Pattern Recognition and Image Processing Group
Institut für rechnergestützte Automation
Technische Universität Wien
Favoritenstr. 9/183-2
A-1040 Wien Österreich
Tel.: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: e0627880@student.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

Bachelorarbeit

Wien, 19. Oktober 2009

Visualisierung von Trajektorien live in Microsoft Bing Maps

Timo Kropp

Unter der Aufsicht von a.o.Univ.-Prof. Robert Sablatnig

Abstract

Hochauflösende Luftaufnahmen der gesamten Erdoberfläche werden mittlerweile von immer mehr interaktiven Kartendiensten im Web angeboten. Diese Webdienste bieten offene Programmierschnittstellen an, die flexible Erweiterungen ermöglichen. Insbesondere für das Verfolgen (engl. Tracking) von bewegten Objekten unter freiem Himmel kann das frei verfügbare Bildmaterial Szeneneinblicke aus der Vogelperspektive bieten, die ansonsten nicht möglich wären. Für diese Arbeit wurde eine Software implementiert, die ein bestehendes tracking System in Microsoft Bing Maps integriert. Mittels einer manuell bestimmten Abbildungsfunktion werden Trajektorien in Echtzeit vom Kamerabild auf Microsofts virtueller Erde abgebildet. Es wird gezeigt, dass trotz starken Unterschieden in der Perspektive und der Auflösung von Kamera und Luftbild in den meisten Situationen die korrekte Abbildungsfunktion bis auf wenige Pixel genau bestimmt werden kann. Dies eröffnet neue Möglichkeiten zur innovativen Visualisierung von Trajektorien in automatischen Videoüberwachungssystemen.

1 Einleitung

In der Videoüberwachung spielt die Echtzeit-Visualisierung von erkannten Ereignissen und bewegten Objekten, wie z.B. Fahrzeuge und Personen eine immer wichtigere Rolle [3]. Insbesondere das Tracking von 50 Objekten und mehr mit einer statischen Überwachungskamera erfordert intuitiv verständlichere Visualisierungen, damit die Bewegungen in dieser Szene für einen Menschen weiterhin nachvollziehbar bleiben. Um solche Anforderungen für bestimmte Szenarien erfüllen zu können, bietet der Webdienst Bing Maps von Microsoft hinreichend genaue Satellitenfotos der Erdoberfläche, welche für die Echtzeit-Visualisierung von Tracking Daten einer Überwachungskamera benutzbar sind.

Mit der Visualisierung von bewegten Objekten auf der Erdoberfläche, haben sich ebenfalls Brown und Dunn [2] beschäftigt. Sie haben die Tracking Daten von Flugzeugen und anderen bewegten Objekten mittels des Global Positioning Systems (GPS) bestimmt und diese auf Landkarten und Bildern der Erdoberfläche dargestellt. Aufgrund der begrenzten Genauigkeit der von GPS ermittelten Positionen wird in dieser Arbeit das Tracking mittels einer Überwachungskamera durchgeführt.

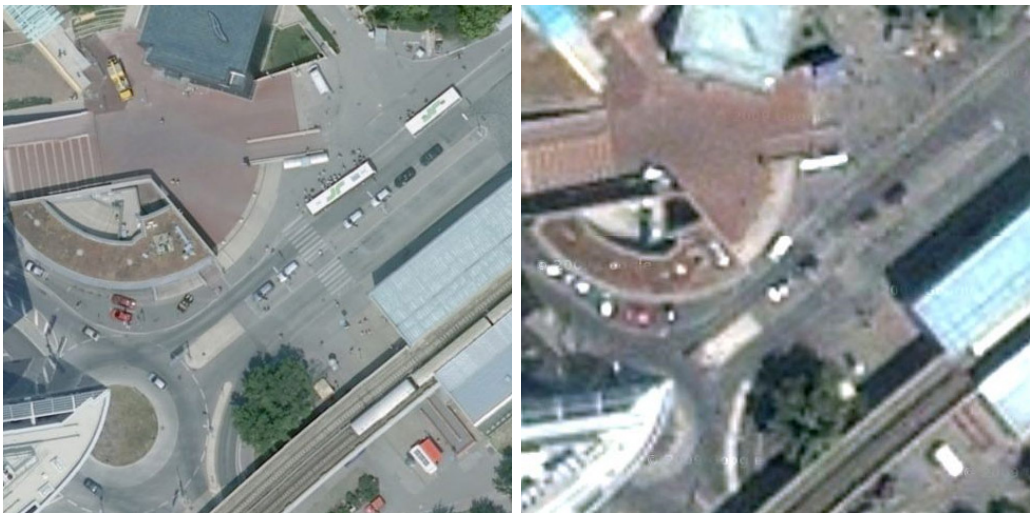
Es wurde eine Anwendung in C++ entwickelt, die ein vorhandenes Tracking System (siehe Kapitel 4.1) in Microsoft Bing Maps integriert. Im Kern der Applikation wird zuerst über mehrere Punktkorrespondenzen im Kamera- und Satellitenbild eine Abbildung zwischen zwei Ebenen ermittelt, die als Homographie bezeichnet wird. Anhand dieser werden anschließend die Tracking Daten in Echtzeit auf die Erdoberflächendarstellung von Bing Maps transformiert. In dieser Arbeit wird mit Hilfe einer Monte Carlo Simulation untersucht, wie genau die Homographie in Hinsicht auf die Auflösung der Satellitenfotos bestimmt werden kann. Des Weiteren wird geprüft, unter welchen Bedingungen die Transformation akzeptable Ergebnisse liefert.

Alle Untersuchungen beziehen sich ausschließlich auf Outdoor-Szenen, da Überwachungskameras innerhalb von Gebäuden keine notwendigen Bezugspunkte zu den Satellitenfotos von Bing Maps liefern können. Außerdem wird vorausgesetzt, dass sich die getrackten Objekte auf einer Ebene der Erdoberfläche bewegen. Anderenfalls ist die Homographie nicht zu bestimmen.

Im Folgenden wird zunächst auf die Funktionsweise des Tracking Systems eingegangen. Anschließend wird erläutert (Kapitel 5), wie die Homographie bestimmt werden kann. Mit diesen Voraussetzungen wird in Kapitel 4 die Implementierung und Funktionsweise der dieser Arbeit zugrunden liegenden

Applikation beschrieben. Den Hauptteil dieser Arbeit bildet Kapitel 5, in dem im ersten Abschnitt die manuelle Homographiebestimmung qualitativ untersucht wird und im zweiten Abschnitt die Abbildungsgenauigkeit mittels der Monte Carlo Simulation evaluiert wird. Abschließend werden die Ergebnisse zusammengefasst und ein Ausblick auf mögliche weiterführende Themen geboten.

1.1 Motivation



(a) Mircosoft Bing Maps (b) Google Earth

Abbildung 1: Screenshot der gleichen Szene in maximaler Zoomstufe (Stand Oktober 2009)

Öffentliche Kartendienste im Internet (z.B. Google Earth [4], Microsoft Bing Maps [8]) bieten neue Möglichkeiten der Visualisierung von Tracking Daten aus Videoüberwachungssystemen. Ihre hochauflösenden Satellitenfotos decken einen Großteil der gesamten Erdoberfläche ab, und innerhalb von einigen Großstädten ist die Auflösung der Aufnahmen bis auf einen Meter genau [6]. Mit diesen Daten als Grundlage können über offene Programmierschnittstellen (siehe Kapitel 1.2), wie sie z.B. von Google Earth und Microsoft Bing Maps angeboten werden, beliebige Erweiterungen entwickelt werden. Für diese Arbeit wurde Bing Maps als Grundlage für die Visualisierung gewählt, da

die Luftaufnahmen beim Standort der verwendeten Überwachungskamera eine bessere Qualität aufweisen als die in Google Earth (siehe Abbildung 1).

Um die Dienste für Visualisierungen nutzen zu können, bietet sich die Integration dieser in ein bestehendes Videoüberwachungssystem an.

1.2 Bing Maps

Microsoft Bing Maps beinhaltet eine 2D- und 3D-Darstellung der Erdoberfläche, auf der interaktiv navigiert werden kann. Im 2D-Modus können sowohl die Satellitenfotos mit oder ohne überlagerten Straßennamen als auch das Kartenmaterial wie in einem Stadtplan angezeigt werden. Es gibt 19 Zoomstufen, wobei jeweils für die Stufen 1 bis 9, 10 bis 14 und 15 bis 19 unterschiedliche Fotos verwendet werden. Für die Visualisierung von Trajektorien sind die Stufen 18 und 19 besonders geeignet, da hier die Auflösung so hoch ist, dass z.B. Autos erkannt werden können (siehe Abbildung 8 in Kapitel 5).

Bing Maps bietet für Entwickler ein „Asynchronous JavaScript and XML Application Programming Interface“ (AJAX API) und ein „Simple Object Access Protocol Application Programming Interface“ (SOAP API) an [9]. Letzteres wurde hauptsächlich für Enterprise Anwendungen entwickelt. Bing Maps wurde mittels der AJAX API in das zu dieser Arbeit gehörende Programm integriert. Diese bietet sich aufgrund der unmittelbaren Schnittstelle zum Frontend von Bing Maps für die Visualisierung von Trajektorien besonders an.

2 Tracking

Das Tracking System, welches für diese Arbeit verwendet wurde, basiert auf einem Point-Tracking Algorithmus [1]. Die besonderen Anforderungen an das Verfahren und deren Implementierung sind durch die Performance und Ressourcen eines Digitalen Signal Prozessors (DSP) bestimmt. Dieser ist in ein eigenständiges Hardware-Modul (AVC) integriert, welches die Tracking Daten nach erfolgter Berechnung in ein Netzwerk an einen Desktop Computer weiterleitet (siehe Abbildung 2).

Abbildung 2: Hardware des Trackingsystems: Desktop Computer, AVC und Kamera

Der Algorithmus besteht aus folgenden Schritten:

1. Vorverarbeitung
2. Segmentierung
3. Tracking

Die Vorverarbeitung beinhaltet die Bestimmung eines Hintergrundbildes. Die Eingabedaten hierfür sind das unbearbeitete Kamerabild im YUV-Farbraum. Das Resultat ist ein Grauwertbild, dessen normierte Werte die

Wahrscheinlichkeit angeben, ob es sich jeweils um ein Vordergrund- oder Hintergrundpixel handelt. Für die Hintergrundmodellierung wird der gleitende Mittelwert verwendet, welcher eine abgewandelte Form des Color Mean and Variance Algorithmus ist [11]. Anstatt die Varianz für jeden Pixel über eine bestimmte Zeit zu berechnen, wird ein fester Schwellwert verwendet.

