Automatic Photo Pop–up

André Jähnig*

9. Februar 2006

Hauptseminar – Graphische Datenverarbeitung
Technische Universität Dresden
Wintersemester 2005/2006
Betreuer: Prof Dr. rer. nat. Stefan Gumhold

*s7006146@mail.inf.tu-dresden.de

Zusammenfassung

Diese Ausarbeitung befasst sich mit einem neuartigen Verfahren, welches aus einem einzigen Bild[1] automatisch ein 3D–Modell erstellt. Dabei ist verständlich, dass sich Komplexität und Detailreichtum nicht mit den Ergebnissen üblicher professioneller 3D–Software messen kann. Viel mehr lässt sich das entstandene Modell mit einer Illustration in einem Kinderbuch vergleichen, welche senkrecht aus dem Buch herausklappt, sobald man die entsprechende Seite aufschlägt. Dazu wird das Bild in die drei Bereiche „Ground“, „Vertical“ und „Sky“ eingeteilt und kann so an den entsprechenden Stellen „eingeschnitten“ und „gefaltet“ werden. Trotz dieser einfachen Annahmen liefert das Verfahren recht gute Ergebnisse für eine Vielzahl von Bildern, die man in einer typischen privaten Fotosammlung finden kann.

[1] eine Fotografie oder ein Gemälde

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Egal ob in Computerspielen, Internetanwendungen oder beispielsweise auf dem Mobiltelefon, überall sind dreidimensionale Welten auf dem Vormarsch oder haben sich bereits etabliert. Dort wo vor einiger Zeit noch schlichte 2D–Grafik oder einfach nur Text vorherrschte, findet man heute aufgepeppte Oberflächen und 3D–Welten mit einem Realismusgrad, der vor ein paar Jahren noch undenkbar gewesen wäre.

Die Technik ist so weit fortgeschritten, dass man nicht nur virtuelle Welten erstellen und aus verschiedenen Blickwinkeln rendern kann, sondern der Benutzer kann diese auch interaktiv erkunden, indem er sich in ihnen bewegt und umschaut. Allerdings benötigt man für das Erstellen solcher möglichst realer oder auch fiktiver Szenen ein gewisses Know-How und oftmals professionelle Software, was wiederum den Einsatz von Spezialisten unumgänglich macht. Hinzu kommt, dass man - speziell für realistische Szenen - eine Vielzahl von Bildern und Fotos aus möglichst unterschiedlichen Blickwinkeln benötigt, um die entsprechenden Texturen zu erstellen.

Es stellt sich also die Frage, ob nicht viel mehr Leute solche Szenen erstellen würden, wenn das Verfahren dazu einfacher wäre, man keine teure Spezialsoftware benötigen würde oder sich in komplizierte Oberflächen einarbeiten müsste. Es gibt sicher viele Leute, die sich beim Betrachten ihrer Urlaubsbilder wünschen, noch einmal an diesen Ort zurück zu kehren. Da dies nicht immer möglich ist, wäre es eine gute Alternative einen virtuellen Rundgang durch seine Bilder bzw. durch die in den Bildern dargestellten Orte machen zu können.

Genau an diesem Punkt soll dieses Paper einen Lösungsansatz bieten. Es wird eine Methode erläutert, die es auch einem nicht versierten Benutzer ermöglicht ein 3D–Modell zu erstellen. Das Verfahren arbeitet dabei voll automatisch und benötigt lediglich ein Bild als Input. Es wird ähnlich wie bei der Erstellung einer Kinderbuch–Illustration vorgegangen, die aus dem Buch heraus klappt. Zuerst wird das Bild auf die Grundfläche des Modells gelegt. Danach werden alle Bereiche, die als senkrechte Regionen betrachtet werden, automatisch hoch geklappt (engl. pop up). Allerdings lässt sich das entstandene Resultat aufgrund der gegebenen Einschränkungen (voll automatisch und basierend auf einem Bild) nicht mit einem professionell erstellten Modell vergleichen. Doch obwohl es einfach und relativ undetailliert ist, liefert es teilweise durchaus anschauliche virtuelle Rundgänge und eine neue Weise seine eigenen Fotos zu betrachten.

Da heutzutage die Digitalfotografie relativ verbreitet ist und dazu verleitet mehr Fotos als früher zu machen, könnte man sich folgende Situation vorstellen: Der Heimanwender lädt seine Bilder von der Digitalkamera auf

den Rechner und betrachtet sie dort in einem 3D–Browser[2]. Nun sucht er einfach die Modelle heraus, die ihm gefallen und verwirft den Rest. Einige werden - wie bei der Digitalfotografie - schlecht oder inkorrekt sein und andere wiederum einfach nur langweilig. Pauschal kann man aber sagen, dass dieses Verfahren bei etwa jedem dritten Bild ein gutes und anschauliches Ergebnis liefert.

1.1 Bisherige Arbeiten

Die Programme oder Methoden, die es bis jetzt auf diesem Gebiet gibt, benötigen viele Ausgangsbilder und spezielle Ausrüstung genauso wie zahlreiche Benutzerinteraktionen. Oftmals sind sie dann auch nur für spezielle Anwendungsgebiete von Nutzen. So zum Beispiel Façade [Debe96], ein Modellierungswerkzeug für Architekturszenen. Um damit ein Modell zu erstellen, benötigt man Fotos von allen Seiten des Gebäudes, aus denen mit Hilfe von Benutzerangaben die Struktur der Szene wiedergewonnen wird. Dazu müssen in den einzelnen Bildern Kanten der Architektur markiert und etwaige andere Bedingungen wie Symmetrie oder Verdeckungen angegeben werden.

Es gibt auch verschiedene automatische Verfahren, die ein 3D–Modell aus mehreren Bildern erstellen können. [Poll04] verarbeitet dabei als Eingabe ein Video, aus dem einzelne Keyframes extrahiert und analysiert werden. Dazu werden einzelne Punkte in den Bildern einander zugeordnet und verfolgt, wodurch Informationen über die Kamerapositionen zu den diversen Zeitpunkten gewonnen werden können. Aus den verschiedenen Ansichten der Szene und der Kenntnis über deren Relation zueinander wird über ein Stereovision-Verfahren die Struktur der Szene wiedergewonnen und ein 3D–Modell generiert.

