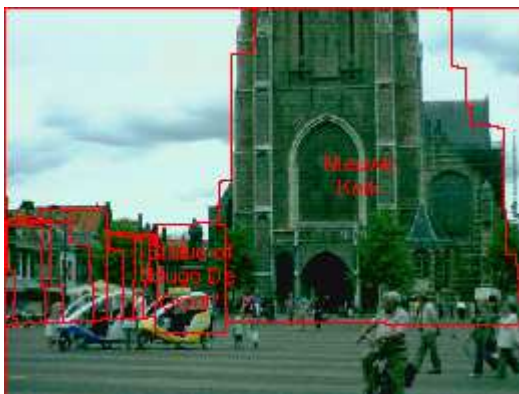
2011



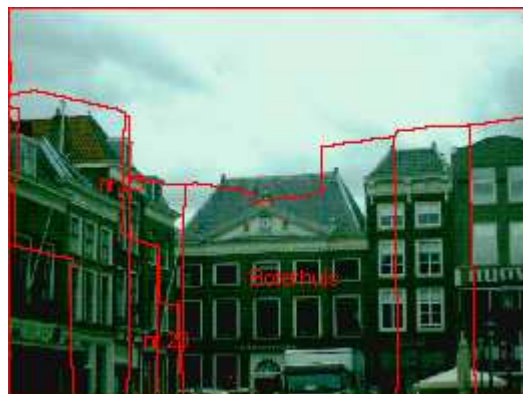
2012



2013



2014



2018

Appendix C

VBA code for automation

C1 VBA code for automation in ESRI ArcScene

```
Private Sub cmdRun_Click()

    Dim pApp As ISxApplication
    Dim pDoc As ISxDocument
    Set pDoc = ThisDocument
    Dim pScene As IScene
    Set pScene = pDoc.Scene
    Dim pSceneGraph As ISceneGraph
    Set pSceneGraph = pScene.SceneGraph

    'View the original photo - Scene Viewer
    Dim pPhotoViewer As ISceneViewer
    Set pPhotoViewer = pSceneGraph.FindViewer("Original photo")

    '=====
    'Open original photo
    Dim pRaster As IRasterDataset
    Set pRaster = OpenRasterDataset(tboFolder, tboFileName)

    Dim pRasLayer As IRasterLayer
    Set pRasLayer = New RasterLayer
    pRasLayer.CreateFromDataset pRaster

    pScene.AddLayer pRasLayer

    'Set to orthogonal view
    Dim pOrigPhotoCam As ICamera
    Set pOrigPhotoCam = pPhotoViewer.Camera
    pOrigPhotoCam.ProjectionType = esriOrthoProjection

    Dim PhotoViewPoint As IPoint
    Set PhotoViewPoint = New Point
    PhotoViewPoint.X = 319.5
    PhotoViewPoint.Y = -239.5

    pOrigPhotoCam.Observer = PhotoViewPoint
    pOrigPhotoCam.ViewingDistance = 1039.6
```

```
pOrigPhotoCam.ViewFieldAngle = 55

pDoc.Scene.SceneGraph.Invalidate pRasLayer, True, True
pDoc.Scene.SceneGraph.RefreshViewers

'=====

MsgBox ("Original photo is loaded!")

'-----
' ObserverPoint and TargetPoint

Dim pObserverPoint As IPoint: Set pObserverPoint = New Point

pObserverPoint.X = CDBl(tboXcoord) 'From form / image metadata
pObserverPoint.Y = CDBl(tboYcoord) 'From form / image metadata
pObserverPoint.Z = CDBl(tboZcoord) 'From form / image metadata

'-----
'Ratio 4:3 - 640x480
If pRasLayer.ColumnCount / pRasLayer.RowCount = 4 / 3 Then

MsgBox ("Ratio of the photo resolution is 4:3")

Dim pTargetPoint As IPoint: Set pTargetPoint = New Point
pTargetPoint.X = pObserverPoint.X + (CDBl(tboViewDist) * Sin(CDBl(tboHeading) / 57.295777951))
pTargetPoint.Y = pObserverPoint.Y + (CDBl(tboViewDist) * Cos(CDBl(tboHeading) / 57.295777951))
pTargetPoint.Z = pObserverPoint.Z + (CDBl(tboViewDist) * Tan(CDBl(tboPitch) / 57.295777951))

'Export 4:3 photo in Scene Viewer
Dim pSceneViewer1 As ISceneViewer
Set pSceneViewer1 = pSceneGraph.FindViewer("Export 4:3 photos (640x480)")
Dim pSV1Cam As ICamera
Set pSV1Cam = pSceneViewer1.Camera

pSV1Cam.ProjectionType = esriPerspectiveProjection

pSV1Cam.ViewFieldAngle = CDBl(tboViewAngle) 'From form / image metadata
pSV1Cam.Azimuth = CDBl(tboHeading)
pSV1Cam.Inclination = -(CDBl(tboPitch))
pSV1Cam.RollAngle = CDBl(tboRoll)