Angewendet wird dieser im zweiten Verarbeitungsschritt der Vorder- und Hintergrund Segmentierung. Die Annahme konstanter Varianz für jeden Pixel liefert zwar etwas schlechtere Ergebnisse in der Hintergrundmodellierung [10], demgegenüber steht jedoch eine deutliche Verringerung der Berechnungszeit auf dem AVC. Diese Optimierung ist aufgrund des Designs des AVC von größerer Bedeutung. Eine weitere Einschränkung ergibt sich für die Integrationszeit des Hintergrundbildes, die festlegt wie schnell ein Objekt vom Zustand „bewegt“ in den Zustand „unbewegt“ in den Hintergrund integriert wird. Wegen der Begrenzung des AVC nur 16 Bit Arithmetik verwenden zu können, ist die Integrationszeit auf den Bereich von 20 bis 60 Sekunden beschränkt. Das Hintergrundbild wird durch folgende Update-Regel adaptiert:

$$\mu(x, y, t) = (1 - \alpha)\mu(x, y, t - 1) + \alpha p(x, y, t), \tag{1}$$

wobei $\mu(x, y, t)$ den berechneten Mittelwert des Pixels an der Stelle x,y zum Zeitpunkt t darstellt und $p(x, y, t)$ der Intensität des Pixels an der Stelle x,y entspricht. $\alpha \in [0, 1]$ wird als Lernrate bezeichnet. Je kleiner der Wert von α ist, desto größer wird die Integrationszeit der Hintergrundadaption.

Der zweite Verarbeitungsschritt, der die Vorder- und Hintergrund Segmentierung beinhaltet, führt ein sogenanntes Labeling auf dem Binärbild durch, welches zuvor aus der Differenz vom Hintergrundbild und dem aktuellen Frame mittels eines Schwellwertes gebildet wurde (siehe Abbildung 3). Der Schwellwert entspricht der Sensitivität bezüglich der Erkennung des Vordergrundes und ist durch den AVC festgelegt. Beim Labeling werden zusammenhängende Vordergrundregionen (Blobs) eine eindeutige Zahl (Label) zugeordnet. Anschließend werden für jeden Blob im Bild die Bounding Box, der Schwerpunkt und die Anzahl der enthaltenen Pixel ermittelt.

Der letzte Verarbeitungsschritt des Algorithmus ist für die Erkennung von Positionsänderungen der erkannten Objekte (Blobs) über die Zeit zuständig. Hierfür werden jeweils zwei aufeinander folgende Frames i und $i + 1$ miteinander verglichen. Unter Verwendung des Kalman-Filters [7] wird für jeden Blob in Frame i eine Vorhersage für seine Position in Frame $i + 1$ getroffen. Diese Berechnung erfolgt auf Grundlage der zuvor ermittelten Positionsände-

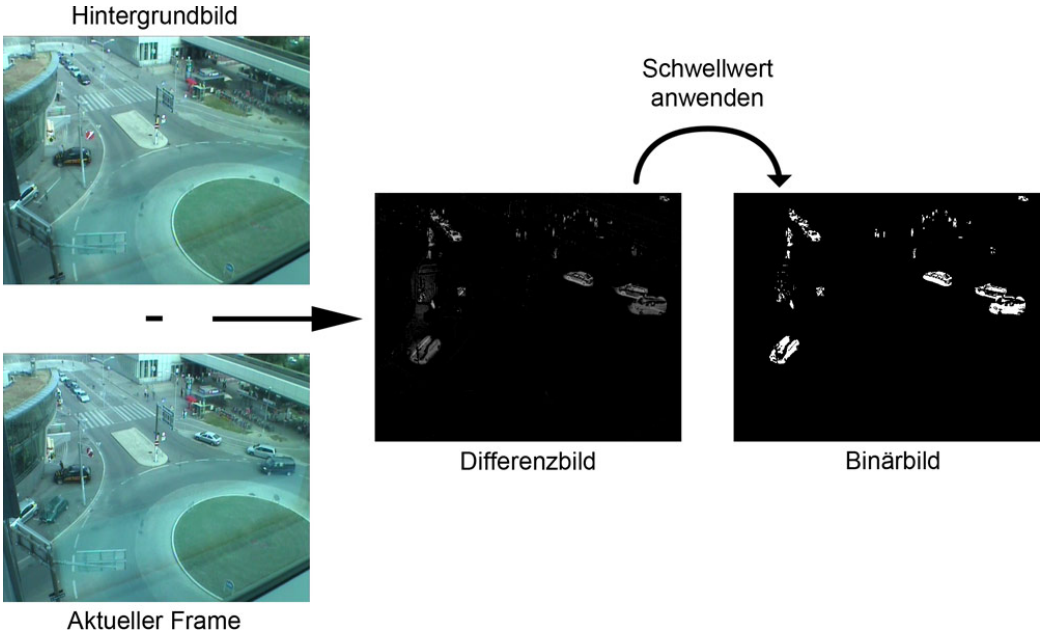


Abbildung 3: Berechnung des Binärbildest mittels der Differenz aus Hintergrundbild und aktuellem Frame mit anschließender Anwendung des Schwellwertes.

rungen in den Frames $< i$. Die vorhergesagte Position in Frame $i+1$ wird als Startpunkt für die nachfolgende Nearest Neighbor-Suche verwendet. Dabei wird der Blob, dessen Entfernung der vorhergesagten Positionen am nächsten ist und dessen Größe (Anzahl der Pixel) ungefähr mit der des gesuchten Blobs übereinstimmt, als neue Objektposition angenommen. Für den Fall, dass kein Blob zugeordnet werden kann, wird angenommen, dass das Objekt entweder verdeckt ist oder das Sichtfeld der Kamera verlassen hat.

Die Tracking-Daten, die an das Netzwerk übermittelt werden, beinhalten für jeden Frame alle erkannten Objekte, ihre zugehörige Position und ihre Bounding-Box.

3 Manuelle Bestimmung der 2D Homographie

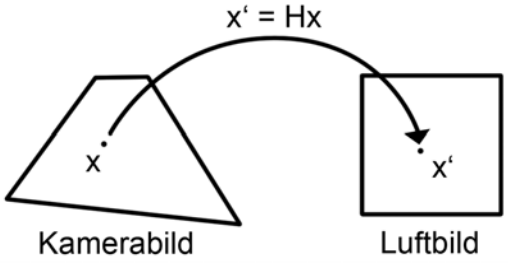


Abbildung 4: Skizze der projektiven Abbildung zwischen den beiden Ebenen des Kamerabildes und des Luftbildes

Um die detektierten Objekte im Kamerabild an der korrekten Position auf dem Satellitenbild darzustellen, ist eine Abbildung notwendig, die jedem Pixel des Kamerabildes einen entsprechenden Punkt im Satellitenbild zuordnet. Im Programmfluss der Visualisierungs-Applikation wird die Bestimmung der Homographie als Kalibrierung bezeichnet. Dieser Schritt muss vom Benutzer manuell durchgeführt werden. Es wird die Annahme getroffen, dass alle getrackten Objekte sich in einer Ebene, auf der Erdoberfläche bewegen. Daraus folgt, dass die Abbildung einer projektiven Transformation (Homographie) $\mathbf{x}' \cong H\mathbf{x}$ entspricht (Abbildung 4). Wobei $\mathbf{x}$ und $\mathbf{x}'$ homogene 3-Vektoren des projektiven Raumes $\mathbb{P}^2$ sind und H eine 3×3 Matrix (2) ist. Alle Vektoren $\mathbf{x} = (x, y, w) \in \mathbb{P}^2$ mit $w \neq 0$ können mit $(x/w, y/w)$ auf ein Element des Vektorraums $\mathbb{R}^2$ zurückgeführt werden. Daraus folgt für $\mathbf{x} \in \mathbb{P}^2$, $t\mathbf{x} \equiv \mathbf{x}$ mit $t \in \mathbb{R} \setminus \{0\}$, d.h. der Vektor $\mathbf{x}$ ist in seiner homogenen Darstellung nur bis auf ein Vielfaches eindeutig bestimmt. Mit der homogenen Darstellung der Vektoren wird die Bestimmung der 2D Homographie zu einem linearen Problem (siehe Kapitel 3.1).

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{2}$$

Es stellt sich nun die Frage, wie viele Punktkorrespondenzen $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ benötigt werden, um die Matrix H zu berechnen. Da aufgrund der homogenen Darstellung h_9 mit einem beliebigen $t \in \mathbb{R} \setminus \{0\}$ multipliziert werden kann, ist H nur auf ein Vielfaches eindeutig bestimmt. Daraus ergibt sich, dass

die Matrix genau 8 Freiheitsgrade besitzt. In der Ebene hat jeder Punkt 2 unabhängige Koordinaten und deswegen müssen vier Punktepaare für die eindeutige Berechnung von H gefunden werden. Diese müssen so gewählt werden, dass das im folgendem aufgestellte Gleichungssystem keine linear abhängigen Gleichungen enthält (siehe Kapitel 3.1).

Mit genau vier Punktepaaren in allgemeiner Lage kann H eindeutig bestimmt werden. Bei der manuellen Wahl der Punktkorrespondenzen können jedoch kleine Abweichungen der korrekten Position, durch Störungen oder Diskretisierungsfehler, auftreten. Aus diesem Grund kann es in solchen Situationen nützlicher sein, mehr als vier Punktepaare zu wählen, damit der Fehler algebraisch minimiert werden kann [5].

3.1 Direct Linear Transformation (DLT) Algorithmus

Gegeben sind $n \geq 4$ homogene 2D Punktkorrespondenzen $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. Gesucht ist die Matrix H, so dass $\mathbf{x}'_i \cong H\mathbf{x}_i$ für alle i gilt. Da die Vektoren $\mathbf{x}'_i$ und $H\mathbf{x}_i$ aufgrund der homogenen Darstellung nur bis auf einen Skalierungsfaktor gleich sind, d.h. in dieselbe Richtung zeigen, jedoch in der Länge variieren, gilt $\mathbf{x}'_i \times H\mathbf{x}_i = 0$. Daraus kann folgendes lineares Gleichungssystem hergeleitet werden [5] mit $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^T$:

$$A_i\mathbf{h} = \mathbf{0} \tag{3}$$

$$\Rightarrow \begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & \mathbf{0}^T & -x'_i\mathbf{x}_i^T \\ -y'_i\mathbf{x}_i^T & x'_i\mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix} \mathbf{h} = \mathbf{0} \tag{4}$$

A_i ist eine 3×9 Matrix und es gilt $\mathbf{h} = (h_1, ..., h_9)^T$.

Der DLT-Algorithmus zur Bestimmung von H besteht nun aus folgenden Schritten [5]:

1. Für jedes Punktepaar $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ ist die Matrix A_i, siehe (3) zu bestimmen. Dabei kann die dritte Zeile von A_i ausgelassen werden, da diese aufgrund der homogenen Koordinaten bis auf einen Skalierungsfaktor von den ersten beiden Zeilen linear abhängig ist.

2. Die n 2×9 Matrizen A_i sind zu einer einzigen $2n$×9 Matrix A zusammen zu setzen.

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix} \tag{5}$$

3. Von der Matrix A sind die Singulärwerte und -vektoren mittels SVD (Singulärwertzerlegung) zu bestimmen. Der singuläre Einheitsvektor, der zum kleinsten Singulärwert gehört, ist die Lösung $\mathbf{h}$. Es gilt $A = UDV^T$ mit der positiven Diagonalmatrix D. Sind dessen Werte in absteigender Größe angeordnet, dann ist $\mathbf{h}$ die letzte Spalte von V.