Neben den Ansätzen, die auf mehreren Ausgangsbildern basieren, gibt es auch Lösungen für die Erstellung virtueller Welten aus nur einem Bild. [Lieb99] oder auch [Crim00] liefern sehr korrekte Modelle, jedoch muss der Benutzer dazu zahlreiche Angaben zum Bild machen, wie etwa parallele Linien, Quadrate oder rechtwinklige Beziehungen markieren. Eine der Hauptinspirationen für Automatic Photo Pop-up war die Idee von „Tour into the Picture“ [Horr97]. Hierbei wird die komplette Szene als eine Art Theaterbühne - eine achsenparallele Box - mit einem Boden, einer Decke, zwei Seitenwänden und einer Hintergrundwand betrachtet. Der Benutzer definiert dabei über ein spinnenartiges Interface die verschiedenen Parameter der Szene. Abbildung 1 verdeutlicht grafisch die einzelnen Schritte, um mit diesem Verfahren ein 3D–Modell zu erstellen. Zuerst definiert der Benutzer im Ausgangsbild (a)

[2]hier wird ein simpler VRML-Viewer genutzt

eine Maske für die Vordergrundobjekte (c), die später auf seperaten Ebenen dargestellt werden. Übrig bleibt das Hintergrundbild (b), bei welchem mit einem entsprechenden Algorithmus die entstandenen Lücken weg retuschiert werden. Im nächsten Schritt wird über das Spinnen–Interface, durch Verschieben des Ausgangspunktes der Spinnenfäden, der Fluchtpunkt der Szene (d) ermittelt, indem einzelne Spinnenfäden mit markanten Fluchtlinien im Bild überein gebracht werden. Danach werden die Koordinaten der Theaterbühne, speziell die der Hintergrundwand festgelegt (e), so dass das Ausgangsbild in die entsprechenden Bereiche für den Boden, die Decke und die restlichen Wände eingeteilt wird. In einem letzten Schritt werden die Vordergrundobjekte (f) und die Kamera (g) neu platziert, so dass die Szene aus einem neuen Blickwinkel gerendert werden kann (h).

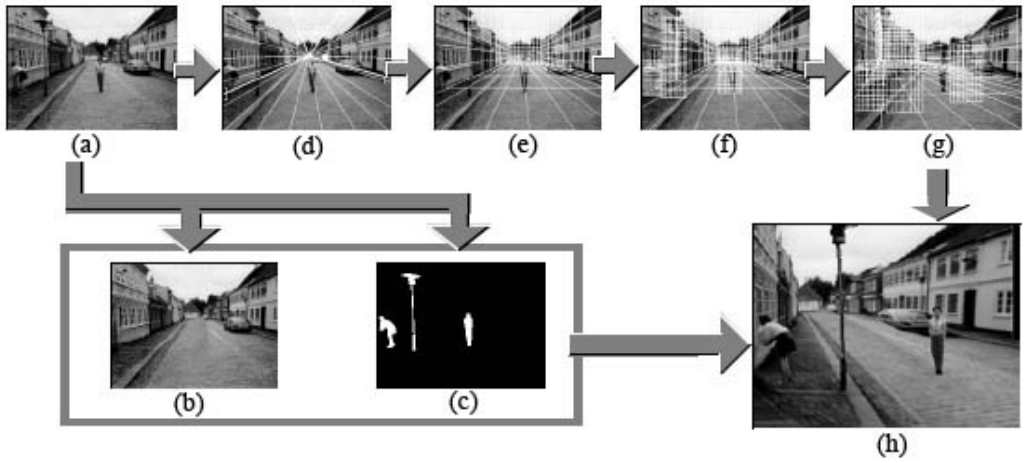

Abbildung 1: Tour into the picture – Ablaufdiagramm

Ein Verfahren, welches automatisch operiert und dabei nur ein Bild benötigt, scheint es aber bisher noch nicht zu geben.

1.2 Der neue Ansatz

Der neue Ansatz, der mit diesem Paper verfolgt werden soll, ähnelt der Handlungsweise eines Menschen. Betrachten wir ein Bild, also eine 2D–Abbildung einer 3D–Szene, können wir uns fast exakt den Ort im Realen vorstellen, wissen wie die Größenrelationen einzelner Objekte zueinander sind oder wie das Ganze aus einem anderen Blickwinkel aussieht. Da wir ein Leben lang unsere Natur beobachten und gewisse Sachverhalte registrieren, lernen wir mit Hilfe von Statistiken was realistisch ist und was nicht. So können wir viele der vielleicht möglichen geometrischen Interpretationen eines Bildes verwerfen.

Das Ziel ist also, die Geometrie im Bild wieder zu erkennen, anstatt diese aufwendig zu berechnen. Analog zum Menschen, der auf sein Wissen zurückgreift, werden dabei, basierend auf einem Set von Trainingsbildern, Entscheidungen getroffen, um ein 3D-Modell zu erstellen. Jedoch wird nicht, wie üblich, versucht bestimmte semantische Klassen, wie etwa Straßen, Bäume oder Gebäude, zu erkennen. Vielmehr sollen einzelne Regionen im Bild geometrischen Klassen (GROUND, VERTICAL oder SKY) zugeordnet werden. So gehört zum Beispiel ein Stück Holz, welches auf dem Boden liegt, und ein gleiches Stück Holz, welches Teil eines Schrankes ist, zu zwei verschiedenen geometrischen Klassen, aber zur selben semantischen Klasse. Aufgrund der richtig bezeichneten Trainingsbilder wird statistisch das Ausgangsbild in geometrische Klassen eingeteilt, um daraus dann das 3D–Modell zu generieren.

Vorerst werden dabei nur Bilder mit Außenaufnahmen betrachtet und, wie bereits oben erwähnt, das Modell, bestehend aus einer Grundplatte mit glatten, senkrecht darauf stehenden Objekten, recht einfach gehalten.

2 Geometrische Merkmale

Zur Generierung eines 3D–Modells anhand eines einzigen Bildes, soll dieses in geometrische Klassen eingeteilt werden. Um zu entscheiden welche Teilbereiche zur Klasse „Ground“, „Vertical“ oder „Sky“ gehören, werden verschiedene geometrische Merkmale des Bildes analysiert. Tabelle 1 auf Seite 13 zeigt eine Übersicht über sämtliche Merkmale, die dazu ausgewertet werden. Die zwei Spalten mit den Ziffern geben einerseits die Anzahl der Variablen an, die das jeweilige Merkmal liefert, und andererseits die Anzahl der Variablen, die tatsächlich für die Auswertung benötigt und auch genutzt werden.

2.1 Farbe

Das Gras ist für gewöhnlich grün oder auch bräunlich und der Himmel ist meistens blau. Man kann also anhand der Farbe einer Fläche im Ausgangsbild bereits einige Aussagen über die eventuelle Oberfläche der zu betrachtenden Region machen. Es werden dazu zwei Farbräume ausgewertet (Zeile C1 – C4 in Tabelle 1). Zum einen der RGB-Farbraum, der Informationen über die Farbigkeit liefert, also wie blau (grün, ...) eine Region ist, und zum anderen der HSV-Farbraum, welcher die Helligkeit angibt oder auch wie grau die Fläche ist.

2.2 Textur

Durch das Betrachten der Oberflächentextur einer Region erhält man weitere Informationen über das Material. So ist es zum Beispiel möglich Gras von den Blättern eines Baumes oder einen Himmel von einer Wasseroberfläche zu unterscheiden. Durch verschiedene Filter, die auf das Bild angewendet werden, wird versucht die Struktur der Szene zu verstärken, um sie besser analysieren zu können. Dazu werden gerichtete Gauss–Filter (T1 – T4) angewandt, um danach einen Vergleich mit den 12 gegenseitig unähnlichsten Universal–Textons (T5–T7) aus dem „Berkeley segmentation dataset“ [Mart01] durchzuführen.

