Universität der Bundeswehr München

Institut für Angewandte Informatik

Aussehensbasierte generative hierarchische Interpretation von Fassaden in terrestrischen Bildsequenzen

Dissertation

Sergej Reznik

Inhaltsverzeichnis

Kapitel 1

Einleitung und Zielsetzung der Arbeit

Mit Google Earth (earth.google.com) und Microsoft Virtual Earth (www.microsoft.com/VirtualEarth) sind topographische Daten in großem Umfang und mit hoher Qualität für Anwendungen, wie z.B. Routenplanung, frei verfügbar. Die momentan stattfindende Ergänzung der Daten um detaillierte dreidimensionale (3D) Modelle für Gebäude in Städten erweitert die möglichen Anwendungen in Richtung Stadtplanung und Architektur aber auch (virtuellen) Tourismus und Filmproduktion.

Wegen des großen Aufwandes für die Datenerfassung sind bisher die 3D Modelle in den meisten Fällen sehr einfach und auf (meist ebene) Flächen für Dächer und Fassaden und somit einfache Anwendungen beschränkt. Im Rahmen dieser Arbeit wird daher gezeigt, wie auf der Grundlage von terrestrischen Bildsequenzen Detailstrukturen auf Fassaden in der Form der Fenster und deren 3D Ausprägung automatisch erfasst werden können. Die Fassaden werden hierzu hierarchisch auf Grundlage von aussehensbasierter Extraktion und statistischer Modellierung interpretiert.

Der primäre Fokus ist die Entwicklung eines lernbasierten aussehensbasierten Verfahrens zur Teile-basierten Detektion von Objekten, die auf der Fassadenebene liegen. Fenster sind wesentliche Bestandteile der meisten Fassaden und charakterisieren diese aufgrund ihrer Gestalt und Anordnung. Sie sind damit die Basis für den Aufbau von generativen Fassadenmodellen. Die verwendeten Ansätze zur aussehensbasierten Objektextraktion sind vereinfachte Implicit Shape Models (ISM) und die Bayes-Modellierung mittels Markov Chain Monte Carlo (MCMC).

Die Fokussierung auf terrestrische Bilddaten in Form von Bildsequenzen mit einer großen Basis zwischen den Bildern (Wide-Baseline) ergab sich zunächst daraus, dass für diese im direkten wissenschaftlichen Umfeld Know-how und leistungsfähige Software vorhanden ist. Darüber hinaus ermöglicht diese Datengrundlage eine äußerst effiziente und wenig aufwändige Datenerfassung.

Grundlagen für die Arbeit werden in Kapitel 2 beschrieben. Dies ist zunächst die aussehensbasierte Objektextraktion, speziell ISM, die eine bild- und anordungsbasierte Detektion aber auch Segmentierung von Objekten ermöglichen, wobei wichtige Teile des Modells aus Daten gelernt werden. Mit Maßstabsräumen werden Bilder auf verschiedenen Maßstäben

repräsentiert. MCMC ist ein Standardverfahren für die statistische Optimierung. Generative statistische Modelle basieren auf einer Repräsentation der Szene, mittels derer das Bild generiert werden kann. Ziel der (statistischen) Optimierung ist es, die Parameter der Repräsentation so anzupassen, dass das vorliegende und das generierte Bild möglichst ähnlich werden. Bei der Modellauswahl wird aus mehreren Modellen das gewählt, welches möglichst gut zu den Daten passt, zugleich aber eine kleine Zahl von Parametern hat. Die letzte Bedingung vermeidet, dass zu komplexe Modelle verwendet werden.

Das vorgestellte Verfahren zur (3D) Modellierung der Fassaden greift eine Reihe von Ideen auf, die in bisherige Arbeiten vorgeschlagen wurden. Diese sind in Kapitel 3 beschrieben.

Kapitel 4 gibt nach der Vorstellung des verwendeten Fassadenmodells einen Überblick über den vorgestellten Ansatz und stellt vorbereitenden Maßnahmen in Form der Orientierung der Bilder und der Generierung der Fassadenebenen vor.

Der Kern der Arbeit wird in den Kapiteln 5, 6 und 7 beschrieben. Kapitel 5 stellt die Hypothesengenerierung für Fenster mittels ISM dar. Hierbei werden Bildausschnitte um markante Punkte gelernt, deren Anordnung in Bezug auf den Mittelpunkt des Fensters beschrieben wird.

Die multiskalige generative Interpretation ist Inhalt von Kapitel 6. Hierbei werden Fenster mittels morphologischer Maßstabsräume abstrahiert. Diese Vorgehensweise ist auf dunkle Fenster auf hellen Fassaden beschränkt.

Der im Kapitel 5 vorgestellte Ansatz zur generativen aussehensbasierten Fassadeninterpretation ist dagegen auf helle und dunkle Fassaden anwendbar. Es wird gezeigt, dass ISM auch für die Segmentierung verwendet werden können und wie die gefundenen Fenster validiert werden. Die hierarchische Modellierung der Fassade verwendet Modellauswahl und die Bestimmung der 3D Struktur der Fenster erfolgt auf Grundlage von Plane Sweeping.

Kapitel 8 beschreibt Experimente zur Modellauswahl und stellt Ergebnisse für die 3D Modellierung von Fassaden vor. Die Evaluierung von Ergebnissen führt zu deren Bewertung.

Kapitel 9 gibt eine Zusammenfassung der Arbeit und schließt sie mit einem Ausblick ab.

Kapitel 2

Grundlagen

2.1 Aussehensbasierte Objektextraktion – Implicit Shape Models (ISM)

Die grundlegende Idee der aussehensbasierten Objektextraktion besteht aus Sicht dieser Arbeit darin, ein Objekt durch Bildausschnitte um interessante Punkte herum und deren Anordnung im Bild zu beschreiben. Welche Ausschnitte und welche Anordnung ein Objekt charakterisieren, wird aus Bilddaten gelernt.

(Weber et al. 2000) benutzen Cluster von Bildausschnitten um Förstnerpunkte (Förstner und Gülch 1987) zur Erkennung von Gesichtern und Autos. In Abbildung 2.1 sind Beispiele für Cluster von Bildausschnitten für Gesichter (Mitte) und Autos (rechts) dargestellt. Die Bildausschnitte für Autos wurden mit einen Gradientenfilter bearbeitet. Es wurden insgesamt 81 Testmuster für Autos und 81 für Gesichter ausgewählt.

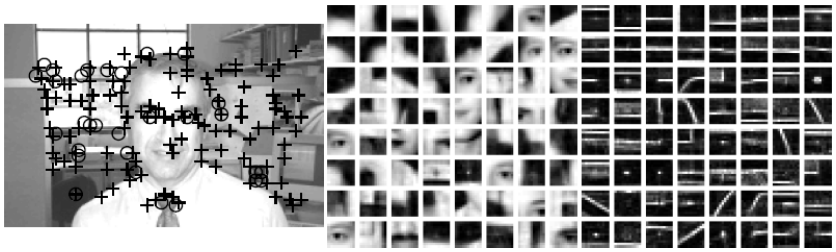


Abbildung 2.1: *Förstner Interest Punkte in einem Trainingsbild eines menschlichen Gesichtes mit verrauschtem Hintergrund (links). Kreuze kennzeichnen eckige Muster während Kreise runde Muster markieren. In der Mitte sind Cluster von Bildausschnitten für Gesicht und rechts für Autos, Letztere in Form der Gradienten, dargestellt. (*Weber et al. 2000).

(Agarwal et al. 2004) verwenden den *normalisierten Kreuzkorrelationskoeffizient* (*Cross Correlation Coefficient – CCC*)

$$\rho_{g_1 g_2} = \frac{\sigma_{g_1 g_2}}{\sqrt{\sigma_{g_1}^2 \cdot \sigma_{g_2}^2}}.$$

Hierbei ist $\sigma_{g_1 g_2}$ die *Kovarianz* zweier Bilder und $\sigma^2_{g_i}$ die *Varianz* oder *mittlere quadratische Abweichung* der Grauwerte $g(r, c)$ eines Bildes vom *Mittelwert* μ. Mittels ρ wird die Ähnlichkeit zweier Bilder bewertet

Für zwei Bilder P_1 und P_2 gilt:
$\rho_{g_1 g_2} = 0$ P_1 und P_2 sind unkorreliert;
$\rho_{g_1 g_2} = 1$ P_1 und P_2 sind zu 100% positiv korreliert;
$\rho_{g_1 g_2} = -1$ P_1 und P_2 sind zu 100% negativ korreliert, d.h. P_2 ist eine inverse Kopie von P_1 .

Der Cross Correlation Coefficient wird in (Agarwal et al. 2004) für Bildausschnitte um Förstner Punkte für das Erkennen von Autos verwendet, die von der Seite aufgenommen wurden. Ähnliche Bildausschnitte der Trainingsbilder, z.B. mit Rädern, werden über Kreuzkorrelation bestimmt und als Cluster gesammelt. Für jedes Cluster werden die Richtungen und die Abstände zu anderen Clustern, d.h. die räumliche Konfiguration, erlernt (siehe Abbildung 2.2). Das Erkennen eines Autos wird auf das Problem der Entscheidung darüber abgebildet, ob ein Bildteil Teile eines Autos enthält. Für den Bildteil werden Förstner Punkte extrahiert und die Ausschnitte um die Punkte werden mit den Clustern über Kreuzkorrelation verglichen. Für die zugeordneten Ausschnitte wird die räumliche Konfiguration berechnet. Auf Grundlage von Korrelation und Konfiguration wird entschieden, wie wahrscheinlich es ist, dass ein Auto vorhanden ist. Um ein Auto in einem Bild zu lokalisieren, wird es in Teile aufgespalten welche ungefähr der Größe des Autos entsprechen. Für jedes Teil wird die Auftretenswahrscheinlichkeit berechnet. Ein Auto gilt als erkannt, wenn die Wahrscheinlichkeit einem lokalen Maximum entspricht und über einem gegebenen Schwellwert liegt. Für Erkennung in Bildern mit unterschiedlichen Auflösungen wird eine Bildpyramide benutzt. Dabei muss eine Hypothese ein räumliches Maximum sowie ein Maximum im Maßstabsraum bilden.

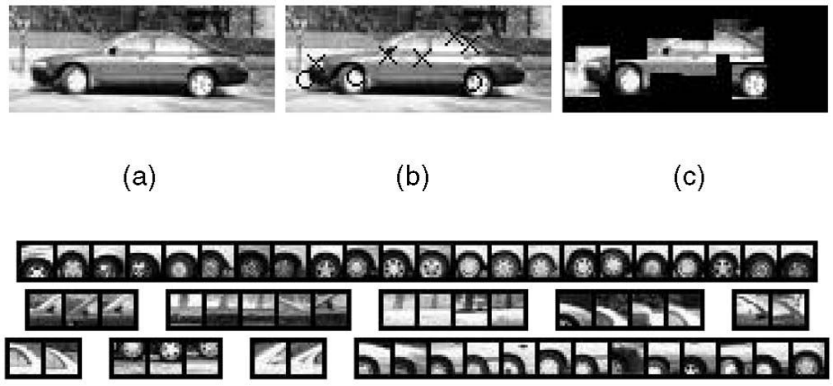


Abbildung 2.2: *(a) Ein Trainingsbild; (b) Förstner Punkte; (c) Bildausschnitte um Förstner Punkte; (d) Beispiele für einige Teile nach Cluster-Bildung für ähnliche Ausschnitte. (*Agarwal et al. 2004).

Implicit Shape Models (ISM) wurden von (Leibe und Schiele 2004b) vorgeschlagen. Auch hier wird für den Vergleich von Bildausschnitten Kreuzkorrelation verwendet. Zusätzlich

werden maßstabsinvariante Eigenschaften verwendet und die Hypothesen für Objekte werden über generalisierte *Hough-Transformation*[1] gebildet (BALLARD 1981). Zusätzlich werden Objekte durch Rückprojektion von erlernten Bildausschnitten in das Bild segmentiert (siehe Abbilung 2.3). Das Verfahren erlaubt die Suche von mehreren Autos, die sich auch gegenseitig verdecken können.

ISM sind dadurch einzigartig, dass sie die Objektkategorisierung und die top-down Segmentation gleichzeitig angehen. ISM vergeben zuerst über Kompatibilität mit lokalen Eigenschaften Wahrscheinlichkeiten und bestimmen mögliche Objektpositionen und -skalen. Für jede solche Hypothese gehen sie zurück ins Bild und stellen pro Pixel fest, ob das Objekt erkannt und vom Hintergrund segmentiert ist. Die Segmentierungsinformationen können auf einem höheren Niveau verwendet werden, um die Genauigkeit der Abfrage zu verbessern. ISM werden verwendet, um gute Resultate und beträchtliche Robustheit zu kombinieren. ISM liefern eine flexible Approximation der Zielkategorie. Weil jeder Bildausschnitt für die Objektmitte unabhängig von den anderen Ausschnitten stimmt, kann das resultierende Modell zwischen lokalen Teilen aus unterschiedlichen Trainingsausschnitten interpoliert. Infolgedessen werden die Objekte der Zielkategorie gut approximiert und es wird gewöhnlich eine hohe Erkennungsrate erzielt. Als Preis für die Flexibilität kann es jedoch z.T. unterschiedliche Modelle nicht auseinander halten und die Fehlerkennungsrate ist hoch.

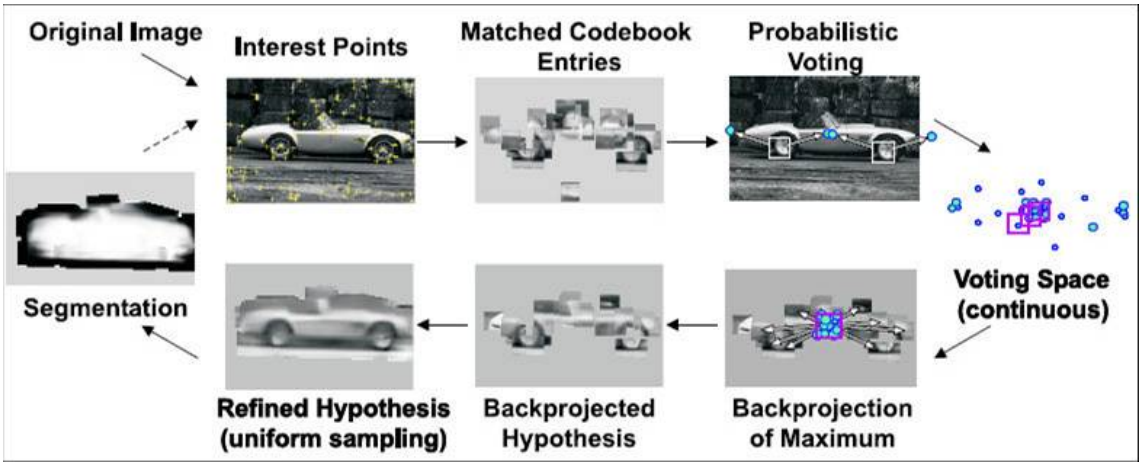


Abbildung 2.3: *Implicit Shape Models. (*LEIBE und SCHIELE 2004b)

2.2 Maßstabsräume

Die Maßstabsraum Theorie bildet einen Rahmen für eine multi-Maßstab Signal Repräsentation, der für Computer Vision sowie Bild- und Signalverarbeitung mit ergänzenden Gedanken aus der Physik und Biologie entwickelt wurde. Sie ist eine formale Theorie für den Umgang

[1] Die Hough-Transformation ist ein robustes globales Verfahren zur Erkennung von Geraden, Kreisen oder beliebigen anderen parametrisierbaren geometrischen Figuren in einem Binärbild. Das Verfahren wurde 1962 von Paul V. C. Hough unter dem Namen *Method and Means for Recognizing Complex Patterns* patentiert (http://de.wikipedia.org/wiki/Hough-Transformation).

mit Bildstrukturen auf verschiedenen Maßstaben. Ein Maßstabsraum wird häufig durch die Größe eines Glättungskernes parametrisiert. Der Glättungskern wird für den Übergang von feinen zum gröberen Maßstab benutzt. Der bedeutendste Maßstabsraum ist der *lineare* oder *gaußsche Maßstabsraum*. Ein Beispiel ist in der Abbildung 2.4 gezeigt.

$\sigma = 1$ $\sigma = 13$ $\sigma = 25$ $\sigma = 49$

Abbildung 2.4: *Gaußscher Maßstabsraum. σ in Pixel.*

Die Repräsentation einer Funktion im gaußschen Maßstabsraum erfolgt durch
$L_\sigma(x,y) = g_\sigma(x,y) * f(x,y)$ mit $g_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ und der Randbedingung $L_0 = f(x,y)$.

Die grundlegende Idee für die formale Behandlung des Maßstabs ist die Multi-Maßstabsrepräsentation mittels auf einem Parameter basierenden Familien von abgeleiteten Signalen. Dies führt zu einer systematischen Vereinfachung der Daten und der Elimination von Details, d.h. Information mit hoher Frequenz (Tiefpass). Der Informationsgehalt des Bildes wird im gröberen Maßstab reduziert, indem Punkte, Kanten, Linien und Flächen beseitigt werden. Damit werden sowohl Rauschen als auch bedeutungstragende Information eliminiert. Der Parameter σ beschreibt den Maßstab. Seine Bedeutung hängt von der Definition des Maßstabsraumes ab. Eine bedeutende Eigenschaft eines Maßstabsraums ist die Kausalität:

- Jedes Merkmal / Extremum im groben Maßstab (großes σ) soll einen nicht notwendigerweise eindeutigen Grund im feinen Maßstab (kleines σ) besitzen.

Ein anderer Typ von Maßstabsräumen ist der *morphologische Maßstabsraum*. Seine Grundoperationen sind *Erosion* und *Dilatation* bzw. *Opening* und *Closing*:

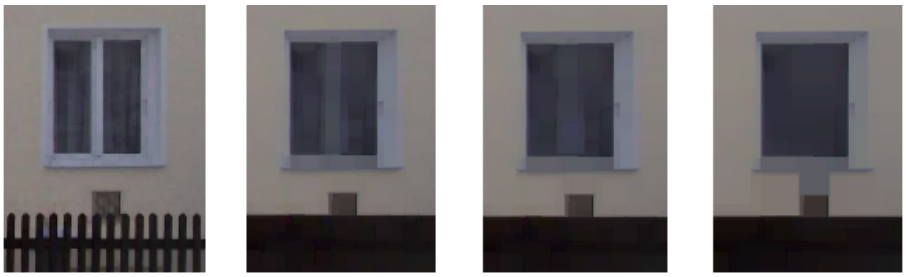
Erosion	$(f \ominus g)(\vec{x}) = \inf_{\vec{x}' \in G}(f(\vec{x} + \vec{x}') - g(\vec{x}'))$
Dilatation	$(f \oplus g)(\vec{x}) = \sup_{\vec{x}' \in G}(f(\vec{x} - \vec{x}') + g(\vec{x}'))$
Opening	$(f \circ g)(\vec{x}) = ((f \ominus g) \oplus g))(\vec{x})$
Closing	$(f \bullet g)(\vec{x}) = ((f \oplus g) \ominus g))(\vec{x})$

Die strukturierende Funktion $g(\vec{x})$ hat als Einzugsbereich die Region G. Wenn $g(\vec{x}) \geq 0$ und $g(\vec{0}) = 0$ dann werden konstante Funktionen $f(\vec{x}) = const$ nicht geändert. Ein kombinierter *Opening-Closing Maßstabsraum* kann wie folgt definiert werden:

$$F(\vec{x}, s) = \begin{cases} (f \bullet g_s)(\vec{x}) & wenn \;\; s > 0\,, \\ f(\vec{x}) & wenn \;\; s = 0\,, \\ (f \circ g_{-s})(\vec{x}) & wenn \;\; s < 0 \end{cases}$$

σ wird durch s ersetzt, das sowohl positiv als auch negativ sein kann und eine Scheibe mit Radius oder Quadrat mit Seitenlänge s beschreibt, die innerhalb den Wert 0 und außerhalb $-\infty$ besitzt.

Der morphologische Maßstabsraum erfüllt die Kausalitätseigenschaft. Während der lineare Maßstabsraum eine Bildfunktion mehr oder weniger kontinuierlich glättet, eliminiert der morphologische Maßstabsraum bezüglich ihrer geometrischen Ausdehnung zu kleine Bereiche (siehe Abbildung 2.5).



$s = 1$ $s = 2$ $s = 5$ $s = 10$

Abbildung 2.5: *Morphologischer Maßstabsraum.*

2.3 Markov Chain Monte Carlo (MCMC)

Die Markovschen Zufallsprozesse wurden nach dem russischen Mathematiker A.A. Markov (1856-1922) benannt, der zum ersten Mal Wahrscheinlichkeitsabhängigkeiten von Zufallsvariablen untersucht hat (Markov 1906). Er hat eine Theorie über *die Dynamik der Wahrscheinlichkeiten* entwickelt. Diese Theorie wurde zur Basis der allgemeinen Theorie der zufälligen Prozesse sowie für angewandte Wissenschaften, wie z.B. die Theorie der diffusen Prozesse, die Theorie der Zuverlässigkeit und die Massenbedingungstheorie. Heute wird die Theorie der Markovprozesse und ihre Ergänzungen in den verschiedensten Bereichen angewendet.

Infolge ihrer Einfachheit und der Übersichtlichkeit der mathematischen Grundlagen sowie der hohen Zuverlässigkeit und der Genauigkeit der abgeleiteten Entscheidungen, eignen sich Markovprozesse besonders für optimale Entscheidungen.

Markovketten-Monte-Carlo (*Markov Chain Monte Carlo – MCMC*) Methoden sind eine Kategorie von Algorithmen für die Abtastung (Sampling) von Wahrscheinlichkeitsverteilungen, die auf der Konstruktion einer Markovkette basieren, welche die gewünschte Verteilung als Gleichgewichtsverteilung besitzt.[1]

Definition:

Eine Folge unabhängiger Zufallsvariablen $\{X_n\}_{n\geq 0}$ ist eine Markovkette erster Ordnung mit diskreter Zeit, wenn gilt

$$P(X_{n+1} = i_{n+1}|X_n = i_n, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = i_{n+1}|X_n = i_n)$$

[1] Siehe z.B. http://de.wikipedia.org/wiki/Markow-Kette

Demgemäß hängt im einfachsten Fall der ersten Ordnung, im Gegensatz zu den Markovketten der höheren Ordnungen, die bedingte Verteilung des folgenden Zustandes der Markovkette nur vom gegenwärtigen Zustand und nicht von allen vorhergehenden Zuständen ab.

Der Wertebereich der Zufallsvariablen $\{X_n\}$ wird Zustandsraum der Kette und n die Anzahl der Schritte genannt. Eine Matrix $P(n)$, mit $P_{ij}(n) \equiv P(X_{n+1} = j \mid X_n = i)$ heißt *Matrix der Übergangswahrscheinlichkeiten* für den Schritt n. Der Vektor $\overrightarrow{P} = (p_1, p_2, \ldots)^\top$, mit $p_i \equiv P(X_0 = i)$ stellt die *Anfangsverteilung* der Markovkette dar. Die Matrix der Übergangswahrscheinlichkeiten ist eine stochastische Matrix. d.h. es gilt:

$$0 \leq P_{ij} \leq 1; \sum_{i=1}^{k} P_{ij}(n) = 1, (i = 1, 2, \ldots, k)$$

Eine Markovkette ist homogen, wenn die *Matrix der Übergangswahrscheinlichkeiten* von der Iterationsnummer unabhängig ist:

$$P_{ij}(n) = P_{ij}, \forall_n \in \mathbb{N}$$

Die Eigenschaften der homogenen Markovketten werden durch den Vektor der Anfangswahrscheinlichkeiten und durch die Matrix der Übergangswahrscheinlichkeiten vollständig bestimmt.

Eine praktisch bedeutende Anwendung der Markovketten ist die Suche nach dem optimalen Zustand eines Systems. Hierauf zielt auch diese Arbeit ab und so wird diese hier näher beschrieben. Es wird angenommen, dass ein System ein globales Kriterium K_i hat, welches den aktuellen Zustand des Systems bei der Iteration i beschreibt und es sei $\acute{P}_i$ der aktuelle Zustand der Markovkette bei der i. Iteration. Wird angenommen, dass je höher der Wert für K_i ist, desto besser der Zustand des Systems ist, so ist folgendes Vorgehen sinnvoll:

$$P_i = \begin{cases} \acute{P}_i & \text{wenn} \quad \delta \geq 0, \\ P_{i-1} & \text{wenn} \quad \delta < 0 \end{cases}$$

wobei $\delta = K_i - K_{i-1}$

D.h., es werden nur solche neuen Zustände akzeptiert, die einen höheren Wert des Entscheidungskriteriums K als bei der vorherigen Iteration aufweisen.

Diese Bedingung zwingt das System, sich schrittweise in einen Zustand mit maximalem Wert für K zu bewegen. Abbildung 2.6 zeigt ein Beispiel.

Das System hängt nur von einem Parameter x ab. Der Prozess startet mit $x = x_0$. Bei jeder Iteration wird zu x ein zufälliger kleiner Wert Δ addiert. Bei der Iteration i befindet sich das System im Zustand $P(i)$ und $x = x_i$. Ist Δ positiv, so gilt $K_{i+1} > k_i$ und der Schritt wird akzeptiert. Anderenfalls wird der Schritt abgelehnt. Nach k Iterationen erreicht das System den Zustand mit dem maximal realisierbaren Wert $K = K_m$. Damit ist das Ziel des Markovprozesses erreicht.

Die gleiche Methode kann für mehrdimensionale Systeme verwendet werden, indem bei jeder Iteration ein Parameter zufällig gewählt und danach zufällig (ein wenig) verändert wird. Nach der Berechnung von K wird der Entscheidungsprozess wiederholt.

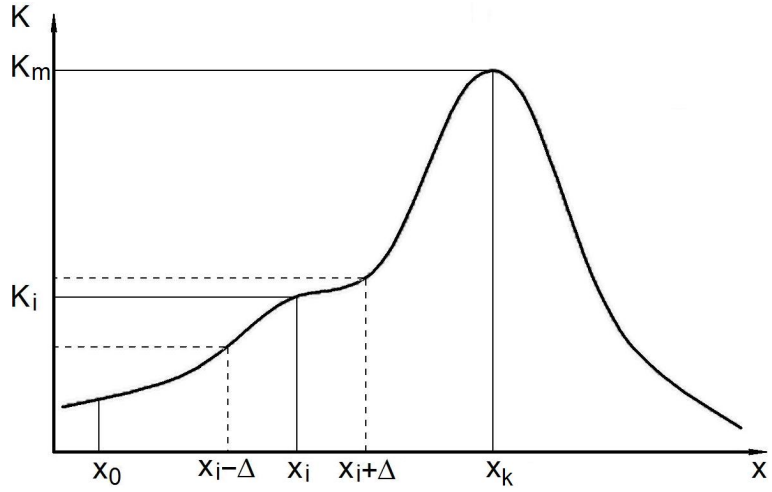


Abbildung 2.6: *Der Entscheidungsprozess für eine eindimensionale Markovkette.*

Im allgemeinen Fall ist eine Situation wie in Abbildung 2.7 dargestellt sehr häufig. Das System befindet sich bei der Iteration i in einem lokalen Maximum. Falls Δ nicht groß genug ist, kann dieser Zustand nicht verlassen werden.