4. Die Matrix H ist aus $\mathbf{h} = (h_1, ..., h_9)^T$ wie in (2) zu bestimmen.

Der DLT-Algorithmus minimiert mit mehr als vier Punktkorrespondenzen die Norm $\|A\mathbf{h}\|$, da das lineare Gleichungssystem (4) in diesem Fall überbestimmt ist. $A\mathbf{h}$ wird auch als „residual vector“ bezeichnet.

3.2 Normalisierter DLT Algorithmus

Der DLT-Algorithmus hat den Nachteil, dass die resultierende 2D Homographie von den jeweils gewählten Koordinatensystemen in dem die Punkte gemessen werden, abhängig ist. Das bedeutet, dass der algebraische Fehler bei mehr als vier Punktkorrespondenzen nicht invariant bzgl. einer Transformation des Bildes ist [5].

Aus diesem Grund wird die Genauigkeit der Ergebnisse, mittels einer vorherigen Normalisierung der Datenpunkte, erheblich verbessert [5]. Der normalisierte DLT-Algorithmus besteht aus folgenden zusätzlichen Schritten:

1. Normalisierung der Punkte $\mathbf{x_1}, ..., \mathbf{x_k}$ mittels einer Transformation T, die aus einer Translation und einer Skalierung besteht. Der Mittelpunkt alle Punkte ist anschließend $(0, 0)^T$ und der durchschnittliche Abstand zum Mittelpunkt beträgt dann $\sqrt{2}$.

2. Normalisierung der Punkte $\mathbf{x_1}', ..., \mathbf{x_k}$ entsprechend des 1. Schrittes mittels einer Transformation T'.

3. Durchführung des DLT-Algorithmus um die Homographie $\tilde{H}$ zu bestimmen (siehe Kapitel 3.1).

4. Denormalisierung mit $H = T'^{-1}\tilde{H}T$.

4 Implementierung

Für diese Arbeit wurde eine Applikation TraVis implementiert, die eine Schnittstelle zwischen Bing Maps und dem Tracking System bildet.

4.1 Systemumgebung des AIT und Integration

Der hardwareseitige Systemablauf besteht aus den Komponenten: Kamera, AVC, Netzwerk und Client-PC (siehe Abbildung 5). Es werden die aufgezeichneten Frames des CCD oder CMOS Sensors einer surveillance camera in Echtzeit auf dem AVC verarbeitet. Nach erfolgtem Tracking pro Frame werden die Ergebnisse inklusive der Bilddaten in ein 100 MBit Ethernet Netzwerk mittels des Real-Time Transport Protocol (RTP) übertragen. Ein beliebiger Desktop Rechner, der in dem Netzwerk integriert ist empfängt die Tracking-Daten und visualisiert diese mit der Applikation TraVis.

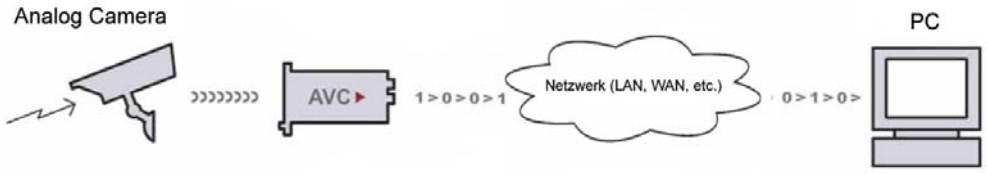


Abbildung 5: Datenfluss des Video Capturing

Durch die kompakte Schnittstelle über das Netzwerk kann die Applikation zur Visualisierung der Trajektorien ohne Änderungen des Tracking Systems unmittelbar integriert werden. Für die Einbindung müssen lediglich die entsprechenden IP-Adressen für die Übertragung festgelegt werden und die eingehenden Daten des AVCs müssen korrekt interpretiert werden.

4.2 TraVis

Die Anwendung ist in 2 Dialogfenster (Kamera Bild und Bing Maps) und einen Menübereich unterteilt. Für dessen Implementierung wurden die Microsoft Foundation Classes (MFC) verwendet, welche eine Sammlung objektorientierter Klassenbibliotheken darstellen.

In dem Dialogfenster, welches das Live Bild der Kamera enthält, werden die Bounding Boxes von erkannten Objekten zusammen mit dessen Trajektorie angezeigt. Ihr Verlauf wird durch den Schwerpunkt der getrackten

Objekte bestimmt. Sobald ein Objekt aus dem Bild verschwindet oder nicht mehr wieder erkannt wird, werden die eingezeichneten Grafikelemente ebenfalls gelöscht. Nach erfolgreicher Kalibrierung werden in der Luftaufnahme der entsprechende Szene, die erkannten Objekte als Punkte und dessen Trajektorien synchron zum Tracking eingezeichnet (siehe Abbildung 6).

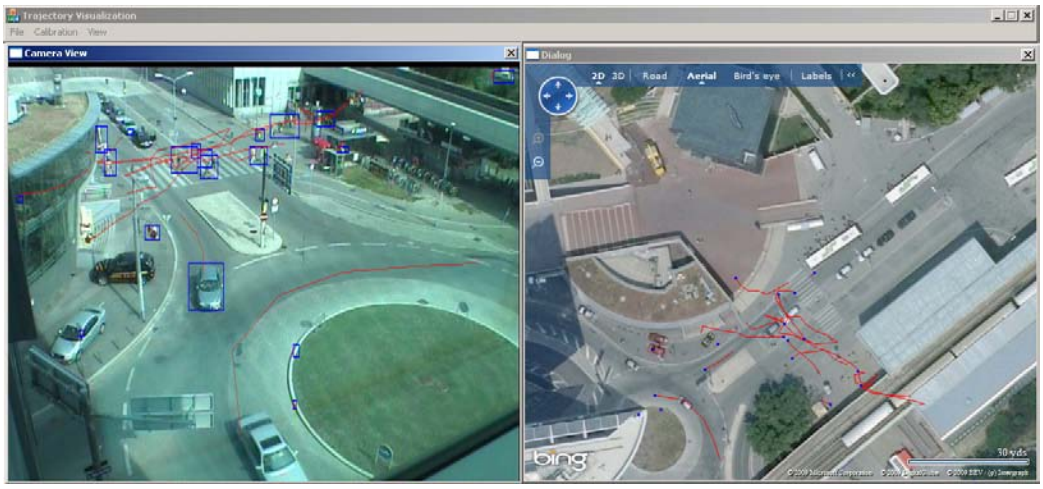


Abbildung 6: Applikation TraVis zur Visualisierung von Trajektorien

Den Kern der Softwarearchitektur von TraVis bilden 8 Klassen (siehe Abbildung 7). Davon stellt die Klasse `CTraVisDlg` den Unterbau dar, über den die Kommunikation zwischen den beiden Dialogen (`CCameraDlg`, `CVirtualEarthDlg`) gesteuert wird. Desweiteren gibt es die Klassen `CTrajectoryManager`, welche für das Verwalten aller Trajektorien und deren Koordinaten in beiden Darstellungen zuständig ist und `CMarkerManager`, der von `CTraVisDlg` im Kalibrierungsmodus verwendet wird. Den mathematischen Teil, die Berechnung der Homographie übernimmt die Klasse `CHomography`. Dessen Methode `caclH()` wird von `CMarkerManager` aufgerufen, sobald eine korrekte Konstellation an Markern vorhanden ist oder diese geändert wurde. Bing Maps wird ausschließlich über ein von Microsoft definiertes AJAX Interface angesprochen. Aus diesem Grund ist `CVirtualEarthDlg` von der MFC Klasse `CDHtmlDialog` abgeleitet. Diese bietet einen Html und JavaScript Interpreter an. Über die Methode `CallJScript()` werden die JavaScript Methoden aufgerufen, die als Schnittstelle zu Bing Maps agieren.

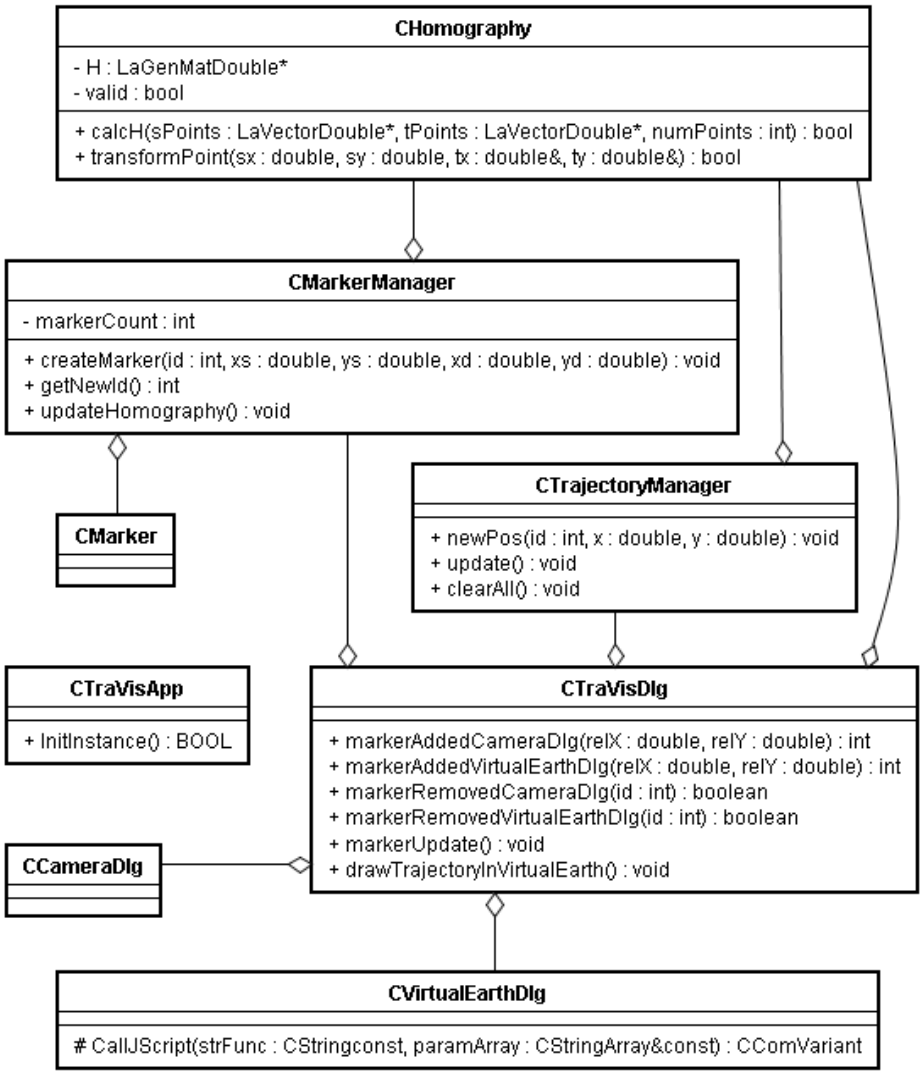


Abbildung 7: Klassendiagramm

5 Evaluierung

Im Folgenden soll untersucht werden, (I) wie genau die manuelle Kalibrierung auf Basis qualitativer Vergleiche durchgeführt werden kann, und (II) mit Hilfe einer Monte Carlo Simulation eine quantitative Aussage über die Fehleranfälligkeit der manuellen Kalibrierung gemacht werden.