2.3 Ort im Bild

Ein weiteres wichtiges Merkmal ist der Ort im Bild, an dem sich die zu betrachtende Region befindet. Zum Beispiel kann man davon ausgehen, dass sich der Boden immer am unteren Rand des Bildes befindet, wohingegen der Himmel immer den oberen Bereich einnimmt. So werden die x– und y–Koordinate, normalisiert nach der Bildbreite bzw. –höhe (L1), und zusätzlich

das 10. und 90. Perzentil (L2) betrachtet. Durch diese Werte kann entschieden werden, wie nah die betrachtete Region am Rand liegt. Zusätzlich zu den Koordinaten wird auch die Form der Region (L4 – L7) analysiert. Eine annähernd konvexe Fläche lässt auf eine senkrechte Region in der Struktur schließen. Eine große und nicht-konvexe Fläche deutet dagegen auf einen Teil des Bodens oder des Himmels hin.

2.4 3D–Geometrie

Dieses Merkmal hilft eine Entscheidung über die Orientierung einer Fläche im Raum zu treffen. Dazu werden markante Fluchtlinien benötigt, wodurch sich die 3D–Lage eindeutig bestimmen lässt. Allerdings ist es schwer, diese Informationen aus einem Bild von einer Außenaufnahme, welche relativ unstrukturiert ist, einfach so zu extrahieren. Hier muss der Umweg über gerade Linien (G1–G2) und deren Schnittpunkt im Bild (G3–G7) gegangen werden. Zur Ermittlung von geraden Linien im Bild wird die Methode von Kosecka und Zhang [Kose02] benutzt.

Hierbei wird das Bild in Graustufen umgewandelt und die Änderung des Grauwertes in x- bzw. y-Richtung betrachtet. Daraus lässt sich, wie in Abbildung 2 gezeigt, für jedes Pixel ein Gradient berechnen, dessen Länge der Stärke der Grauwertänderung entspricht. Befinden sich im Bild mehrere zusammenhängende Pixel mit annähernd derselben Gradientenrichtung und -länge, kann davon ausgegangen werden, dass sich in diesem Bereich eine Kante bzw. eine Linie befindet.

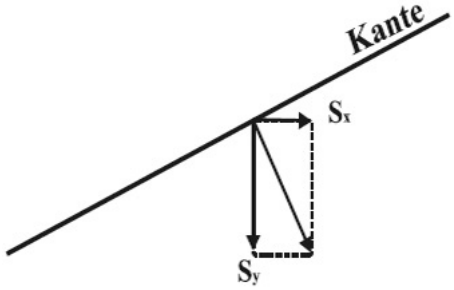

Abbildung 2: Graustufen–Gradient

Wie bereits oben erwähnt, werden auch die Schnittpunkte der Linien, die annähernd parallel zueinander sind, betrachtet. Diese werden in acht Richtungen und in „weit“ oder „sehr weit“ entfernt, ausgehend von der Bildmitte, eingeteilt.

Bei Bildern mit natürlichen Oberflächen, die keine Kanten enthalten, kann man die Textur zur Bestimmung des Fluchtpunktes nutzen. Eine Textur, die

sich über eine größere Fläche des Bildes erstreckt, nimmt mit der Kameraentfernung an Detailiertheit ab. So erkennt man im nahen Bereich noch einzelne Strukturen, wobei diese immer mehr verwaschen, je weiter sich der betrachtete Punkt von der Kamera entfernt. Durch Vergleichen des Bereiches der „stärksten“ Texturierung mit dem Zentrum der Fläche, die man betrachtet, wird ein Textur–Gradient (G8) berechnet, aus welchem man Informationen über Fluchtpunkt und –linie erhalten kann.

2.5 Horizont

Das letzte Merkmal, welches betrachtet wird, ist der Horizont. Um zu ermitteln, auf welcher Höhe im Bild sich dieser befindet, werden wieder die langen, nahezu parallelen Linien und deren Schnittpunkte genutzt. Die Horizontposition wird so gewählt, dass der $L\frac{1}{2}$–Abstand zu allen Schnittpunkten im Bild minimal wird. Dies liefert besonders gute Ergebnisse bei bebauten Szenen, da diese viele zum Boden parallele Linien enthalten, die sich in einer 2D–Abbildung auf dem Horizont schneiden. Oft ist die Lage einer Region in Relation zum Horizont wichtiger als die absoluten Bildkoordinaten. Deshalb wird die Information separat durch das Merkmal G3 geliefert.

Merkmalsbeschreibung	**Anzahl**	**genutzt**
Farbe	**15**	**15**
C1. RGB-Werte	3	3
C2. HSV-Werte: konvertiert aus den RGB-Werten	3	3
C3. Farbton: Histogramm (5 bins) & Mittelwert	6	6
C4. Sättigung: Histogramm (3 bins) & Mittelwert	3	3
Textur	**29**	**13**
T1. DOOG Filter: absolute Rückgabewerte	12	3
T2. DOOG Filter: Hauptvariablen	1	0
T3. DOOG Filter: ID der Max-Variable	1	1
T4. DOOG Filter: $(Max - Mittelwert)$ der Var.	1	1
T5. Textons: absolute Rückgabewerte	12	7
T6. Textons: Maximalwert	1	0
T7. Textons: $(Max - Mittelwert)$ der Variablen	1	1
Ort und Form	12	10
L1. Ort: normalisierter x- & y-Wert	2	2
L2. Ort: 10. & 90. Perzentil der x- & y-Werte	4	4
L3. Ort: 10. & 90. Perzentil des Horizontes	2	2
L4. Form: Superpixelanzahl in der Gruppierung	1	1
L5. Form: Anzahl der Seiten der konvexen Hülle	1	0
L6. Form: Anzahl der Pixel/Fläche (konv. Hülle)	1	1
L7. Form: ob die Region zusammenhängend ist	1	0
3D Geometrie	**35**	**28**
G1. Lange Linie: Anzahl in der Region	1	1
G2. Lange Linie: % der nahezu parallelen Linien	1	1
G3. Schnitt: Anz. für 12 Richtungen + Mittelwert	13	11
G4. Schnitt: % rechts vom Zentrum	1	1
G5. Schnitt: % über dem Zentrum	1	1
G6. Schnitt: % weit vom Zentrum (8 Richtungen)	8	4
G7. Schnitt: % sehr weit v. Zentrum (8 Richt.)	8	5
G8. Textur-Gradient: x- & y-Kantenzentrum (T2)	2	2

Tabelle 1: geometrische Merkmale – Übersicht

3 Festlegen der geometrischen Klassen

Mit den soeben vorgestellten geometrischen Merkmalen soll nun versucht werden das Bild so in die drei Bereiche „Ground“, „Vertical“ und „Sky“ einzuteilen, dass daraus dann in einem weiteren Schritt das 3D-Modell generiert werden kann. Wie bereits anfangs erwähnt, wird bei der Entscheidungsfindung auf einen Satz von Trainingsbildern zurückgegriffen.

3.1 Superpixel erstellen

Wenn man jetzt versucht die geometrischen Merkmale auf das Bild als Ganzes oder auf jedes Pixel einzeln anzuwenden, bekommt man keinerlei verwertbare Information zu welcher geometrischen Klasse das Pixel gehört. Deshalb werden in einem ersten Schritt sogenannte Superpixel gebildet. Dies sind kleine, nahezu uniforme Regionen im Bild. Dadurch sind komplexere Statistiken effizienter anwendbar, um mehr über die Bildstruktur zu erfahren.

