

Oliver Lau

Magic Eye Movies

Bewegte Autostereogramme mit der Microsoft Kinect

Mitte der Neunzigerjahre war das „Magische Auge“ groß in Mode, eine spezielle Abbildungstechnik, die in scheinbar gleichförmigen Mustern Bilder mit 3D-Effekt versteckte. Dank der Tiefenkamera der Kinect lassen sich diese Bilder nun in Bewegung darstellen.

Großer Beliebtheit erfreut sich räumliche Darstellung nicht erst, seitdem Hollywood verstärkt 3D-Filme produziert. Bereits 1838 ergründete der britische Wissenschaftler Sir Charles Wheatstone das stereoskopische Sehen, also die Wahrnehmung räumlicher Tiefe durch das Betrachten der Umwelt mit zwei Augen. Dabei stellte er fest, dass sich der räumliche Eindruck durch die Unterschiede in der horizontalen Lage korrespondierender Bildpunkte aus linkem und rechtem Bild ein-

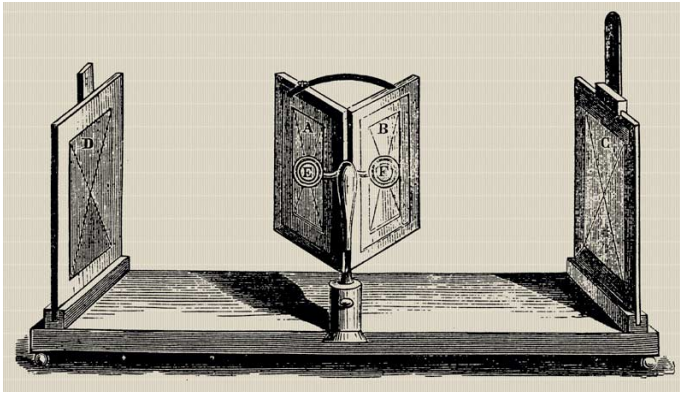
stellte. Mit dieser Erkenntnis bastelte er das erste Stereoskop, eine Konstruktion mit zwei Spiegeln, die den Blick trennten, womit das linke Auge auf eine Fotografie schaute, die aus einer leicht nach links versetzten Perspektive aufgenommen wurde, und das rechte auf ein Foto mit einer leicht nach rechts versetzten. Schaut man aus der richtigen Entfernung auf die Spiegel, setzt das Gehirn die beiden unterschiedlichen Fotos zu einem räumlichen Bild zusammen.

Elf Jahre später verbesserte der schottische Wissenschaftler David Brewster das Stereoskop, indem er die beiden Spiegel durch Linsen ersetzte, was eine wesentlich kompaktere Bauform erlaubte. Bei seinen Experimenten stellte er den sogenannten Tapeteneffekt fest: Starrt man auf wiederkehrende Muster, wie sie häufig auf Tapeten zu sehen sind, beginnt das Gehirn nach kurzer Zeit, die Muster paarweise zu vereinen. Dabei entsteht der Eindruck, dass die Muster zu

einem Objekt gehören, das sich auf einer gedachten Fläche hinter der tatsächlichen befindet.

Vereinfacht man das Tapeteneffektszenario so weit, dass das Muster auf ein winziges Quadrat reduziert wird und man nur noch eine horizontale Linie entlang der Wand betrachtet, gelangt man zu der Abbildung rechts in der Mitte. Darin stehen P1 und P2 für zwei Punkte auf dieser horizontalen Linie, die sich über die Wölbungen eines dreidimensionalen Objekts schlängelt. Stellt man nun eine durchsichtige Folie zwischen Betrachter und Objekt, lassen sich P1 und P2 durch Linien zum Auge auf der Folie abbilden. Wiederholt man das Verfahren für alle weiteren (diskreten) Punkte und für alle möglichen (diskreten) horizontalen Linien, ist das Stereogramm fertig.

Nun kann man die räumliche Szene entfernen, denn die Abbildung enthält sämtliche für den Raumeindruck relevanten Informationen daraus. Schaut man mit einem neutralen, das heißt auf unendlich fokussierten Blick



Das erste Stereoskop von 1838 teilt den Blick über zwei im Winkel von 45 Grad aufgestellte Spiegel auf zwei horizontal leicht versetzt aufgenommene Fotografien auf.