Es wird angenommen, dass auf die Pixelgröße des Kamerabildes ein kleinerer oder gleich großer Bereich abgebildet wird, als auf die Größe der Pixel in der Luftaufnahme. Mit den aktuell vorhandenen Luftbildern von Bing Maps trifft diese Annahme im Allgemeinen zu, solange die statische Kamera in einem nicht zu flachen Winkel auf die Erdoberfläche schaut. Unter diesen Vorrausetzungen ist die Positionsgenauigkeit der Marker für die Kalibrierung durch die Auflösung der Luftaufnahmen von Bing Maps beschränkt.

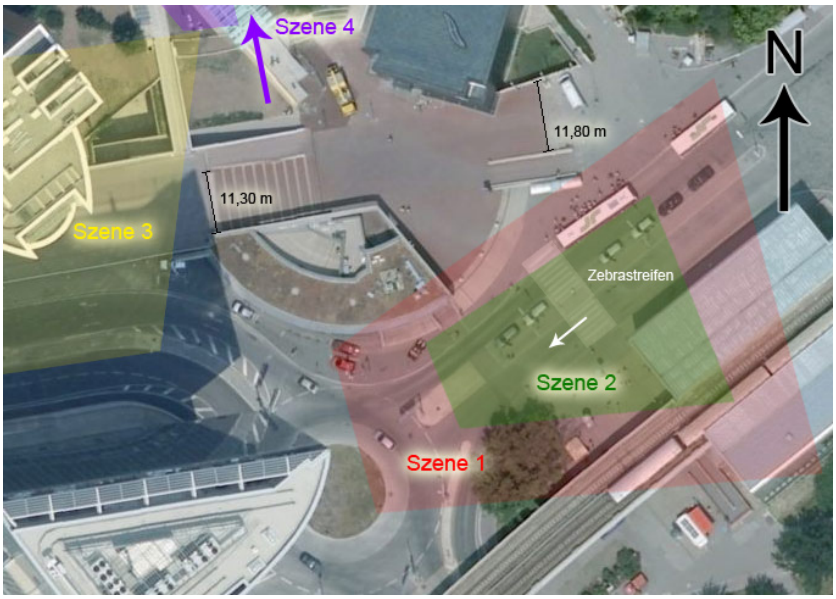


Abbildung 8: Luftbild aus Bing Maps mit gekennzeichneten Szenen

Der Standort der Überwachungskamera, die für die Evaluierung verwendet wurde, befindet sich im Tech Gate Tower in Wien. In Abbildung 8 sind die untersuchten Sichtfelder der verschiedenen Szenen dargestellt. Zu beachten ist, dass die Luftbilder schon mindestens drei Jahre alt sind und nicht mehr vollständig mit der aktuellen Situation übereinstimmen. So ist z.B. der Zebrastreifen auf der Straße heute ein paar Meter südwestlicher als auf dem

Bild. Alle Luftaufnahmen sind wie Abbildung 8 nach Norden ausgerichtet. Um die tatsächliche Auflösung in Bing Maps von dieser Szene in der maximalen Zoomstufe zu bestimmen, haben zwei unabhängige Messungen von 11,30 m bzw. 11,80 m eine abgebildete Pixelgröße von 0,24 m bzw. 0,21 m ergeben. Durchgeführt wurden diese auf zwei Streckenabschnitten (Abbildung 8) und anschließend wurde die mittlere Pixelgröße über die Anzahl der Pixel, die sich auf der Gerade befinden, bestimmt. Für das Kamerabild kann auf Grund der Perspektive keine konstante Pixelgröße angegeben werden.

5.1 Setup / Experimentieraufbau

Abbildung 9: Kamera und Szenenausschnitt

Alle Aufnahmen, die für die Evaluierung verwendet wurden, stammen von einer statischen Kamera, die sich im 4. Stock befindet und auf die umliegenden Straße und Plätze blickt (Abbildung 9). Mit diesem Setup ist es möglich sowohl Szenen mit einem großen also auch welche mit einem flachen Kamerawinkel zu konstruieren.

5.2 Qualitative Studie

Für die qualitative Evaluierung der Kalibrierung wurden vier Testfälle mit jeweils unterschiedlichen Szenen erstellt, drei von ihnen mit einem großen Kamerawinkel (Abbildung 11, 12 und 13) und eine mit einem sehr flachen Winkel. Um zu prüfen wie sich die Anzahl der Marker auf die Kalibrierung auswirkt, werden die abgebildeten Sichtfelder der Kamera im Luftbild miteinander verglichen. Dessen Kanten können für einen anschaulichen Vergleich mit dem Kamerabild herangezogen werden. Der Testfall vier stellt mit seinem sehr großen Sichtfeld eine besondere Konfiguration dar. Es soll ermittelt werden, welchen Einfluss dieses Setup auf die Kalibrierung hat.

(a) Kamerabild (b) Luftbild von Bing Maps

Abbildung 10: Positionsvergleich eines Markerpaares mit Bildausschnitten von 70 x 70 Pixeln. Im Luftbild ist die Gesamtgröße 14,0 x 14,0 Metern mit durchschnittlicher Pixelgröße von 0,2 Metern. Das Kamerabild entspricht einem deutlich kleineren Ausschnitt.

In Abbildung 10 wird der Anfangs erwähnte Auflösungsunterschied deutlich. In Abhängigkeit der Schärfe von markanten Merkmalen im Bild wie z.B. der Ecke in dieser Abbildung können die Marker bis auf ein oder zwei Pixel genau platziert werden. In den drei nachfolgenden Abbildungen 11, 12 und 13 zeigt die erste Spalte jeweils ein Kamerabild von der zugehörigen Szene und in der zweiten Spalte befindet sich der entsprechende Ausschnitt von Bing Maps. Alle Bilder sind im Kalibrationsmodus der Software TraVis entstanden. Dabei nimmt die Anzahl der Marker von oben nach unten zu. Begonnen wird immer mit vier Markern.

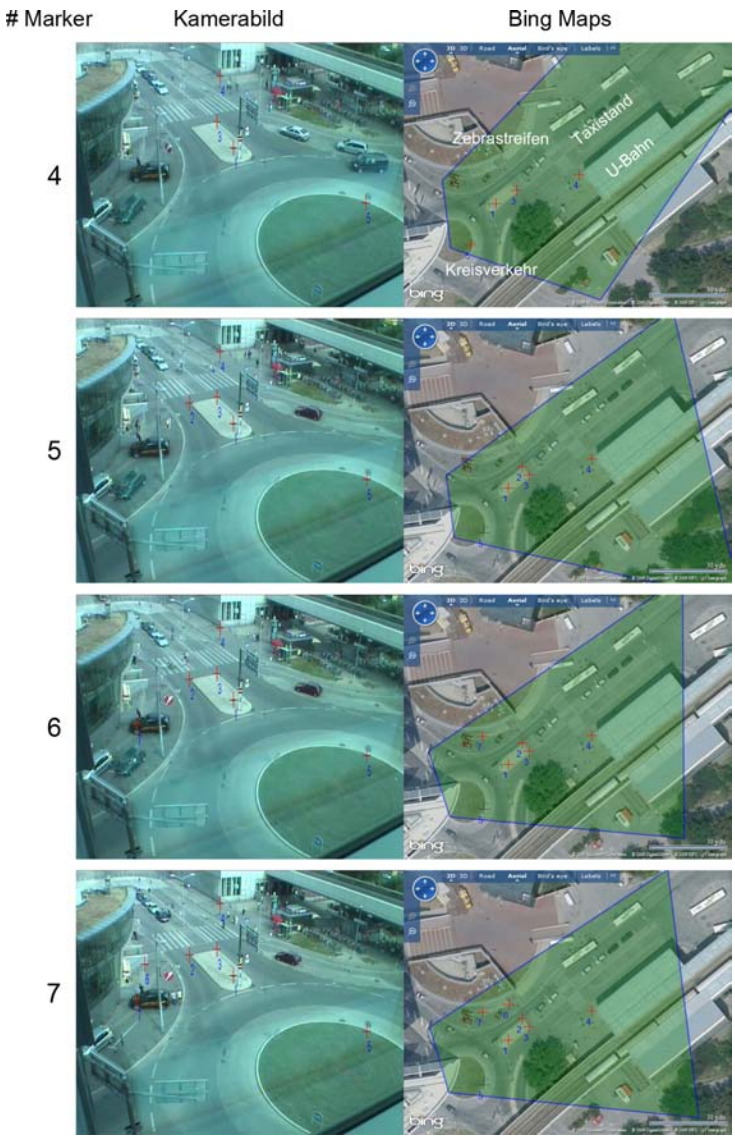


Abbildung 11: Szene 1, Kalibrierung mit unterschiedlicher Anzahl der Markern (rote Kreuze). In der rechten Spalte ist das projizierte Field of View abgebildet. Dessen Grenzen entsprechen mit steigender Anzahl an Markern immer besser der exakten Position, wie beim Vergleich mit den Rändern des Kamerabildes deutlich wird.