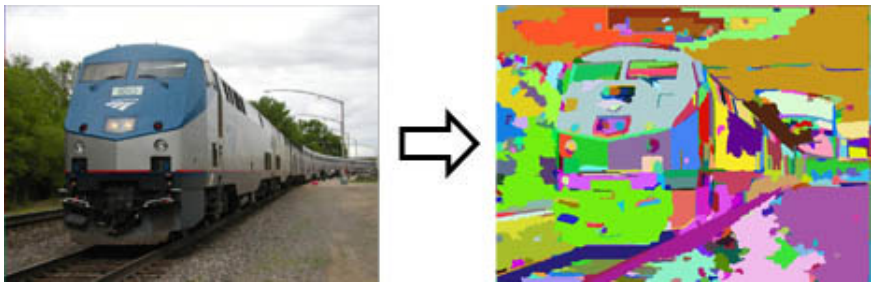

Abbildung 3: Bildung von Superpixel

Für die Superpixelgenerierung wird die Over-Segmentation-Technik von Felzenszwalb und Huttenlocher [Felz04] genutzt. Hierbei wird das Bild als Graph betrachtet, wobei jeder Pixel ein Knoten darstellt und sich je eine Kante zwischen zwei benachbarten Pixeln befindet. Jede dieser Kanten hat eine Wichtung, welches ein Maß für den Unterschied dieser beiden Pixel in Bezug auf ein bestimmtes Merkmal (z.B. Farbe, Intensität, ...) ist. Der Algorithmus teilt diesen Graphen nun so in Teilgraphen, dass die Wichtung innerhalb eines Teilgraphen klein ist und im Gegensatz dazu die Wichtung zwischen zwei Pixeln, welche zu verschiedenen Teilgraphen gehören, groß. Abbildung 4 verdeutlicht noch einmal den Algorithmus.

3.2 Superpixel gruppieren

Auch die Superpixel allein reichen noch nicht zur Bestimmung der geometrischen Klassen. Bis jetzt können nur die einfacheren Merkmale wie etwa

Gegeben ist ein Graph $G = (V, E)$, mit n Knoten und m Kanten. Gesucht ist eine Segmentierung von V in die Komponenten $S = (C_1, ..., C_2)$.

1. Sortiere E in $\Pi = (o_1, ..., o_m)$ nach aufsteigender Kantenwichtung.
2. Starte mit einer Segmentierung S^0, bei der sich jeder Pixel v_i in einer eigenen Komponente befindet.
3. Wiederhole Schritt 3 für $q = 1, ..., m$.
4. Konstruiere S^q aus S^{q-1} wie folgt:
   Seien v_i und v_j die Pixel, die durch die q-te Kante in der Reihenfolge verbunden sind, also $o_q = (v_i, v_j)$. Wenn sich v_i und v_j in unterschiedlichen Komponenten von S^{q-1} befinden und $w(o_q)$ klein im Vergleich zu den internen Differenzen von beiden Komponenten ist, so verbinde diese beiden, andernfalls mache nichts.
   Formaler ausgedrückt: C_i^{q-1} ist die Komponente von S^{q-1}, die v_i enthält und C_j^{q-1} die, die v_j enthält. Wenn $C_i^{q-1} \neq C_j^{q-1}$ und $w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$, so erhält man S^q aus S^{q-1}, indem man C_i^{q-1} und C_j^{q-1} zusammenfügt. Andernfalls ist $S^q = S^{q-1}$.
5. Gebe $S = S^m$ zurück.

Abbildung 4: Over–Segmentation–Algorithmus

Farbe oder Textur ausgewertet werden. Um die komplexeren Merkmale zu berechnen, muss eine weitere Zusammenfassung der Superpixel vorgenommen werden. Bis jetzt ist es lediglich möglich zu entscheiden, ob zwei Superpixel zur selben geometrischen Klassen gehören oder nicht. Deshalb werden sie nun so zusammengefasst, dass alle Superpixel einer Gruppierung wahrscheinlich die gleiche Bezeichnung bzw. die gleiche geometrische Klasse haben. Idealerweise entspricht eine Gruppierung einem Objekt in der Szene, wie etwa einem Haus, dem Himmel oder einem Baum. Da dies aber nicht zu 100 Prozent sicher ist und auch nicht garantiert werden kann, dass alle Superpixel eine Gruppierung wirklich die selbe Bezeichnung haben, werden mehrere solcher Gruppierungsmöglichkeiten über das Bild erstellt. Die verschiedenen Möglichkeiten werden dann genutzt, um die richtige geometrische Klasse zu finden. Durch Variieren der Anzahl von Gruppierungen N_C im Bild, ist es möglich, einen guten Kompromiss zu erreichen zwischen zu wenig Erkenntnis über die Bildstruktur und der Gefahr verschiedene Bezeichnungen in einer Gruppierung zu haben.

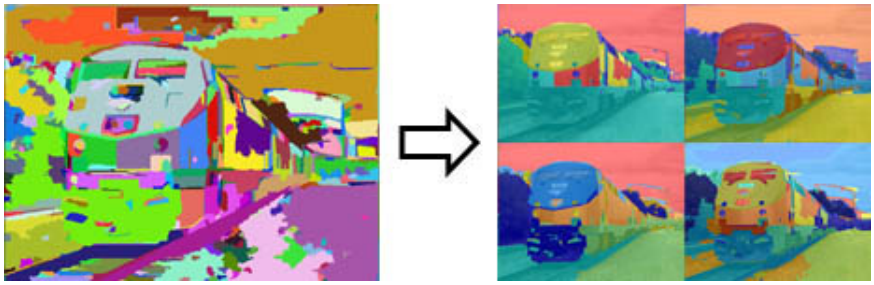

Abbildung 5: Gruppieren der Superpixel

Die Generierung der Gruppierungen geschieht iterativ. Zuerst wird jeder Gruppierung ein zufällig gewähltes Superpixel zugeordnet. Danach werden die restlichen Superpixel nacheinander jeweils der Gruppierung zugeordnet, bei der es am wahrscheinlichsten ist, dass beide die selbe Bezeichnung haben. Dazu wird mit Hilfe der Gleichung (1) die durchschnittliche paarweise Wahrscheinlichkeit zu den anderen Superpixeln ausgewertet, welche bei der entsprechenden Gruppierung maximal ist:

$$S(C) = \sum_{k}^{N_C} \frac{1}{n_k(1-n_k)} \sum_{i,j \in C_k} \log P(y_i = y_j | \, |z_i - z_j|) \tag{1}$$

Der erste Summentherm ist dabei für die Durchschnittsbildung verantwortlich, wobei n_k die Anzahl der Superpixel in der Gruppierung C_k ist. Die Wahrscheinlichkeit, dass beide Superpixel die gleiche Bezeichnung haben, wird im zweiten Summentherm auf Basis der absoluten Differenz der beiden Merkmalsvektoren z_i und z_j berechnet. Wie genau diese ermittelt wird zeigt Kapitel 4 auf Seite 18.