pSV1Cam.ViewingDistance = CDBl(tboViewDist)

pSV1Cam.Target = pTargetPoint
pSV1Cam.Observer = pObserverPoint
pDoc.Scene.SceneGraph.RefreshViewers

'Export Snapshot
Dim iSV1Width As Long: iSV1Width = CDBl(pRasLayer.ColumnCount)
Dim iSV1Height As Long: iSV1Height = CDBl(pRasLayer.RowCount)
Dim iSV1Output As esri3DOutputImageType: iSV1Output = 1
Dim iSV1FileName As String: iSV1FileName = tboFolder + tboFileName + "_export" + ".bmp"
pSceneViewer1.GetSnapshot iSV1Width, iSV1Height, iSV1Output, tboFolder + tboFileName + "_export" + ".bmp"

'-----
'Ratio 3:2 - 640x427

Else
If pRasLayer.ColumnCount / pRasLayer.RowCount = 3 / 2 Then

MsgBox ("Ratio of the photo resolution is 3:2")

'Export Nikon Image - Scene Viewer
Dim pSceneViewer2 As ISceneViewer
Set pSceneViewer2 = pSceneGraph.FindViewer("Viewer 7")
Set pSceneViewer3 = pSceneGraph.FindViewer("Export 3:2 photos (640x427)")
pSceneViewer2.Caption = "Export 3:2 photos (640x427)"
Dim pSV2Cam As ICamera
Set pSV2Cam = pSceneViewer2.Camera
```

```

pSV2Cam.ProjectionType = esriPerspectiveProjection

pSV2Cam.ViewFieldAngle = CDBl(tboViewAngle) 'From form / image metadata
pSV2Cam.Azimuth = CDBl(tboHeading)
pSV2Cam.Inclination = -(CDBl(tboPitch))
pSV2Cam.RollAngle = CDBl(tboRoll)

Dim ViewDist As Double
Dim bellowext As Double

If tboFocalLenght = 12 Then
    bellowext = 0.012001195
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 15 Then
    bellowext = 0.015001867
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 17 Then
    bellowext = 0.017002398
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 20 Then
    bellowext = 0.02000332
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 24 Then
    bellowext = 0.024004781
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 34 Then
    bellowext = 0.034009596
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 50 Then
    bellowext = 0.050020756
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

If tboFocalLenght = 70 Then
    bellowext = 0.070040689
    tboViewDist = (0.001 * tboFocalLength * bellowext) / (0.01 * tboFocalLength - bellowext)
Else
    End If

pTargetPoint.x = pObserverPoint.x + (CDBl(tboViewDist) * Sin(CDBl(tboHeading) / 57.295777951))
pTargetPoint.Y = pObserverPoint.Y + (CDBl(tboViewDist) * Cos(CDBl(tboHeading) / 57.295777951))
pTargetPoint.Z = pObserverPoint.Z + (CDBl(tboViewDist) * Tan(CDBl(tboPitch) / 57.295777951))

pSV2Cam.ViewingDistance = ViewDist

pSV2Cam.Target = pTargetPoint
pSV2Cam.Observer = pObserverPoint
pDoc.Scene.SceneGraph.RefreshViewers

'Export Snapshot
Dim iSV2Width As Long: iSV2Width = CDBl(pRasLayer.ColumnCount)