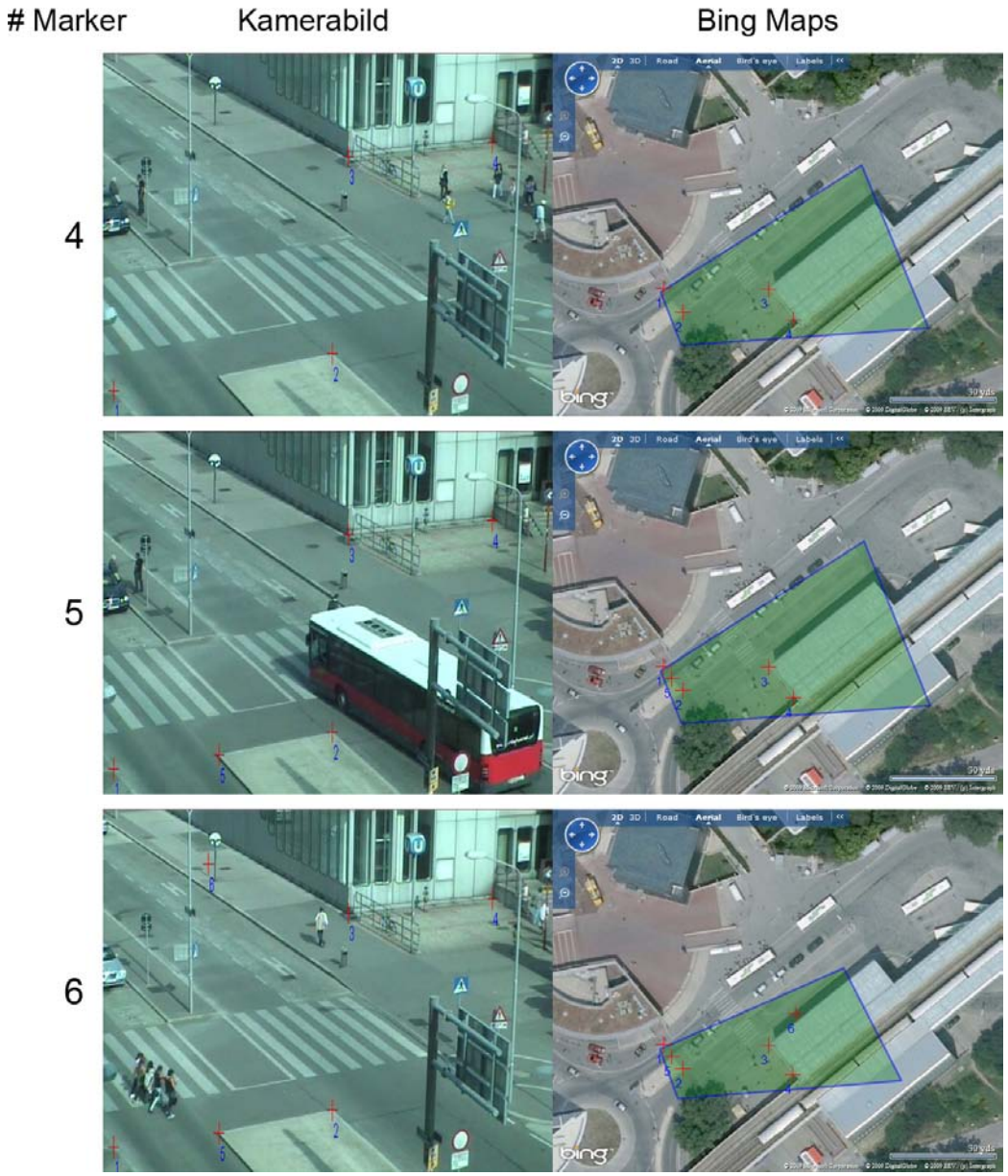


Abbildung 12: Szene 2, Kalibrierung mit unterschiedlicher Anzahl der Markern (rote Kreuze). In der rechten Spalte ist das projizierte Field of View abgebildet. Dessen Grenzen entsprechen mit steigender Anzahl an Markern immer besser der exakten Position, wie beim Vergleich mit den Rändern des Kamerabildes deutlich wird.

Testfall 1 (Abbildung 11) stellt eine typische Konfiguration mit einem großen Sichtfeld der Kamera dar. Es ist ein Kreisverkehr mit einer abzweigenden Straße, über die ein Zebrastreifen verläuft, zu sehen. Angrenzend, oben rechts befindet sich eine U-Bahn Station aus der periodisch große Mengen an Menschen heraus strömen und zum Teil die Straße überqueren. Etwas weiter oben die Straße entlang befindet sich ein Taxistand (siehe Abbildung 11 erste Zeile, zweite Spalte). Aufgrund der Kameraperspektive und der nicht konsistenten Position des Zebrastreifens, gibt es nur relativ wenige Features im Bild die als präzise Markerpositionen genutzt werden können. Drei der vier Marker in der ersten Zeile liegen mit kleinen Abweichungen auf einer Gerade. Dadurch wird die Berechnung der Homographie numerisch instabil und das dargestellte Field of View weist entsprechend deutliche Fehler auf. Durch die Hinzunahme eines weiteren Markers wird die Transformation deutlich verbesssert. Der sechste Marker schließlich beeinflusst den rechten oberen Bereich des Kamerabildes und bildet diese Ecke auf einen Bereich hinter der U-Bahn Station ab. Durch die Verdeckung der U-Bahnstation Kamerabild kann nicht die Korrektheit bestimmt werden, aber aufgrund des rechten Bildrandes der Kamera scheint diese Position schon auf wenige Meter genau zu sein. In der letzten Zeile wurden sieben Marker für die Kalibrierung verwendet. Im Vergleich zu der vorherigen Konfiguration mit sechs Marker gibt es nur noch minimale Änderungen des Field of Views. Am stärksten wird die linke obere Ecke des projizierten Kamerabildes verändert.

Folgender Testfall 2 (Abbildung 12) vermittelt einen ähnlichen Eindruck der Ergebnisse wie der Erste. Hier wurde die Kameraposition nicht geändert, jedoch mit Hilfe eines optischen Zooms ein deutlich kleinerer Bildbereich gewählt. Die größten Änderungen bilden hier die Positionen der Marker im Kamerabild. Im Unterschied zur ersten Szene, in der die Marker im Kamerabild relativ gleichmäßig über den mittleren Bereich verteilt waren, befinden diese sich nun deutlich dichter am Rand. Das Ergebnis daraus ist, dass das projizierte Field of View deutlich schneller in Abhängigkeit der Marker-Anzahl auf einen stabilen Zustand hin konvergiert.

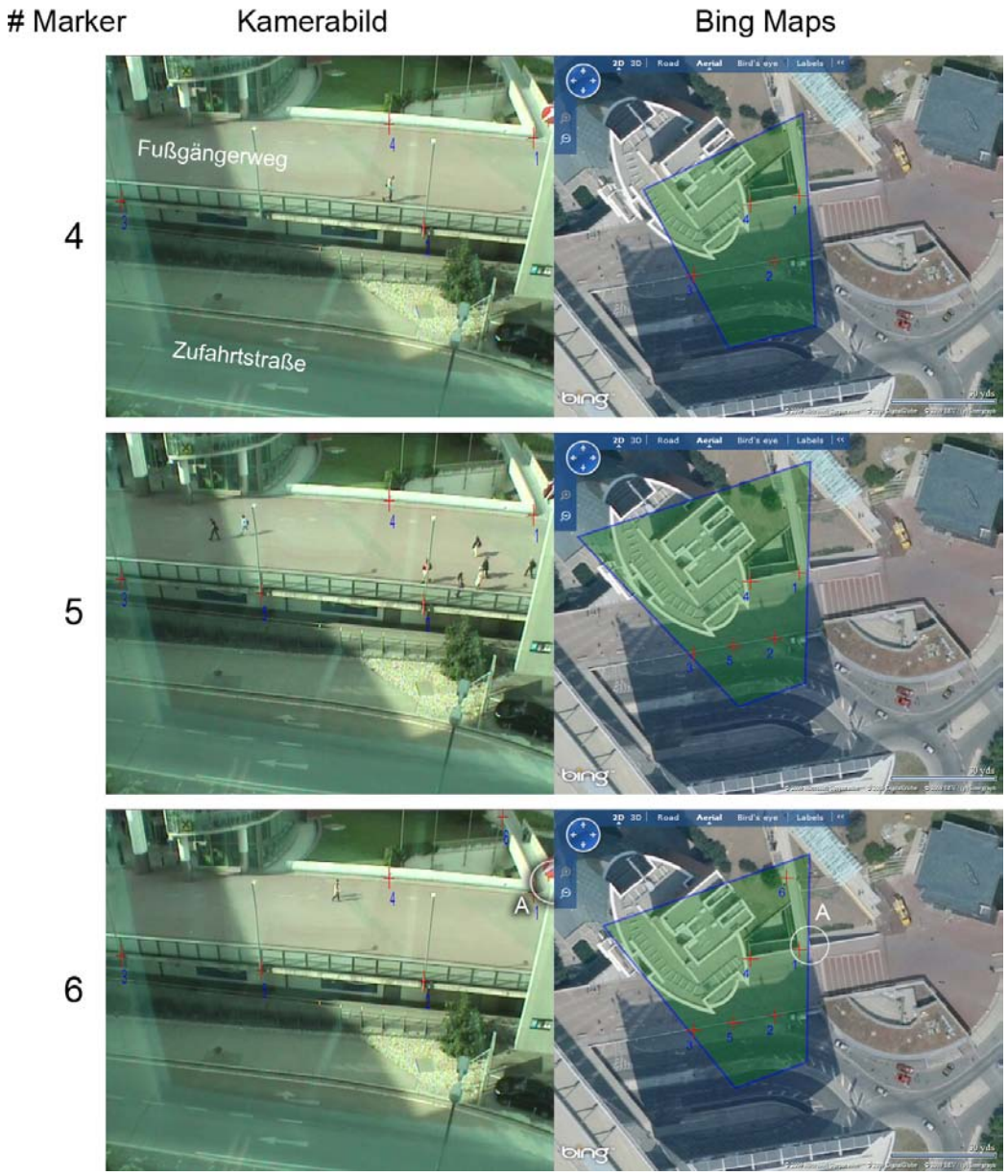


Abbildung 13: Szene 3, Kalibrierung mit unterschiedlicher Anzahl der Markern (rote Kreuze). In der rechten Spalte ist das projizierte Field of View abgebildet. Dessen Grenzen entsprechen mit steigender Anzahl an Markern immer besser der exakten Position, wie beim Vergleich mit den Rändern des Kamerabildes deutlich wird.

Wie bereits weiter oben erläutert, transformiert die Homographie alle Punkte einer Ebene in eine zweite Ebene. Aus diesem Grund wurde als dritten Testfall (Abbildung 13) ein Schauplatz gewählt der nicht nur aus einer planaren Fläche besteht. Es befindet sich im unteren Teil des Bildes eine Zufahrtstraße und weiter oben eine breiter Fußgängerweg (Abb. 13, erste Zeile, erste Spalte). Die Marker wurden in diesem Fall auf dem Fußgängerüberweg platziert, um Trajektorien in diesem Bereich abbilden zu können. Objekte die sich auf der Zufahrtstraße befinden, können mit dieser Kalibrierung nicht korrekt transformiert werden, da diese sich nicht in der gleichen Ebene befinden. Auch in dieser Szene stabilisiert sich die Homographie von vier bis sechs Markern sehr genau. Zum Beispiel schneidet der rechte Rand des Kamerabildes eine kleine Brücke (siehe Abb. 13, 3. Zeile, Markierung A). Auch im Luftbild schneiden die Grenzen des projizierten Sichtbereiches diesen Punkt sehr genau.

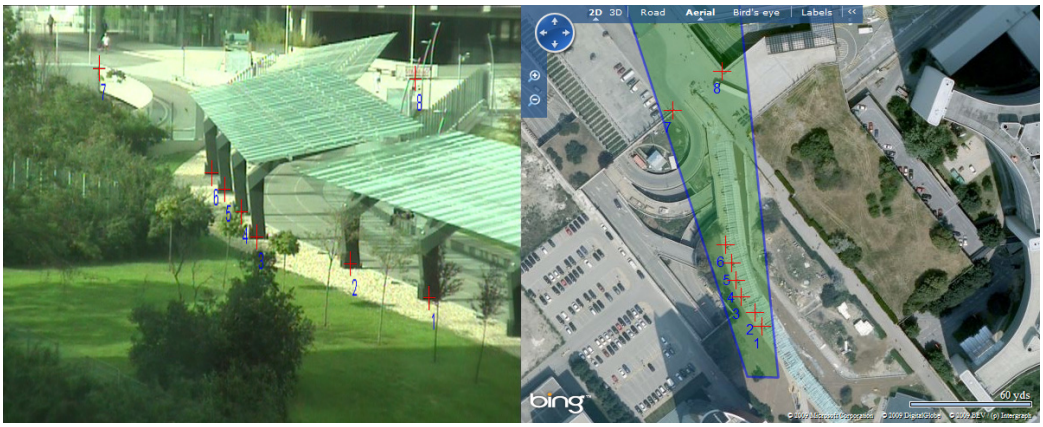


Abbildung 14: Szene 4, Kalibrierung mittels 8 Markern. In der rechten Spalte ist aufgrund des flachen Kamerawinkels der projizierte Field of View sehr schmal und in die Länge gezogen. Das Luftbild stammt aus der zweit höchsten Zoomstufe aus Bing Maps und deshalb entspricht jeder Pixel einem größeren Bereich als 0,2 Meter.