3.3 Superpixel bezeichnen

Anhand der nun gebildeten Gruppierungen kann eine Entscheidung getroffen werden, welche Bezeichnung jedes Pixel bekommt. Dazu wird zum einen für jede Gruppierung die Zugehörigkeit zu den drei geometrischen Klassen geschätzt und zum anderen wie wahrscheinlich es ist, dass alle Superpixel einer Gruppierung die selbe Bezeichnung haben. Aus diesen beiden Wahrscheinlichkeiten wird dann mit Hilfe der Gleichung (2) die Bezeichnung der Superpixel und somit der darin enthaltenen Pixel bestimmt.

$$P(y_i = t|x) = \sum_{k:s_i \in C_k} P(y_k = t|x_k, C_k) P(C_k|x_k) \tag{2}$$

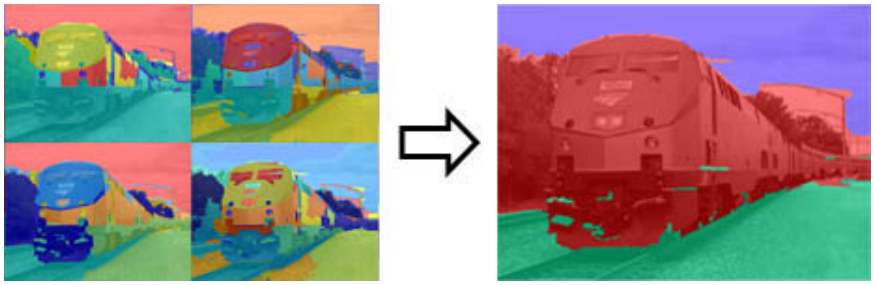

Abbildung 6: Bezeichnung der Superpixel

Es wird also für alle Gruppierungen C_k, die das Superpixel s_i enthalten, einmal die Wahrscheinlichkeit berechnet, dass die Gruppierung die Bezeichnung $y_k = t$ für $t = \{$„Ground“, „Vertical“, „Sky“$\}$ hat und außerdem wie wahrscheinlich es ist, dass die Bezeichnung innerhalb der Gruppierung homogen ist. Dazu wird diesmal der Merkmalsvektor x_k der Gruppierung genutzt. Durch Aufsummieren der einzelnen Produkte erhält man für jede geometrische Klasse einen Wahrscheinlichkeitswert, woraufhin das Superpixel die wahrscheinlichste Bezeichnung bekommt. Somit erhält man eine Einteilung des Bildes in „Ground“, „Vertical“ und „Sky“ und kann nun das 3D–Modell erstellen.

4 Trainingsdaten

In der Einleitung wurde bereits erwähnt, dass die Grundlage für die Einteilung des Bildes in „Ground“, „Vertical“ und „Sky“ ein Set von Trainingsbildern sein soll. Somit wird analog zum menschlichen Handeln ein Gedächtnis geschaffen, auf welches bei der Erstellung des neuen 3D–Modells zurück gegriffen werden kann.

Genauer gesagt werden die Trainingsdaten benutzt, um die Wahrscheinlichkeitsfunktionen zum Gruppieren und Bezeichnen von Superpixeln zu bestimmen bzw. diese zu lernen. Dazu wurden 82 repräsentative Bilder ausgesucht, die einen breiten Mix von bebauten, halb-bebauten und Natur-Szenen enthalten.

Jedes dieser Bilder wurde mit der Over-Segmentation-Technik [Felz04] in Superpixel eingeteilt, die danach manuell mit der korrekten geometrischen Klasse bezeichnet wurden. Basierend auf den so entstandenen 2500 gleich und unterschiedlich bezeichneten Superpixeln wird später die paarweise Wahrscheinlichkeit $P(y_i = y_j | |z_i - z_j|)$ zweier Superpixel bestimmt (vgl. Kapitel 3.2). Die Berechnung verwendet dazu eine logistischer Regression in Form von Adaboost [Coll02].

$$f_m(z_1, z_2) = \sum_{i}^{n_f} \log \frac{P_m(y_1 = y_2 | |z_{1i} - z_{2i}|)}{P_m(y_1 \neq y_2 | |z_{1i} - z_{2i}|)} \tag{3}$$

Jedes Superpixelpaar der Trainingsdaten liefert in Bezug auf ein einzelnes Merkmal eine Aussage über die Merkmalsdifferenz und inwiefern beide die selbe bzw. nicht die selbe Bezeichnung haben. Durch die 2500 Trainingssamples kann nun auch für jede neue Merkmalsdifferenz über eine Dichteschätzung eine Wahrscheinlichkeit (P_m) ermittelt werden. Aus diesen Wahrscheinlichkeiten werden, wie in Gleichung (3) gezeigt, schwache Klassifikatoren gebildet. So liefert jedes Merkmal einen schwachen Klassifikator und durch Aufsummieren dieser wird ein starker Klassifikator für die Wahrscheinlichkeitsberechnung gebildet. Das besondere an Adaboost ist, dass für alle schwachen Klassifikatoren eine Wichtung entsprechend der Fehleranfälligkeit vorgenommen wird. Dazu wird für ein Trainingssample mittels des gebildeten starken Klassifikators die Wahrscheinlichkeit ausgerechnet und analysiert. Die anfangs gleichverteilten Wichtungen der schwachen Klassifikatoren werden nun anhand des Fehlers[3] korrigiert. Und zwar so, dass die fehleranfälligen Merkmalsklassifikatoren eine geringere Wichtung bekommen und die genaueren eine höhere. Durch mehrmaliges Wiederholen mit verschiedenen

[3] durch Verwendung der Trainingssample, kann eine Aussage über den Fehler getroffen werden

Trainingssamplen und jeweils entsprechender Anpassung der Gewichte kann die Genauigkeit der Wahrscheinlichkeitsfunktion verbessert werden.

Zur Bestimmung der Bezeichungs– und Homogen–Wahrscheinlichkeiten (siehe Kapitel 3.3) wird ähnlich vorgegangen. Zunächst werden auf den Trainingsbildern, wie in Kapitel 3.2 beschrieben, mehrere Gruppierungsmöglichkeiten gebildet und diese dann korrekt mit „Ground“, „Vertical“, „Sky“ oder „Mixed“[4] bezeichnet. Nun kann wieder mittels Adaboost eine Wahrscheinlichkeitsfunktion aufgrund der Merkmale gebildet werden. Hier werden allerdings nicht alle Merkmale genutzt, sondern jeder schwache Klassifikator entscheidet mittels eines Entscheidungsbaumes, welche weiteren Merkmale benutzt werden sollen.