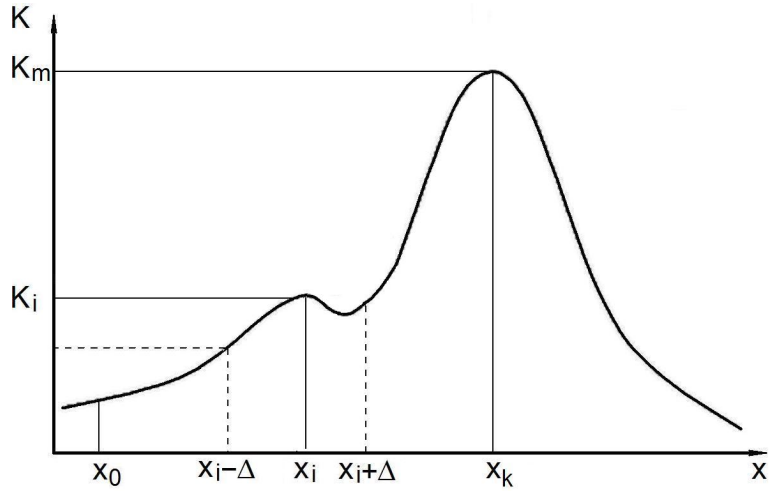


Abbildung 2.7: *Metropolis-Hastings Sampling*.

Um diesen Zustand zu vermeiden, wurden verschiedene Methoden entwickelt. In Rahmen dieser Arbeit wurden zwei von ihnen verwendet: *Simulated Annealing* (Kirkpatrick et al. 1983) und *Metropolis-Hastings sampling* (Metropolis et al. 1953) und (Hastings 1970).

Die Grundidee von *Simulated Annealing (simulierte Abkühlung)* ist wie folgt: Wenn, wie oben beschrieben, bei jeder Iteration ein Parameter des Systems P_j zufällig um einen Wert Δ_j verändert wird, dann hängt die maximale Größe dieses Wertes exponentiell von der Iterationsanzahl i ab:

$\Delta_j(i) = Ae^{-Bi}$,

A ist hierbei das maximale Inkrement Δ_j zu Anfang des Markovprozesses. Der Parameter B steuert die Geschwindigkeit der Abkühlung und wird meist empirisch bestimmt. Je höher

die Iterationszahl ist, desto kleiner werden die Änderungen der Parameter.

Beim *Metropolis-Hastings Sampling* werden, um ein lokales Maximum wie in Abbildung 2.7 zu verlassen, auch (einige) negative Fälle akzeptiert.

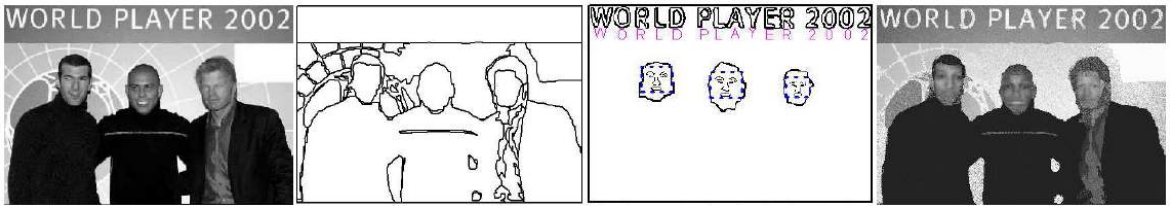
$$P_i = \begin{cases} \acute{P}_i & \text{wenn} \quad \delta \geq 0, \\ \acute{P}_i \text{ mit } Q(\delta), & \text{wenn} \quad \delta < 0 \end{cases}$$

$Q(\delta)$ ist die Wahrscheinlichkeit, mit der ein schlechterer Zustand des Systems akzeptiert wird. Sie ist klein genug und indirekt proportional zu δ zu wählen. Diese Bedingung erlaubt es, mit gewisser Wahrscheinlichkeit schlechtere Zustände zu akzeptieren und somit lokale Maxima zu verlassen. *Metropolis-Hastings Sampling* erhöht damit die Robustheit eines Markovprozesses deutlich und wird deswegen häufig als Teil von stochastischen Methoden verwendet.

Eine bedeutende Erweiterung für MCMC wurde von (Green 1995) in Form von *Reversible Jump MCMC (RJMCMC)* vorgeschlagen. RJMCMC ermöglicht es durch geeignete Normierung bezuglich der Metrik der Modellparameter sowie über die Einbeziehung der Vorschlagswahrscheinlichkeit, Modelle unterschiedlicher Komplexität, d.h. Parameterzahl, mittels einer Markovkette zu sampeln. Dies erlaubt es, beim statistische Sampling neue Objekte einzufügen bzw. vorhandene (inkorrekte) zu löschen, Aufgaben, die bei der Analyse eines Bildes praktisch zwingend auftreten.

2.4 Generative statistische Modelle

Die grundlegende Idee von generativen (statistischen) Modellen besteht darin, dass beschrieben wird, wie das Bild aus der Repräsentation der Szene generiert wird. Bei (Tu et al. 2005) besteht die Repräsentation der Szene aus Buchstaben, Gesichtern sowie homogenen und texturierten Regionen. Diese werden so eingefügt und ihre Parameter so statistisch modifiziert, dass das generiertes Bildes dem gegebenen Bild möglichst ähnlich wird (siehe Abbildung 2.8).

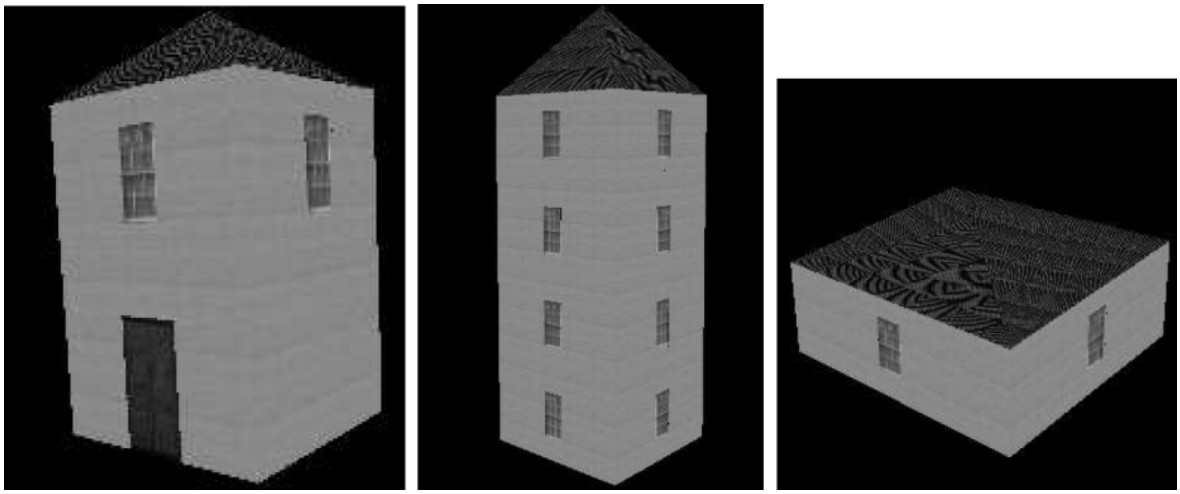


a) b) c) d)

Abbildung 2.8: *Ergebnis von Segmentierung und Erkennung. a) – Gegebenes Bild; b) – Segmentierung; c) – Objekterkennung; d) – Synthetisiertes Bild (*Tu et al. 2005)

Die Repräsentation der Szene von (Dick et al. 2004) ist ein Gebäudemodell aus Teilen, wie z.B. Fassaden und Fenstern. Abbildung 2.9 zeigt ein Beispiel für Modelle, die aus Wänden und Fenstern bestehen. Für diese werden Parameter, wie z.B. Breite und Höhe der Fenster

sowie die Helligkeit der Fassade, statistisch mittels RJMCMC (siehe Abschnitt 2.3) verändert, um ein Aussehen herzustellen, das dem gegebenen Bildern ähnelt. Das Ergebnis für die Simulation einer städtischen Szene, die vollständig mittels des Priors dieses Verfahrens erzeugt wurde, ist in Abbildung 3.2 dargestellt.



a) b) c)

Abbildung 2.9: *Hütte, Turm und Bungalow als Startpunkt für MCMC Algorithmus. (*Dick et al. 2004)

Generative Modelle sind aus verschiedenen Gründen im Kontext der Objektextraktion besonders interessant. Z.B. können sie schrittweise gelernt werden und sie können in allgemeinen Fall sogar mit fehlenden Daten arbeiten. Sie erlauben die modulare Konstruktion zusammengesetzter Lösungen für komplizierte Probleme und eignen sich zur hierarchischen Klassifizierung. Insbesonders kann Vorwissen / Prior Information einfach berücksichtigt werden. Generative Modelle zeigen beträchtliche Robustheit in Bezug auf partielle Verdeckungen und starke Veränderungen des Objektaussehens innerhalb der Objektkategorie. Der Preis für diese Robustheit ist allerdings, dass generative Modelle u.U. eine größere Anzahl von falschen Hypothesen erzeugen. Das ist besonders für Objektklassen der Fall, die eine hohe visuelle Ähnlichkeit besitzen, wie z.B. Fenster und Türen oder Pferde und Kühe.

2.5 Modellauswahl

Das zentrale Ziel dieser Arbeit ist eine Beschreibung von Fassaden. Diese Beschreibung ist im Allgemeinen nicht eindeutig. D.h., für eine Fassade existiert eine Vielzahl von Modellen. Deswegen ist es nötig, ein möglichst plausibles Modell aus mehreren möglichen auszuwählen.

Wenn für einen Datensatz eine Menge von Modellen existiert, so hängt die Entscheidung oft von der Modellqualität und der Modellkomplexität ab. Modellqualität bedeutet hierbei die Anpassung eines Modells an die Daten. Die beste Anpassung ist praktisch immer für

das Modell mit der höchsten Parameteranzahl möglich. Im extremen Fall besteht das Modell selbst aus den gegebenen Daten. Deshalb ist es oft sinnvoll, einfachere Modelle mit einer kleineren Parameteranzahl bzw. Modellkomplexität für eine bessere Verallgemeinerung und damit Prognose zu bevorzugen. Ein Prinzip für die Auswahl eines Modells, das nicht nur gut zu den Daten passt, sondern auch einfach ist, heißt *Ockham's Razor principle (Ockhams Rasiermesser)*[1]. Meist wird die Modellkomplexität in der Form eines probabilistischen Informationskriteriums beschrieben.

Ein Informationskriterium dient zur Auswahl eines Modells in der angewandten Statistik. Dabei geht die Anpassungsgüte des geschätzten Modells an die vorliegenden empirischen Daten (Stichprobe) und die Komplexität des Modells, gemessen in Form der Anzahl der Parameter, in die Beurteilung ein. Die Anzahl der Parameter wird dabei "strafend" berücksichtigt, da sonst umfassende Modelle mit vielen Parametern bevorzugt werden.

Praktisch alle heute verwendeten Informationskriterien sind ähnlich. Sie liegen in zwei verschiedenen Formulierungen vor. Entweder ist das Maß für die Anpassungsgüte die maximale Likelihood oder die minimale Varianz der Residuen. Hieraus ergeben sich unterschiedliche Vorgehensweisen. Beim Ersten ist das Modell "am besten", bei dem das Informationskriterium den höchsten Wert hat. Die "strafende" Anzahl der Parameter muss abgezogen werden. Beim zweiten ist das Modell mit dem niedrigsten Wert des Informationskriteriums am besten, die Anzahl der Parameter muss "strafend " addiert werden. Eine Analyse der Beziehungen solche Ansätze, sowie ihre Eigenschaften, Stärken und Schwächen werden in (Schindler und Dellaert 2006) beschrieben.

Das älteste Kriterium wurde 1973 von Akaike als "an information criterion" vorgeschlagen (Akaike 1973). Heutzutage ist es als *Akaikes Informationskriterium* (**Akaike's Information Criterion, AIC**) bekannt. Es lässt sich mit der logarithmierten Likelihood-Funktion L wie folgt darstellen:

$$AIC = 2k - 2\ln(L)$$

wobei k die Anzahl der Parameter im statistischen Modell darstellt.

Wenn angenommen werden kann, dass die Modellfehler normal und unabhängig verteilt sind, dann lässt sich AIC mit n der Anzahl der Beobachtungen und RSS (*residual sum of squares*) der Summe der quadratischen Residuen formulieren als:

$$AIC = 2k + n\ln\left(\frac{\mathrm{RSS}}{n}\right)$$

Die Erhöhung der Parameterzahl k bewirkt in den meisten Fällen die Verbesserung der Anpassungsgüte. AIC belohnt daher nicht nur die Anpassungsgüte, sondern bestraft gleichzeitig die Anzahl der geschätzten Parameter. Diese verhindert *Overfitting*. Das bevorzugte Modell

[1] *Ockhams Rasiermesser* beschreibt das Sparsamkeitsprinzip in der Wissenschaft. Es besagt, dass von mehreren Theorien, die den gleichen Sachverhalt erklären, die einfachste zu bevorzugen ist. (http://de.wikipedia.org/wiki/Ockhams-Rasiermesser)

ist jenes mit dem niedrigsten AIC Wert. Die AIC Methodik versucht ein Modell zu finden, welches am besten zu den Daten mit einer minimalen Parameteranzahl passt.

Das **Bayesian Information Criterion (BIC)** ist ebenfalls ein statistisches Kriterium für die Modellauswahl. Es ist auch als *Schwarz criterion (SIC)* (Schwarz 1978) bekannt. Das BIC ist ein asymptotisches Ergebnis unter der Annahme, dass die Datenverteilung zu einer exponentiellen Familie gehört. Die mathematische Definition für BIC lautet:

$$\mathrm{BIC} = -2 \cdot \ln(L) + k \ln(n)$$

mit n – Anzahl Beobachtungen
k – Anzahl Parameter
L – *Likelihood*.

Wenn die Modellfehler normalverteilt sind, dann gilt:

$$\mathrm{BIC} = n \ln\left(\frac{\mathrm{RSS}}{n}\right) + k \ln(n).$$

BIC bestraft die Parameteranzahl bei der oft vorliegenden größeren Zahl von Beobachtungen stärker als AIC.

Minimum Description Length (MDL) wurde 1978 von Jorma Rissanen (Rissanen 1978) zur Beschreibung von Regelmäßigkeiten in gemessenen Daten eingeführt: Je stärker die Daten komprimiert werden können, desto größer ist der Ordnungsanteil im Signal.

Das *Minimum Description Length* (MDL) Prinzip besteht darin, dass die Länge der Beschreibung des Modells plus die Länge der Beschreibung der Aussagen minimal sein soll:

$$\mathrm{MDL} = -\log_2 \prod_{n=1}^{n} P(d_i|\pi) + \frac{K}{2} \log_2(n).$$

MDL gibt hierbei die zur Kodierung des Modells (π, K) minimal benötigte Anzahl von Bits an. Gesucht ist ein Modell (π, K), das mit der geringsten Komplexität K und der größten Datenwahrscheinlichkeit $\prod_{n=1}^{n} P(d_i|\pi)$ die beobachteten Daten d_i beschreibt.

Kapitel 3

Bisherige Arbeiten

Das in vorgegebener Arbeit vorgestellte Verfahren zur 3D Fassadenmodellierung ist in erster Linie von (Dick et al. 2000, Dick et al. 2001, Dick et al. 2002, Dick et al. 2004) inspiriert.

Die Eingabedaten von (Dick et al. 2004) sind zwei bis sechs Bilder einer architektonischen Szene. Hieraus wird ein 3D Modell der Szene produziert, das aus verschiedenen architektonischen Komponenten wie z.B. Wände, Fenster, Säulen und Türen besteht. Ein Beispiel ist in Abbildung: 3.1 dargestellt.

Abbildung 3.1: *Das 3D Modell aus Fenstern, Säulen, Dach und Türe wird aus mehreren Bildern generiert, in diesem Fall die Downing College Library, Cambridge (*Dick et al. 2004).

Der Ansatz von (Dick et al. 2004) ist statistisch generativ (siehe Abschnitt 2.4). Für die Kombination von Information über die Struktur und die Identität wird ein probabilistisches *Bayesian Framework* benutzt. Eine prior Verteilung für die Parameter des Modells beschreibt die Annahmen über die Struktur des Modells, wie z.B. die (typische) Breite und Höhe von Fenstern oder Türen aber auch die Anordnung der Fenster auf der Fassade. Die Gültigkeit der Annahmen kann durch Simulation überprüft werden. Hierbei wird die Frage beantwor-

tet, ob die durch Sampling der prior Verteilung generierten Gebäude plausibel sind (siehe Abbildung 3.2).

Für gegebene Bilder werden die *maximum a posteriori* (MAP) Modell Parameter auf der Grundlage der prior Verteilung und der Likeihood bestimmt. Die Likelihood beschreibt hierbei die Übereinstimmung von aus der Repräsentation der Szene generierten und gegebenen Bilder. Für die Bestimmung der MAP Schätzung wird RJMCMC (siehe Abschnitt 2.3) verwendet. Damit können neue Objekte eingefügt und (inkorrekte) vorhandene Objekte gelöscht werden. *Modellauswahl* (siehe Abschnitt 2.5) wird benutzt, um zwischen ähnlichen Modellen, wie z.B rechteckige und oben abgerundete Fenster, in Abwägung der Parameterzahl und der Anpassung an das Bild (*Likelihood*), auszuwählen.

Abbildung 3.2: *Gebäude im klassischen Stil, die mittels Priors (siehe auch Abbildung 2.9) generiert wurden. (*Dick et al. 2004).

Die architektonische Szene wird grundlegend als eine Menge von Wänden modelliert. Jede Wand ist als ein Rechteck beschrieben und enthält eine Reihe von volumetrischen Primitiven, insbesondere Türen und Fenster. Mittels einer Reihe von globalen Parametern Θ_G, werden die Eigenschaften, die alle Wände betreffen, wie z.B. den Stil des Gebäudes, beschrieben. Letzterer enthält genau einen einzigen Stilparameter, der entweder auf klassischen oder gotischen Stil gesetzt wird.

Das Modell wurde für terrestrische Bilder entworfen. Daher ist es zu einem entsprechenden *level of detail* (LOD) konsistent. Objekte für einen feineren LOD, wie einzelne Ziegel, Türgriffe oder feine Ornamente sind nicht modelliert. Die Modellierung der Dächer wird wenig Aufmerksamkeit gewidmet, weil die meisten terrestrischen Bilder sehr wenig Information über die Dachkonstruktion enthalten.

(Alegre und Dallaert 2004) nutzen als Eingabedaten auf die Fassadenebenen entzerrte, vertikal ausgerichtete hoch aufgelöste Bilder von Fassaden mit sehr regelmäßigen Strukturen und wenig Störungen. Die Bilder werden analysiert, indem sie rekursiv in reguläre Teilstücke einer Bestandteilhierarchie zerlegt werden (siehe Abbildung 3.3). Die Zerlegung wird mittels stochastisch kontextfreier Grammatiken beschreiben und es wird wie in (Dick et al. 2004) RJMCMC verwendet, um die optimale Zerlegung in Fassadenteile zu bestimmen. Hiermit wird formale Korrektheit mit über die Produktionen der Grammatik steuerbare Flexibilität gekoppelt.

In (Van Gool et al. 2007, Müller et al. 2007) wird ein Verfahren für die (halb-) automatische

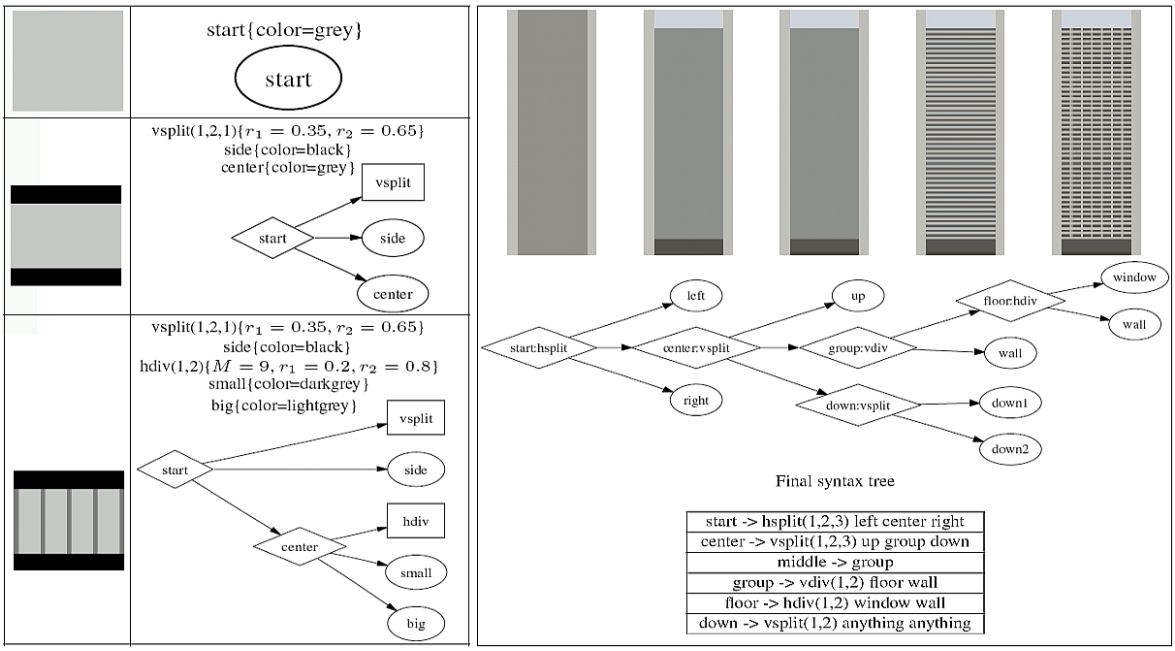
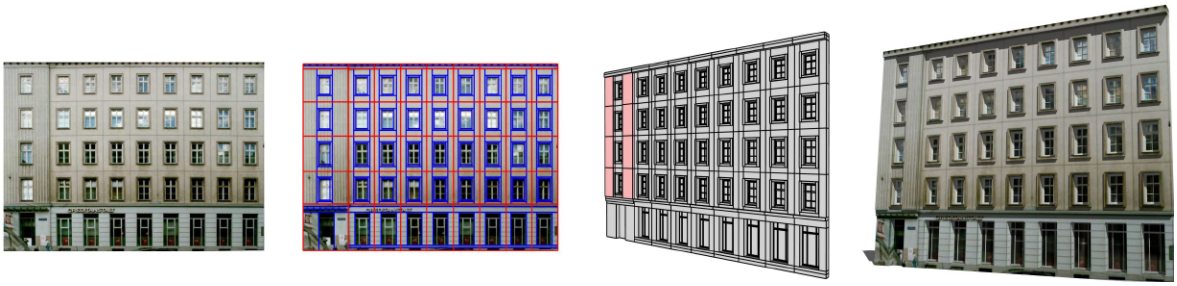


Abbildung 3.3: *Modellierung der Bestandteilshierarchie für eine Fassade mittels einer stochastischen kontextfreien Grammatik (*Alegre und Dallaert 2004).

Generierung von 3D Modellen von Fassaden aus Einzelbildern basierend auf Grammatiken vorgestellt. Hierbei wird die 3D Orientierung mittels Fluchtpunkten aus dem Bild bestimmt unter der Annahme, dass Fassaden zumeist eine rechteckige Form haben und senkrecht ausgerichtet sind. Falls die Fassade eine ausgeprägte 3D Struktur besitzt und mit starker Perspektive aufgenommen wurde, kann die Orientierung automatisch bestimmt und damit die Orthogonalisierung der Fassadenebene erfolgen. Wenn die Fassade sich praktisch eben darstellt, muss die Orthogonalisierung manuell durchgeführt werden.

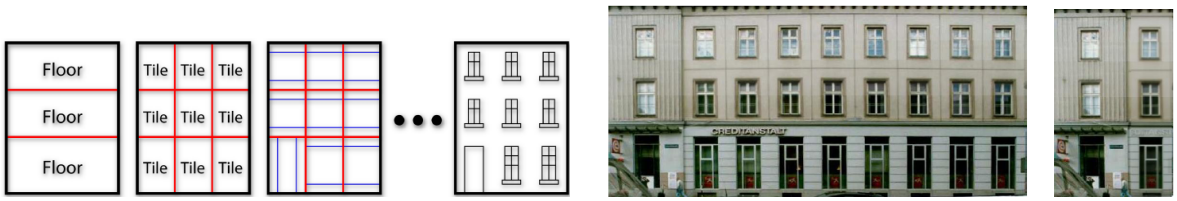
Abbildung 3.4 zeigt ein Beispiel, beginnend mit dem Eingabebild auf der linken Seite bis zum 3D Ergebnis auf der rechten Seite. Das Verfahren besteht aus einer Pipeline mit vier Stufen. Die Pipeline transformiert das Bild in ein texturiertes 3D Modell mit einer semantischen Struktur in Form eines *shape tree*. Es wird eine hierarchische Unterteilung verwendet. Ein Beispiel ist in Abbildung 3.5 dargestellt.

Ein Ansatz für die Detektion von regelmäßigen Strukturen auf orthogonalisierten Fassadenbildern und eine minimale Beschreibung der Struktur der Fassadenelemente in Form von achsenparallelen Basiselementen wird in (Wenzel et al. 2007) vorgeschlagen. Es wird angenommen, dass auf einer Fassade oftmals ähnliche Fenster liegen, die in regulären Strukturen angeordnet sind. Dafür werden Basiselemente, wie z.B. ein Fenster oder eine Fenstergruppe, gesucht und durch verschiedene Translationsvektoren ein Fassadenmodell konstruiert (siehe Abbildung 3.6). Für die Bestimmung der optimalen Beschreibung der Struktur der Fassadenelemente in Form von achsenparallelen Basiselementen wird MDL verwendet (siehe Abschnitt 2.5).



a b c d

Abbildung 3.4: *Mittels der grammatikbasierten Rekonstruktion können Fassadentexturen in ein semantisches 3D Modell hoher Qualität konvertiert werden. a) – Orthogonalisiertes Fassadenbild als Eingabe; b) – Automatisch unterteilte Fassade; c) – Polygonales 3D Modell als Ergebnis; d) – Visualisierung mit Schatten und Spiegelungen (*Van Gool et al. 2007).



a b

Abbildung 3.5: *a) – Hierarchische Unterteilung der Fassade. b) – Links: die Fassade aus Abbildung 3.4 nach der Entfernung der senkrechten Symmetrie. Rechts: Die Beseitigung der horizontalen Symmetrie resultiert in einer nicht reduzierbaren Fassade (*Van Gool et al. 2007).

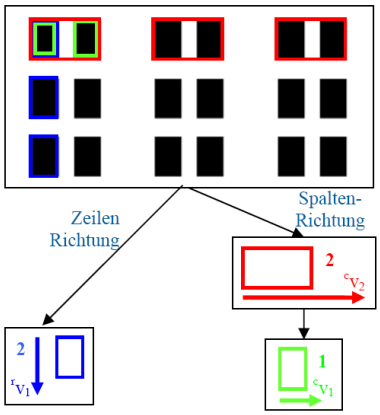


Abbildung 3.6: *Eine typische Fassade, die in vertikaler Richtung durch eine einfache Wiederholung charakterisiert ist. Die horizontale Struktur lässt sich durch eine einfache Translation der Elemente der Doppelfenster und deren Beziehung wiederum durch eine weitere zweifache Translation beschreiben. Die kompakte Bildbeschreibung besteht in horizontaler Richtung somit aus einer Hierarchie (K = 2) von Basiselementen mit dem Betrag der Translation und der Anzahl der Wiederholungen. (*Wenzel et al. 2007)

Ein Ansatz für die Bestimmung der 3D Struktur von Fassaden inklusive der Fensterlage relativ zur Fassadenebene wurde in (Werner und Zisserman 2002) vorgeschlagen. Die Bestimmung der Fensterlage mittels *Plane Sweeping* ist in der Abbildung 3.7 dargestellt.