```

```
Dim iSV2Height As Long: iSV2Height = CDBl(pRasLayer.RowCount)
Dim iSV2Output As esri3DOutputImageType: iSV1Output = 1
Dim iSV2FileName As String: iSV2FileName = tboFolder + tboFileName + "_export" + ".bmp"
pSceneViewer2.GetSnapshot iWidth, iHeight, iOutput, iSV2FileName

'-----
'Ratio unknown
Else

MsgBox ("Ratio unknown - Unable to create and export scene. Press OK to End")
End
End If
End If

Dim pExpScene As IRasterDataset
Set pExpScene = OpenRasterDataset(tboFolder, tboFileName + "_export" + ".bmp")

Dim pExpSceneLayer As IRasterLayer
Set pExpSceneLayer = New RasterLayer
pExpSceneLayer.CreateFromDataset pExpScene

pScene.AddLayer pExpSceneLayer

pDoc.UpdateContents
pSceneGraph.RefreshViewers
MsgBox ("Exported scene is created and saved as " + tboFolder + tboFileName + "_export" + ".bmp" + "!")

Dim pRasLayerEffects As ILayerEffects
Set pRasLayerEffects = pRasLayer
If pRasLayerEffects.SupportsTransparency Then
    pRasLayerEffects.Transparency = 25
End If
pDoc.Scene.SceneGraph.Invalidate pRasLayer, True, True

Dim pExpSceneLayerEffects As ILayerEffects
Set pExpSceneLayerEffects = pExpSceneLayer
If pExpSceneLayerEffects.SupportsTransparency Then
    pExpSceneLayerEffects.Transparency = 50
End If
pDoc.Scene.SceneGraph.Invalidate pRasLayer, True, True
pDoc.Scene.SceneGraph.Invalidate pExpSceneLayer, True, True

pDoc.UpdateContents
pSceneGraph.RefreshViewers

MsgBox ("Please continue in ArcMap with the labeling of the photo")

pScene.DeleteLayer pExpSceneLayer
pScene.DeleteLayer pRasLayer
pDoc.UpdateContents
pSceneGraph.RefreshViewers

End Sub
```

```
Public Function OpenRasterDataset(sPath As String, sFileName As String) As IRasterDataset
    ' sPath: directory where dataset resides
    ' sFileName: name of the raster dataset
    On Error GoTo ErrorHandler

    ' Create RasterWorkspaceFactory
    Dim pWSF As IWorkspaceFactory
    Set pWSF = New RasterWorkspaceFactory

    ' Get RasterWorkspace
    Dim pRasWS As IRasterWorkspace
    If pWSF.IsWorkspace(sPath) Then
        Set pRasWS = pWSF.OpenFromFile(sPath, 0)
```

```

        Set OpenRasterDataset = pRasWS.OpenRasterDataset(sFileName)
    End If

    ' Release memory
    Set pRasWS = Nothing
    Set pWSF = Nothing
    Exit Function
ErrorHandler:
    Set OpenRasterDataset = Nothing
End Function

```

```

Private Sub UserForm_Click()

End Sub

```

```

Private Sub UserForm_Initialize()
    ' Add items to the dropdown list.
    cbxDeviceType.AddItem "<unknown>"
    cbxDeviceType.AddItem "Nikon D100"
    cbxDeviceType.AddItem "Topcon GPT-7003i"

    cbxLensType.AddItem "<unknown>"
    cbxLensType.AddItem "Standard"
    cbxLensType.AddItem "Tele"
    cbxLensType.AddItem "Wide-Angle"

End Sub