[4] die Gruppierung enthält verschieden bezeichnete Superpixel

5 3D–Modell erstellen

Das Bild ist nun also in die Bereiche für den Boden, die senkrechten Regionen und den Himmel eingeteilt. Wenn man jetzt wieder den anfangs erwähnten Vergleich mit der Kinderbuch–Illustration hinzuzieht, müsste das Bild nur noch an den entsprechenden Stellen eingeschnitten und gefalten werden, so dass die senkrechten Bereiche hochklappen. Jedoch besteht noch folgendes Problem: Wo im Bild muss ein Knick oder Schnitt gemacht werden? Bis jetzt wurde zwar ermittelt, welcher Bereich senkrecht erscheinen soll, allerdings ist noch keine Aussage darüber getroffen worden, wo sich die einzelnen Objekte befinden. Das heißt, es muss eine Möglichkeit gefunden werden, um die senkrechte Region in einzelne Objekte aufzuteilen. Das ist besonders schwer, wenn sich die einzelnen Objekte überlappen. Und es muss die Stelle gefunden werden, an der die senkrechte Region hochgeknickt wird, also die Stelle wo ein Objekt auf den Boden trifft. Das ist unmöglich, wenn diese Stelle durch ein anderes Objekt verdeckt wird. Die aktuelle Implementierung versucht noch nicht die Überlappungen zu trennen, sondern lieber einen Schnitt oder Knick weniger zu machen anstatt zu viele.

Bevor dieses Problem angegangen wird, werden zunächst eventuelle Bezeichnungsfehler im Bild korrigiert. Dazu werden sämtliche Superpixel, welche als „Ground“ oder „Sky“ bezeichnet wurden und komplett von Nicht–„Ground“ bzw. Nicht–„Sky“ Superpixel umrandet sind, so umbenannt wie die Mehrheit der angrenzenden Superpixel (vgl. Abbildung 10).

Danach wird eine erste Trennung der senkrechten Superpixel in Regionen vorgenommen, die gar nicht oder nur sehr lose miteinander verbunden sind. Um besser entscheiden zu können, wo einzelne Regionen nur lose miteinander verbunden sind, wird ein morphologischer Algorithmus angewandt, der zuerst die Regionen schrumpft (Erosion), so das kleine Gebiete und Ausläufer verschwinden. Danach werden die übrigen Regionen wieder erweitert (Dilation), so dass sie ihre ursprüngliche Größe zurück erhalten. Abbildungen 7 und 8 zeigen noch einmal an einem kleinen Beispiel, wie durch das Schrumpfen und Wachsen kleine Regionen beseitigt werden.

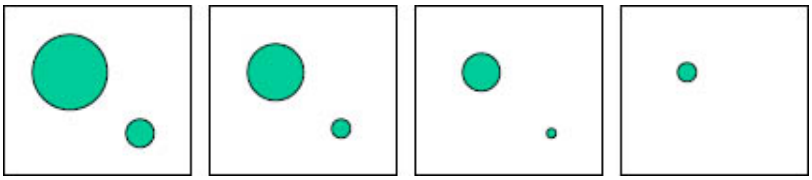

Abbildung 7: Erosion einer Region

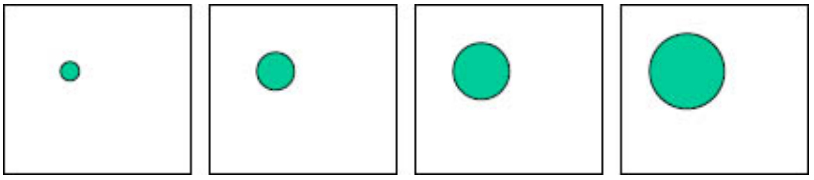

Abbildung 8: Dilation einer Region

Nun wird für jede Region versucht die Grenze zum Boden durch eine Reihe von Liniensegmenten mit Hilfe der Hough–Transformation [Duda72] darzustellen. Die Hough-Transformation ist ein effizientes und robustes Verfahren um in einem Bild parametrisierte geometrische Figuren, wie etwa Geraden oder Kreise, zu erkennen. Dazu wird das Bild zuerst in ein schwarz–weiß Bild umgewandelt und eine Kantenerkennung durchgeführt. Um nun geometrische Objekte im Bild zu finden, wird das Bild in einen Dualraum (Hough–Raum) transformiert. Für jeden Punkt, der auf einer Kante im Bild liegt, werden alle Parameter der gesuchten geometrischen Figur ausgewertet und im Dualraum eingetragen. Somit entspricht jeder Punkt im Dualraum einem Objekt im Bildraum.

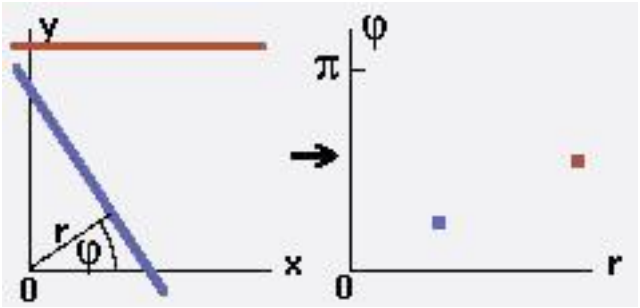

Abbildung 9: Transformation einer Geraden in einen Hough–Raum

Eine Gerade lässt sich mathematisch beschreiben durch ihren senkrechten Abstand r zum Koordinatenursprung, sowie durch den Winkel φ zwischen der entsprechende Verbindungsstrecke und der x-Achse (siehe Abbildung 9). Jedes Wertepaar (r, φ) entspricht demnach einer Geraden. Um Geraden im Bild zu finden, werden nun für jedes Wertepaar (r, φ) die Bildpunkte mit den Koordinaten (x, y) gezählt, die nahe der jeweiligen gedachten Linie liegen. Die (r, φ)–Kombinationen entsprechen dann einer Geraden im Bild, wenn zu ihnen viele Bildpunkte gehören.

Wenn ein Punkt (x, y) vorgegeben ist, gilt für eine Gerade mit dem Winkel φ, die durch diesen Punkt geht:

$$r = x \cos \varphi + y \sin \varphi \tag{4}$$

Für jeden der Punkte, die zu einer Kante im Bild gehören, wird nun mittels der Gleichung (4) der Abstand r für verschiedene φ–Werte, die in kleinen Schritten von 0 bis 2Π gehen, berechnet. Die Schrittweite darf nicht zu groß und nicht zu klein gewählt werden, damit weder die Rechenzeit zu lang ist, noch Geraden übersehen werden. Der Abstand r muss sinnvoll gerundet werden, damit ähnlich wie bei φ nicht zu viel und nicht zu wenig Werte entstehen. Damit erhält man die (r, φ)–Werte der Geraden, die durch diese Punkte gehen. Die Summe der Bildpunkte, die jeweils zu denselben (r, φ)–Paaren gehören, geben an, wieviele Bildpunkte auf den gedachten Geraden liegen. Somit erhält man ein Histogramm mit dem Ergebnis der Hough–Transformation des Originalbildes. Eine große Summe lässt vermuten, dass im Bild tatsächlich eine Gerade mit dem zugehörigen (r, φ) verläuft. Abbildung 10 zeigt die für eine Beispielszene gefundenen Liniensegmente.