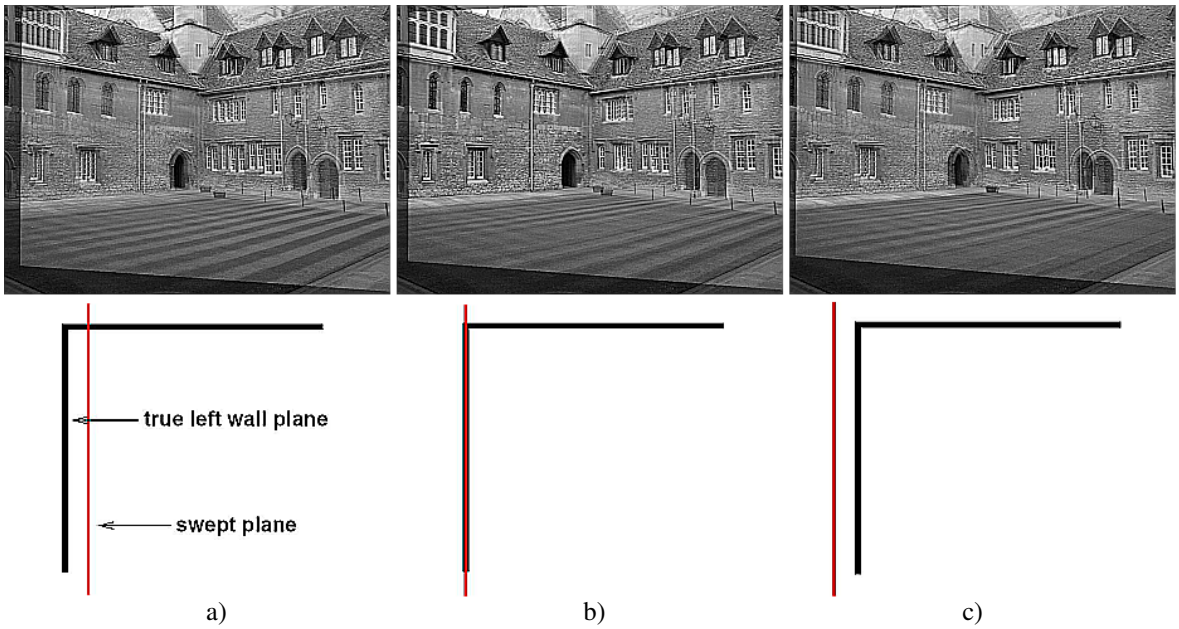


Abbildung 3.7: *Eine virtuelle Ebene, die parallel zur Fassadenebene ist, wird senkrecht zur linken Fassadenebene verschoben – Plane Sweep (*Werner und Zisserman 2002).

Abbildung 3.7 zeigt oben die Überlagerung zweier Bilder auf Grundlage ebener Homographie. Bei der mittleren Verschiebung (b) stimmen virtuelle und reale Ebenen für die Fassade auf der linken Seite überein und damit auch die überlagerten Bilder. Die Abbildung 3.8 zeigt, wie dieses Verfahren für die Bestimmung der Fensterlage verwendet werden kann, indem die Ähnlichkeit der überlagerten Bilder in Abhändigkeit von der Translation senkrecht zur Ebene mittels eines Scores bewertet wird (a).

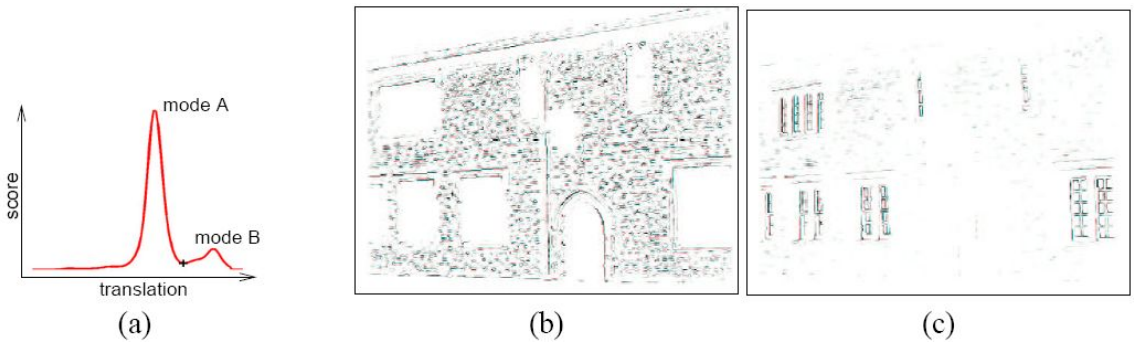


Abbildung 3.8: *(a) Bewertung der Ähnlichkeit der drei gegebenen Ansichten bei Translation der virtuelle Ebene senkrecht zur Fassade. Mode A entspricht der Fassadenebene und Mode B der Fensterebene. Die Ähnlichkeiten für Punkte auf der Fassade sind in (b) für den Mode A und in (c) für den Mode B dargestellt (*Werner und Zisserman 2002).

Kapitel 4

Überblick und vorbereitende Maßnahmen

4.1 Fassadenmodell

Jedes Fassadenmodell besteht aus einer Sammlung von Parametern, mit denen die Lage der Objekte, d.h. z.B. Fenster, auf der Fassade eindeutig beschrieben wird. Ein Parameter kann z.B. eine Länge oder eine Anzahl von Elementen sein.

Fassaden bestehen aus einer begrenzten Anzahl von Elementen, wie z.B. Fenster, Türen und Regenfallrohre. Die Fassadenelemente können in eine *hierarchische* Struktur mit geometrischen Relationen untereinander organisiert werden. Das ist die Basis für die Konstruktion von generativen statistischen Modellen. Ein Beispiel hierfür ist in Abbildung 4.1 dargestellt.

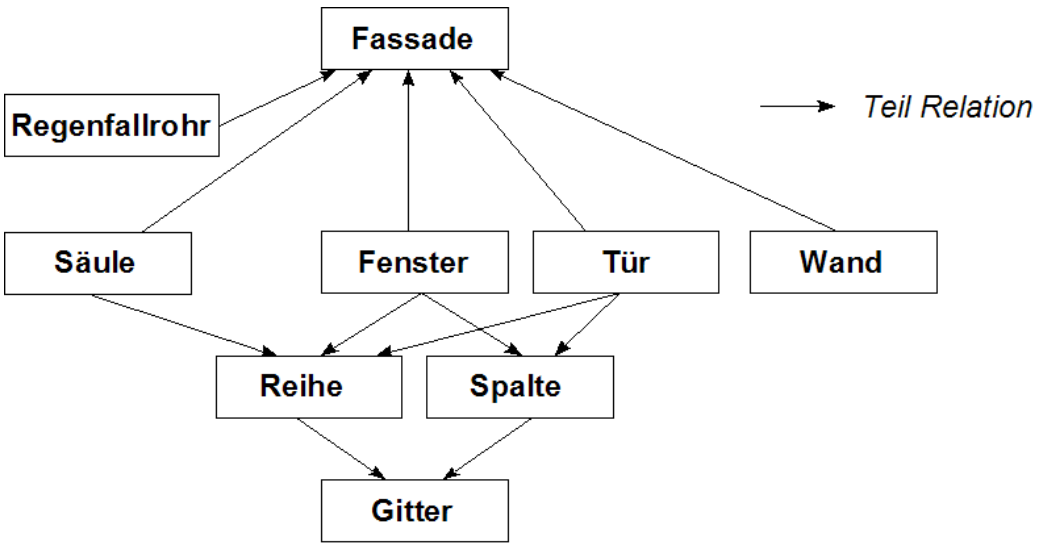


Abbildung 4.1: *Generatives Fassadenmodell*

In Abbildung 4.1 wird definiert, dass z.B. Säulen nur in Reihen organisiert sein können,

Fenster aber in Reihen oder Spalten und sogar als Gitter. Türen können ein Bestandteil von einer Fensterreihe oder /-spalte sein.

Im Rahmen dieser Arbeit wurden als Fassadenobjekte nur Fenster untersucht, da sie auf fast allen Fassaden existieren. Es wird angenommen, dass Fenster in der Regel auf der Fassadenebene oder in der Nähe dieser Ebene liegen. Fenster sind sehr häufig aus Standardelementen aufgebaut, die für eine bestimmte zeitliche Epoche, wie für moderne Gebäude, charakteristisch sind. Fenster bestehen nahezu immer aus einer oder mehreren Glasscheiben und haben häufig eine rechteckige Form (Van Gool et al. 2007). Die Größen einer Fensterart liegen in bestimmten Intervallen (Dick et al. 2000).

Andere Fassadenobjekte, wie z.B. Balkone, Bogen, Türen, Säulen und Regenfallrohre, haben meist nicht so viele gemeinsame Eigenschaften wie Fenster. Deshalb sind für sie aufwändigere Verfahren für die Erkennung und Modellierung notwendig.

Normalerweise liegen die Fenster auf den Fassaden in regelmäßigen Strukturen, die eine Hierarchie bilden. Sie sind oft in vertikale, horizontale oder in beide Richtungen (Gitter) geordnet. Im Rahmen der vorgelegten Arbeit werden *Einzelfenstermodell*, *Spaltenmodell* und *Zeilenmodell* definiert.

Das *Einzelfenstermodell* besteht aus unabhängigen Fenstern. Weil für die Bestimmung der Lage und Größe eines rechteckigen horizontal ausgerichteten Fensters 4 Parameter notwendig sind, ist die gesamte Anzahl der Parameter für dieses Modell $Np_e = N * 4$, mit N – Anzahl der Objekte im Modell.

Das *Spalten- / Zeilenmodell* besteht aus *gleichen* Objekten, die in Spalten (Zeilen) mit *gleichem* Abstand zwischen den Objekten innerhalb einer Spalte (Zeile) organisiert sind. Die Abstände zwischen Spalten (Zeilen) sind voneinander *unabhängig*. Für die Definition einer Spalte (Zeile) sind 6 Parameter notwendig: 4 für die Bestimmung des Fensters, 1 für den Abstand zwischen den Fenstern und 1 für die Anzahl der Fenster in der Spalte (Zeile). Die gesamte Anzahl der Parameter ist daher: $Np_r = N_s * 6$, mit N_s – Anzahl der Spalten (Zeilen) im Modell. Die praktischen Untersuchungen haben gezeigt, dass viele Fassaden unter Verwendung dieser Modelle adäquat beschrieben werden können.

Es ist möglich, eine Vielzahl anderer Modelle zu definieren, wie z.B. ein Spaltenmodell, bei dem alle Objekte einer Spalte gleich, aber die Abstände zwischen den Objekten unterschiedlich sind. Solche Modelle besitzen eine höhere Anzahl an Parametern als das definierte *Spaltenmodell*, aber weniger als das *Einzelfenstermodell*. In dieser Arbeit wurde hierauf verzichtet, weil der Schwerpunkt auf der grundlegenden Machbarkeit lag. Darüber hinaus ist zu beachten, dass jedes weitere Modell die Kombinatorik bei der Selektion verschlechtert und dass die Modellauswahl kompliziert werden kann, weil die Unterschiede zwischen den Modellen z.T. sehr gering sind.

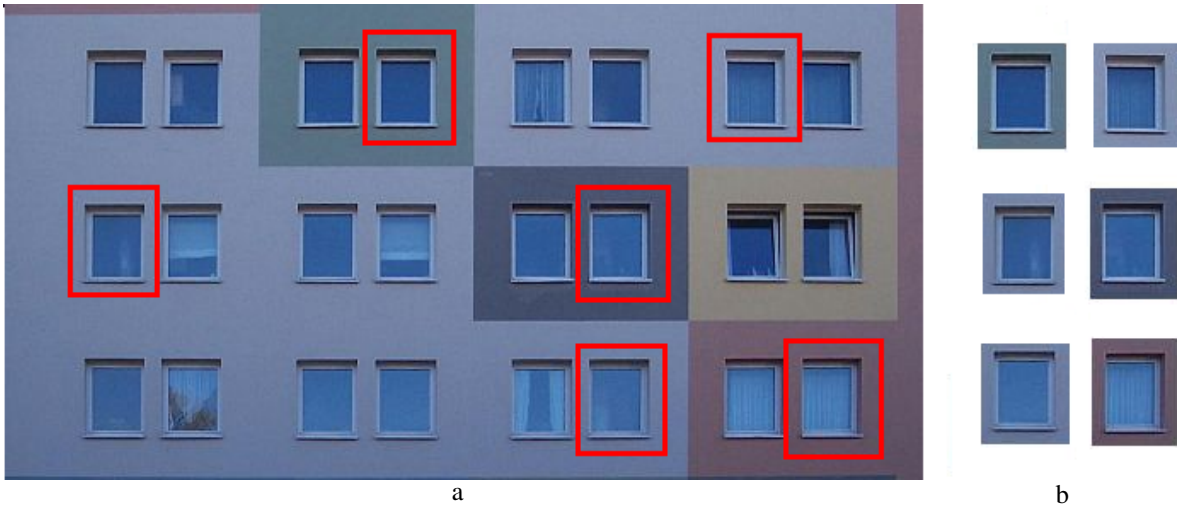
4.2 Überblick über den Ansatz

Gebäude sind künstliche Objekte, die für bestimmte Zwecke entworfen und gebaut werden. Aus praktischen und ästhetischen Gründen liegen die Größen der inneren Räume in einem bestimmten Bereich. Normalerweise gibt es auf einem Stockwerk keine Stufen und die Höhe eines Raumes ist mindestens 2,3 Meter. Der untere Rand von Fenstern ist oft ein Fensterbrett, das meist einen Abstand vom Boden besitzt. Dies alles führt dazu, dass die Anordnung der Fenster auf Fassaden häufig den internen Aufbau der Gebäude widerspiegelt. Oft sind Objekte auf Fassaden auch in regulären Strukturen angeordnet.

Der in dieser Arbeit vorgestellte Ansatz zielt darauf ab, ein Fassadenmodell durch Analyse terrestrischer Bilder zu generieren. Wenn nur eine Aufnahme vorliegt, ist das Ergebnis ein flaches Fassadenmodell. Falls mehrere Aufnahmen einer Fassade, die von unterschiedlichen Standpunkten aufgenommen wurden, ausgewählt werden, ist das Ergebnis ein 3D Modell der sichtbaren Fassaden.

Als Vorbereitung wird ein Trainingprozess durchgeführt. Er besteht aus den folgenden zwei Stufen:

1. *Erstellung eines Trainingsdatensatzes*:
Es werden terrestrische Fassadenbilder mit Fenstern einer Art, zum Beispiel “modern”, selektiert. Die Bilder werden auf die Fassadenebene transformiert, d.h. orthogonalisiert und auf eine Auflösung von ca. 1 cm pro Pixel skaliert. Dann werden Bildteile mit Fenstern ausgewählt (siehe Abbildung 4.2). Diese sind die Basis zum Lernen von Strukturen mittels *Implicit Shape Models* (ISM), siehe Abschnitt 2.1.



a b

Abbildung 4.2: *a) Orthogonalisierte Fassade; b) Bildteile mit Fenstern.*

2. *Lernen von Strukturen für ISM.*
Die Bildausschnitte um *Förstner Punkte* – FP (Förstner und Gülch 1987), die in der Nähe der manuell vorgegeben Fensterecken liegen, und die Differenzvektoren von den FP zum Fensterzentrum sowie zu den Fensterecken werden gespeichert. Sie bilden den *Trainingsdatensatz* zur weiteren Erkennung und Segmentierung von Fenstern.

Die Detektion und Segmentierung von Fenstern basiert auf *ISM* mit zwei Stufen:

1. *Generierung von Fensterhypothesen und Fensterumrissen*
Als Grundlage wird manuell oder automatisch ein orthogonalisiertes und skaliertes Fassadenbild erzeugt (siehe Abschnitt 4.3). Auf Basis des Trainingsdatensatzes wird die Lage und der Umriss der Fensterhypothesen mittels ISM bestimmt.

2. *Validierung der Hypothesen.*
Für die Validierung in Form von *Selbstdiagnose* werden mittels eines Bewertungssystems alle Hypothesen nach ihrer Qualität sortiert. Schwache Fensterhypothesen werden auf Grundlage von starken Fensterhypothesen überprüft. Hierfür wird angenommen, dass auf einer Fassade gewöhnlich zumindest mehrere ähnliche Fenster liegen.

Mittels *Modellauswahl* wird das am besten geeignete Modell für die Fensterkonfiguration in Form von Zeilen, Spalten oder einzelnen Fenstern ausgewählt. Wenn mehrere Aufnahmen einer Fassade vorhanden sind, wird für die gefundenen Fenster oder aus diesen gebildeten Zeilen oder Spalten durch *Plane Sweeping* die Tiefe auf der Fassade dreidimensional (3D) bestimmt (Werner und Zisserman 2002).

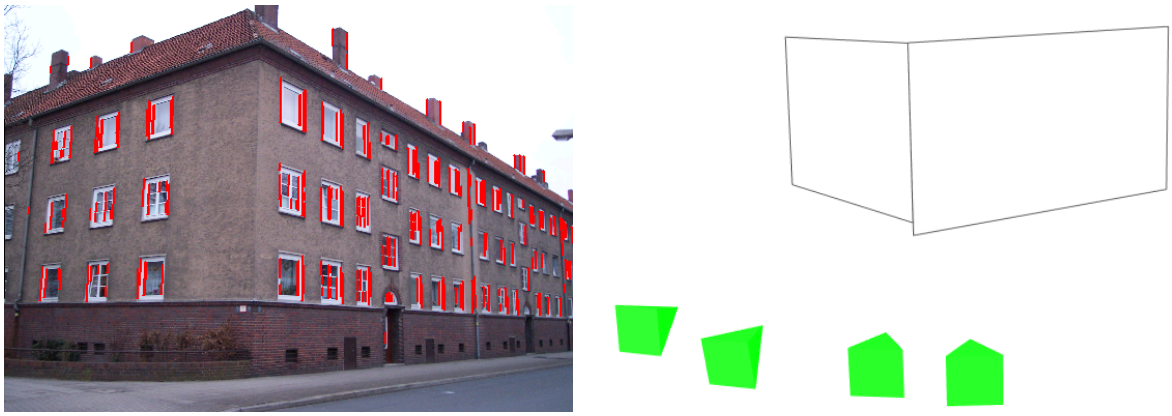
4.3 Generierung von Fassadenebenen aus Bildsequenzen

Die vorgelegte Arbeit basiert auf einen Ansatz zur 3D Rekonstruktion aus un- oder schwach kalibrierten Wide-Baseline Bildsequenzen ohne Näherungen und ohne die Verwendung von Passpunkten (Mayer 2005). Die Grundlage ist eine projektive oder im kalibrierte Fall euklidische Rekonstruktion auf Grundlage von Förstner Punkten (Förstner und Gülch 1987) mittels Fundamentalmatrizen und Trifokaltensoren (Hartley und Zisserman 2003) bzw. dem 5-Punkt Algorithmus von (Nistér 2003), die durch *Random Sample Consensus* – RANSAC (Fischler und Bolles 1981) robustifiziert wird. Die mittels Trifokaltensor verknüpften Bildtripel werden über die Direkte Lineare Transformation (DLT) bzw. eine euklidische Transformation im kalibrierten Fall in ein einheitliches Koordinatensystem gebracht. Um große Bilder effizient verarbeiten zu können, werden Bildpyramiden verwendet, wobei Fundamentalmatrizen bzw. essentielle Matrizen und Trifokaltensoren nur auf den oberen Ebenen der Bildpyramiden berechnet und die Punkte dann auf die originale Auflösung verfolgt werden. Eine hohe Genauigkeit garantieren einerseits die affine Bildzuordnung mittels kleinster Quadrate und andererseits die robuste Bündelausgleichung nach allen Schritten.

Im unkalibrierten Fall wird die projektiv ausgeglichene Sequenz mittels des Ansatzes von (Pollefeys et al. 2004) kalibriert. Abschließend erfolgt eine euklidische Bündelausgleichung inkl. der Bestimmung der Verzeichnung. Das Ergebnis sind hoch präzise Orientierungen für die Kameras sowie 3D Punkte inkl. Kovarianzmatrizen. Abbildung 4.3 zeigt links drei von vier Bildern eines Gebäudes in Hannover und rechts das Ergebnisse für Orientierung und 3D Rekonstruktion. Die hohe Qualität zeigt sich darin, dass der rechte Winkel an der Gebäudeecke gut rekonstruiert ist.

Abbildung 4.3: *Drei von vier Bildern der Wide-Baseline Bildsequenz und euklidische Rekonstruktion (Punkte in rot, Kameras als grüne Pyramiden).*

Gebäude sind meist durch eine horizontale Ausrichtung geprägt. Um die Gebäude zu horizontieren, werden die auf Fassaden fast immer vorhandenen vertikalen Linien genutzt. Deren Bilder schneiden sich, wie bei allen parallelen Linien, im Fluchtpunkt. Es werden Linien mittels Burns-Linienextraktor (Burns et al. 1986) detektiert und hieraus mittels RANSAC Schnittpunkte bestimmt, die sehr vielen Geraden entsprechen. Aus Letzteren wird derjenige Fluchtpunkt ausgewählt, der am besten zu der vom Nutzer vorzugebenden Ausrichtung der Kamera passt (siehe Abbildung 4.4 a)). Aus dem Fluchtpunkt wird über die bekannte Kamerakalibrierung die entsprechende Raumrichtung berechnet und hiermit die Szene orientiert.

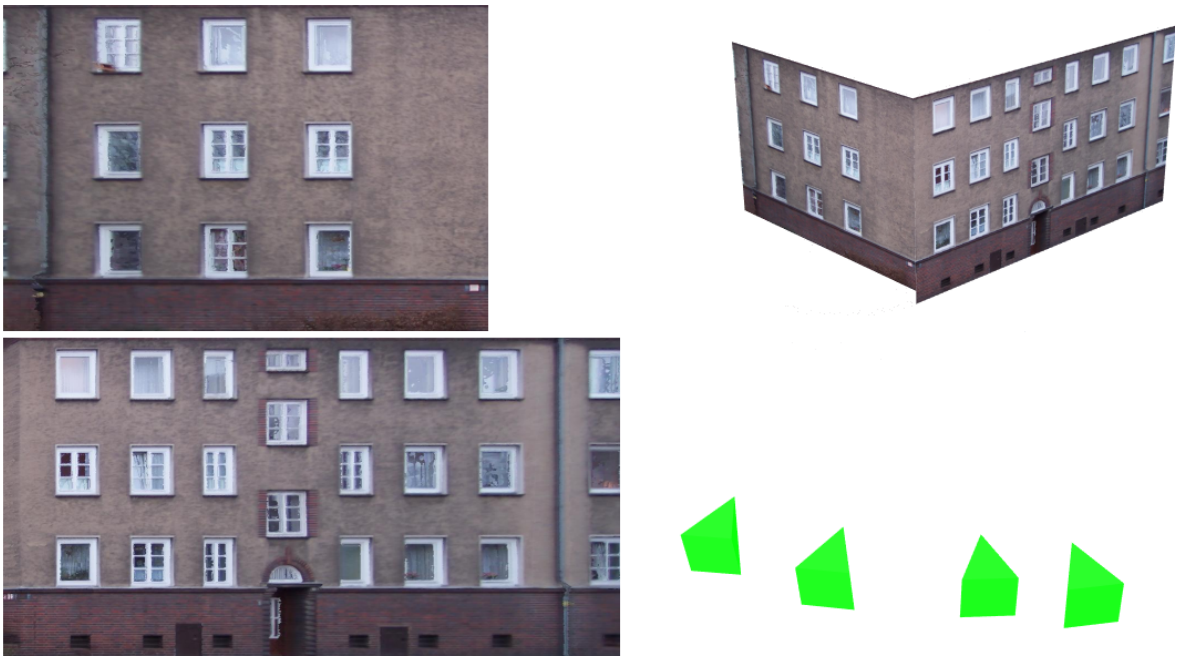


a b

Abbildung 4.4: *a) – Rote Linien, die den vertikalen Fluchtpunkt beschreiben; b) – Die zwei dominanten vertikale Ebenen und die Kameras (grüne Pyramiden).*

Fassaden bestehen in den meisten Fallen aus Ebenen. Diese werden, wie in (Bauer et al. 2003, Schindler und Bauer 2003) vorgeschlagen, mittels RANSAC extrahiert (Mayer 2007). Hierzu werden aus den euklidischen 3D Punkten jeweils drei Punkte zufällig ausgewählt und alle Punkte bestimmt, die auf der entsprechenden Ebene liegen. Zwar sind für die 3D Punkte die Genauigkeiten in Form der Kovarianzmatrizen bekannt, allerdings hängt die Frage, welche Punkte zu den Fassadenebenen gehören, von weiteren Faktoren, wie z.B. dem Maß-

stab, der geometrischen Anordnung der Kameras, der Anordnung der Punkte auf der Ebene, sowie der Baugenauigkeit ab. Deswegen wird die Genauigkeit der Ebenen sowie der maximale Abstand von auf die Ebene projizierten Punkten untereinander manuell vorgegeben. Letzteres vermeidet, dass Ebenen aus weit voneinander entfernten Punkten gebildet werden. Das Ergebnis sind die horizontierten Fassadenebenen sowie die Fassadenbilder, die sich als (robuste) Mittel aus allen, auf die Ebene projizierten Bildern ergeben (MAYER 2007) (siehe Abbildung 4.5 a)). Das Ergebnis für die Bildsequenz Hannover ist in Abbildung 4.5 b) dargestellt.



a b

Abbildung 4.5: *a) Zwei horizontierten Fassadenebenen; b) – Ergebnis für die Bildsequenz Hannover.*

Anmerkung: Für Einzelbilder werden die Parameter für eine Homographie auf Grundlage manuell gemessener Punkte bestimmt. Damit wird das Bild auf die Fassadenebene entzerrt.

Kapitel 5

Hypothesenbildung mittels Implicit Shape Models

5.1 Lernen von Strukturen

Für die generative Modellierung von Fassaden wird implizit angenommen, dass die Bestandteile, wie z.B. die Fassadenebene, die Fenster und die Türen, vollständig bekannt sind. Eine triviale Lösung bestände darin, eine große Menge von Fenstern zu nehmen und sie vollständig zu erlernen. Problematisch ist dabei, dass reale Fenster auf den Fassaden extrem unterschiedliche Formen, Größen, Farben und interne Strukturen haben (siehe Abbildung 5.1). So werden im Rahmen dieser Arbeit für die Detektion und die Segmentierung von Objekten auf Fassaden *Implicit Shape Models* — ISM (Leibe und Schiele 2004b, Leibe und Schiele 2004a, Borenstein und Ullman 2004, Agarwal und Roth 2002) verwendet, siehe auch Abschnitt 2.1.

ISM eignen sich für die Detektion von Objekten, welche gleiche Bestandteile und Größe aufweisen und ähnlich aussehen. Für eine Fassade ist zudem die Annahme von Bedeutung, dass ihre Orientierung bekannt ist. Somit hat ein rechteckiges Fenster eine vertikale Orientierung. Damit kann man sich auf Verschiebungen beschränken und auf Drehungen verzichten.

a b c d e f g

Abbildung 5.1: *Fenstertypen.*

Da es auf Fassaden sehr häufig andere Objekte, wie z.B. Balkone, Türen oder Ornamente gibt, die wie Fenster teilweise eine rechteckige Form besitzen, ist es sinnvoll, Fenster mit

gemeinsamen Merkmalen zu gruppieren. Im Rahmen dieser Arbeit wurden die Fenster in drei Typen unterteilt: *Modern*, *Klassik* und *Bogen*.