Die vierte Szene (Abbildung 14) stellt einen Grenzfall bzgl. der Homographie Bestimmung dar. Der am weitesten entfernte Bildbereich des Kamerabildes wird von einem Gebäude verdeckt, wodurch dort keine Marker für die Kalibrierung platziert werden können. Um für diese Szene die bestmöglichste projektive Transformation zu bestimmen wurden 8 Marker platziert. Das

Field Of View vermittelt in dieser Abbildung den Eindruck, dass zumindest im vorderen Bereich die Kalibrierung ähnlich genau wie in den Szenen zuvor bestimmt werden konnte. Ausschließlich der hintere Bereich verläuft sehr weit aus dem Bildausschnitt heraus.

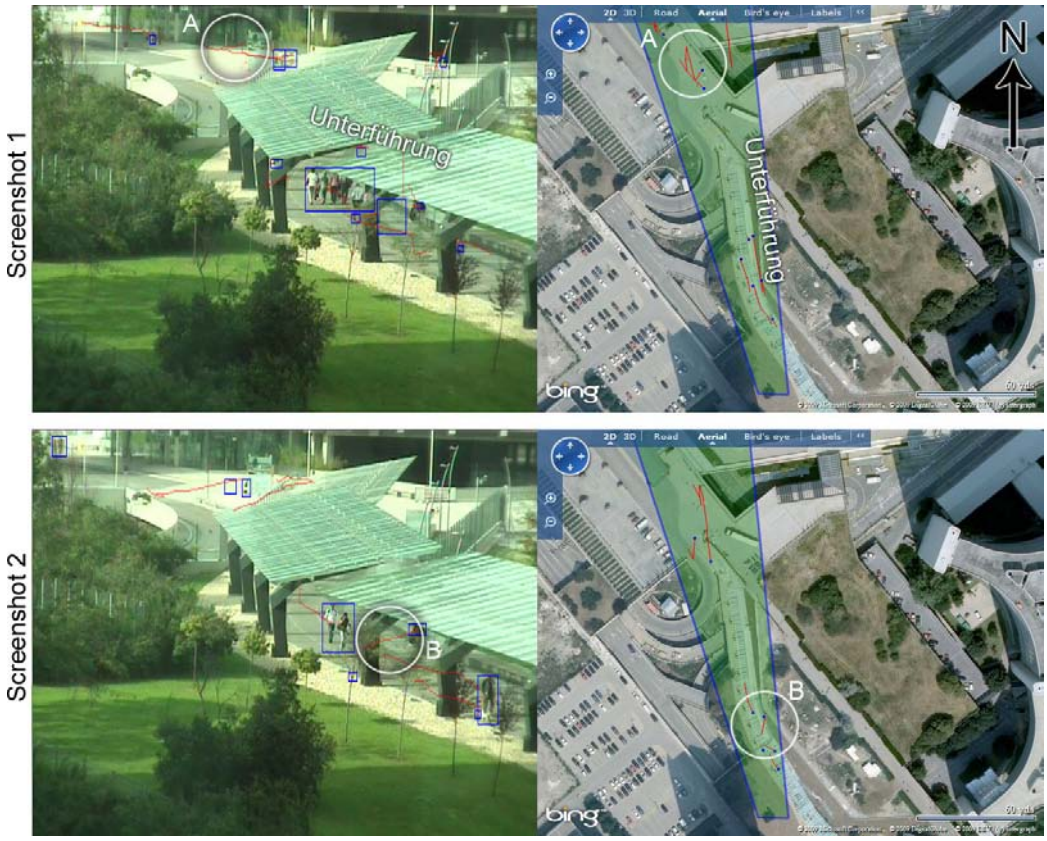


Abbildung 15: Szene 4, Screenshots aus TraVis für einen qualitativen Vergleich der abgebildeten Trajektorien. Bei den Markierungen A und B sind deutliche Fehler in der transformierten Position vorhanden.

Um die Kalibration der Szene 4 genauer evaluieren zu können, werden zwei Screenshots (Abbildung 15) aus der Applikation TraVis verglichen, denen die Kalibration aus Abbildung 14 zugrunde liegen. Beim genaueren Betrachten dieser Bilder wird deutlich, dass die Positionen der getrackten Personen im Bereich der Unterführung (siehe Abbildung 15, Screenshot 1) sehr präzise im Luftbild dargestellt werden. Dies gilt insbesondere für Personen die sich in vertikaler Richtung auf dem Luftbild bewegen. Der zweite Screens-

hot enthält im vorderen Bereich eine kurze Trajektorie (Abbildung 15, B) die senkrecht zur Unterführung verläuft. Diese wird im Satellitenbild durch eine Bewegung Richtung Norden jedoch mit einem großen Fehler abgebildet. Weitere größere Fehler treten vor allem im hinteren Bereich des Sichtfeldes der Kamera auf. Im ersten Screenshot zum Beispiel, befindet sich dort eine anormale Trajektorie (Abbildung 15, A). Für die transformierte Trajektorie ergibt dies ein sehr große Abweichung Richtung Norden, was einer Rückwärtsbewegung des Objektes entspricht. Tatsächlich hat sich die getrackte Person in diesem Bereiche ausschließlich in gerader Richtung bewegt. Ähnliche Resultate können bei den benachbarten Trajektorien beobachtet werden.

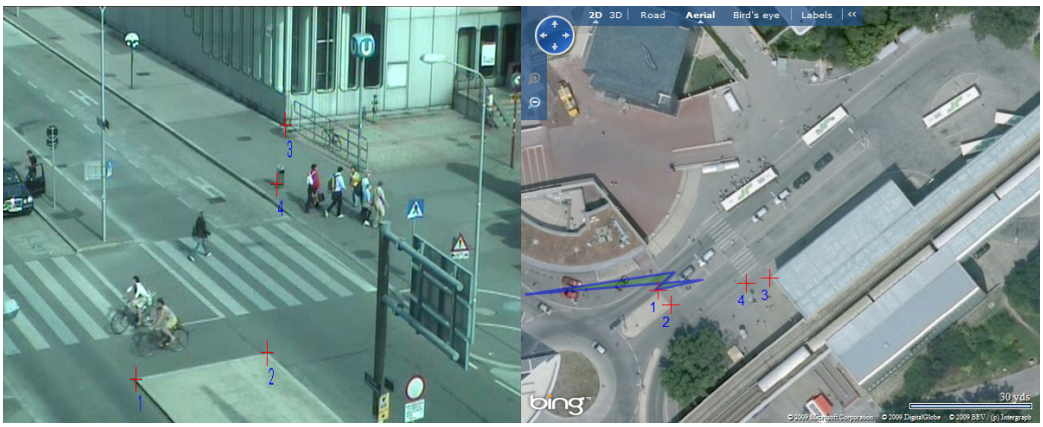


Abbildung 16: Fehlerhafte Kalibrierung aufgrund von mangelhaften Markerpositionen. Marker 2,3 und 4 liegen fast auf einer Linie. Das projizierte Field of View ist vollständig deformiert.

Weitere fehlerhafte Transformationen können aus numerisch instabilen Lösungen des Gleichungssytems zur Bestimmung der Homographie resultieren. Dieser Fall liegt vor, wenn zu wenige Gleichungen numerisch stabil, linear unabhängig sind. Für die Marker entspricht dies einer Anordnung auf einer Geraden, d.h. mehrere Marker sind kollinear (Abbildung 16).

5.3 Quantitative Evaluierung

Die Ergebnisse aus Kapitel 5.2 haben visuell gezeigt, dass die Transformation sehr genau bestimmt werden kann, solange die Kamera mit einem großen Winkel auf die Szene schaut. Unter der Voraussetzung einer bestmöglichen Kalibrierung wird im Folgenden die Streuung transformierter Punkte innerhalb des Field of Views und somit die Robustheit der projektiven Transformationen untersucht. Hierfür wird jeweils für die vier Szenen (siehe 5.2) eine Monte Carlo Simulation durchgeführt.

In einem regelmäßigen rechteckigen Gitter auf dem Kamerabild werden Punkte ausgewählt, die mit Gaussian noise gestört und ins Luftbild transformiert werden. Die Varianz der zweidimensionalen Gauß Verteilung beträgt in jedem Fall, sowohl für die x- also auch für die y-Komponente 3 Pixel. Um stabile Schätzungen zu erhalten, werden mindestens 10.000 Samples pro Pixel generiert. Die Ergebnisse werden auf zwei Arten visualisiert. In den folgenden Abbildungen befindet sich auf dem linken Bild die Darstellung der Fehlerellipsen pro gestörten Pixel. Hierbei handelt es sich um den Bereich im Luftbild auf den mehr als 99 Prozent der Gauß-Verteilung transformiert wird. Damit stellt sich heraus, wie störempfindlich die Homographie ist und auf welchen Bereich im Luftbild besonders kleine bzw. große Fehler gemacht werden. Die zweite Darstellungsform bildet eine interpolierte Farbcodierung der gemittelten Varianzen der Fehlerellipsen ab. Normiert ist diese auf den maximalen Wert. Hier wird die relative Störempfindlichkeit innerhalb des projizierten Field of Views deutlich.

Alle vier untersuchten Szenen spiegeln ein grundsätzlich ähnliches Ergebnis wider. Aufgrund der Perspektive nimmt die Varianz von Vorne nach Hinten im Kamerabild zu. Dies ist durch die tatsächliche Pixelgröße im Bild bestimmt. Absolute Unterschiede ergeben sich durch die tatsächliche Größe des Bildbereiches und durch den Kamerawinkel.

Die erste untersuchte Szene beinhaltet einen sehr großen Field of View, d.h. die Kamera bildet einen großen Bereich ab. Die Fehlerellipsen in Abbildung 17 sind im vorderen Bereich noch Kreisrund und mit zunehmender Entfernung vom Projektionszentrum steigt ihre Exzentrizität und ihre Größe. Daraus resultiert, dass ein getracktes Objekt, welches sich von der Kamera weg bewegt, mit einem immer größer werdenden Fehler abgebildet wird und eines, welches sich parallel zur Kamera bewegt einen konstanten Fehler erfährt. In diesem Fall ist der Fehler ebenfalls von der Entfernung des Objektes zur Kamera abhängig ist. In der farbcodierten Darstellung stellt

sich heraus, dass die mittlere Standardabweichung in der vorderen Bildhälfte (Abbildung 17, A) maximal 2 Pixel beträgt und bis zur linken oberen Bildecke (Abbildung 17, B) auf über 8 Pixel ansteigt.