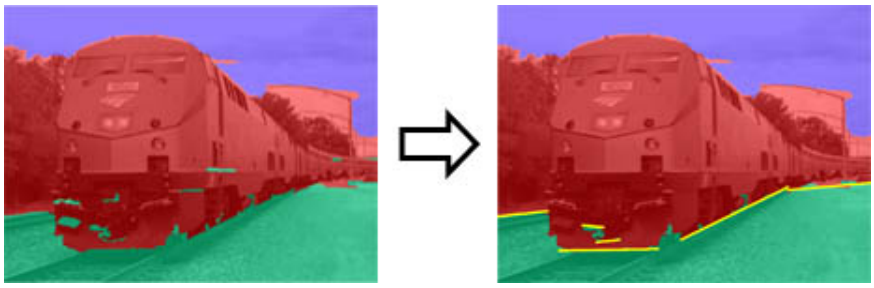

Abbildung 10: durch Hough–Transformation gefundene Liniensegmente

Die so gefundenen Liniensegmente in jeder Region werden dann zu einer oder mehreren Polylinien verbunden. Zur effizienteren und fehlerfreieren Generierung werden die Liniensegmente vorher sortiert und eventuelle Überlappungen entfernt (siehe auch Abbildung 13).

Falls mittels der Hough–Transformation keine Liniensegmente für eine Region gefunden werden können, so wird eine einfache Polylinie angenommen, die als Start- und Endpunkt den jeweils äußerst linken bzw. rechten Grenzpunkt nimmt. Danach wird die Polylinie eventuell noch in maximal drei Segmente aufgeteilt, so dass der $L1$–Abstand zwischen Polylinie und der Grenze zwischen den „Ground“– und „Vertical“–Superpixeln minimal wird.

Durch die Erstellung der Polylinien wurde nun eine weitere und finale Trennung der senkrechten Regionen erreicht. Jede Polylinie für sich repräsentiert ein Objekt im 3D–Modell. Jedes Objekt wird aus einer Anzahl von zusammenhängenden, senkrecht zum Boden stehenden Ebenen modelliert,

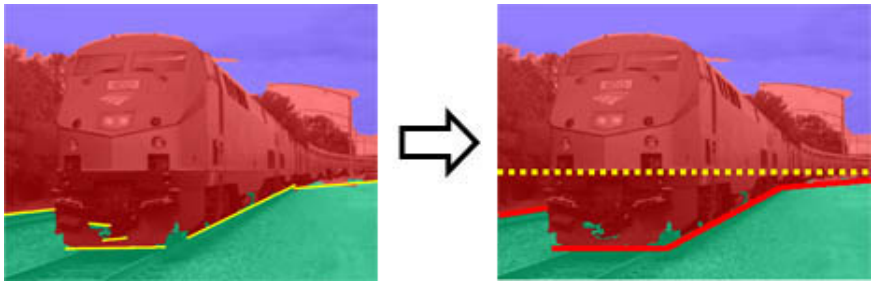

Abbildung 11: Polylinie und geschätzte Horizontposition

nämlich genauso viele wie die Polylinie Segmente hat. Die Polylinie gibt außerdem an, wo und in welchem Winkel zueinander die Ebenen auf der Bodenplatte stehen. Die Höhe der Ebenen ist gegeben durch die maximale Höhe der Region im Bild über dem jeweiligen Segment und den Kameraparametern. Der Boden des Bildes wird ebenfalls mit Hilfe der Kameraparameter und der geschätzten Horizontposition in 3D–Koordinaten transformiert. Anschließend wird das fertig erstellte 3D–Modell texturiert, indem aus dem Ausgangsbild zwei Texturen erstellt und auf das Modell gemappt werden. Für die beiden Texturen wird jeweils der Alphawert einer der beiden Regionen („Ground“ bzw. „Vertical“) auf Null gesetzt, so dass nur die entsprechenden Pixel in der Textur enthalten sind. Der Bereich des Himmels wird bei der Generierung des 3D–Modells nicht beachtet.

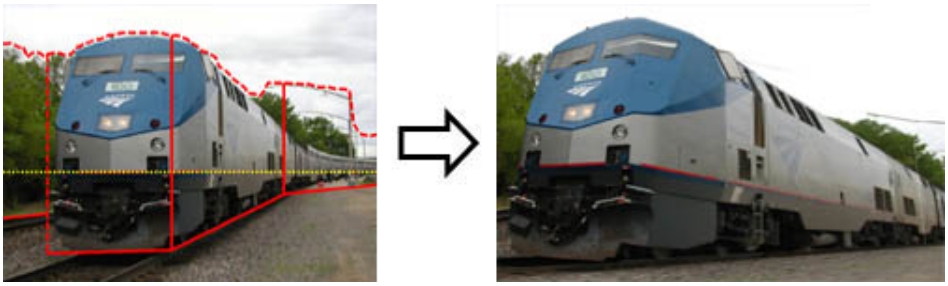

Abbildung 12: Erstellung des texturierten 3D–Modells

Ausgegeben wird ein fertig modelliertes und texturierts VRML–Modell in dem der Benutzer sein Bild „erkunden“ kann.

Teile die als senkrecht markierten Pixel in einzelne Regionen.
Für jede einzelne Region:

1. **Finde die „Ground“–„Vertical“–Grenzpunkte** $p(x, y)$
2. **Finde iterativ die besten Liniensegmente** bis kein Segment aus mehr als m_p Punkten besteht:
   (a) Finde die beste Linie L in p mittels der Hough–Transformation
   (b) Finde die größte Menge von Punkten $p_L \in p$ von L innerhalb der Distanz d_t, welche zwischen aufeinander folgenden Punkten keine Lücken größer g_t haben
   (c) Entferne p_L aus p
3. **Bilde eine Menge von Polylinien aus den Liniensegmenten**
   (a) Entferne das kleinere von vollständig überlappenden Segmenten (in x-Richtung)
   (b) Sortiere die Segmente nach der Mitte der Punkte auf dem Segment in x-Richtung
   (c) Verbinde zwei aufeinander folgende, sich schneidende Liniensegmente zu einer Polylinie, wenn sich der Schnittpunkt zwischen den Segmentmitten befindet
   (d) Entferne die kleinere von allen sich überlappenden Polylinien

Falte entlange der Polylinie
Schneide aufwärts an den Polylinienenden und entlang den „Ground“–„Sky“– bzw. „Vertical“–„Sky“–Grenzen
Projizieren die Ebenen in 3D–Koordinaten und texturiere das Modell

Abbildung 13: Algorithmus zur Erstellung des 3D–Modells anhand der geometrischen Klassen

6 Umsetzung

Für die Implementierung dieses Verfahrens wurde größtenteils die kommerzielle mathematische Software MATLAB der Firma The MathWorks[5] genutzt. Zusätzlich kam noch eine Statistik–Toolbox für den Lernalgorithmus und den Entscheidungsbaum zum Einsatz. Die Superpixel–Generierung wurde mit dem öffentlich Code von [Felz04] realisiert. Für das Lernen der einzelnen Wahrscheinlichkeitsfunktionen wurden zwanzig Adaboost–Iterationen durchgeführt.

Desweiteren wurden verschiedene Schwellenwerte für die Generierung des 3D–Modells festgelegt. So hat jedes Segment mindestens $s/20$ Grenzpunkte m_p, wobei s die Länge der Bilddiagonale ist. Zudem ist der Mindestabstand d_t, den ein Punkt haben muss, damit er noch für ein Segment in Betracht kommt, $s/100$. Die maximale horizontale Lücke g_t, die zwischen zwei aufeinanderfolgenden Punkten erlaubt ist, entspricht dem Maximum von der Länge des Segmentes und $s/20$ (vgl. Algorithmus in Abbildung 13).