```

```

Private Sub cbxDeviceType_Change()

End Sub

```

```

Private Sub cmdCancel_Click()
End
End Sub

```

C2 VBA code for automation in ESRI ArcMap

```

Private Sub cmdRun_Click()

Dim pApp As IMxApplication
    Dim pDoc As IMxDocument
    Set pDoc = ThisDocument
    Dim pMap As IMap
    Set pMap = pDoc.ActiveView

'open raster file and different bands

Dim praster As IRasterDataset
Set praster = OpenRasterDataset(tboFolder, tboFileName + "_export.bmp")

Dim pRasLayer As IRasterLayer
Set pRasLayer = New RasterLayer
pRasLayer.CreateFromDataset praster

Dim pOrigPhoto As IRasterDataset

```

```
Set pOrigPhoto = OpenRasterDataset(tboFolder, tboFileName)

Dim pOrigPhotoLayer As IRasterLayer
Set pOrigPhotoLayer = New RasterLayer
pOrigPhotoLayer.CreateFromDataset pOrigPhoto

pMap.AddLayer pOrigPhotoLayer

pDoc.ActiveView.Refresh

MsgBox ("Original photo loaded")
Dim pBandCol As IRasterBandCollection
Set pBandCol = praster
' Get the bands of the raster
Dim pRasterBandRed As IRasterBand
Set pRasterBandRed = pBandCol.Item(0)
Dim pRasterBandGreen As IRasterBand
Set pRasterBandGreen = pBandCol.Item(1)
Dim pRasterBandBlue As IRasterBand
Set pRasterBandBlue = pBandCol.Item(2)

'Create two constant rasters
Dim pRasterMakerOp As IRasterMakerOp
Set pRasterMakerOp = New RasterMakerOp

Dim pEnv As IRasterAnalysisEnvironment
Set pEnv = pRasterMakerOp
pEnv.SetCellSize esriRasterEnvValue, 1

Dim pExt As IEnvelope
Set pExt = New Envelope
pExt.PutCoords 0, -480, 640, 0
pEnv.SetExtent esriRasterEnvValue, pExt

' Declare the output raster object
Dim pConstRaster65536 As IRaster
Set pConstRaster65536 = pRasterMakerOp.MakeConstant(65536, True)
Dim pConstRaster256 As IRaster
Set pConstRaster256 = pRasterMakerOp.MakeConstant(256, True)

'-----

' Weighted Sum of the rasters
Dim pMathOp As IMathOp
Set pMathOp = New RasterMathOps

Dim pTimesBand1 As Raster
Set pTimesBand1 = pMathOp.Times(pConstRaster65536, pRasterBandRed)

Dim pTimesBand2 As Raster
Set pTimesBand2 = pMathOp.Times(pRasterBandGreen, pConstRaster256)

Dim pPlusBand1and2 As Raster
Set pPlusBand1and2 = pMathOp.Plus(pTimesBand2, pTimesBand1)

Dim pSumRaster As Raster
Set pSumRaster = pMathOp.Plus(pPlusBand1and2, pRasterBandBlue)

'---- conversion
Dim pConversionOp As IConversionOp: Set pConversionOp = New RasterConversionOp

'Dim pCVRas As IRaster: Set pRas01 = getRasterFromFileFunction("c:\data\myRaster")
Dim pFClassOut As IObjectClass
Dim pWSF As IWorkspaceFactory: Set pWSF = New ShapefileWorkspaceFactory
Dim pWS As IWorkspace: Set pWS = pWSF.OpenFromFile(tboFolder, 0)
Dim sOutFCname As String: sOutFCname = tboFileName + "_label.shp"

Set pFClassOut = pConversionOp.RasterDataToPolygonFeatureData(pSumRaster, pWS, sOutFCname, False)
```

```

Dim pLayer As IFeatureLayer
Set pLayer = New FeatureLayer
Set pLayer.FeatureClass = pFClassOut

pMap.AddLayer pLayer

MsgBox ("Layer added!")

Dim pLayerEffects As ILayerEffects
Set pLayerEffects = pLayer
If pLayerEffects.SupportsTransparency Then
    pLayerEffects.Transparency = 100

End If
pDoc.ActiveView.Refresh

MsgBox ("Layer is now transparant!")

'JOIN DATA
Dim strJnField As String
Dim TableIndex As Long
Dim strJnField2 As String: strJnField2 = "GRIDCODE"
If chkBuild = True Then

    strJnField = "building_f"
    TableIndex = 0
    JoinToLayer
Else
    End If

If chkStreet = True Then
    strJnField = "building_f"
    TableIndex = 1
    JoinToLayer
Else
    End If

If chkWater = True Then
    strJnField = "building_f"
    TableIndex = 2
    JoinToLayer
Else
    End If

'Label photo
Label
ChangeMxDocumentLabelEngine

End Sub