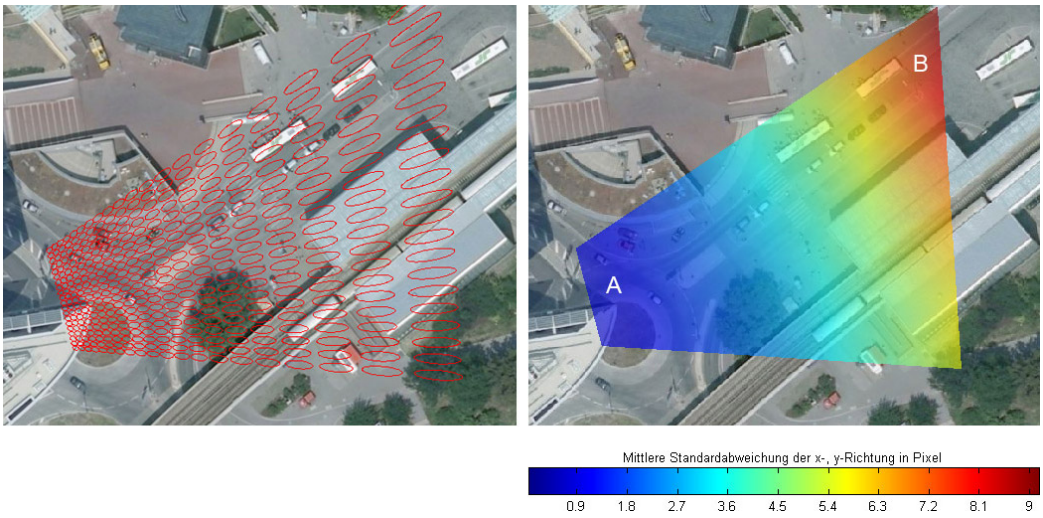


Abbildung 17: Szene 1, linkes Bild zeigt Fehlerellipsen von abgebildeten Punkten, welche mit Gauss-Noise ($\sigma = 3.0$) gestört wurden. Rechtes Bild zeigt die mittlere Standardabweichung für die x- und y-Richtung in einer farbkodierten Darstellung des projizierten Sichtbereiches der Kamera.

Ein maximal halb so großen projizierten Sichtbereich der Kamera als in der ersten Szene, hat das zweite Szenario (Abbildung 18). Die Fehlerellipsen sind hier sehr ähnliche angeordnet, haben aber einen deutlich kleineren Radius. Ihre zugehörigen Gauß-Verteilungen haben eine mittlere Standardabweichung von anfangs 0,6 bis 1,2 Pixel (blau, Abbildung 18, A) bis maximal knapp über 2,2 Pixel (rot, Abbildung 18, B)). Auch hier ist ein zunehmendes Wachstum von vorne nach hinten zu beobachten. Aufgrund des kleinen projizierten Field of Views können die Trajektorien im blauen Bereich bis auf ca. 0,2 m (1 Pixel) genau dargestellt werden.

In Szene 3 kann die maximale mittlere Standardabweichung auf weniger als 2 Pixel reduziert werden (Abbildung 18), jedoch ist der gesamte vordere Bildbereich aufgrund der tiefer gelegenen Straße nicht korrekt kalibriert. Der relevante Bereich dieser Szene, auf den die Kalibrierung angepasst wurde (siehe Kapitel 5.2, hat eine Abweichung von 1 bis 1,5 Pixel. Dieses sehr gute Ergebnis kann auf den steilen Kamerawinkel zurück geführt werden.

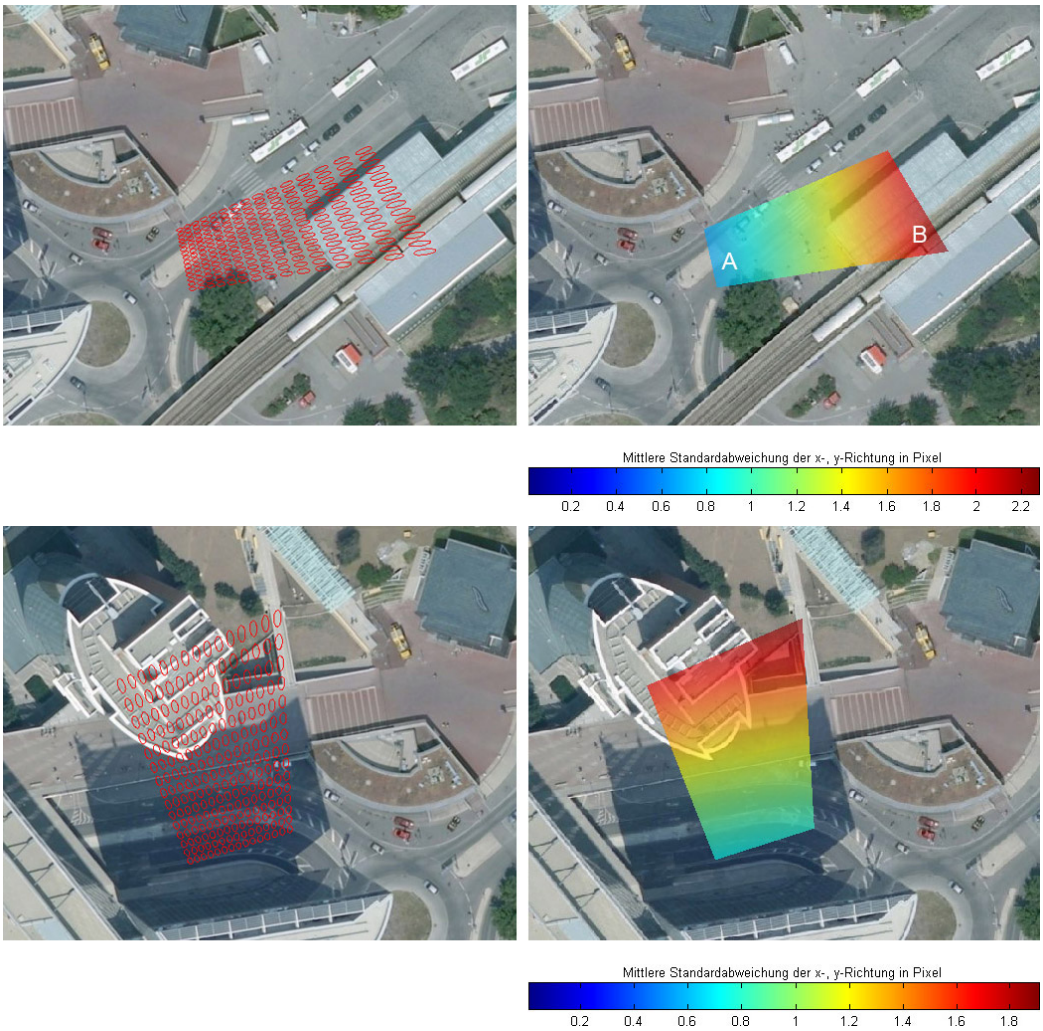


Abbildung 18: Szene 2 (oben) und 3 (unten), linkes Bild zeigt Fehlerellipsen von den abgebildeten Punkten, welche mit Gauss-Noise ($\sigma = 3.0$) gestört wurden. Rechtes Bild zeigt die mittlere Standardabweichung für die x- und y-Richtung in einer farbkodierten Darstellung des projizierten Sichtbereiches der Kamera.

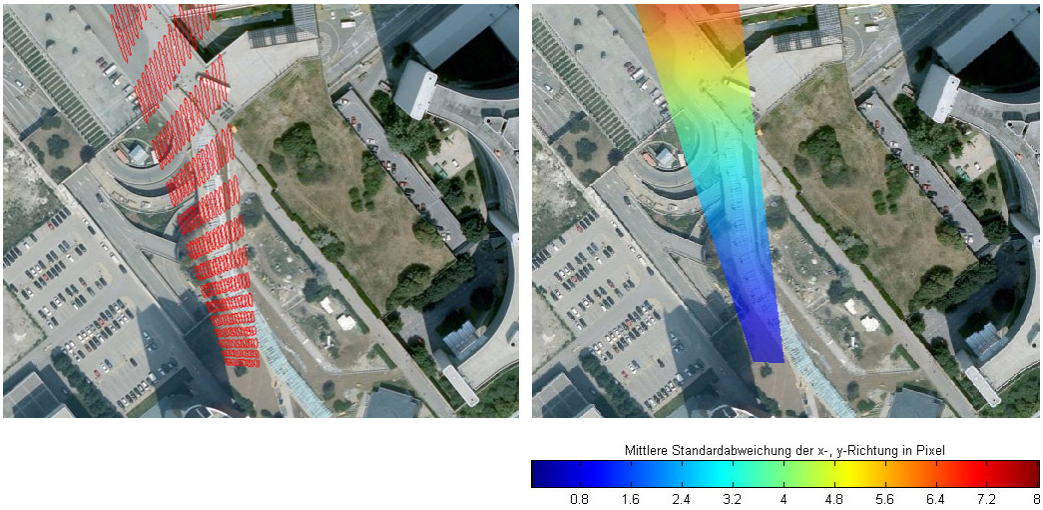


Abbildung 19: Szene 4, linkes Bild zeigt Fehlerellipsen von abgebildeten Punkten, welche mit Gauss-Noise ($\sigma = 3.0$) gestört wurden. Rechtes Bild zeigt die mittlere Standardabweichung für die x- und y-Richtung in einer farbkodierten Darstellung des projizierten Sichtbereiches der Kamera.

Das letzte der vier untersuchten Szenarien zeigt die Grenzen dieses Verfahrens auf (Abbildung 19). Der sehr niedrige Kamerawinkel und der länglich verzerrte projizierte Sichtbereich resultieren in sehr großen Standardabweichungen von über 8 Pixel im hinteren Bildbereich. Unter Verwendung der zweitgrößten Zoomstufe von Bing Maps entspricht hier ein Pixel einem größeren Bereich als der eines Pixels aus den ersten drei Szenen. Die Fehlerellipsen im hinteren Bereich weisen eine sehr große Exzentrizität auf, was den starken Fehler in vertikaler Richtung in Kapitel 5.2 erklärt. Die scheinbar korrekten Ergebnisse aus 5.2 im blauen Bereich beinhalten tatsächlich einen geringen Fehler in horizontaler und große Abweichungen in vertikaler Lage. Da Aufgrund der Perspektive der zurückgelegte Weg der Objekte nicht sehr genau abgeschätzt werden kann, fällt der Fehler in vertikaler Richtung kaum auf.

5.4 Ergebnisse

In den nachfolgenden Abbildungen 20 und 21 sind Screenshots der Software TraVis dargestellt. Sie beinhalten Livebilder der ersten drei Szenen mit jeweils der optimalsten Kalibrierung.