Der Typ *Modern* umfasst Fenster, die eine rechteckige Form, wenig interne Struktur (Sprossen u.Ä.) sowie wenig Struktur außerhalb der Fensterscheiben haben (siehe Abbildung 5.1 d) und f)). Zum Typ *Klassik* gehören Fenster, die eine rechteckige Form, viel interne Struktur und häufig Außenstrukturen besitzen (siehe Abbildung 5.1 a), b), c) und g)). Dem Typ *Bogen* werden Fenster zugeordnet, die Bögen, runde, oder andere komplizierte Form aufweisen (siehe Abbildung 5.1 e)).

Die meisten Fenster gehören zu den ersten zwei Typen. Im Rahmen dieser Arbeit wurden in der Mehrzahl *moderne* und teilweise *klassische* Fenster analysiert. Zum Lernen wurden Bilder mit Fassaden gesammelt, auf die Fassadebene transformiert und horizontiert (siehe Abschnitt 4.3 und Abbildung 5.2.). Alle Bilder wurden auf die selbe Pixelgroße von ca. 1cm pro Pixel skaliert. Nur Bildteile mit Fenstern wurden für den Trainingsprozess verwendet. Insgesamt wurden 119 Fenster des Typs *Modern* und 100 Fenster des Typs *Klassik* ausgewählt.

a b

Abbildung 5.2: *a) Originalaufnahme einer Fassade; b) Orthogonalisiertes Bild.*

Alle im Trainingsprozess verwendeten Fenster haben eine rechteckige Form und sind vertikal orientiert, d.h. ihre Form ist nur von den Ecken abhängig. Deshalb wurden die Bereiche um die Ecken als wesentliche Teile eines Fenstes sowie deren relative Lagen, genauer gesagt die Fensterecken relativ zur Fenstermitte erlernt (siehe Abbildung 5.3 a)). Als Punktmerkmale werden Förstnerpunkte (Förstner und Gülch 1987) benutzt. Diese sind robust gegen Kontrast- und Helligkeitsunterschiede und damit liegen Punkte für ähnliche Bildstrukturen am gleichen Ort. Für Fenster bedeutet das, dass in der Nähe der Fensterecken meist die selben Punkte extrahiert werden und die Punktlage relativ zum Fenstereck sehr ähnlich ist.

Die Bildausschnitte um die Förstnerpunkte und ihre relative Lage in Form von Differenzvektoren zur zugehörigen Fenstermitte sind die Basis für ISM (siehe Abbildung 5.3 b)) und bilden den *Trainingsdatensatz*.

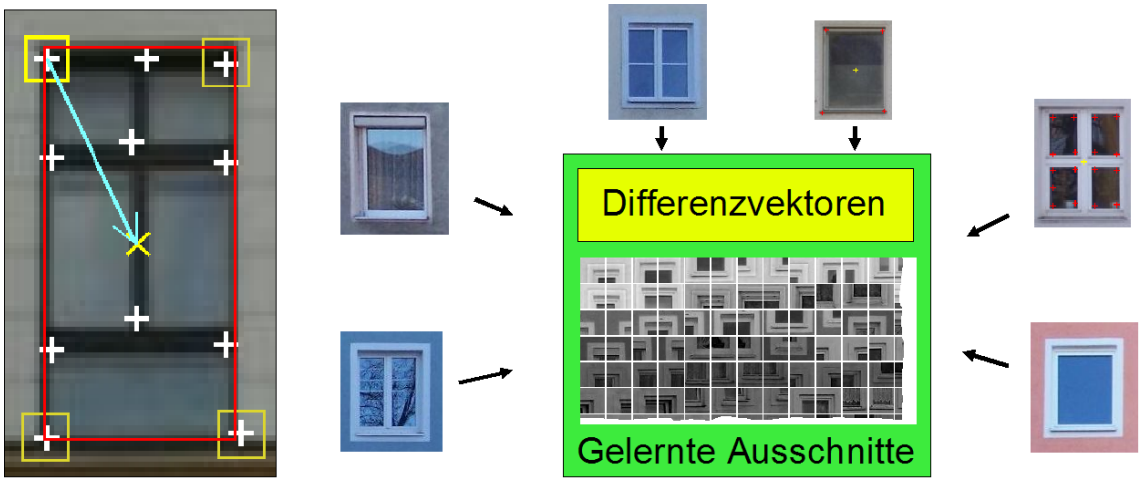


Abbildung 5.3: *a) – Fenster mit manuell markiertem Umriss (rotes Rechteck), Förstnerpunkte (weiße Kreuze) und ein Differenzvektor eines Förstnerpunktes in der Nähe einer Fensterecke zur Fenstermitte (blauer Pfeil); b) – Gelernte Differenzvektoren und Bildausschnitte um Förstnerpunkte für eine Fenstermenge (Trainingsdatensatz).*

Die bis hierher vorgestellte Vorgehensweise erzeugt Trainingsdatensätze, die nur für eine Detektion der Fenstermitte geeignet sind ((Mayer und Reznik 2005, Mayer und Reznik 2006, Reznik und Mayer 2007)). ISM können aber auch zur Segmentierung verwendet werden, indem das Verhältnis von Bildausschnitt zu Objektgrenze gelernt wird. Nach einem in (Mayer und Reznik 2006) vorgestellten Zwischenschritt wurde letztendlich der in (Reznik und Mayer 2008) und im Weiteren vorgestellte Weg entwickelt (siehe Abbildung 5.4).

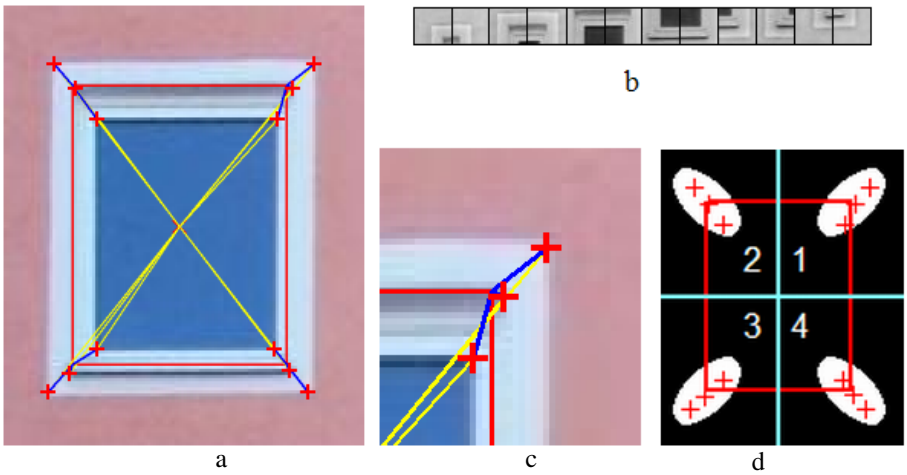


Abbildung 5.4: *Training für Fensterdetektion und Segmentierung - a) Bildausschnitt mit manuell vorgegebenem Fensterumriss (rotes Rechteck), Förstnerpunkte an den Ecken des Fensterumrisses (rote Kreuze) sowie Vektoren von den Förstnerpunkten zum Fenstermittelpunkt (gelbe Linien) und zu der korrespondierenden Ecke des Umrisses (kurze blaue Linien); b) Bildausschnitte um die Förstnerpunkte; c) Detail von a) zeigt das Verhältnis der Förstnerpunkte zu einer Ecke des Fensters; d) Elliptische Flächen um die Fensterecken (weiß) in denen die Förstnerpunkte extrahiert werden.*

Auf jedem Trainingsbild wird der Umriss des Fensters manuell markiert (rotes Rechteck in Abbildung: 5.4 a)) und Förstnerpunkte werden extrahiert. Da empirisch gefunden wurde, dass vor allem die Bereiche in der Nähe der Fensterecken für die Detektion und Segmentierung von Bedeutung sind, werden nur Förstnerpunkte akzeptiert, die in elliptischen Flächen in der Nähe der Umrissecken liegen (siehe Abbildung 5.4 d)). Die Ausmaße der Ellipse sind für jedes Fenster gleich, weil alle auf den gleichen Maßstab skaliert wurden. Die Halbachsen der Ellipsen wurden auf 20 und 10 Bildpunkte gesetzt. Die resultierenden quadratischen Bildausschnitte um die Förstnerpunkte sind in Abbildung 5.4 b) dargestellt. Sie sind die Basis für die weitere Fensterdetektion und Segmentierung. Die Bildausschnitte werden zusammen mit den relativen Koordinaten (*Differenzvektoren*) zwischen dem Förstnerpunkt und dem Zentrum des Fensters gespeichert. Das Zentrum wird aus dem manuell markierten Umriss des Fensters berechnet. Einer der Differenzvektoren ist als gelbe Linie in Abbildung 5.4 a) dargestellt. Für das Lernen der Fensterumrisse werden zusätzlich die Differenzvektoren zwischen jedem Förstnerpunkt und den zugehörigen Ecken des Umrisses gespeichert. Diese Vektoren sind als blaue Linien in Abbildung 5.4 a) und c) dargestellt.

Die gespeicherten Daten für das Fenster aus Abbildung 5.4 sind in der Tabelle 5.1 zusammengefasst. Alle anderen notwendigen Parameter wie die Ausschnittsgröße werden ebenfalls gespeichert.

Tabelle 5.1: *Gespeicherte Daten für gelernte Bildausschnitte. ΔXc und ΔYc – relative Abstände zwischen Förstnerpunkt und Zentrum des Fensters; Q – Quadrant für Förstnerpunkt; ΔxN und ΔyN – relative Abstände zwischen Förstnerpunkt und Ecken des Umrisses.*

ΔXc	ΔYc	Q	$\Delta x1$	$\Delta y1$	$\Delta x2$	$\Delta y2$	$\Delta x3$	$\Delta y3$	$\Delta x4$	$\Delta y4$
41	53	2	6	98	6	7	76	7	76	98
-44	53	1	-79	98	-79	7	-9	7	-9	98
34	45	2	-1	90	-1	-1	69	-1	69	90
-37	45	1	-72	90	-72	-1	-2	-1	-2	90
27	35	2	-8	80	-8	-11	62	-11	62	80
-32	35	1	-67	80	-67	-11	3	-11	3	80
27	-40	3	-8	5	-8	-86	62	-86	62	5
-31	-40	4	-66	5	-66	-86	4	-86	4	5
-36	-47	4	-71	-2	-71	-93	-1	-93	-1	-2
36	-48	3	1	-3	1	-94	71	-94	71	-3
43	-54	3	8	-9	8	-100	78	-100	78	-9
-42	-54	4	-77	-9	-77	-100	-7	-100	-7	-9

Für rechteckige Objekte ist Speicherung redundant. Es wäre ausreichend, nur einen Differenzvektor, nämlich jenen zur nächstliegenden Umrissecke, zu speichern. Für Objekte mit komplizierterer Form ist es aber sinnvoll, zusätzliche Information zu speichern.

Die manuelle Markierung der Fensterumrisse erfolgt nur einmal. Damit wird es möglich, Parameter, wie z.B für den Förstneropetator oder die Ausschnittsgröße, zu ändern, und trotzdem einen vergleichbaren Trainingsdatensatz zu erhalten, der sich auf die gleichen Umrisse

bezieht.

Die Ausschnittsgröße eines Trainingdatensatzes ist eine ungerade Zahl zwischen 9 und 35 Pixel. Eine ungerade Zahl wird gewählt, weil der Ausschnitt damit einen zentralen Punkt hat. Für jede Größe wird ein entsprechender Trainingsdatensatz erstellt. Hierbei wird auf Auschnitte verzichtet, die näher als die Hälfte der Ausschnittgröße zur Bildgrenze liegen. Je größer der Ausschnitt ist, desto höher ist die Zahl solcher Punkte. Insgesamt haben Trainigsdatensätze für moderne Fenster folgende Anzahl von Ausschnitten: 9×9 Pixel – 1210; 15×15 Pixel – 1203; 25×25 Pixel – 1079; 35×35 Pixel – 827 Ausschnitte. Trainingsdatensätze für verschiedene Ausschnittsgrößen sind in Abbildung 5.5 dargestellt.

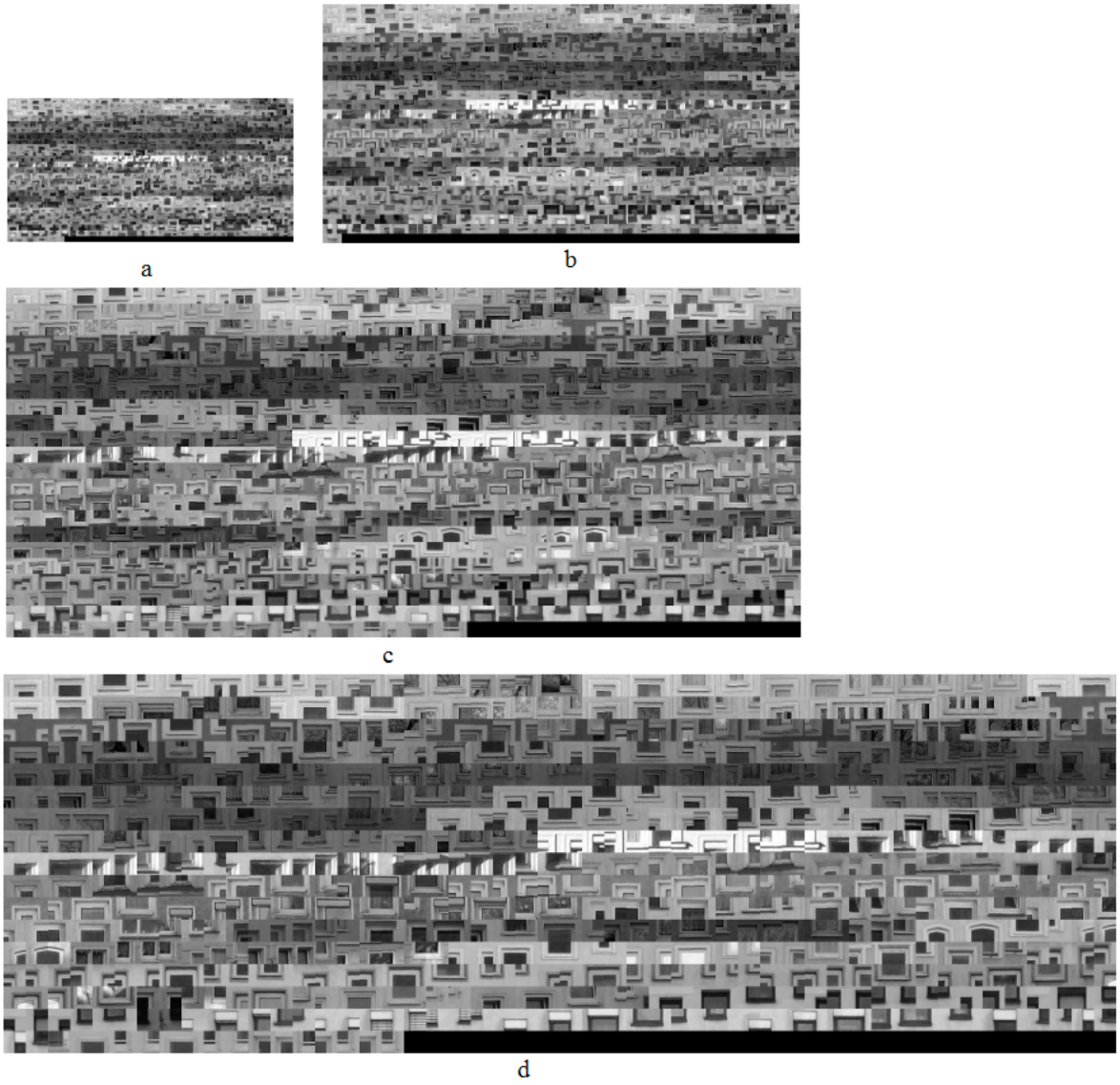
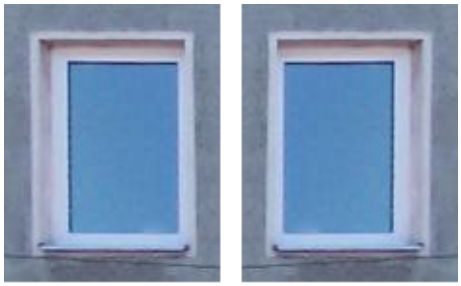


Abbildung 5.5: *a) – Bildausschnitte für moderne Fenster für die Ausschnittsgrößen 9×9 Pixel; b) – 15×15 Pixel; c – 25×25 Pixel; d) – 35×35 Pixel.*

Jeder Bildausschnitt des Trainingsdatensatzes wird in Form seiner vertikal gespiegelten Kopie zweimal verwendet. Dabei wird die horizontale Koordinate des Differenzvektors entspre-

chend invertiert. Dies ist sinnvoll, weil ein Fenster zumeist eine vertikale Symmetrie besitzt. Fenster liegen so weit vor (oder hinter) der Fassade und Fassaden selbst sind so große Objekte, dass durch die Zentralperspektive von nicht zu weit entfernten Aufnahmen die Leibungen der Fenster z.T. von links und z.T. von rechts gesehen werden (siehe Abbildung 5.6).



a b

Abbildung 5.6: *Fenster – a) und gespiegelte Kopie – b).*

5.2 Hypothesengenerierung

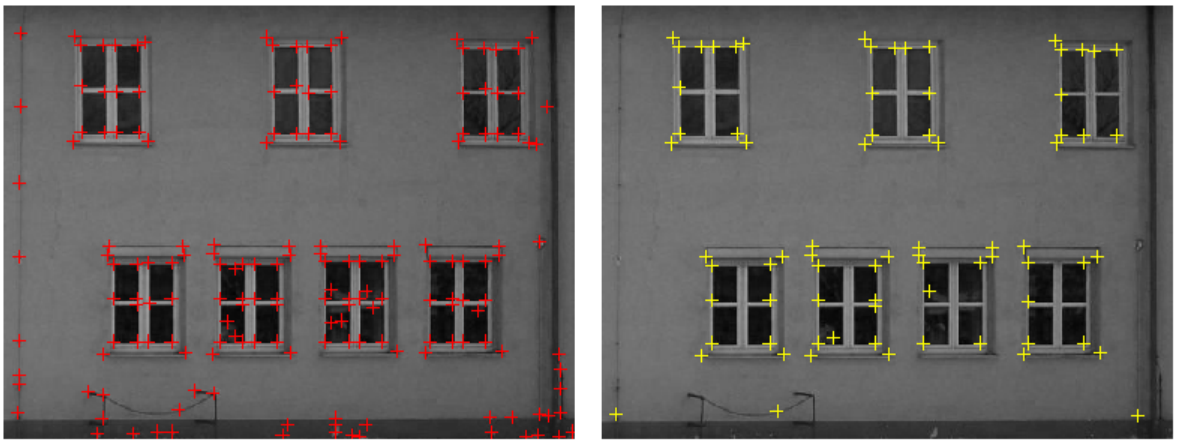
Liegt ein Trainingsdatensatz vor, so können Fenster detektiert werden. Da aber eine nicht vernachlässigbare Wahrscheinlichkeit besteht, dass ein erkanntes Objekt kein Fenster ist, sondern z.B. eine Tür oder eine Störung, werden die Objekte als *Hypothesen*, d.h. *Fensterhypothesen* bezeichnet. Zur Demonstration der Fensterdetektion und -segmentierung werden nur Fenster von Fassaden benutzt, die nicht Teil des Trainingsdatensatzes sind.

Vor der Detektion wird das zu untersuchende Bild auf die Fassadenebene transformiert (siehe Abschnitt 4.3), vertikal ausgerichtet und auf die Pixelgröße des Trainingssatzes von ca. 1 cm pro Bildpunkt skaliert. Damit sind die Objektgrößen im Trainingsdatensatz und auf der zu untersuchenden Fassade ungefähr gleich, was die Wahrscheinlichkeit einer korrekten Fenstererkennung erhöht.

Experimente mit einer großen Zahl von Fassaden haben gezeigt, dass es keine optimale Größe für die Bildausschnitte gibt, welche für alle Fälle geeignet ist. Sie hängt von der Komplexität der Fenster ab, den vorhandenen Störungen, der Qualität der Aufnahme, u.s.w. In der Praxis wurden Trainingsdatensätze für Ausschnittsgrößen von 15, 25 und 35 Punkten verwendet (siehe Abbildung 5.5 b), c) und d)).

Für die Wiederherstellung d.h., für die Detektion der Fenster, werden Förstnerpunkte mit den gleichen Parametern wie für den Trainingsdatensatz, aber im gesamten Bild extrahiert (siehe Abbildung: 5.7 a)).

Bildausschnitte um die extrahierten Punkte, z.B. der Größe 35 x 35 Pixel, werden mit allen gelernten Ausschnitten mittels Kreuzkorrelation verglichen. Wenn der Kreuzkorrelationskoeffizient (CCC – siche Abschnitt 2.1) über dem empirisch gefundenen Schwellwert von 0,8 liegt, wird der Ausschnitt als Hypothese für eine Fensterecke akzeptiert. Der *Differenzvektor*, der sich auf die Fenstermitte bezieht, wird verwendet, um eine Hypothese für die Mitte des

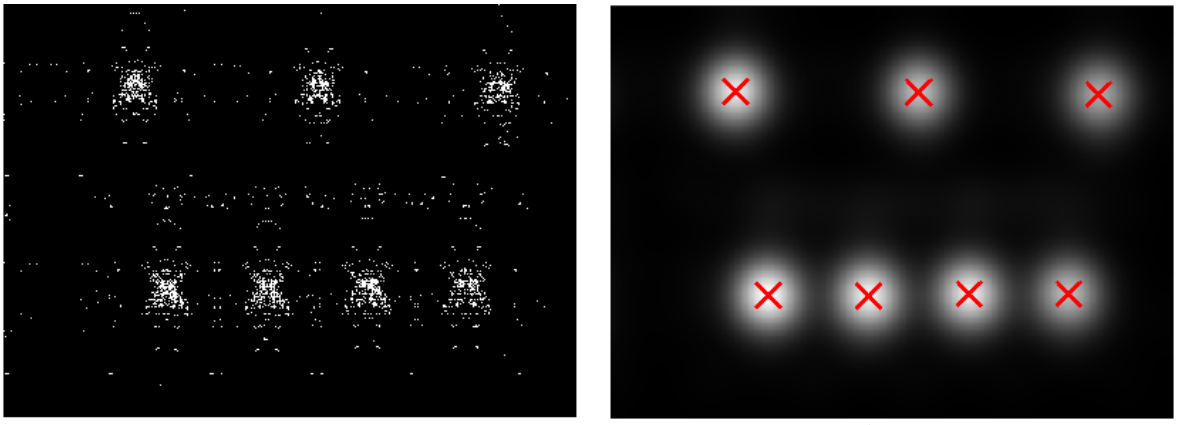


a b

Abbildung 5.7: *Extrahierte Förstnerpunkte; b) akzeptierte Förstnerpunkte.*

Fensters zu erzeugen, indem in einem anfänglich leeren Akkumulationsbild Evidenz inkrementiert wird. Weil Bildausschnitte u.U. mit mehreren Bildausschnitten der Trainingsdaten stark korreliert sein können, kann jeder Punkt mehrfach für mögliche Mittelpunkte stimmen. Die Förstnerpunkte, die mindestens einmal akzeptiert worden sind, sind in Abbildung 5.7 b) als gelbe Kreuze gekennzeichnet. Die Enden der Differenzvektoren als Evidenz für die Fenstermittelpunkte sind für das verwendete Beispiel in der Abbildung 5.8 a) dargestellt.

Die akkumulierte Evidenz für Fenstermittelpunkte wird über einen Gaußfilter mit der durchschnittlichen Größe der Trainigsfenster (ca. 80 Pixel) integriert und die lokalen Maxima dieser Funktion werden als Hypothesen für Fenster angesehen (siehe Abbildung 5.8 b)).



a b

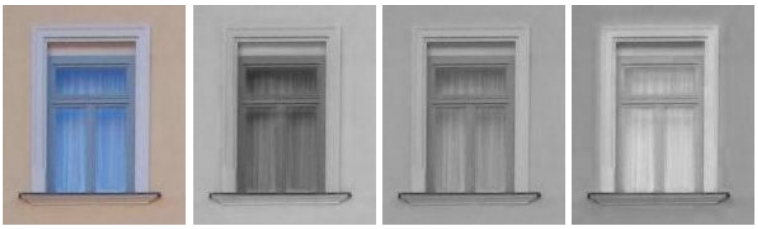
Abbildung 5.8: *a) Evidenz für die Fenstermittelpunkte; b) Gauß-gefilterte lokale Maxima.*

Kapitel 6

Multiskalige generative Interpretation von hellen Fassaden

Im Rahmen der Entwicklung des vorgestellten Ansatzes wurde als Zwischenschritt (Mayer und Reznik 2005, Mayer und Reznik 2006, Reznik und Mayer 2007) ein auf einem generativen statistischen Modell (siehe Abschnitt 2.4) und Maßstabsräumen (siehe Abschnitt 2.2) beruhender Ansatz zur Extraktion von Fenstern entwickelt, der auf den im Folgenden genannten Annahmen basiert:

- Fensterscheiben sehen während des Tages, wenn die Aufnahmen gemacht werden, meist dunkel aus. Dies gilt besonders für den roten Kanal, da Fenster aus Glas bestehen, das besonders rotes Licht durchlässt, und weil der Himmel, der in den Fenstern reflektiert wird, meistens blau ist. Ein typisches Beispiel ist in Abbildung: 6.1 dargestellt.



a b c d

Abbildung 6.1: *Farbkanäle für ein Fenster. a) – Farbbild, b) - Roter Kanal; c) – Grüner Kanal; d) – Blauer Kanal.*

- Die meisten Fenster und Türen haben eine rechteckige Form. Eine Seite ist vertikal, um das Fenster leicht öffnen zu können. Aus der Untersuchung einer großen Zahl Fenster ergibt sich, dass das Verhältnis von Höhe zur Breite eines Fensters meist zwischen 0,25 und 5 liegt (Dick et al. 2002). Sehr schmale Fenster sind häufiger als sehr breite. Fenster und Türen bestehen häufig aus komplizierten Bestandteilen, wie z.B. horizontalen und vertikalen Fenstersprossen, aber auch Blumentöpfen.

Der letzte Punkt bedeutet, dass sich Fenster oder Türen auf verschiedenen Skalen, bzw. Ebenen eines Maßstabsraumes (siehe Abschnitt 2.2), durch Abstraktion unterschiedlich ausprägen. Dies führt zu einer Spezialisierungshierarchie, die sich auch für die Visualisierung eines Objektes als Prototyp eignet. Für Fenster gilt:

- Auflösung ca. 1 cm pro Pixel: Fenster mit allen Details (Visualisierung: Bild des Fensters, reale Tiefen)
- Auflösung ca. 5 – 10 cm pro Pixel: Fenster als dunkle Fläche mit wenigen Details (Visualisierung: durchschnittliches Bild und Tiefe des Fensters als Kombination mehrerer ähnlicher Fenster)
- Auflösung ca. 10 – 50 cm pro Pixel: Fenster als dunkles Rechteck (Visualisierung: Rechteck mit durchschnittlicher Farbe auf Ebene hinter Fassade)

Das im Weiteren zur Extraktion verwendete Modell für Fenster besteht aus dunklen Rechtecken auf einem hellen Hintergrund. D.h., es entspricht der Auflösung von 10 – 50 cm pro Pixel. Für den Vergleich von Modell und Bild wird wegen der Effizienz und weil sich dies empirisch bewährt hat, die normierte inkrementell implementierte Kreuzkorrelation (CCC – siehe Abschnitt 2.3) verwendet. Um das Ergebnis robuster zu machen, wird auf das Modell ein Rauschen von zehn Grauwerten gelegt, das nicht modellierte Strukturen implizit repräsentieren soll.