```

```

Public Function OpenRasterDataset(sPath As String, sFileName As String) As IRasterDataset
    ' sPath: directory where dataset resides
    ' sFileName: name of the raster dataset
    On Error GoTo ErrorHandler

    ' Create RasterWorkspaceFactory
    Dim pWSF As IWorkspaceFactory
    Set pWSF = New RasterWorkspaceFactory

    ' Get RasterWorkspace
    Dim pRasWS As IRasterWorkspace
    If pWSF.IsWorkspace(sPath) Then
        Set pRasWS = pWSF.OpenFromFile(sPath, 0)
        Set OpenRasterDataset = pRasWS.OpenRasterDataset(sFileName)
    End If

    ' Release memeory

```

```
Set pRasWS = Nothing
Set pWSF = Nothing
Exit Function
ErrorHandler:
Set OpenRasterDataset = Nothing
End Function
```

```
Public Sub JoinToLayer()
```

```
On Error GoTo EH
```

```
Dim pDoc As IMxDocument
Dim pMap As IMap
Set pDoc = ThisDocument
Set pMap = pDoc.FocusMap
```

```
' Get the first layer in the table on contents
Dim pFeatLayer As IFeatureLayer
Dim pDispTable As IDisplayTable
Dim pFCLayer As IFeatureClass
Dim pTLayer As ITable
If pMap.LayerCount = 0 Then
    MsgBox "Must have at least one layer"
    Exit Sub
End If
Set pFeatLayer = pMap.Layer(0)
Set pDispTable = pFeatLayer
Set pFCLayer = pDispTable.DisplayTable
Set pTLayer = pFCLayer
```

```
' Get the first table in the table on contents
Dim pTabCollection As IStandaloneTableCollection
Dim pStTable As IStandaloneTable
Dim pDispTable2 As IDisplayTable
Dim pTTable As ITable
Set pTabCollection = pMap
If pTabCollection.StandaloneTableCount = 0 Then
    MsgBox "Must have atleast one table"
    Exit Sub
End If
```

```
Set pStTable = pTabCollection.StandaloneTable(0)
Set pDispTable2 = pStTable
Set pTTable = pDispTable2.DisplayTable
```

```
Dim strJnField As String: strJnField = "building_f"
Dim strJnField2 As String: strJnField2 = "GRIDCODE"
```

```
' Create virtual relate
Dim pMemRelFact As IMemoryRelationshipClassFactory
Dim pRelClass As IRelationshipClass
Set pMemRelFact = New MemoryRelationshipClassFactory
Set pRelClass = pMemRelFact.Open("TabletoLayer", pTTable, strJnField, pTLayer, _
strJnField2, "forward", "backward", esriRelCardinalityOneToMany)
```

```
' use Relate to perform a join
Dim pDispRC As IDisplayRelationshipClass
Set pDispRC = pFeatLayer
pDispRC.DisplayRelationshipClass pRelClass, esriLeftOuterJoin
```

```
Exit Sub
EH:
```

```
MsgBox Err.Number & " " & Err.Description
```

```
End Sub
```

```
Sub Label()
```



```

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
    Dim pMap As IMap
    Set pMap = pMxDoc.ActiveView

    Dim pAnnotateMap As IAnnotateMap
    Set pAnnotateMap = New AnnotateMap
    Set pMap.AnnotationEngine = pAnnotateMap

Dim pFLayer As IGeoFeatureLayer
Set pFLayer = pMap.Layer(0)

'+++ setup LabelEngineProperties for the FeatureLayer
' get the AnnotateLayerPropertiesCollection for the FeatureLayer
Dim pAnnoLayerPropsColl As IAnnotateLayerPropertiesCollection
Set pAnnoLayerPropsColl = pFLayer.AnnotationProperties
pAnnoLayerPropsColl.Clear

' create a new LabelEngineLayerProperties object
Dim aLELayerProps As ILabelEngineLayerProperties
Set aLELayerProps = New LabelEngineLayerProperties
aLELayerProps.Expression = "[building_footprints_attrib.building_2]"

' assign it to the layer's AnnotateLayerPropertiesCollection
pAnnoLayerPropsColl.Add aLELayerProps

' show labels
pFLayer.DisplayAnnotation = True

'+++ refresh the map
pMxDoc.ActiveView.Refresh

End Sub