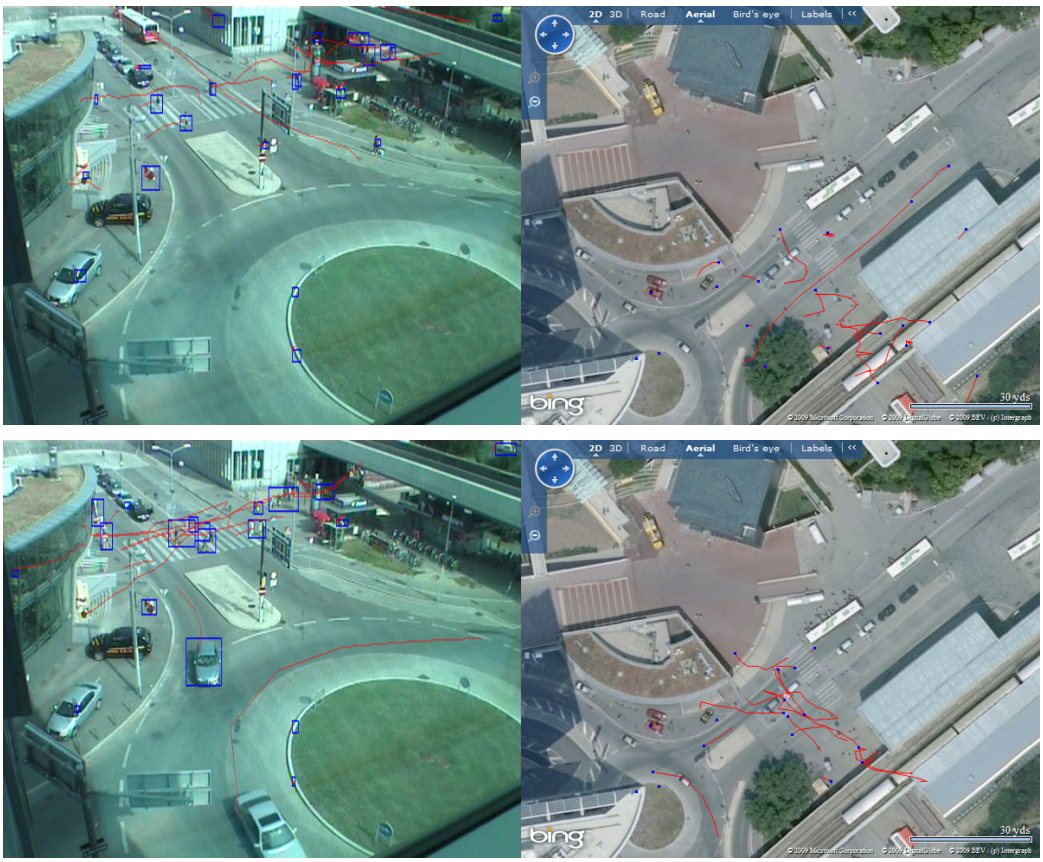


Abbildung 20: Szene 1, Screenshots aus TraVis mit live Darstellung der verfolgten Objekte und dessen Trajektorien im Kamerabild und in Bing Maps.

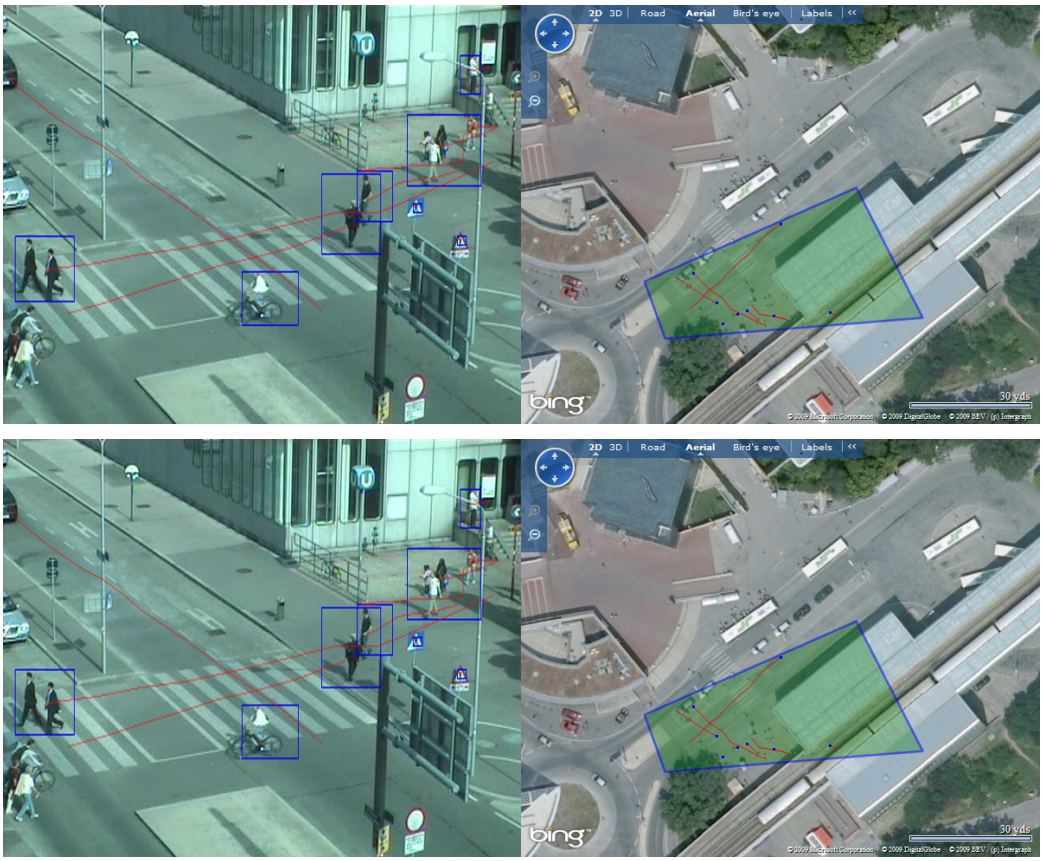


Abbildung 21: Szene 2, Screenshots aus TraVis mit live Darstellung der verfolgten Objekte und dessen Trajektorien im Kamerabild und in Bing Maps.

6 Ergebnisse und Ausblick

Der Online-Kartendienst Bing Maps von Microsoft wurde für die Entwicklung einer Applikation zur Echtzeitvisualisierung von Trajektorien verwendet. Die Auflösung der Luftbilder, die Microsoft zu Verfügung stellt, hat eine Grenze überschritten, die es ermöglicht zu Kamerabildern adäquate Ansichten einer Szene von oben zu bieten. Um diese Möglichkeit effektiv nutzen zu können, muss zunächst eine Kalibrierung zwischen Kamerabild und der Luftaufnahme durchgeführt werden. Hierfür müssen mindestens 4 Punktkorrespondenzen gefunden werden, um die notwendige projektive Transformation bestimmen zu können. Aufgrund der Auflösungsdifferenzen und teilweise älteren Satellitenfotos ist das Finden von gleichen Bildpunkten nicht trivial. Aus diesem Grund wird dieser Schritt in der entwickelten Applikation manuell durchgeführt.

Mit dieser Arbeit konnte gezeigt werden, dass die Kalibrierung von Kamera- und Luftbild in den meisten Fällen sehr genau bestimmt werden kann. Hierbei sind im Mittel 7 Punktkorrespondenzen notwendig. Wenn die Kamera sehr steil von schräg oben auf die Szene gerichtet ist, kann mittels der manuell bestimmten Kalibrierung eine Transformation vom Großteil des Kamerabildes bis auf einen Pixel genau durchgeführt werden. Dies entspricht einer Genauigkeit von ca. 0,2 m im Luftbild. Ebenfalls konnten die Grenzen des Verfahrens aufgezeigt werden. Diese sind durch den Kamerawinkel auf die Szene bestimmt. Sobald die Kamera mit einem sehr flachen Winkel ausgerichtet ist und somit einen sehr großen Bereich nach hinten aufnimmt, steigt der Fehler auf über 8 Pixel im Luftbild an, was keine akzeptablen Ergebnisse mehr darstellt.

Sollten zukünftig Luftbilder mit noch höheren Auflösungen angeboten werden, könnte die Kalibrierung automatisch, z.B. mittels Extraktion von Features aus den Bildern, durchgeführt werden. Des Weiteren hat Microsoft Bing Maps einen 3D-Modus integriert, dessen Programmierschnittstelle jedoch noch sehr unflexibel ist. Sollte seine API erweitert werden, könnte der Webdienst für noch komplexere Visualisierungen genutzt werden. Es wäre dann möglich, die Szene nicht nur von oben zu betrachten, sondern aus jeder beliebigen Perspektive oberhalb der Erdoberfläche. Der 3D-Modues bietet zudem eine große Anzahl an 3D-Modellen der Gebäude einiger Großstädte an. Denkbar wäre diese Information in ein Kamerabild zu transformieren um damit auf einen Teil der drei dimensionalen Szeneninformation im Bild zu schließen.

7 Danksagung

Ganz besonders möchte ich mich bei Roman Pflugfelder bedanken! Seine fachliche Unterstützung und die vielen anregenden Ideen haben mir während dieser Arbeit sehr geholfen, mich inspiriert und mich auch in meinem Studium weitergebracht.

Ich bedanke mich bei Herrn Markus Clabian, dem Leiter des Geschäftsfeldes „Video and Security Technology“ am AIT, für den Vorschlag und die angebotenen Möglichkeiten dieses Thema bearbeiten und umsetzen zu können. Ich freue mich ebenfalls über die kompetente Unterstützung des gesamten Teams des Geschäftsfeldes und bedanke mich dafür, dass wann immer ein scheinbar unüberwindbares Problem auftrat, ich hilfreiche Ratschläge bekomme habe. Namentlich möchte ich Michael Dittrich erwähnen, mit dessen Hilfe ich viele Hürden während der Implementierung überwinden konnten.

Bedanken möchte ich mich bei meinen Eltern, die mich stets motivieren meine Ziele zu erreichen! Ein sehr großer Dank geht an meine Schwester Jana Kropp für ihre kurzfristige und fachlich kompetente Unterstützung beim Korrekturlesen.

Ein besonderer Dank geht an Natascha Husar, die mir während des Verfassens dieser Arbeit immer zur Seite stand und mich hilfreich Unterstützt hat!

Wien, den 19.10.2009
Timo Kropp

Literatur

[1] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):90–99, 1986.

[2] D. Brown, D.R. Dunn. Trajectory visualization by using global positioning systems (gps). In *SoutheastCon, 2005. Proceedings. IEEE*, pages 183–186, Piscataway, NJ, USA, 2005. IEEE.

[3] S. Fleck, C. Vollrath, F. Walter, and W. Strasser. An integrated visualization of a smart camera based distributed surveillance system. In *ACST'07: Proceedings of the third conference on IASTED International Conference*, pages 234–242, Anaheim, CA, USA, 2007. ACTA Press.

[4] Google. Google earth, August 2009. `http://earth.google.com/`.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[6] heise online. Microsofts virtuelle erde mit realen details, Mai 2007. `http://www.heise.de/newsticker/Microsofts-virtuelle-Erde-mit-realen-Details--/meldung/90238`.

[7] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[8] Microsoft. Bing maps, August 2009. `http://www.bing.com/maps/`.

[9] Microsoft. Bing maps for enterprise from microsoft - developer resources, online map apis, sdks, and map web services, August 2009. `http://www.microsoft.com/maps/developers/`.

[10] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, pages 255–261 vol.1. IEEE, 1999.

[11] S. Veigl. Visuelle Fahrzeugverfolgung mittels Embedded Systems. Diplomarbeit, Technische Universität Wien, Austria, März 2008.