Für eine verlässliche Zuordnung von abstrahiertem Modell und Bild hat sich eine Abstraktion, d.h., die Nutzung eines Maßstabsraumes (siehe Abschnitt 2.2) als essenziell erwiesen. Die Wahl fiel nach einigen Experimenten auf Grauwertmorphologie, implementiert als Dual Rank Filter (Eckstein und Munkelt 1995). Aus weiteren Experimenten ergab sich, dass es sinnvoll ist, zuerst mittels *Opening* mit einer Seitenlänge des quadratischen Strukturelements von ca. 7 cm dunkle Teile zu beseitigen und dann mit einem *Closing* mit einer Seitenlänge von ca. 15 cm die hellen Teile zu eliminieren. Das *Opening* vor dem *Closing* ist notwendig, um zu vermeiden, dass helle Teile nicht eliminiert werden können, weil sie durch kleine dunkle Teile gestört werden. Insgesamt ergibt sich eine Abstraktionshierarchie, die aus dem entzerrten Bild, dem gefiltertem, d.h., abstrahierten Bild, sowie dem verrauschten Modell besteht (siehe Abbildung 6.2).

Die Parameter des aus dunklem Fenster und hellem Hintergrund bestehenden generativen Modells werden mittels MCMC (siehe Abschnitt 2.3) geschätzt. Für jede Iteration von MCMC wird entweder die Breite, die Höhe, oder die Position des dunklen Vierecks, das ein Fenster darstellt, verändert. Die Wahrscheinlichkeit beträgt 30% für die Änderung von Breite oder Höhe und 20% für die horizontale oder vertikale Position. Dies berücksichtigt, dass mehr über die Position als über die Größe bekannt ist, was sich aus der Verwendung eines Verfahrens zur Bestimmung der Näherungen ergibt, das nur den Mittelpunkt schätzt (siehe Kapitel 5). Für die Robustifizierung der Suche wird *Simulated Annealing* (siehe Abschnitt 2.3) verwendet.

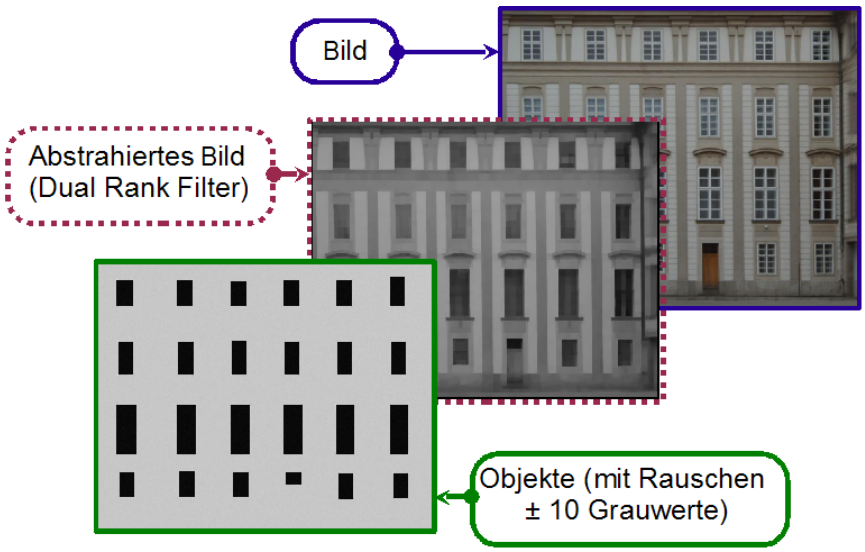


Abbildung 6.2: *Abstraktionshierarchie bestehend aus dem ursprünglichen Bild, dem mittels Dual-Rank Filter abstrahierten Bild sowie dem verrauschten Modell.*

Abbildung 6.3: *Dunkle Fenster auf heller Fassade - Hypothesen (weiße Quadrate) und Fenster (grüne Rechtecke).*

Abbildung: 6.3 zeigt das Ergebnis von Fensterdetektion und -extraktion auf einem mittels Grauwertmorphologie abstrahierten Bild. Hypothesen sind in Form von kleinen weißen Quadraten an den Positionen der Maxima der Evidenz des Implicit Shape Models dargestellt. Die Größe der die anfänglichen Hypothesen bildenden Quadrate wurde empirisch bestimmt. Die extrahierten Fenster zeigen, dass die Modellierung zu einem sinnvollen Ergebnis führt. In der Abbildung 6.4 sind weitere Ergebnisse dargestellt. Diese dokumentieren sowohl die Leistungsfähigkeit des vorgestellten Ansatzes, als auch Defizite in Form von nicht ideal extrahierten Fenstern. Besonders eklatant sind die Verbreiterung des Fensters unten in der Mitte der mittlerer Fassade durch die links von ihm befindliche dunkle Stange sowie die viel zu kleine Extraktion des Fensters in der Mitte unten der unteren Fassade. Eine Evaluierung der Ergebnisse erfolgt in Kapitel 8.

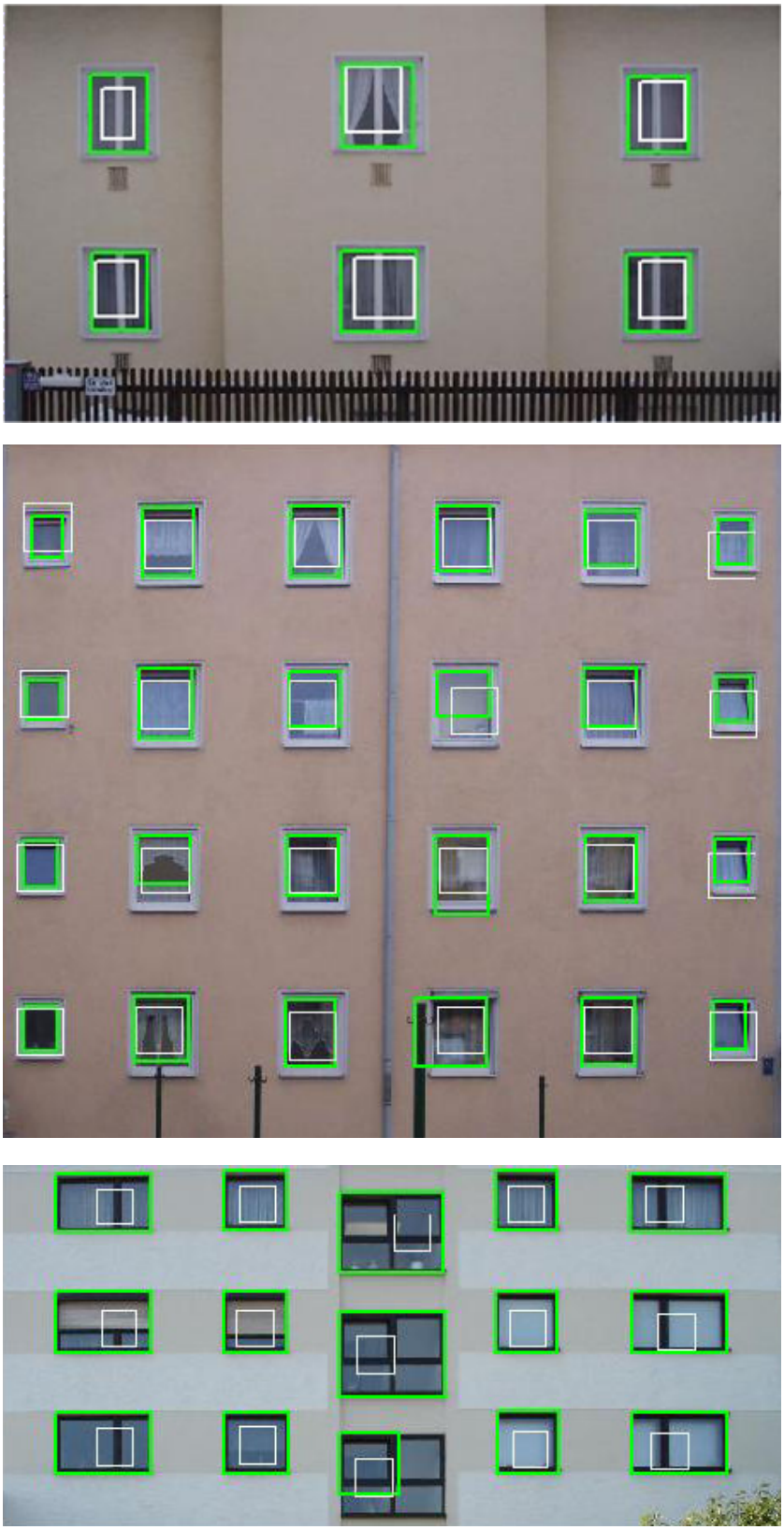


Abbildung 6.4: *Weitere Ergebnisse für dunkle Fenster auf hellen Fassaden – Hypothesen (weiße Quadrate) und Fenster (grüne Rechtecke).*

Kapitel 7

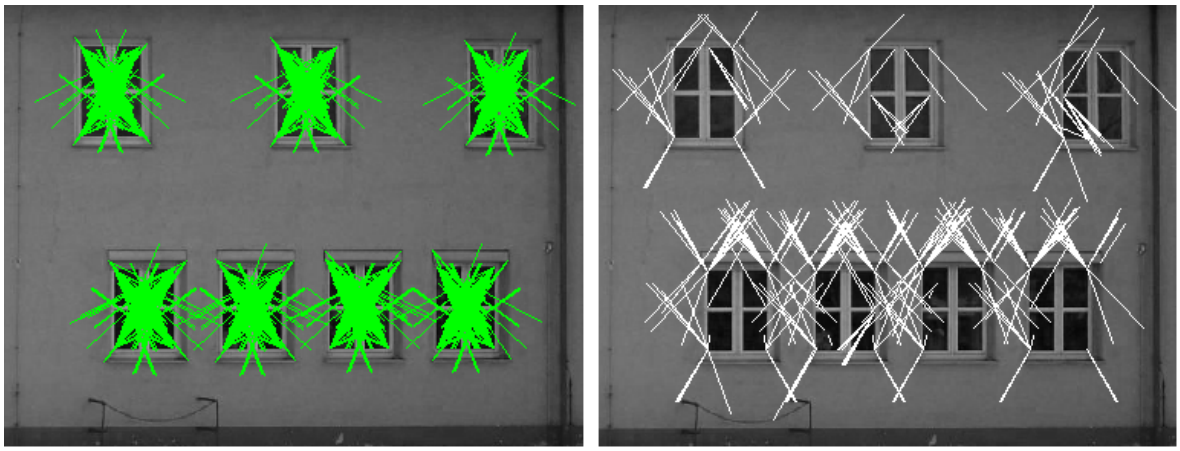
Generative aussehenbasierte Fassadeninterpretation

Der im letzten Kapitel präsentierte Ansatz ist auf Grund des verwendeten Modells auf helle Fassaden mit dunklen Fenstern beschränkt. Der im Weiteren vorgestellte Ansatz vermeidet dieses Defizit indem eine aussehensbasierte Modellierung in Form eines *Implicit Shape Models* – ISM (siehe Abschnitt 2.1 und Kapitel 5) verwendet wird. An die Objektextraktion schließt sich die Validierung der Objekte an (siehe Abschnitt 7.2). Abschnitt 7.3 beschreibt, wie die validierten Objekte hierarchisch in Zeilen und Spalten modelliert werden und zuletzt wird die 3D Ausprägung der Objekte bestimmt (siehe Abschnitt 7.4).

7.1 Objektextraktion

Abschnitt 5.2 zeigte, wie Mittelpunkte von Fenstern bestimmt werden. Ziel dieses Abschnittes ist die Segmentierung der Fenster auf Grundlage von ISM.

Die gelernten Ausschnitte, welche zu den Maxima der Evidenz führen (z.B. Abbildung 5.8 b)), sind Hypothesen für Bildbereiche in der Nähe von Ecken von Fensterumrissen. Bildausschnitte, die für ein Maximum votieren, werden in entsprechenden Mengen zusammengefasst. Für diese kann die Lage eines Bildausschnittes relativ zum entsprechenden Zentrum der Fensterhypothese bestimmt werden. Für die Wiederherstellung eines Fensterumrisses werden alle Trainingsausschnitte und Bildausschnitte erneut analysiert. Hierbei wird gefordert, dass ein Bildausschnitt und der entsprechende Trainingsausschnitt zum gleichem Quadranten gehören (siehe Abbildung 5.4 d) und Tabelle 5.1). D.h., jeder Differenzvektor muss vom Bildausschnitt in Richtung des Zentrums der Fensterhypothese zeigen. Ein entsprechendes Ergebnis ist in Abbildung 7.1 dargestellt. Die Anzahl der akzeptierten Vektoren ist deutlich höher, als die Anzahl der abgelehnten Vektoren. Damit ist beim gezeigten Beispiel eine große Mengen von Punkten für die erfolgreiche Wiederherstellung der Fensterumrisse verfügbar.



a b

Abbildung 7.1: *Filterung der Zuordnung von Bild- und Trainingsausschnitten: a) Akzeptierte Differenzvektoren; b) Abgelehnte Differenzvektoren.*

Um die Fensterumrisse genau abzugrenzen, wird die aus Trainingsdaten (siehe Abschnit 5.1) bekannte Relation zwischen den Mitten der Trainingsausschnitte und dem gegebenen Umriss der Fenster verwendet.

Diese Relationen sind als blaue Linien in der Abbildung 5.4 a) und c) gekennzeichnet. Ein Beispiel für die grundlegende Idee der Bestimmung des Fensterumrisses ist in Abbildung 7.2 gegeben. Der Kreuzkorrelationskoeffizient zwischen dem Bildausschnitt um den roten

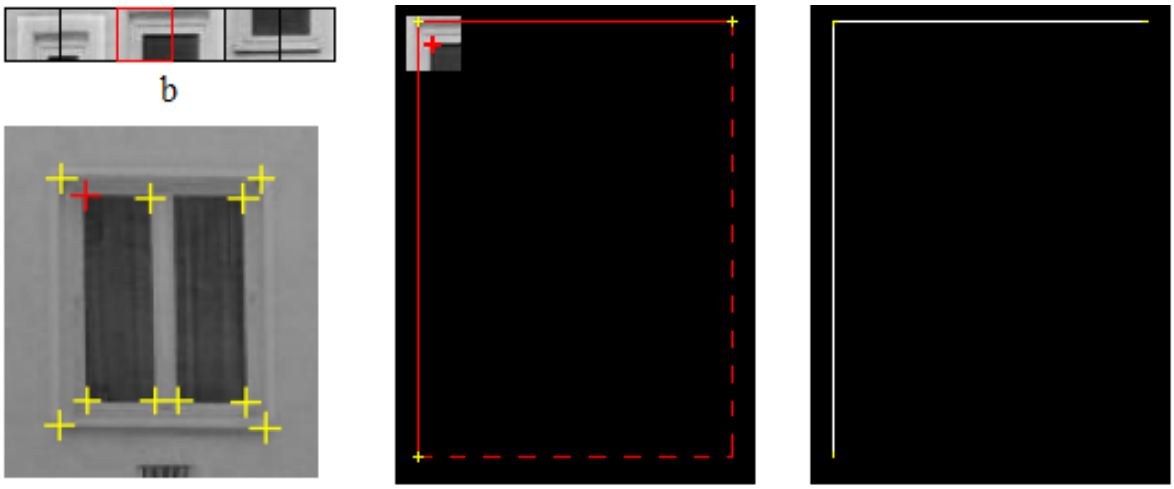
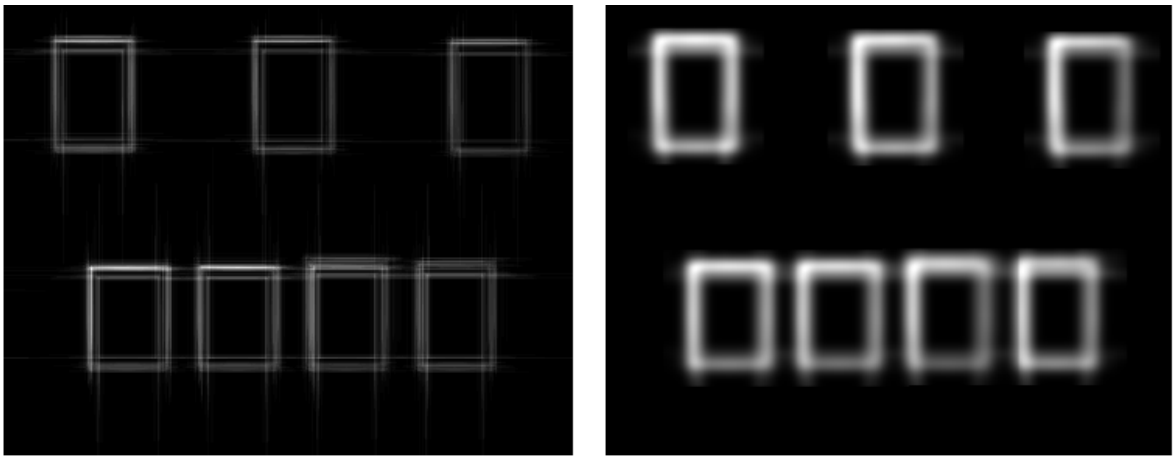


Abbildung 7.2: *Bestimmung der Fensterumrisse - a) Bildpunkte; b) Trainingsausschnitte - der Ausschnitt links über dem b) wurde dem Ausschnitt um das rote Kreuz an der oberen linken Ecke der Fensterscheibe in a) zugeordnet; c) Relation zwischen dem Zentrum des Ausschnitts (dickes rotes Kreuz) und dem Fensterumriss in den Trainingsdaten (linkes Kreuz für Position – Längen der Seiten aus Trainingsdaten); d) Hypothese für Teile des Fensterumrisses.*

Punkt in der oberen linken Ecke in der Abbildung 7.2 a) und dem roten Trainingsausschnitt in der Abbildung 7.2 b) liegt über dem empirisch bestimmten Schwellwert von 0,8. Abbildung

7.2 c) zeigt die Relation zwischen der Mitte des Ausschnittes, der durch ein dickes rotes Kreuz gekennzeichnet ist, und der Ecke des Umrisses des Fensters, die durch ein kleines gelbes Kreuz markiert ist. Von der Ecke des Umrisses aus werden die zwei benachbarten Seiten des Rechtecks aus den Trainingsdaten gezeichnet (siehe Abbildungen 7.2 c) und d)). Das Resultat ist eine Hypothese für Teile des Fensterumrisses.

Die Hypothesen für die Fensterumrisse werden über alle Bildausschnitte und alle Trainingsausschnitte akkumuliert. Wie in Abschnitt 5.1 wird jeder Trainingsausschnitt zusammen mit seiner gespiegelten Kopie verwendet. Das Resultat ist eine Verteilung für die Fensterumrisse wie diese in der Abbildung 7.3 a) dargestellt ist, die geglättet wird (siehe Abbildung 7.3 b)). Die geglättete Verteilung wird relativ zu ihrem maximalen Wert normalisiert, der zu eins gesetzt wird.



a b

Abbildung 7.3: *a) Verteilung für Fensterumrisse auf ganzer Fassade – a) Akkumulation; b) Nach Glättung und Normalisierung.*

7.2 Validierung der Objekte

So gute Ergebnisse für die Fensterumrisse wie in Abbidung 7.3 b) ergeben sich nur bei sehr einfachen standardisierten Fenstern und hochqualitativen Bildern. Viele Fassadenbilder sind aber nicht so einfach: Fenster sind nicht einheitlich oder von Vegetation und anderen Gegenständen teilweise oder sogar ganz verdeckt. Sehr oft werden für ein Fenster nicht alle vier Ecken gefunden und somit ist der Umriss nicht vollständig definiert. Dazu kommen falsche Objekte: Auf Fassaden liegen sehr häufig andere Objekte und Störungen mit teilweise rechteckiger Form. Diese sehen z.T. Fenstern ähnlich. Für solche Fälle wurde ein Verfahren zur Validierung entwickelt.

Die Grundidee, bzw. Annahme für die Validierung ist folgende: In der Regel liegen auf einer Fassade zumindest einige gleiche Fenster. Deswegen ist es sinnvoll, mit Hilfe von gut erkannten Fenstern alle anderen Objekte zu testen, ob sie übereinstimmen. Eine ähnliche Idee wird in (Van Gool et al. 2007, Müller et al. 2007) verwendet.

Da die Suche auf rechteckige Fenster beschränkt ist, reicht ein Notensystem für die Bewertung von rechteckigen Objekten, wie es in Abbildung 7.4 dargestellt ist.

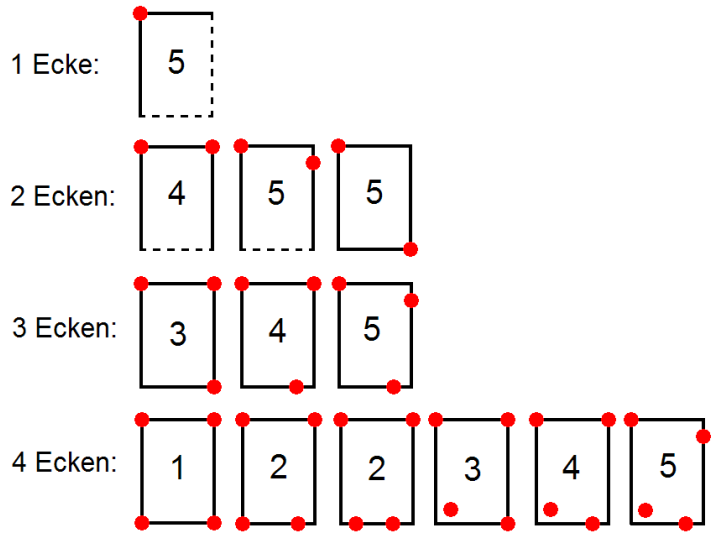


Abbildung 7.4: *Notensystem für rechteckige Objekte – eine bessere Note bedeutet eine höhere Verlässlichkeit. Für symmetrische Konfigurationen ist jeweils nur eine Möglichkeit angegeben.*

Dargestellt sind alle möglichen Konfigurationen wenn 1, 2, 3 oder 4 Ecken, die für ein Objekt gefunden wurden. Die empirisch definierten Noten von 5 bis 1 spiegeln die Verlässlichkeit der Objekte wider. Wenn es nicht möglich ist, den gesamten Umriss wiederherzustellen, so sind in Abbildung 7.4 Varianten mit Punktlinien gekennzeichnet. Je rechteckiger eine Variante ist, desto besser ist ihre Note. Es wurde definiert, dass zwei Punkte, z.B. in der Nähe der oberen rechten und linken Ecke, als auf einer Linie liegend gelten sollen, wenn die Differenz zwischen ihren (vertikalen) Koordinaten kleiner als 5 Pixel ist. 5 Pixel bedeuten ca. 5 cm auf der realen Fassade und berücksichtigen mögliche Verzerrungen (vor allem der Orthogonalität) des Bildes. Wenn die Differenz größer als 5 Pixel ist, dann ist die Wahrscheinlichkeit, diesen Fall zu akzeptieren, proportional zur Normalverteilung. Ein typisches Beispiel ist in Abbildung 7.5 a) dargestellt. Alle Objekte mit der Note 1 (grüne Rechtecke) beschreiben die reale Fensterlage gut.

Normalerweise haben falsche Objekte oder Störungen weniger als zwei erkannte Ecken (siehe z.B. Abbildung 7.6). Deshalb ist die Wahrscheinlichkeit hoch, dass ein Objekt mit vier oder drei erkannten Ecken ein Fenster ist. Die Validierung wird deswegen für das Beispiel in Abbildung 7.6 in zwei Stufen durchgeführt: Zuerst werden alle Objekte mit den Noten 2 und 3 (hellblaue Rechtecke) mit den Objekten mit der Note 1 (grüne Rechtecke) verglichen. D.h., es werden alle Objekte geprüft, die mindestens drei gut definierte Ecken haben (siehe Abbildung 7.4). Das Ziel ist, die Anzahl von korrekt erkannten Fenstern so weit wie möglich zu erhöhen. Der Vergleich wird mittels Bildzuordnung durchgeführt und das Entscheidungskriterium ist der *Kreuzkorrelationskoeffizient* (*CCC* siehe Abschnitt 2.3), mit einem hohen Schwellwert von 0,8. Für die Bestimmung der Lage, an der *CCC* den maximalen Wert annimt, wird *MCMC* (siehe Abschnitt 2.3) verwendet.