```

```

Public Sub ChangeMxDocumentLabelEngine()

'This simple subroutine will change the label engine for all data frames in a map to be the maplex label engine
'You should have a maplex license enabled before running this code

Dim pMxDoc As IMxDocument
Dim pMaps As IMaps
Dim pMap As IMap
Dim index As Long
Dim pAnnotateMap As IAnnotateMap

Set pMxDoc = ThisDocument
Set pMaps = pMxDoc.Maps

'loop through all the maps
For index = 0 To pMaps.Count - 1
    Set pAnnotateMap = New esriMaplex.MaplexAnnotateMap 'cocreate a new MaplexAnnotateMap object
    Set pMap = pMaps.Item(index) 'get the map at the current index
    Set pMap.AnnotationEngine = pAnnotateMap 'set the map AnnotationEngine to be MaplexAnnotateMap
    'after setting the AnnotationEngine, the Map automatically translates all labeling properties to Maplex.
Next index

End Sub

```

```

Private Sub cmdCancel_Click()
End
End Sub

```

```

Private Sub UserForm_Click()

```

End Sub

C3 VBA code for colouring of objects with OIDs color values

```
Private Sub UIButtonControl1_Click()  
    *** Paste into VBA  
    *** Creates a UniqueValuesRenderer and applies it to first layer in the map.  
    *** Layer must have "FID_nlzahnw" field  
    Dim pApp As Application  
    Dim pDoc As ISxDocument  
    Set pDoc = ThisDocument  
    Dim pScene As IScene  
    Set pScene = pDoc.Scene  
    Dim pLayer As ILayer  
    Set pLayer = pScene.Layer(0)  
    Dim pFLayer As IFeatureLayer  
    Set pFLayer = pLayer  
    Dim pLyr As IGeoFeatureLayer  
    Set pLyr = pFLayer  
  
    Dim pFeatCls As IFeatureClass  
    Set pFeatCls = pFLayer.FeatureClass  
    Dim pQueryFilter As IQueryFilter  
    Set pQueryFilter = New QueryFilter 'empty supports: SELECT *  
    Dim pFeatCursor As IFeatureCursor  
    Set pFeatCursor = pFeatCls.Search(pQueryFilter, False)  
    *** Make the renderer  
    Dim pRender As IUniqueValueRenderer, n As Long  
    Set pRender = New UniqueValueRenderer  
  
    Dim symd As ISimpleFillSymbol  
    Set symd = New SimpleFillSymbol  
    symd.Style = esriSFSSolid  
    symd.Outline.Width = 0.4  
  
    Dim symdOutlineColor As IRgbColor  
    Set symdOutlineColor = New RgbColor  
    symdOutlineColor.Red = 255  
    symdOutlineColor.Blue = 255  
    symdOutlineColor.Green = 255  
    symd.Outline.Color = symdOutlineColor  
    *** These properties should be set prior to adding values  
    pRender.FieldCount = 1  
    pRender.Field(0) = "FID_nlzahnw"  
    pRender.DefaultSymbol = symd  
    pRender.UseDefaultSymbol = True  
  
    Dim pFeat As IFeature  
    n = pFeatCls.FeatureCount(pQueryFilter)  
    *** Loop through the features  
    Dim i As Integer  
    i = 0  
    Dim ValFound As Boolean  
    Dim NoValFound As Boolean  
    Dim uh As Integer  
    Dim pFields As IFields  
    Dim iField As Integer  