In verschiedenen Tests mit unterschiedlicher Gruppierungsanzahl ($N_C = \{3, 4, 5, 6, 7, 9, 12, 15\}$) wurde festgestellt, dass der Kennzeichnungsalgorithmus auf Parameteränderungen oder kleine Änderungen in der Berechnung der Bildstatistiken ziemlich unempfindlich reagiert. Bei 62 verschiedenen Bildern hat er ca. 87% der Pixel korrekt mit „Ground“, „Vertical“ oder “Sky“ bezeichnet. Aus etwa jedem dritten Bild entstand ein anschauliches 3D–Modell. Die Fehlerquote hängt allerdings nicht nur von der korrekten Bezeichnung der Pixel ab. Selbst wenn das Bild zu 100% richtig bezeichnet wird, können unbrauchbare 3D–Modelle daraus entstehen. Das liegt zum einen daran, dass einige Bilder schlichtweg ungeeignet sind, zum anderen an eventuellen Verdeckungen, so dass sich Objekt–Grenzen nur sehr schwer oder gar nicht bestimmen lassen und somit verschiedene vertikale Objekte auf nur einer Ebene abgebildet oder einfach ignoriert werden. Ein weiterer Fehler könnte eine falsche Polylinie sein, so dass die Orientierung der einzelnen Ebenen im Raum und auch untereinander nicht mehr der Realität entspricht. Durch die anfangs getroffenen Annahmen, ist es momentan auch nicht möglich schräge oder mehrere boden–parallele Ebenen (Berge, Treppen, Plateaus, ...), sowie vollgestopfte Szene (Leute, Bäume, ...) zu modellieren, was wiederum zu einer fehlerhaften Darstellung führt. Auch eine schlechte Abschätzung der Horizontlinie führt zu einem verzerrten und somit falschen Modell. Führt man die Methode auf ein und dem selben Bild mehrmals aus, kann es durchaus vorkommen, dass leicht unterschiedliche Modelle entstehen. Dies hängt mit der zufälligen Initialisierung bei der Gruppierung der Superpixel zusammen.

[5] www.mathworks.com

Ein Athlon mit 2,13 GHz benötigt für ein Bild mit einer Auflösung von 800x600 Pixel ungefähr 1,5 Minuten, um ein VRML-Modell der Szene zu erstellen. Dabei wurde ein nicht–optimierter MATLAB-Code verwendet.

1. Bild → Superpixel durch die Over–Segmentierung (Kap. 3.1)
2. Superpixel → mehrere Gruppierungsmöglichkeiten (Kap. 3.2)
   (a) Berechne Merkmale für jedes Superpixel (Kap. 2)
   (b) Berechne die paarweisen Wahrscheinlichkeiten
   (c) Variiere die Anzahl der Gruppierungen:
   maximiere die durchschnittliche paarweise Wahrscheinlichkeit innerhalb jeder Gruppierung (Gleichung 1)
3. Gruppierungsmöglichkeiten → Superpixel–Bezeichnung (Kap. 3.3)
   (a) Für jede Gruppierung:
      i. Berechne die geometrische Merkmale (Kap. 2)
      ii. Berechne die Bezeichnungswahrscheinlichkeit
      iii. Berechne Homogen–Wahrscheinlichkeit
   (b) Für jedes Superpixel: Berechne Bezeichnungswahrscheinlichkeit und ordne die wahrscheinlichste zu
4. Superpixel–Bezeichnung → 3D–Modell (Kap. 5)
   (a) Teile senkrechte Region in einzelne Objekte
   (b) Für jedes Objekt: Finde eine Gerade für die Grenze zum Boden
   (c) Erstelle VRML–Modell durch Ausschneiden des Himmels und Hochklappen der Objekte

Abbildung 14: Erstellung eines VRML–Modells

6.1 Ergebnisse

Abbildung 15 auf der nächsten Seite zeigt ein paar Beispiele, die dieses Verfahren liefert. Zu sehen sind jeweils die Ausgangsbilder und Ansichten der Szene aus anderen Blickwinkeln, die mit Hilfe des automatisch generierten 3D–Modells gemacht wurden.

Abbildung 15: Automatic Photo Pop-up – Ergebnisse

7 Ausblick

Durch das Festlegen von gewissen Annahmen und der Nutzung eines statistischen Frameworks, wurde eine Methode entwickelt, die aus einem einzigen Bild ein relativ anschauliches, wenn auch einfaches, 3D-Modell erstellen kann. Dies ist allerdings erst der Anfang. Der nächste Schritt wäre das Anwendungsgebiet auf Innenaufnahmen zu erweitern oder generell die Effizienz des ganzen Verfahrens zu steigern. Dafür gibt es verschiedene Ansatzpunkte. So könnte zum Beispiel die Segmentierungstechnik verbessert werden, damit die Bezeichnung, besonders an den Regionsgrenzen, genauer wird oder auch um Vordergrundobjekte zu erkennen. Durch eine bessere und robustere Erstellung der Polylinie könnte außerdem die Orientierung der senkrechten Regionen im Raum besser aus dem Bild gewonnen werden.

Abschließend kann man sagen, dass die automatische Generierung von 3D–Modellen und damit die Erstellung virtueller Welten aus einzelnen Bildern, vorzugsweise privaten Fotografien des Heimanwenders, dem Betrachter eine vollkommen neue Dimension liefert, seine Bilder zu betrachten.

Literatur

[Debe96] Debevec, P.E., Taylor, C.J., und Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *ACM SIGGRAPH 96*, 11–20.

[Poll04] Pollefeys, M., Gool, L. V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., und Koch, R. 2004. Visual modeling with a hand-held camera. *Int. J. of Computer Vision 59*, 3, 207–232

[Lieb99] Liebowitz, D., Criminisi, A., und Zissermann, A. 1999. Creating architectural models from images. In *Proc. Eurographics*, Vol. 18, 39–50.

[Crim00] Criminisi, A., Reid, I., und Zisserman, A. 2000. Single view metrology. *Int. Journal of Computer Vision 40*, 2, 123–148

[Horr97] Horry, Y., Anjyo, K.-I., und Arai, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *ACM SIGGRAPH 97*, 225–232.

[Mart01] Martin, D., Fowlkes, C., Tal, D., und Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Int. Conf. on Computer Vision*, Vol. 2, 416–423.

[Kose02] Kosecka, J., und Zhang, W. 2002. Video compass. In *European Conf. on Computer Vision*, Springer-Verlag, 476–490.

[Felz04] Felzenszwalb, P., und Huttenlocher, D. 2004. Efficient graph-based image segmentation. *Int. Journal of Computer Vision 59*, 2, 167–181.

[Duda72] Duda, R., und Hart, P. 1972. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM 15*, 1, 11–15.

[Coll02] Collins, M., Schapire, R., und Singer, Y. 2002. Logistic regression, adaboost and bregman distances. *Machine Learning 48*, 1-3, 253–285.