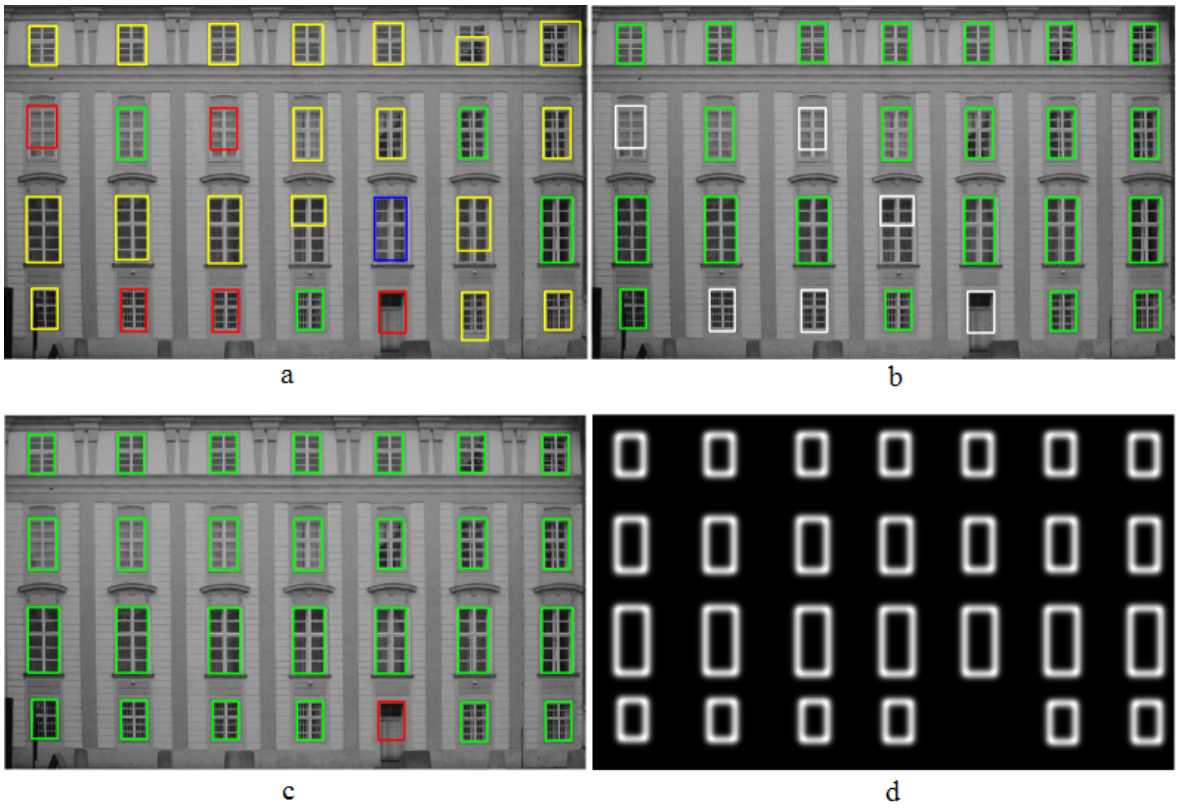
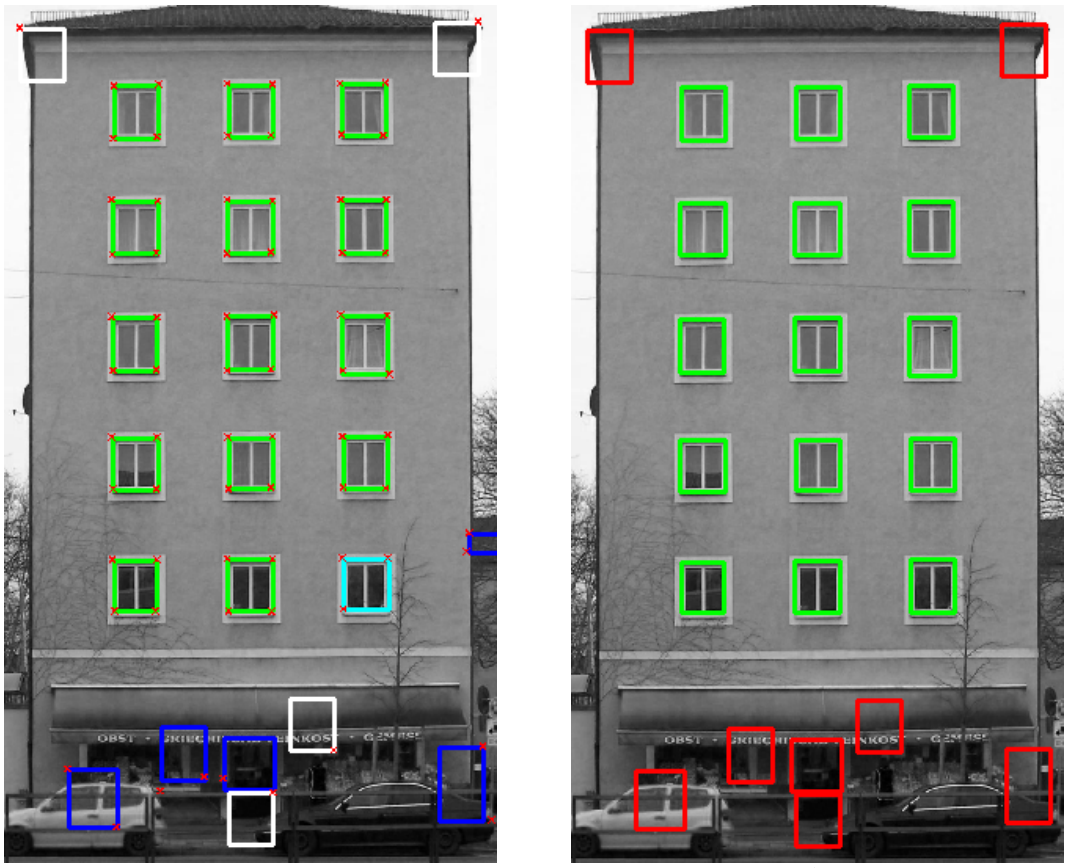


Abbildung 7.5: *a) Validierung der Objekte: Grün – Sehr gut (Note 1), Blau – Gut (Note 2), Gelb – Befriedigend (Note 3), Rot – Ausreichend (Note 4); b) Ergebnis nach dem Vergleich der Objekte mit den Noten 2 und 3 mit den Hypothesen mit Note 1; c) Endgültiges Ergebnis; d) Umrissbild.*

Eine große Anzahl von Experimenten, die in Kapitel 8 zusammengefasst sind, zeigt, dass ein Optimum, d.h. Maximum von CCC, des Markov Prozesses meist nach 400 Iterationen erreicht wird.

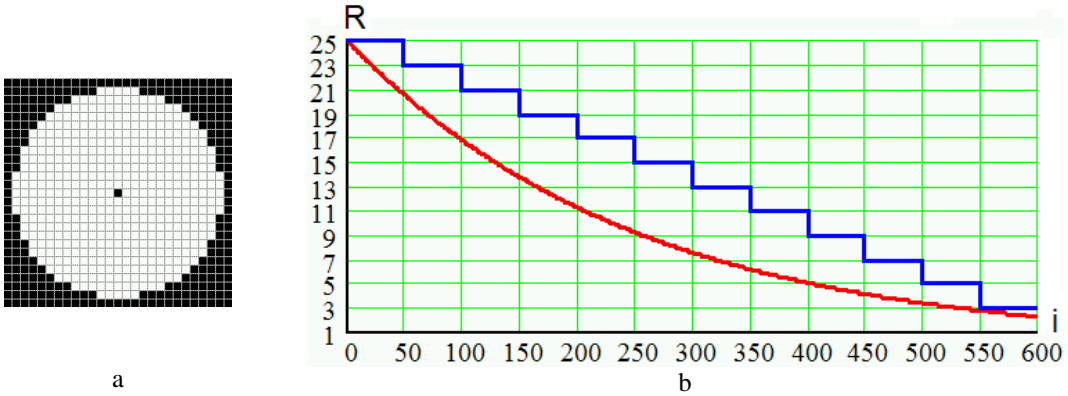
Im gegebenen Fall hängt der Markovprozess von der Verschiebung in der Ebene, d.h. von zwei Koordinaten ab. Deshalb werden bei jeder Iteration zufällig zwei ganze Zahlen gewählt, welche die Änderungen der kartesischen Koordinaten bestimmen. Diese Zahlen liegen im Interval $[-R, \ldots, R]$, mit R dem maximalen Verschiebungsradius. Am Anfang des MCMC Prozesses wird $R = 25$ Bildpunkte gesetzt. Um Isotropie zu garantieren, wird die Verschiebung durch eine Maske geprüft (siehe Abbildung 7.7 a)).

Der MCMC Prozess wird zusammen mit *Simulated Annealing* (siehe Abschnitt 2.3) in Form einer stufenförmigen Abkühlung verwendet (siehe Abbildung 7.7 b)). Das bedeutet Folgendes: Nach jeweils 50 Iterationen wird der maximale Radius R um 2 verkleinert. Experimente haben gezeigt, dass exponentielles *Simulated Annealing* keine Vorteile gegenüber der beschriebenen Methode hat, aber langsamer ist, weil es keine ganzen Zahlen verwendet. Zusätzlich wurde beim dem MCMC Prozess *Metropolis-Hastings Sampling* (siehe Abschnitt 2.3) genutzt, wodurch mit einer höheren Wahrscheinlichkeit das globale Maximum erreicht wird.



a b

Abbildung 7.6: *Fenstererkennung: a) Vor der Validierung. Rote Kreuze – Erkannte Ecken, Grüne Rechtecke – Sehr gut, Hellblaue – Ausreichend, Blau – Schlecht, Weiß – Sehr schlecht; b) Nach der Validierung. Grüne Rechtecke – akzeptierte Objekte, rote Rechtecke – abgelehnte Objekte.*



a b

Abbildung 7.7: *a) – Maske für den Verschiebungsvektor; b) Exponentielle Abkühlung – rot, stufenförmige Abkühlung – blau. R – maximaler Verschiebungsradius, i – Anzahl der Iterationen*

Z.B. für die in Abbildung 7.11 dargestellte Fassade wurden insgesamt 374 MCMC Prozesse mit einer durchschnittlichen Anzahl von 423 Iterationen benutzt. Von diesen haben nur 55 Prozesse den vorgegebenen Schwellwert 0,8 für den Keuzkorrelationskoeffizienten erreicht.

Ein Prozess startet immer an der vorgegebenen Näherung, d.h. im Zentrum des Bildausschnittes für das Beispiel in Abbildung 7.8, und läuft bis zum maximalen Wert des Keuzkorrelationskoeffizienten CCC. Die weißen Linien stellen die Verschiebungen dar, welche auf Grund einer positiven Differenz des CCC akzeptiert wurden. Rote Linien entsprechen Verschiebungen, welche mittels *Metropolis-Hastings Sampling* angenommen wurden, d.h. bei denen zwischenzeitlich ein schlechteres Ergebnis akzeptiert wurde. Für jede Gruppe j aus 50 Iterationen wurde der maximale Wert CCC_j gespeichert. Ein Prozess wird gestoppt, wenn entweder $|CCC_j - CCC_{j-1}| < 0,01$ beträgt oder die Anzahl der Iterationen $I = 1000$ erreicht. Die letzte Bedingung wurde verwendet, um einen Prozess zu stoppen, wenn kein Maximum für CCC größer als $0,8$ gefunden wurde. Insgesamt ist feststellbar, dass die Anzahl der akzeptierten Schritte oft ziemlich klein ist. Z.B. wurden in Abbildung 7.8 d) nur 5 Schritte von 100 Iterationen angenommen. Dabei wurde ein Zustand mit $CCC = 0,81$ gefunden.

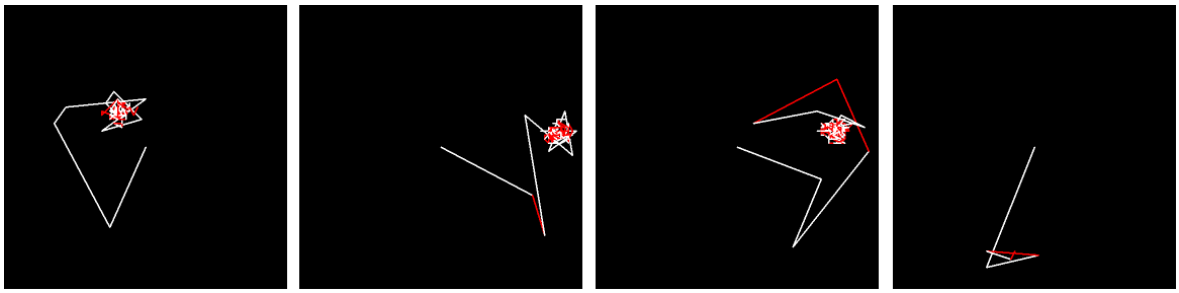


Abbildung 7.8: *Beispiele für MCMC Prozesse: a)* $CCC = 0,82$, $I = 500$*; b)* $CCC = 0,82$, $I = 600$*; c)* $CCC = 0,84$, $I = 600$*; d)* $CCC = 0,81$, $I = 100$

Abbildung 7.8 gibt auch einen Eindruck, wie *Simulated Annealing* funktioniert. Am Anfang des Prozesses sind die Verschiebungen relativ groß. Je länger ein Prozess läuft, desto kleiner werden die erlaubten Verschiebungen. Am Ende ist es nur noch 1 Pixel.

Jedes schwache Objekt wird mit allen bereits gefundenen Fenstern überprüft. In Abbildung 7.9 ist das Prüfungsschema für ein schwaches Objekt mit drei erkannten Ecken dargestellt.

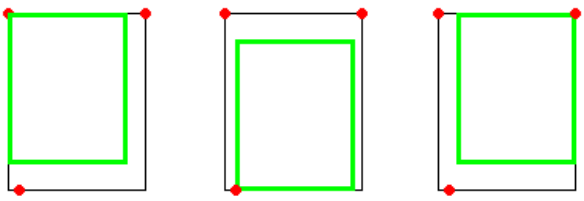


Abbildung 7.9: *Validierung eines schwachen Objektes. Rote Punkte – erkannte Ecken, grünes Rechteck – bereits akzeptiertes Fenster.*

Das grüne Rechteck zeigt ein bereits gefundenes Fenster. Wenn der CCC größer als 0,8 ist und das Fenster damit akzeptiert wird, erhält man als Ergebnis die Position des Objektes mit dem maximalen Wert des CCC. Der MCMC Prozess startet bei jeder erkannten Ecke. Alle akzeptierten Objekte bekommen die Note 1 (siehe Abbildung 7.5 b)). Nach dieser Stufe hat die Anzahl von korrekt erkannten Fenster oft deutlich zugenommen.

Im Weiteren werden die übrigen Hypothesen mit dem gleichen Verfahren geprüft. Der Schwellwert für den *CCC* wird auf zwischen 0,5 und 0,8 gesetzt. In der Abbildung 7.5 ist der Schwellwert für den $CCC = 0,8$, weil keine Störungen auf der Fassade auftreten.

Abbildung 7.10 zeigt die Detektion und Segmentierung eines teilweise verdeckten Fensters. Vor der Validierung wurde für das Fenster oben rechts (purpurrotes Rechteck) nur eine Ecke detektiert (a). Der Schwellwert für den CCC wurde auf 0.7 gesetzt.

a b

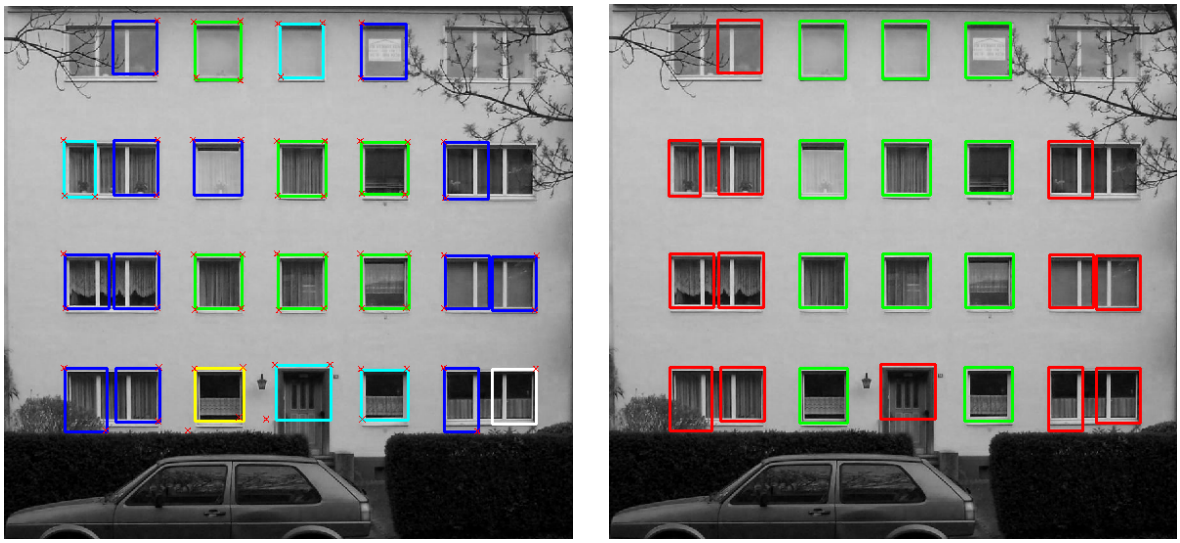
Abbildung 7.10: *a) Ergebnis vor der Validierung der Objekte; b) Endgültiges Ergebnis.*

In der Abbildung 7.11 ist ein Beispiel dargestellt, bei dem mittels ISM drei falsche Hypothesen und ein Objekt mit falscher Lage detektiert wurden (siehe rote Rechtecke in der Abbildung 7.11 a)). Nach der Validierung wurde das Fenster mit der falschen Lage korrekt erkannt. Die drei falschen Hypothesen wurden abgelehnt (siehe rote Rechtecke in der Abbildung 7.11 b)), weil die Fenster zu klein sind.

a b

Abbildung 7.11: *a) Ergebnis vor der Validierung der Objekte; b) Endgültiges Ergebnis.*

Zuletzt macht Abbildung 7.12 an einem Beispiel Schwächen des Ansatzes klar. Es wurden nur die Fenster erkannt, die in ähnlicher Form im Trainingsdatensatz vorhanden sind, was für die breiten Fenster nicht gilt. Für diese müssten spezielle Trainingsdatensätze mit entsprechenden Fenstern erstellt werden.

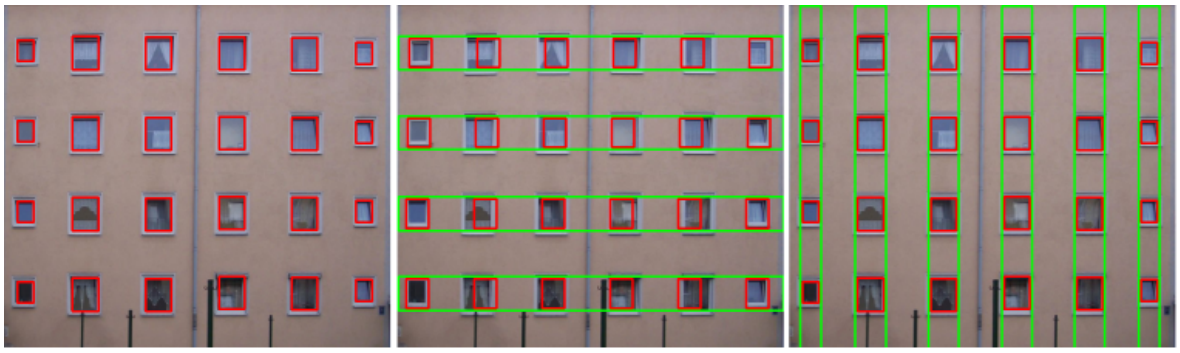


a b

Abbildung 7.12: *Fenstererkennung: a) Vor der Validierung. Rote Kreuze – Erkannte Ecken, Grüne Rechtecke – Sehr gut, Gelb – Gut, Cyan – Ausreichend, Blau – Schlecht; b) Nach der Validierung. Grüne Rechtecke – akzeptierte Objekte, rote Rechtecke – abgelehnte Objekte. Breite Fenster sind nicht im Trainingsdatensatz enthalten und werden deswegen nicht korrekt erkannt.*

7.3 Hierarchische Modellierung

Die vorhergehenden Abschnitte beschreiben, wie einzelne Fenster, wie z.B. in der Abbildung 7.13 a), ermittelt und abgegrenzt werden. Fenster auf einer Fassade sind allerdings oft nicht zufällig, sondern in Reihen, Spalten oder Gittern angeordnet. d.h. es ergibt sich eine Hierarchie der Modelle, wie dies in Abschnitt 4.1 diskutiert wurde und in Abbildung 7.13 dargestellt ist.



a b c

Abbildung 7.13: *Modellauswahl – Repräsentation der Fassade durch a) Einzelfenster; b) Fensterzeilen; c) -spalten. Letztere bestehen aus Fenstern gleicher Größe mit konstantem horizontalen bzw. vertikalen Abstand.*

Reihen und Spalten werden auf Grundlage der Analyse von horizontaler oder vertikaler Anordnung der Fenster gebildet. Allerdings ist häufig nicht klar, ob eine Fassade besser mittels einzelner Fenster oder durch Reihen oder Spalten von Fenstern repräsentiert wird. Z.B. zeigt die Abbildung 7.13 eine Konfiguration, welche gut mittels Spalten, aber nicht durch Reihen repräsentiert werden kann. Im allgemeinen Fall wird bezüglich der optimalen Anpassung an die Daten immer das Einzelfenstermodell bevorzugt, weil dieses auf Grund der großen Zahl an zur Beschreibung verwendeten Parametern am flexibelsten ist. Will man, dass Reihe oder Spalte gewählt werden, so benötigt man ein Kriterium, welches einerseits die Anpassung an die Daten, aber anderseits auch eine kleine Anzahl von Parametern honoriert.

Das vorgestellte Problem wird daher als ein Problem der Modellauswahl angesehen. Wegen seiner Einfachheit und guter Ergebnisse für den vorgestellten Ansatz wurde *Akaike's Information Criterion* – AIC eingesetzt, obwohl neue Arbeiten, wie z.B. (Geman et al. 2002) und (Wenzel et al. 2007), *Minimum Description Length* MDL bevorzugen (siehe Abschnitt 2.5). Im Besonderen wird

$$AIC = k - 2n \ln(L)$$

benutzt, mit k – Anzahl der Parameter des Modells, n – Anzahl der Beobachtungen und L – Wahrscheinlichkeit (Likelihood) der Umrisse der Fenster. Für Einzelfenster ist die Parameterzahl vier (Breite, Höhe und Koordinaten des Zentrums) mal Zahl der Fenster und für eine Reihe oder eine Spalte ist sie sechs (Parameter Einzelfenster plus Abstand und Zahl der Fenster). Die Likelihood wird in dem in Abschnitt 7.1 beschriebenen normalisierten Verteilungsbild (siehe z.B. Abbildung 7.3 b)) bestimmt. Abbildung 7.14 a) zeigt die Verteilung für eine Fassade zusamen mit den Umrissen in Form von roten Rechtecken. Jedes Pixel auf diesen Rechtecken stellt eine Beobachtung der Likelihood dar.

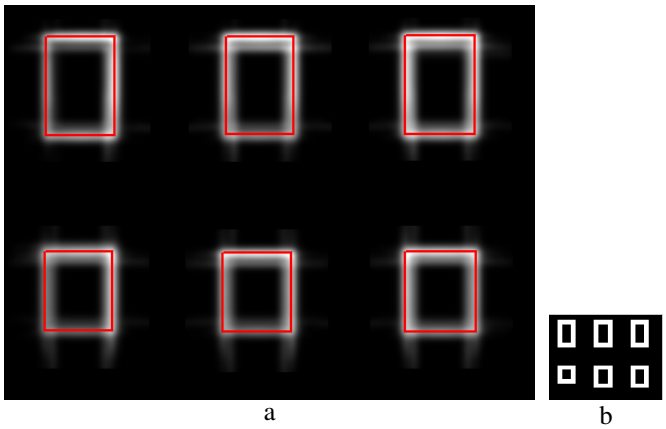


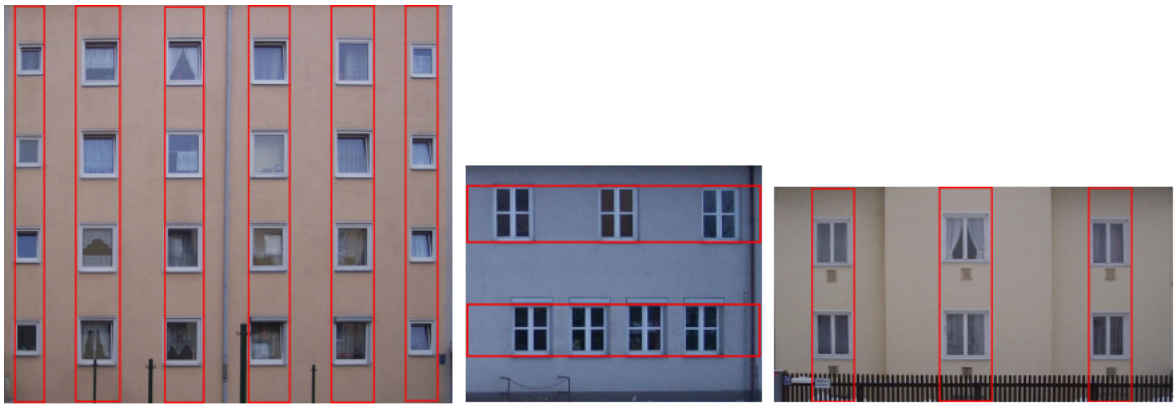
Abbildung 7.14: *Definition von Likelihood für den Fensterumriss – a) Verteilungsbild mit Fensterumrissen als rote Rechtecke; b) Auf minimale Größe reduziertes Verteilungsbild.*

Aus der Formel für AIC ergibt sich, dass die Anzahl der Beobachtungen, die maßstabsabhängig ist, eine große Bedeutung für die Likelihood und damit die Auswahl anhand der Modellkomplexität besitzt. Zahlreiche Experimente und deren Analyse (siehe Abschnitt 8.1) führten zu der Erkenntnis, dass die Ermittlung der Likelihood auf die minimale Umrisslänge bezogen werden muss. Das Sampling Theorem impliziert, dass für ein rechteckiges Objekt

die minimale Seitenlänge ca. drei Pixel betragen sollte. Das Verteilungsbild wird dementsprechend zur Berechnung der Likelihood für AIC auf diese minimale Größe reduziert (siehe Abbildung 7.14 b)).

Der Umriss für Einzelfenster bzw. für Fenster in Reihen oder Spalten im Verteilungsbild wird mittels MCMC bestimmt, indem die vier mal Fensteranzahl bzw. sechs Parameter variiert werden. Analog zum Vorgehen in der Objektverifikation im letzten Abschnitt 7.2 wird wiederum *Simulated Annealing* sowie der *Metropolis-Hastings Algorithm* verwendet.[1] Um eine höhere Genauigkeit zu erzielen, wird für die Bestimmung des Umrisses die ursprüngliche Auflösung des Verteilungsbildes verwendet.

Ergebnisse für Modellauswahl und damit eine hierarchische Modellierung mit Zeilen und Spalten sind in der Abbildung 7.15 dargestellt. In allen drei gegebenen sowie vielen anderen Bildern, mit denen das Verfahren geprüft wurde, wurde das korrekte Modell gewählt. Wenn es eine offensichtliche Struktur auf der Fassade gibt, spiegelt sie sich in erheblich unterschiedlichen AIC Werten (siehe Abbildung 7.15) wider.



a – **S: 265**, R: 404, E: 359 b – S: 115, **R: 65**, E: 94 c – **S: 76**, R: 83, E: 87

Abbildung 7.15: *Modellauswahl unter Verwendung von AIC – S: Spalten, R: Reihen, E: Einzelfenster; gewählte Modelle sind fett dargestellt.*

Die bisher vorgeschlagene Modellauswahl funktioniert nur für ausreichend regulär auf der Fassade organisierte Fenster. Sie arbeitet nicht zuverlässig für nur teilweise regelmäßig in Spalten oder Reihen angeordnete Fenster. Hierfür wird im Folgenden eine Erweiterung vorgestellt: Regelmäßige Strukturen innerhalb einer Spalte oder einer Reihe werden dafür als *Fensterketten* bezeichnet. Eine Fensterkette wird als eine Menge von Fenstern definiert, die innerhalb einer Spalte oder Reihe liegen und die aus gleichgroßen Fenstern mit dem gleichen Abstand zwischen den Fenstern bestehen. Die Suche wird zufällig mit einem Fenster innerhalb einer Spalte/Reihe gestartet. Es wird geprüft, ob ein benachbartes Fenster zusammen mit dem gewählten Fenster einer Kette entspricht. Die optimale Anpassung der Fenster wird wiederum mittels MCMC durchgeführt. Dieser Prozess wird wiederholt, bis alle benachbarten Fenster geprüft sind. Im allgemeinen Fall gibt es, abhängig vom ersten geprüften

[1] Eine graphische Darstellung für mehrdimensionales MCMC ist kompliziert. Eine Vorstellung über den Verlauf des Prozesses gibt die Abbildung 7.8.

Fenster, u.U. eine Menge von unterschiedlichen Ergebnissen (siehe z.B. Abbildung 7.16). Deshalb wird für jede Reihe oder Spalte die Suche mehrmals zufällig gestartet und am Ende das Ergebnis mit dem niedrigsten AIC ausgewählt (hier Abbildung 7.16 a)).

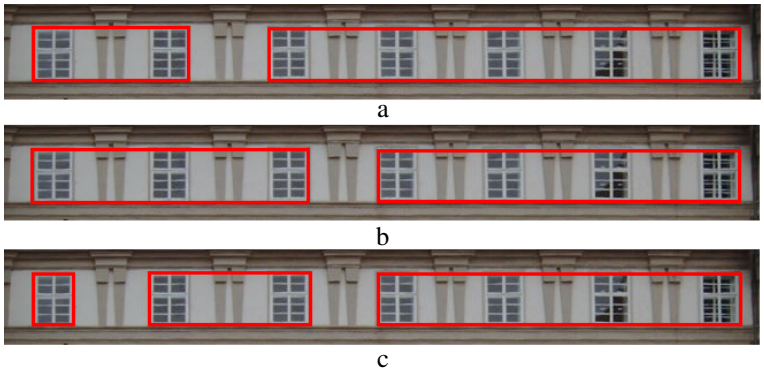


Abbildung 7.16: *Verschiedene Varianten von Fensterketten. Variante a) ist optimal.*

Das Ergebnis für die gesamte Fassade ist in Abbildung 7.17 dargestellt. Nach der Fensterdetektion (siehe Abbildung 7.5 c) und d)) wurde das Einzelfenstermodell gewählt, weil die Abstände zwischen den Fenstern z.T. nicht gleich sind. Die Suche nach Fensterketten innerhalb der Reihen führte zur Beschreibung dieser Fassade mit 54 gegenüber 108 Parametern für Einzelfenster.