    Set pFields = pFeatCursor.Fields  
    iField = pFields.FindField("FID_nlzahnw")  
    Do Until i = n  
        Dim symx As ISimpleFillSymbol  
        Set symx = New SimpleFillSymbol  
        symx.Style = esriSFSSolid  
        symx.Outline.Width = 0  
        symx.Outline.Color = symdOutlineColor  
        Set pFeat = pFeatCursor.NextFeature  
        Dim x As String  
        x = pFeat.Value(iField) '*new Cory*  
        *** Test to see if we've already added this value  
        *** to the renderer, if not, then add it.  
        ValFound = False  
        For uh = 0 To (pRender.ValueCount - 1)
```

```

        If pRender.Value(uh) = x Then
            NoValFound = True
            Exit For
        End If
    Next uh
    If Not ValFound Then
        pRender.AddValue x, "FID_nlzhnw", symx
        pRender.Label(x) = x
        pRender.Symbol(x) = symx
    End If
    i = i + 1
Loop
' ** now that we know how many unique values there are
' ** we can size the color ramp and assign the colors.

For ny = 0 To (pRender.ValueCount - 1)
    Dim xv As String
    xv = pRender.Value(ny)
    If xv <> "" Then
        Dim jsy As ISimpleFillSymbol
        Set jsy = pRender.Symbol(xv)
        Dim pNextColorLong As Long
        Dim pNextColorRGB As IRgbColor
        Set pNextColorRGB = New RgbColor
        pNextColorLong = xv
        pNextColorRGB.Blue = pNextColorLong Mod &H100

        pNextColorLong = pNextColorLong \ &H100
        pNextColorRGB.Green = pNextColorLong Mod &H100
        pNextColorLong = pNextColorLong \ &H100
        pNextColorRGB.Red = pNextColorLong Mod &H100

        jsy.Color = pNextColorRGB
        Dim pNextOutLine As ICartographicLineSymbol
        Set pNextOutLine = New CartographicLineSymbol
        With pNextOutLine
            .Width = 0.00000000001
            .Color = pNextColorRGB
        End With

        jsy.Outline = pNextOutLine

        pRender.Symbol(xv) = jsy
    End If
Next ny

' ** If you didn't use a color ramp that was predefined
' ** in a style, you need to use "Custom" here, otherwise
' ** use the name of the color ramp you chose.
pRender.ColorScheme = "Custom"
pRender.FieldType(0) = True
Set pLyr.Renderer = pRender
pLyr.DisplayField = "FID_nlzhnw"

' ** This makes the layer properties symbology tab show
' ** show the correct interface.
Dim hx As IRendererPropertyPage
Set hx = New UniqueValuePropertyPage
pLyr.RendererPropertyPageClassID = hx.ClassID

' ** Refresh the TOC
'pDoc.ActiveView.ContentChanged
pDoc.UpdateContents
Dim p3Dprop As I3DProperties
Dim pLyrExt As ILayerExtensions: Set pLyrExt = pLayer
Set p3Dprop = pLyrExt.Extension(0)
Dim pFeatLyr As IFeatureLayer: Set pFeatLyr = pLayer
Set pLyrExt = pFeatLyr

With p3Dprop
    .Illuminate = False

    .Apply3DProperties pFeatLyr
End With

```

```
pDoc.Scene.SceneGraph.RefreshViewers
```

```
End Sub
```