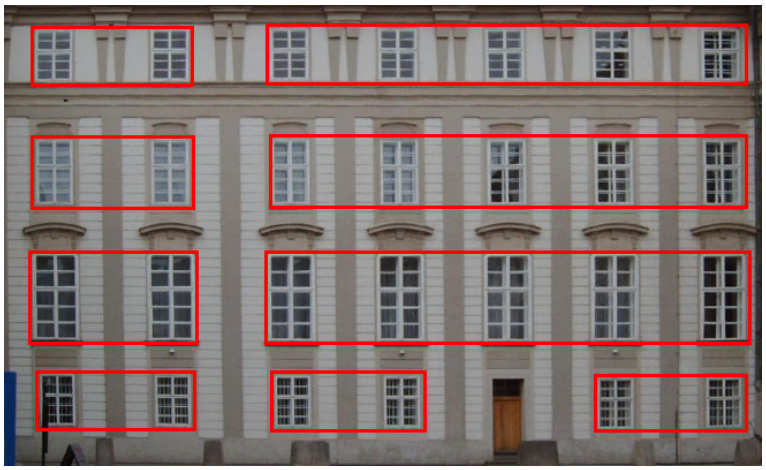


Abbildung 7.17: *Ergebnis für die Modellauswahl bei Suche nach Fensterketten.*

7.4 Bestimmung der 3D Struktur

Wenn Bildsequenzen als Datengrundlage verwendet werden, dann ist die dreidimensionale (3D) Lage der Fenster relativ zur Fassadenfläche bestimmbar. Als Grundlage für die Bestimmung der 3D Struktur wird der *Plane Sweeping* Ansatz von (Baillard und Zisserman 1999)

bzw. (Werner und Zisserman 2002) verwendet (siehe Kapitel 3). Für ein Fenster bzw. eine Reihe oder Spalte werden das / die 3D Rechteck(e) auf der bekannten Fassadenebene ausgeschnitten. Die Rechtecke werden in Richtung der Normalen der Fassadenfläche verschoben ("gesweept"). Die Ermittlung der Tiefe basiert auf der Summe der kleinsten Quadrate der Differenzen zwischen den Projektionen der Fläche(n) in die einzelnen Bilder und ihrem Durchschnittbild. Diese Summe wird für einen sinnvollen Tiefenbereich berechnet. Das Resultat ist die Tiefe für das Minimum der Summe. Für Zeilen oder Spalten werden die Beiträge der Fenster für eine Zeile oder Spalte für eine bestimmte Tiefe aufaddiert. Abbildung 7.18 zeigt automatisch segmentierte und orthogonalisierte Fassadenbilder.

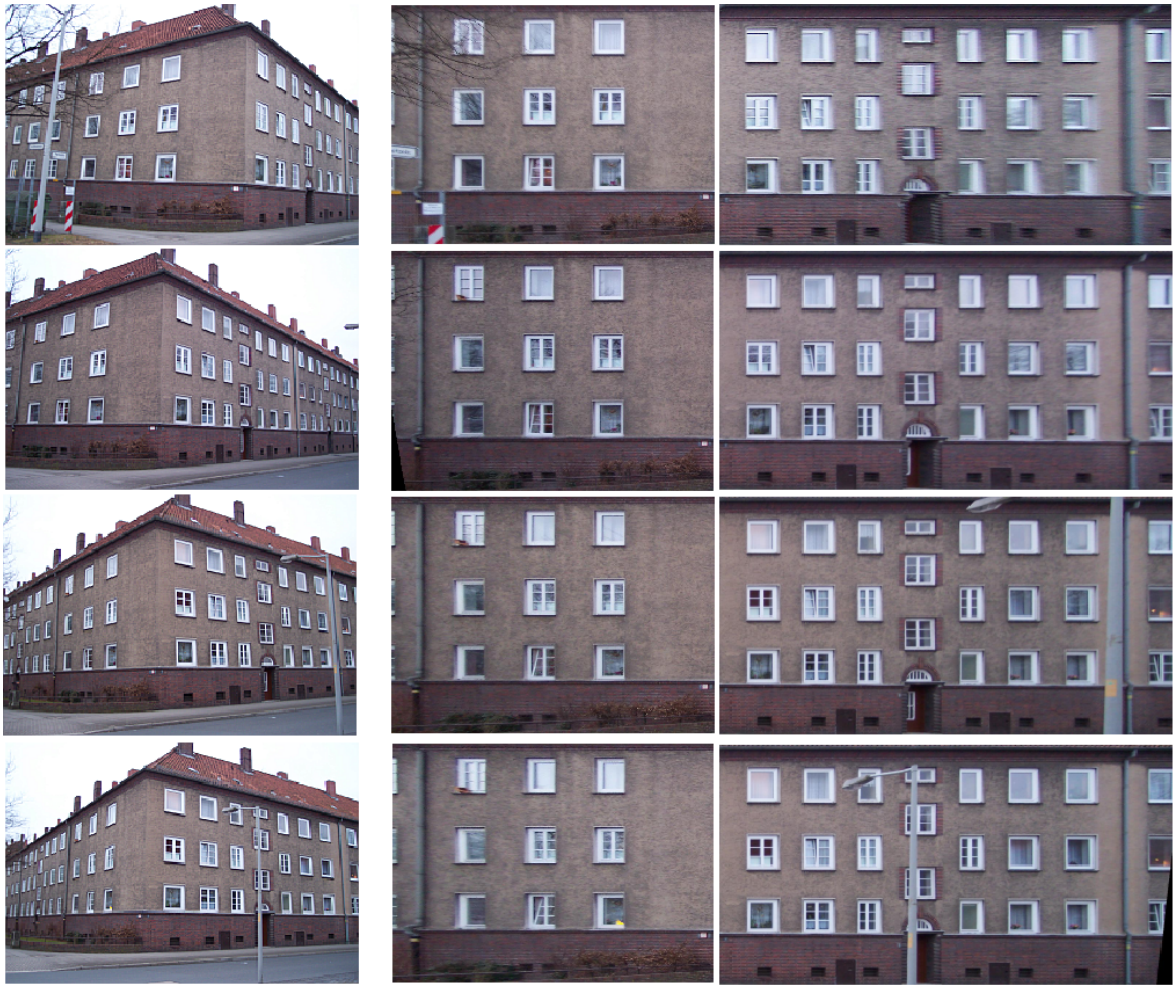


Abbildung 7.18: *links – Bildsequenz; rechts – automatisch orthogonalisierte Fassadenbilder.*

Die Bilder der Fassade in Abbildung 7.18 wurden durch einen Medianfilter kombiniert. Das Ergebnis ist in Abbildung 7.19 dargestellt. Es enthält die Information, welche die Mehrheit der Bilder zeigt. Damit werden die meisten Störungen entfernt, welche sich zwischen Kamera und Fassade befinden, wie z.B. die Straßenlaterne in Abbildung 7.18 unten rechts.

In Abbildung 7.19 ist rechts ein einzelnes Fenster dargestellt. Man kann sehen, dass die Bereiche, die auf der Fassadenebene liegen, scharf abgebildet sind. Dagegen passen im Bereich des Fensters die projizierten Bilder nicht zueinander, weil sie nicht auf der Fassadenebene

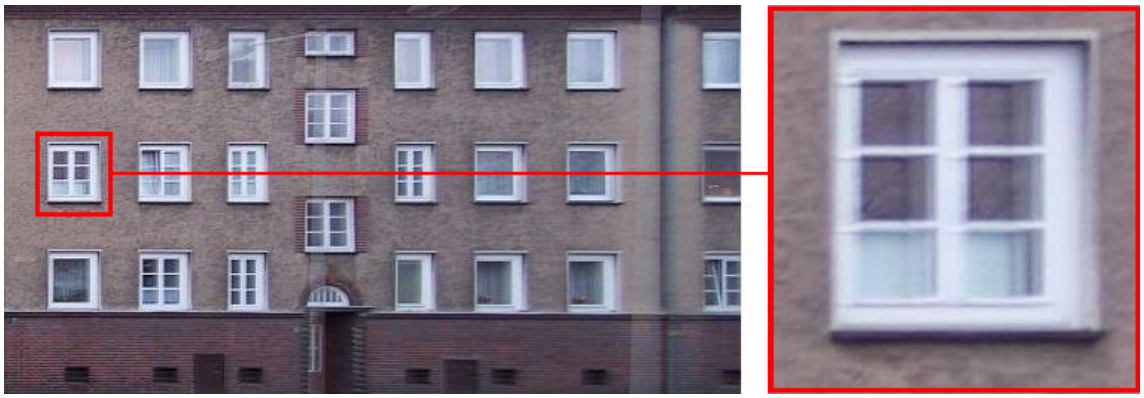


Abbildung 7.19: *Links – Kombination der Bilder durch Medianfilter; rechts – vergrößertes Fenster.*

liegen. Die Idee von *Plane Sweeping* ist es, eine virtuelle Ebene zu erstellen, die hier parallel zur Fassadenebene bewegt wird. Die Aufgabe besteht darin, die Lage der Ebene zu finden, bei der die Bildteile innerhalb des Fensters zueinander passen. Die Differenz zwischen der Fassadenebene und der virtuellen Ebene ist die Tiefe des Fensters relativ zur Fassadenebene. Die Tiefenschätzung ist in Abbildung 7.20 skizziert.

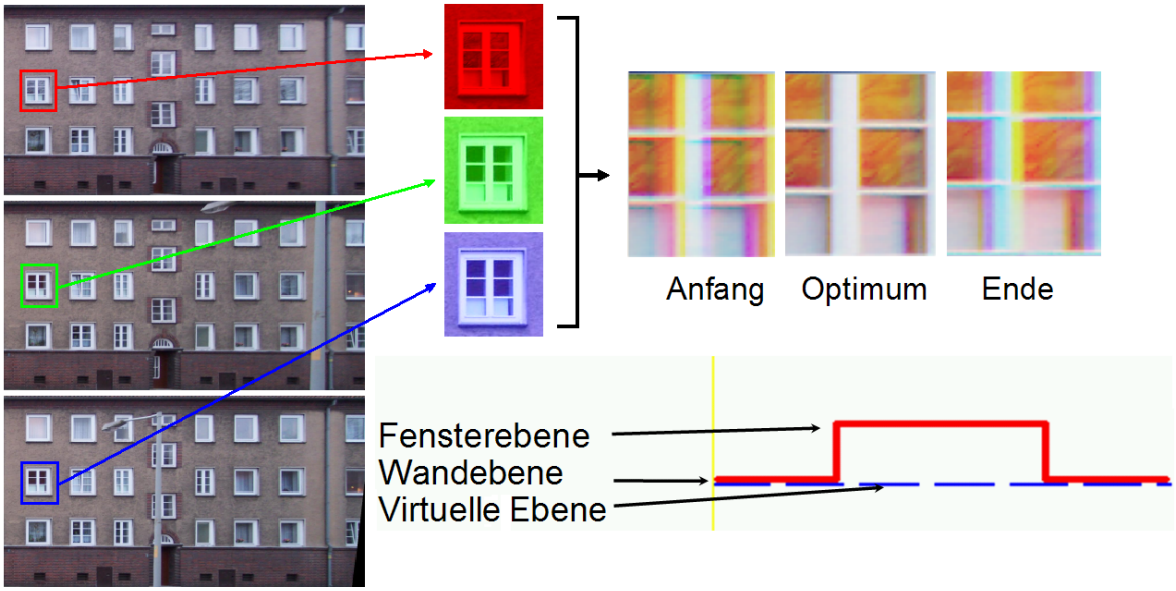


Abbildung 7.20: *Planesweeping demonstriert an drei Bildern. Die Helligkeitswerte der drei Bilder wurden für eine gute Visualisierung den drei Farbkanälen Rot, Grün und Blau zugeordnet. Unterschiede zeigen sich damit in Form von Farben.*

Für die Berechnungen wurden Grauwertbilder benutzt, die für eine bessere Sichtbarkeit für die Graphik den drei Farbkanälen rot, Grün und Blau zugeordnet wurden. Die virtuelle Ebene wird in kleinen Stufen parallel zur Fassadenebene verschoben. Der Abstand, bei dem die Standardabweichung σ der Bilder minimal und das Bild am schärfsten ist, entspricht der Tiefe des Fensters.

Für die Tiefenschätzung werden aus einer Bildsequenz automatisch mindestens drei Bilder gewählt, welche einen räumlichen Winkel zwischen der Verbindungslinie von Kameraposition und Fassadenmitte und der Normalen der Fassadenebene haben, der kleiner als 90° ist (siehe Abbildung 7.21). D.h., es werden nur Bilder verwendet, in denen die Fassade sichtbar ist.

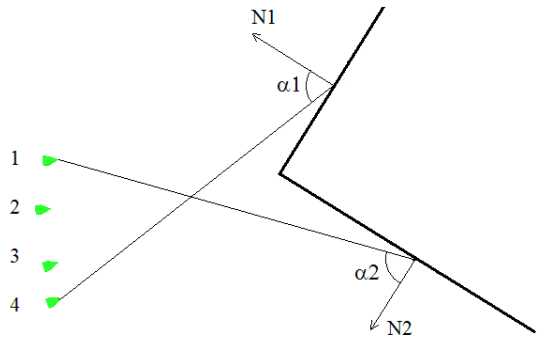


Abbildung 7.21: *Winkel zwischen Verbindungslinien von Kameraposition und Fassadenmitte zu den Normalen der Fassadenebenen N1 und N2. Die Kamerapositionen sind als grüne Pyramiden dargestellt.*

In der Abbildung 7.22 ist das Ergebnis für das 3D Modell (Bilder siehe Abbildung 7.18) dargestellt.

a b

Abbildung 7.22: *a) Fensterumrisse für zwei Fassaden als rote Rechtecke, 3D Fensterpositionen als grüne Rechtecke und Kamerapositionen als grüne Pyramiden; b) Detail für eine Fassade.*

Kapitel 8

Experimente und Evaluierung

Das Kapitel 8 stellt zuerst Experimente zur Modellauswahl mittels AIC vor. In Abschnitt 8.2 werden Ergebnisse für die 3D Modellierung präsentiert. Die Evaluierung von Ergebnissen in Abschnitt 8.3 führt zu deren Bewertung in Abschnitt 8.4.

8.1 Modellauswahl mittels AIC

In diesem Abschnitt werden Experimente zur Modellauswahl beschrieben. Das Ziel ist die Überprüfung der Formel für *Akaike's Information Criterion* (AIC):

$$AIC = k - 2n \ln(L)$$

mit k – *Anzahl der Parameter des Modells*, n – *Anzahl der Beobachtungen* und L – *Wahrscheinlichkeit (Likelihood)*. Die Formel besteht aus zwei Summanden: Der erste steht für die Einfachheit des Modells und der zweite für die Anpassung des Modells an die Daten. Der Parameter n ist ein Normierungskoeffizient und beschreibt die *Anzahl der Beobachtungen*. Insbesonders wird im Weiteren die sich aus einer Vielzahl von Experimenten ergebende Annahme geprüft, dass n auf die minimale Umrisslänge bezogen werden muss (siehe Abschnitt 7.3).

Alle dargestellten Ergebnisse wurden mit Hilfe von Umrissbildern erzielt (siehe z.B. Abbildung 7.14), die mittels Implicit Shape Models – ISM (siehe Abschnitt 2.1) erstellt wurden. Um eine gute Bewertbarkeit der Qualität zu gewährleisten, werden die Ergebnisse im Weiteren zusammen mit den orthogonalisierten Originalbildern dargestellt.

Als Grundlage für die Modellauswahl werden neben dem Umrissbild die genäherten Koordinaten der Fenster benutzt. Die Analyse der Fensterkoordinaten führt zum *Einzelfenster-*, *Spalten-* und *Zeilenmodell*.[1] Nach der Optimierung der Modelle mittels MCMC (siehe z.B. Abbildung 8.2) wird für jedes Modell der AIC berechnet. Zuletzt wird das Modell mit dem minimalen AIC gewählt.

[1] Definitionen der Modelle siehe Abschnitt 4.1.

Im Weiteren bezeichnen L_E – Likelihood für das Einzelfenstermodell, L_S – Likelihood für das Spaltenmodell und L_R – Likelihood für das Reihenmodell. In den Graphiken wird das Einzelfenstermodell als *schwarze* Linie, das Spaltenmodell als *rote* Linie und das Reihenmodell als *blaue* Linie dargestellt. Als vertikale *grüne* Linie wird der Wert für n dargestellt, der der minimalen Umrisslänge für diese Fassade entspricht.

Für die dargestellten Experimente wurden nur Fassaden gewählt, bei denen das gewählte, d.h. optimale, Modell nicht die optimale *Anzahl der Parameter* k hat. D.h. *AIC* hat den minimalen Wert, obwohl es ein Modell für diese Fassade mit kleinerem k gibt.

Für eine verbesserte Darstellung wird das *relative Akaike's Information Criterion RAIC* definiert:

$$RAIC_i(n) = \frac{AIC_i(n)}{AIC_E(n)}$$

mit $i = (S, R, E)$ – Spalten, Reihen oder Einzelfenster. Die graphischen Darstellungen für RAIC erfolgt in Form von Geraden. Im Weiteren wird $RA_E(n) = 1$ definiert, d.h. die Gerade für das Einzelfenstermodell liegt horizontal (siehe die schwarzen Linien in den Abbildungen 8.1 d), 8.2 d) und 8.3 d)).

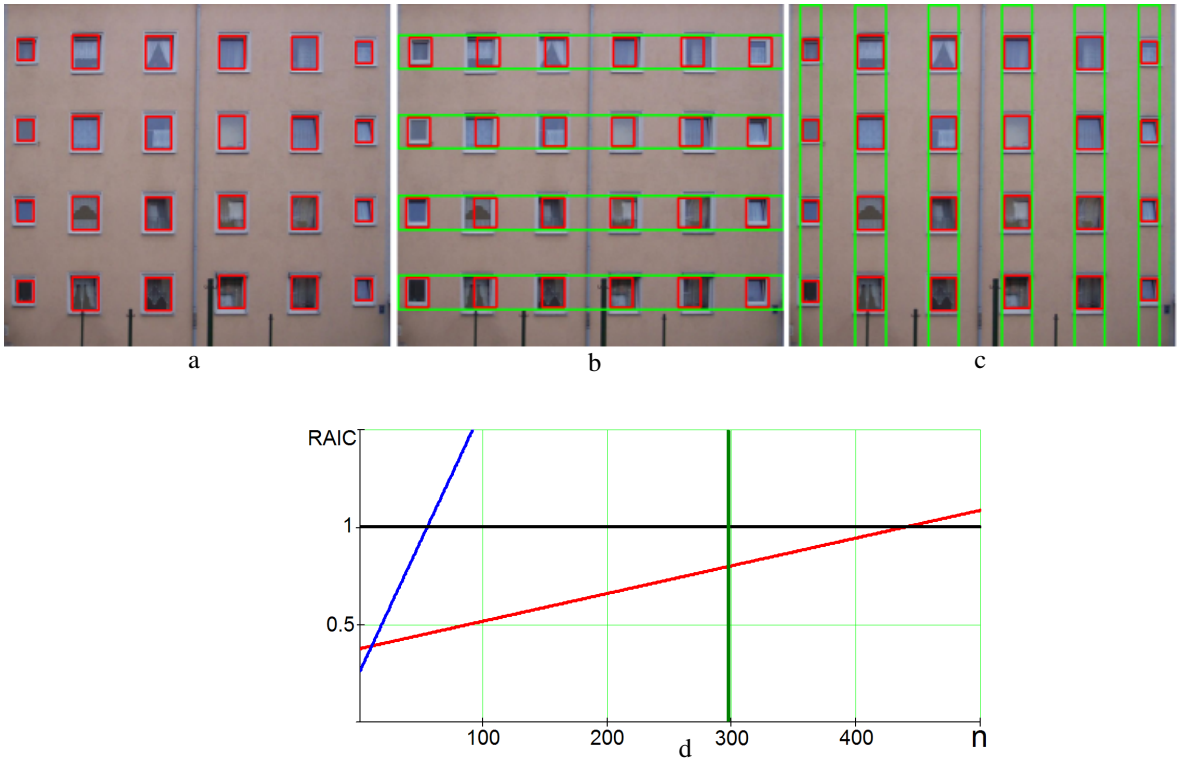


Abbildung 8.1: *Ergebnis der Optimierung mittels MCMC a) Einzelfenstermodell, Anzahl Parameter für Modell* $k = 96$*; b) Spaltenmodell,* $k = 36$*; c) Zeilenmodell,* $k = 24$*. d) Abhängigkeit RAIC von n (Einzelfenster – schwarz, Spalten – rot, Reihen – blau, minimale Umrisslänge – grün)*

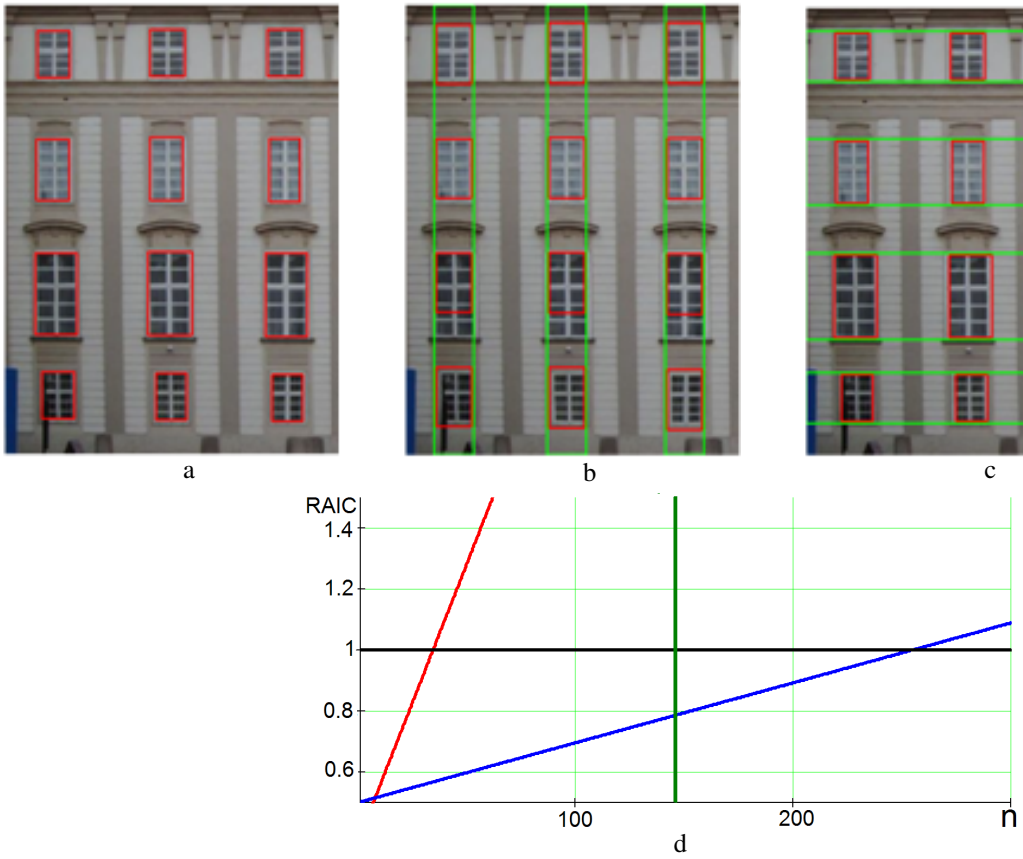


Abbildung 8.2: *Ergebnis der Optimierung mittels MCMC a) Einzelfenstermodell, $k = 48$; b) Spaltenmodell, $k = 18$; c) Zeilenmodell, Anzahl von Parameter für Modell $k = 24$. d) Abhängigkeit RAIC von n (Farben siehe Abbildung 8.1)*

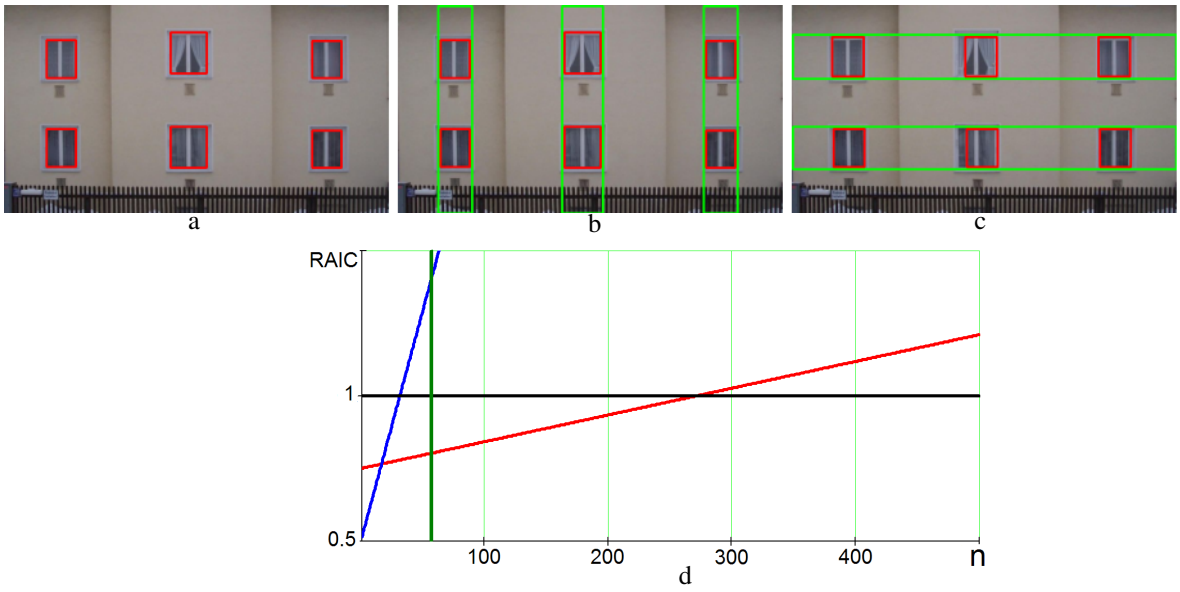


Abbildung 8.3: *Ergebnis der Optimierung mittels MCMC a) Einzelfenstermodell, $k = 24$; b) Spaltenmodell, $k = 18$; c) Zeilenmodell, Anzahl von Parameter für Modell $k = 12$. d) Abhängigkeit RAIC von n (Farben siehe Abbildung 8.1)*

Die in den Abbildung 8.1, 8.2 und 8.3 dargestellten Ergebnisse der Experimente zeigen, dass es einen Wertebereich für n gibt, bei dem die Formel für den AIC korrekte Ergebnisse für die Modellauswahl liefert. In diesem Wertebereich liegt auch die *minimale Umrisslänge* für die *Anzahl von Beobachtungen*. Deswegen wurde diese für die Modellauswahl in dieser Arbeit verwendet.

8.2 Ergebnisse für die 3D Modellierung

Die Abbildung 8.4 zeigt vier Bilder einer Sequenz mit sieben Bildern.

Abbildung 8.4: *Bilder Nummer eins, drei, fünf und sieben der Sequenz Ostbahnhof-1.*

Abbildung 8.5 stellt das Resultat für die Fensterextraktion und 3D Bestimmung für drei manuell grob abgegrenzte Fassadenbereiche ohne Erdgeschoss dar.

Abbildung 8.5: *Ergebnis für die Sequenz Ostbahnhof-1 (Bilder siehe Abbildung 8.4) – Fensterumrisse für drei Fassaden mit Fensterzeilen als rote Rechtecke, 3D Fensterpositionen als grüne Rechtecke und Kamerapositionen als grüne Pyramiden.*

In Abbildung 8.6 ist das Ergebnis für zwei weitere grob abgegrenzte Fassaden zu sehen. Abbildung 8.7 zeigt zwei weitere grob abgegrenzte Fassaden jeweils mit Erdgeschoss.

Abbildung 8.6: *Ergebnis für die Sequenz Ostbahnhof-2 mit zehn Bildern und Fensterspalten – Erläuterungen siehe Abbildung 8.5.*

Abbildung 8.7: *Ergebnis für die Sequenz Bordeauxplatz mit elf Bildern und Einzelfenstern – Erläuterungen siehe Abbildung 8.5.*

Die Fenster sind in den Abbildungen 8.5, 8.6 und 8.7 jeweils durch rote Vierecke gekennzeichnet (siehe auch Abbildung 8.8). Während sich die Modellauswahl bei den drei Fassaden des ersten Beispiels für Fensterreihen entschied, fiel die Auswahl für die zwei Fassaden des zweiten Beispiels auf Fensterspalten und für die zwei Fassaden des dritten Beispiels auf Einzelfenster. Zu beachten ist, dass Reihen- und Spaltenmodelle aus Fenstern mit gleicher Form und Abstand entweder in horizontaler oder vertikaler Richtung bestehen und die Entscheidung für die vollständige Fassade durchgeführt wurde. D.h., die in Abschnitt 7.3 vorgeschlagenen Fensterketten kamen hier nicht zum Einsatz. Die 3D Rekonstruktion war in großen Teilen zuverlässig und bestimmt die Fenster hinter der Fassade (siehe auch grüne Rechtecke in der Abbildung 8.8).

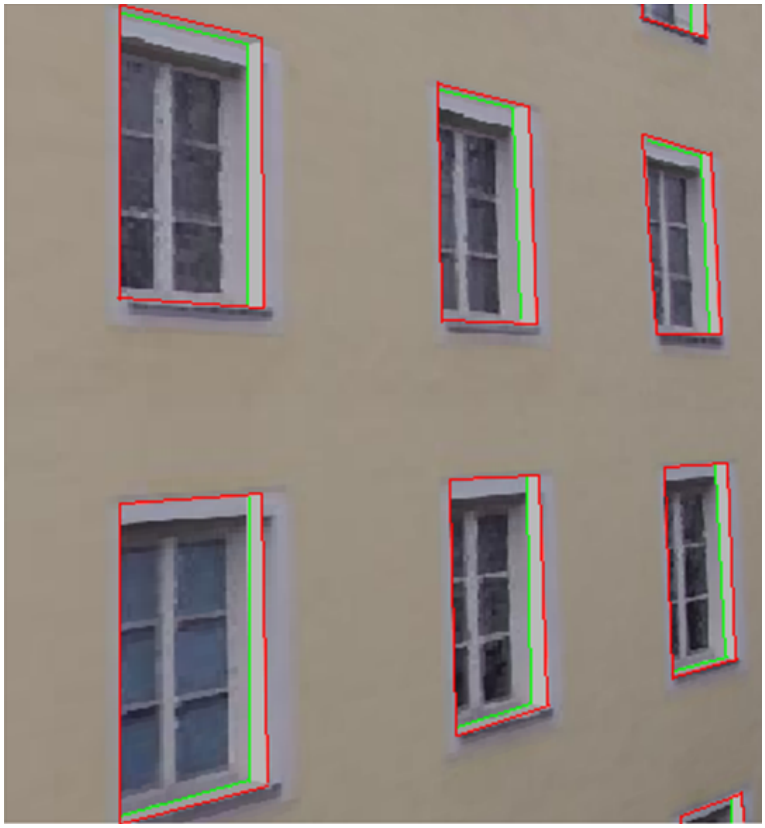
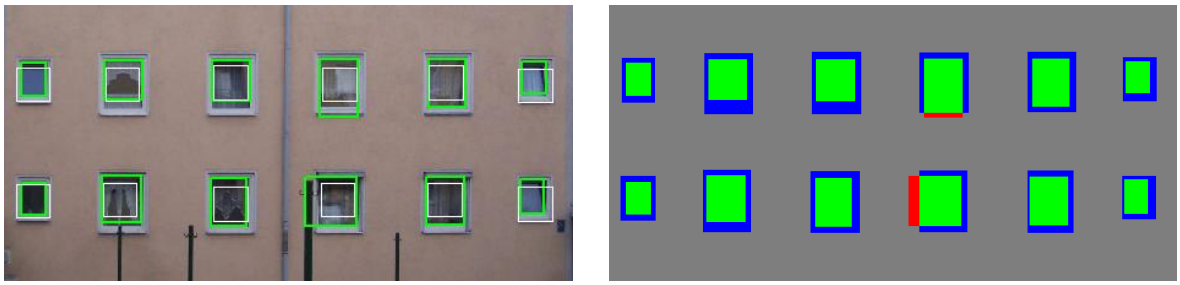


Abbildung 8.8: *Detail einer Fassade der Sequenz Ostbahnhof-1 (siehe Abbildung 8.5)*

8.3 Evaluierung von Ergebnissen

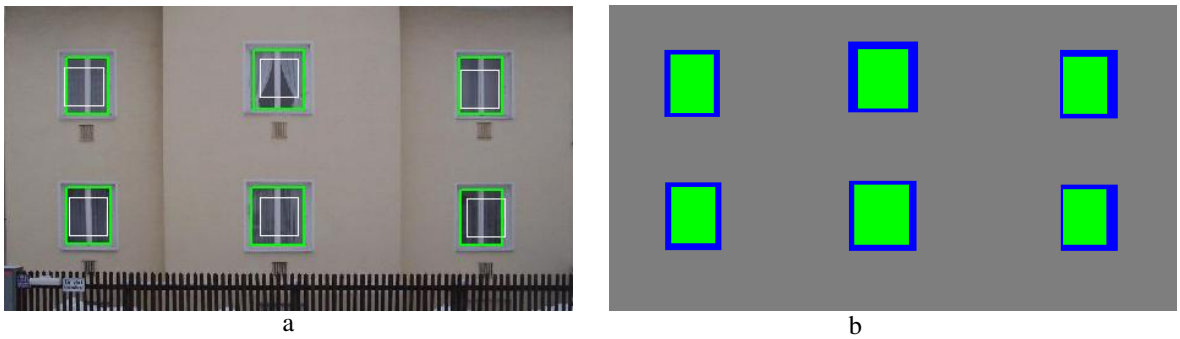
In diesem Abschnitt ist die Evaluierung der Qualität der Fenstersegmentierung mittels Grauwert-Morphologie und mittels ISM für zwei Fassaden dargestellt.

Das Resultat der Evaluierung der Ergebnisse für die Fensterextraktion mittels Grauwert Morphologie zeigen Abbildungen 8.9 und 8.10. Grundlage für die Bewertung sind manuell extrahierte Rechtecke für die Fenster (Referenz). In den Abbildungen entsprechen grüne Pixel korrekt extrahierten Fensterpixeln, blaue Pixel der Referenz und rote Pixel inkorrekt extrahierten Fensterpixeln.



a b

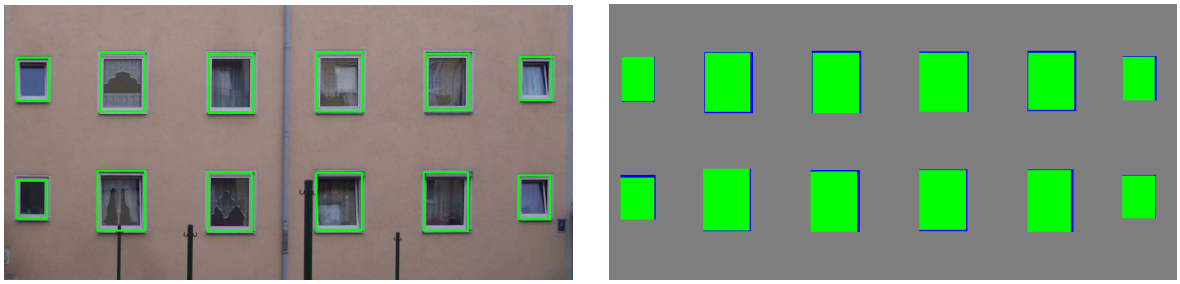
Abbildung 8.9: *Ergebnisse für Fassade 1: a) Start von MCMC (weiße Rechtecke) und Endergebnis (grüne Rechtecke); b) Evaluierung (korrekt – grün, Referenz – blau, inkorrekt – rot)*



a b

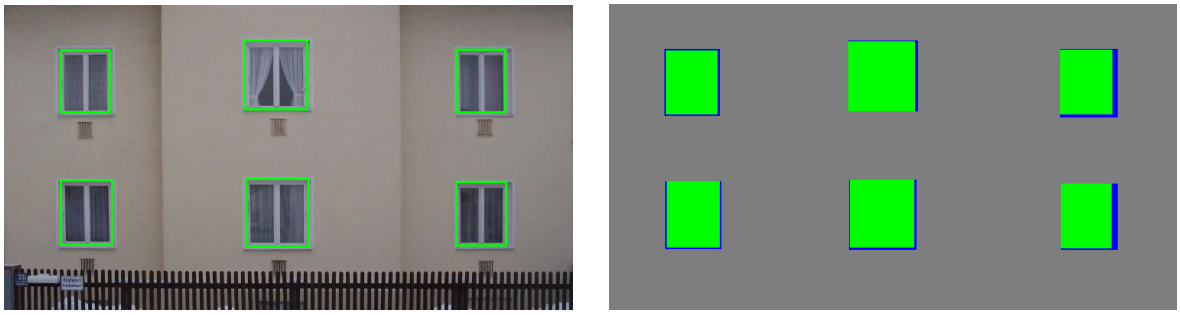
Abbildung 8.10: *Ergebnisse für Fassade 2: a) Start von MCMC (weiße Rechtecke) und Endergebnis (grüne Rechtecke); b) Evaluierung (Farben siehe Abbildung 8.9)*

Die Evaluierung der Ergebnisse für die Segmentierung der Fenster mittels ISM für die beiden selben Fassaden ist in den Abbildungen 8.11 und 8.12 dargestellt.



a b

Abbildung 8.11: *Ergebnisse für Fassade 1: a) Segmentierung durch ISM (grüne Rechtecke); b) Evaluierung (Farben siehe Abbildung 8.9)*



a b

Abbildung 8.12: *Ergebnisse für Fassade 2: a) Segmentierung durch ISM (grüne Rechtecke); b) Evaluierung (Farben siehe Abbildung 8.9)*

Die Tabellen 8.1 und 8.2 stellen die quantitative Bewertung im Sinn von (Heipke et al. 1998) dar. Hierbei gilt

$$\text{Korrektheit} = \frac{\text{Zahl korrekt als Fenster extrahierter Pixel}}{\text{Zahl aller als Fenster extrahierter Pixel}}$$

$$\text{Vollständigkeit} = \frac{\text{Zahl korrekt als Fenster extrahierter Pixel}}{\text{Zahl aller Fensterpixel der Referenz}}$$

Tabelle 8.1: *Evaluierung der Ergebnisse für die Abbildungen 8.9 und 8.10*

	Korrektheit	Vollständigkeit
Fassade 1	97%	93%
Fassade 2	100%	95%

Tabelle 8.2: *Evaluierung der Ergebnisse für die Abbildungen 8.11 und 8.12*

	Korrektheit	Vollständigkeit
Fassade 1	100%	99%
Fassade 2	100%	97%

Es ist erkennbar, dass für die dargestellten Fassaden die Qualität bei der Verwendung von Grauwert-Morphologie deutlich schlechter ist. Dafür gibt es folgende Gründe:

- Die Grauwert-Morphologie eignet sich nur für eine beschränkte Anzahl von Fällen, nämlich hochqualitative Bilder ohne Störungen: Der Kontrastunterschied zwischen dem Fassadenhintergrund und den Fensterbereichen muss groß sein, anderenfalls sind die Fensterbereiche nicht vom Hintergrund trennbar. Außerdem ist dieses Verfahren auf Fenster beschränkt, die wenig interne Strukturen besitzen.
- Die erfolgreiche Segmentierung mittels ISM funktioniert deutlich besser, weil sie für die Extraktion nur markante Objektteile verwendet. Viele Störungen, z.B. durch Strukturen innerhalb des Objektes, haben damit wenig Auswirkung auf den Umriss.

8.4 Bewertung der Ergebnisse

Das in dieser Arbeit beschriebene Verfahren für die hierarchische 3D Modellierung von Fassaden hat folgende Stärken:

- Als Eingangsdaten werden terrestrische *un-* oder *schwach kalibrierte* Wide-Baseline Bildsequenzen von Gebäuden benutzt. Durch Analyse werden die vertikalen Fluchtpunkte und evtl. die Kalibrierung der Kamera automatisch bestimmt. Auch die Fassadenebenen werden *automatisch detektiert, segmentiert und orthogonalisiert*. Die Fenster werden *automatisch aussehensbasiert* mittels *Implicit Shape Models (ISM) segmentiert*. *Regularitäten* der Fensterpositionen werden *in Form von Zeilen, Spalten oder Fensterketten* beschrieben. Die *3D Lage der Fenster* wird relativ zur Fassadenebene *automatisch bestimmt*. Das Verfahren erlaubt es auch, z.B. von Vegetation, *teilweise verdeckte Fenster zu detektieren*.

Zu den Schwächen dieses Verfahrens gehören:

- Das 3D Modell wird nur für die Fassaden, d.h. ohne Dächer generiert, weil die Dächer meistens vom Boden aus nicht oder kaum sichtbar sind. Das Verfahren ist derzeit auf die Fenstererkennung begrenzt. Andere Objekte, wie z.B. Türen, Balkone, Säulen, Regenfallrohre, wurden nicht modelliert, weil sie eine zu komplizierte Form und / oder eine schwer zu prognostizierende 3D Lage besitzen. Zu den Schwächen gehören auch die notwendige manuelle Auswahl der Fensterart, z.B. *Modern*, *Klassisch* oder *Bogenfenster* und die manuelle Normierung der orthogonalisierten Fassadenbilder auf einen Maßstab.

Im Vergleich zu den in Kapitel 3 vorgestellten bisherigen Arbeiten ergeben sich folgende Defizite aber auch Stärken:

- Auch (Dick et al. 2004) benutzen terrestrische Bildsequenzen architektonischer Szenen, um 3D Modelle zu generieren. Es werden nicht nur Fenster, sondern auch Türen, Säulen und Regenfallrohre statistisch mittels RJMCM modelliert. Dagegen werden keine Regularitäten der Objekte bestimmt und auch die Tiefen der Fenster werden nicht geschätzt.

- Der Ansatz von (Alegre und Dallaert 2004) verwendet eine mehrfache Objekthierarchie. Allerdings nutzt er manuell orthogonalisierte Fassadenbilder und eignet sich nur für Fassaden mit einer fast idealen regulären Fensterstruktur.

- Das Verfahren von (Van Gool et al. 2007, Müller et al. 2007) erlaubt es, 3D Modelle von Fassaden durch Analyse von Einzelaufnahmen (halb-) automatisch zu generieren. Bei diesem Verfahren sind die Fenster aber nur Elemente der Fassade und werden nicht explizit, z.B. wie in der vorliegenden Arbeit via ISM, erkannt.

- In (Wenzel et al. 2007) werden regelmäßige Strukturen auf der Fassade detektiert. Fenster werden aber nicht explizit erkannt, sondern nur reguläre Teile der Fassade, und es findet keine 3D Modellierung statt.

- Der Ansatz von (Ripperda 2008) detektiert regelmäßige Strukturen auf der Fassade mittels RJMCMC. Außerdem wurde eine Objekthierarchie definiert. Dieser Ansatz verwendet keine 3D Modellierung, obwohl auch Laserscanner-Daten genutzt werden.

Alle angesprochenen Verfahren zur 3D Modellierung verwenden als Eingangsdaten terrestrische Aufnahmen. Diese erlauben die Modellierung auf einem sehr feinen *Level Of Detail* (LOD). Sie sind aber nicht vollautomatisch und es fehlt abgesehen vom letzten die Kopplung mit anderen Datenquellen, wie z.B. Laserdaten. Letztere ermöglichen eine bessere Qualität der 3D Modelle, besonders die Modellierung von Objekten wie Balkonen und Säulen, die meist nicht auf der Fassadenebene liegen.

Kapitel 9

Zusammenfassung und Ausblick

Diese Arbeit stellt einen Ansatz vor, mit dem für Fassaden in terrestrischen Bildsequenzen, auf denen einzelne Fenster unterscheidbar sind, d.h. keine reinen Glasfassaden, die Fenster erkannt, ihr Umriss bestimmt, in Zeilen und Spalten angeordnet und 3D rekonstruiert werden können. Aussehensbasierte Modellierung in Form von Implicit Shape Models (ISM) wird zusammen mit Markov Chain Monte Carlo – MCMC kohärent sowohl für die Detektion als auch für die Umrissbestimmung der Fenster verwendet. Die Fenster werden validiert und Modellselektion basierend auf Akaike's Information Criterion – AIC wird zur Auswahl zwischen Einzelfenstern und aus diesen gebildeten Fensterzeilen bzw. -spalten verwendet. Dies erhält das allgemeine Aussehen der Fassade ohne die Notwendigkeit, jedes Objekt getrennt zu beschreiben, was Ressourcen spart und die Effizienz der Visualisierung erhöht. Für eine 3D Interpretation der Fassade wird *Plane Sweeping* genutzt.

Die vorgestellte Methode kann in mehrere Richtungen erweitert werden:

- In terrestrischen Aufnahmen ist meist genug Information über die interne Struktur der Fenster vorhanden. ISM sollte es ermöglichen, Fensterteile, wie z.B. Fensterrahmen, Fensterscheiben, Fenstersprossen und Fensterbretter, zu extrahieren. Damit ergäbe sich z.B. die Möglichkeit, ein Fenster zu modellieren, welches geöffnet werden kann.

- Auf einer Fassade liegen nicht nur Fenster, sondern auch andere Objekte, wie z.B. Türen oder verschiedene Verzierungen. Mit entsprechenden Trainigsdatensätzen könnten diese Objekte zumindest teilweise mittels ISM detektiert und segmentiert werden.

- Balkone sind häufig ein Fassadenbestandteil, aber ihre Größe, Form, Farbe und 3D Lage variieren stark. Deshalb ist ISM nicht gut für die Modellierung solcher Objekte geeignet und es ist sinnvoll, andere Verfahren, wie z.B. eine Kombination von *3D Gruppierung* und *Plane Sweeping* und / oder eine Kupplung mit anderen Datenquellen, die die 3D Ausprägung der Objekte expliziter repräsentieren, z.B. Laserdaten, zu verwenden.

- Alle Bestandteile der Fassade, von den Fensterteilen bis hin zu aus Fenstern, Türen, Verzierungen u.ä. Teilen gebildeten Gittern, sollten in einem hierarchischen statisti-

schen Modell beschrieben werden. Die wahrscheinlichste Interpretation könnte mittels *Reversible Jump Markov Chain Monte Carlo (RJMCMC)* bestimmt werden, wobei einfache Interpretation mit Hilfe von Modellauswahl favorisiert werden sollten.

- Die zusätzliche Verwendung von Luftbildaufnahmen und / oder luftgetragenen Laserdaten würde es erlauben, 3D Modelle von Gebäuden inklusive der Dächer zu erstellen.

Neben der verbesserten 3D Modellierung von Gebäuden könnte der Ansatz auch in folgender Weise ergänzt werden:

- Weil die Außenansicht einer Fassade den internen Aufbau eines Gebäudes widerspiegelt, könnten die Innenräume dem 3D Modell hinzufügt werden.
- Eine realistische 3D Darstellung eines Stadtmodells sollte die gesamte urbane Umgebung beinhalten. D.h., dass Bäume, Zäune, Kinderspielplätze und evtl. sogar Fahrzeuge modelliert werden sollten.

Literaturverzeichnis

Agarwal, S., Awan, A. und Roth, D. (2004): Learning to Detect Objects in Images via a Sparse, Part-Based Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(11): 1475–1490.

Agarwal, S. und Roth, D. (2002): Learning a Sparse Representation for Object Detection, *Seventh European Conference on Computer Vision*, Band IV, 113–127.

Akaike, H. (1973): Information Theroy and an Extension of the Maximum Likelihood Principle, *Second International Symposium on Information Theory*, 267–281.

Alegre, F. und Dallaert, F. (2004): A Probalistic Approach to the Semantic Interpretation of Building Facades, *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 1–12.

Baillard, C. und Zisserman, A. (1999): Automatic Reconstruction of Piecewise Planar Models from Multiple Views, *Computer Vision and Pattern Recognition*, Band II, 559–565.

Ballard, D. (1981): Generalizing the Hough Transform to Detect Arbitrary Shapes, *Pattern Recognition* **13**(2): 111–122.

Bauer, J., Karner, K., Schindler, K., Klaus, A. und Zach, C. (2003): Segmentation of Building Models from Dense 3D Point-Clouds, *27th Workshop of the Austrian Association for Pattern Recognition*.

Borenstein, E. und Ullman, S. (2004): Learn to Segment, *European Conference on Computer Vision*, 315–328.

Burns, J., Hanson, A. und Riseman, E. (1986): Extracting Straight Lines, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(4): 425–455.

Dick, A., Torr, P., Ruffle, S. und Cipolla, R. (2001): Combining Single View Recognition and Multiple View Stereo for Architectural Scenes, *International Conference on Computer Vision*, 268–274.

Dick, A., Torr, P. und Cipolla, R. (2000): Automatic 3D Modelling of Architecture, *British Machine Vision Conference*, 273–289.

Dick, A., Torr, P. und Cipolla, R. (2002): A Bayesian Estimation of Building Shape Using MCMC, *Seventh European Conference on Computer Vision*, Band II, 852–866.

Dick, A., Torr, P. und Cipolla, R. (2004): Modelling and Interpretation of Architecture from Several Images, *International Journal of Computer Vision* **60**(2): 111–134.

Eckstein, W. und Munkelt, O. (1995): Extracting Objects from Digital Terrain Models, *Remote Sensing and Reconstruction for Three-Dimensional Objects and Scenes*, 2572, SPIE, 43–51.

Fischler, M. und Bolles, R. (1981): Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of the ACM* **24**(6): 381–395.

Förstner, W. und Gülch, E. (1987): A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features, *ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, Interlaken, Schweiz, 281–305.

Geman, S., Potter, D. und Chi, Z. (2002): Composition Systems, *Quarterly of Applied Mathematics* **LX**: 707–736.

Green, P. (1995): Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination, *Biometrika* **82**: 711–732.

Hartley, R. und Zisserman, A. (2003): *Multiple View Geometry in Computer Vision – Second Edition*, Cambridge University Press, Cambridge, Großbritannien.

Hastings, W. K. (1970): Monte Carlo Sampling Methods Using Markov Chains and Their Applications, *Biometrika* **57**(1): 97–109.

Heipke, C., Mayer, H. und Wiedemann, C. (1998): External Evaluation of Automatically Extracted Road Axes, *Photogrammetrie – Fernerkundung – Geoinformation* **2/98**: 81–94.

Kirkpatrick, S., Gelatt, C. D. und Vecchi, M. P. (1983): Optimization by Simulated Annealing, *Science, Number 4598* **220**: 671–680.

Leibe, B. und Schiele, B. (2004a): Combined Object Categorization and Segmentation with an Implicit Shape Model, *ECCV'04 Workshop on Statistical Learning in Computer Vision*, 1–15.

Leibe, B. und Schiele, B. (2004b): Scale-Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search, *Pattern Recognition – DAGM 2004*, Springer-Verlag, Berlin, 145–153.

Markov, A. A. (1906): Propagation of Law of Large Numbers to the Values, Which Depend on Each Other, *Proceedings of Physical-Mathematical Society by Kazan University, Part 2* **15**: 135–156.

MAYER, H. (2005): Robust Least-Squares Adjustment Based Orientation and Auto-Calibration of Wide-Baseline Image Sequences, *ISPRS Workshop in conjunction with ICCV 2005 "Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images" (BenCos), Beijing, China*, 1–6.

MAYER, H. (2007): 3D Reconstruction and Visualization of Urban Scenes from Uncalibrated Wide-Baseline Image Sequences, *Photogrammetrie – Fernerkundung – Geoinformation* **3/07**: 167–1761.

MAYER, H. und REZNIK, S. (2005): Building Façade Interpretation from Image Sequences, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Band (36) 3/W24, 55–60.

MAYER, H. und REZNIK, S. (2006): MCMC Linked with Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Band (36) 3, 130–135.

METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A. und TELLER, E. (1953): Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics* **21**: 1087–1092.

MÜLLER, P., ZENG, G., WONKA, P. und VAN GOOL, L. (2007): Image-based Procedural Modeling of Facades, *Proceedings of ACM SIGGRAPH 2007 / ACM Transactions on Graphics*, Band 26, ACM Press, New York, NY, USA.

NISTÉR, D. (2003): An Efficient Solution to the Five-Point Relative Pose Problem, *Computer Vision and Pattern Recognition*, Band II, 195–202.

POLLEFEYS, M., VAN GOOL, L., VERGAUWEN, M., VERBIEST, F., CORNELIS, K. und TOPS, J. (2004): Visual Modeling with a Hand-Held Camera, *International Journal of Computer Vision* **59**(3): 207–232.

REZNIK, S. und MAYER, H. (2007): Implicit ShapeModels, Model Selection, and Plane Sweeping for 3D Facade Interpretation, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Band (36) 3/W491, 173–178.

REZNIK, S. und MAYER, H. (2008): Implicit Shape Models, Self Diagnosis, and Model Selection for 3D Facade Interpretation, *Photogrammetrie – Fernerkundung – Geoinformation* **3/08**: 187–196.

RIPPERDA, N. (2008): Grammar Based Facade Reconstruction Using RjMCMC, *Photogrammetrie – Fernerkundung – Geoinformation* **2/08**: 83–92.

RISSANEN, J. (1978): Modeling by Shortest Data Description, *Automatica* **14**: 465–471.

SCHINDLER, G., K. P. und DELLAERT, F. (2006): Line-Based Structure From Motion for Urban Environments, *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

SCHINDLER, K. und BAUER, J. (2003): Detailed Building Reconstruction with Shape Templates, *27th Workshop of the Austrian Association for Pattern Recognition*.

SCHWARZ, G. (1978): Estimating the Dimension of a Model, *The Annals of Statistics* **6**(2): 461–464.

TU, Z., CHEN, X., YUILLE, A. und ZHU, S.-C. (2005): Image Parsing: Unifying Segmentation Detection and Recognition, *International Journal of Computer Vision* **63**(2): 113–140.

VAN GOOL, L., ZENG, G., VAN DEN BORRE, F. und MULLER, P. (2007): Towards mass-Produced Building Models, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Band (36) 3/W491, 209–220.

WEBER, M., WELLING, M. und PERONA, P. (2000): Unsupervised Learning of Models for Recognition, *Sixth European Conference on Computer Vision*, Band 1, 18–32.

WENZEL, S., DRAUSCHKE, M. und FÖRSTNER, W. (2007): Detection and Description of Repeated Stuctures in Rectified Facade Images, *Photogrammetrie – Fernerkundung – Geoinformation* **7/07**: 485–494.

WERNER, T. und ZISSERMAN, A. (2002): New Techniques for Automated Architectural Reconstruction from Photographs, *Seventh European Conference on Computer Vision*, Band II, 541–555.