Beim vergleichsweise kompakten Brewster-Stereoskop von 1849 betrachtet man die beiden Fotos durch Linsen.



auf die Folie, sieht man die Szene trotzdem, denn das Gehirn setzt jedes Punktepaar wie beim Tapeteneffekt zu einem Punkt zusammen und ordnet ihnen eine Tiefe zu.

Derartige Stereogramme nennt man Autostereogramme oder Single-Image-Stereogramme (SIS), weil keine Hilfsmittel zum Betrachten notwendig sind und weil man nicht mehr zwei Bilder, sondern nur noch eines für den 3D-Effekt braucht.

Bereits 1994 haben wir das Thema in einem Artikel aufgegriffen [1]. Tiefenbilder waren seinerzeit rar oder mussten aufwendig produziert werden. Heute kommt man in Echtzeit an sie heran: Die Microsoft Kinect liefert 30 pro Sekunde – aus einer realen Szenerie. Damit kann man zum Beispiel sich selbst dreidimensional in einem Autostereogramm betrachten, und zwar bewegt wie in einem Film und nicht wie Anno Tobak statisch auf einem Bild.

Die für diesen Artikel in C++ programmierte Beispiel-Software „Bineqt“ setzt das in die Tat um. Sie erfordert das Framework Qt [2] und das Microsoft Kinect SDK [3] für Entwickler oder die Kinect Runtime für Anwender. Letzteres bedingt Windows als Betriebssystem. Alle Komponenten sowie der Quelltext stehen wie gewohnt unter dem c't-Link am Artikelende zum Download bereit. Wer nicht selbst kompilieren will, kann eine fertige EXE-Datei herunterladen.

Tiefenbilder

Die Software geht davon aus, dass der Betrachter frontal auf die Mitte der 3D-Szene sieht. Wie im Bild unten zu sehen, liegt die Abbildungsebene auf halber Strecke

zwischen Augen und dem entferntesten Punkt der 3D-Szene. Der Faktor μ gibt in Abhängigkeit von D den Punkt der 3D-Szene an, der am nächsten zum Betrachter liegt. Erfahrungsgemäß eignet sich ein Wert von $1/3$ bis $2/3$ am besten für μ .

Der Abstand s zwischen dem rechten und linken Abbildungspunkt ergibt nach den Ähnlichkeitssätzen für Dreiecke zu:

$$s = E \cdot (1 - \mu \cdot z) / (2 - \mu \cdot z)$$

Welche Farbe die Abbildungspunkte an den Koordinaten $\text{left} = (x - s, y)$ und $\text{right} = (x + s, y)$ haben, ist dem Prinzip nach egal, nur

identisch muss sie sein, damit beide Augen das Gleiche sehen.

Für den Fall, dass sich die Abbildungen mehrerer Punkte überdecken, müssen alle betroffenen Projektionspunkte die gleiche Farbe bekommen. Dafür sorgt ein Algorithmus, der ein Array mit ganzen Zahlen berechnet, das so viele Elemente hat, wie das zu erzeugende Autostereogramm in Pixeln breit sein soll.

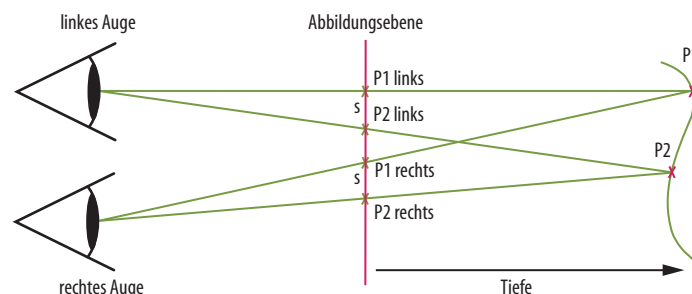
Darin wird in jedes Feld sein Index eingetragen. Das bedeutet, dass zu Beginn für einen Eintrag i mit dem Wert i im Feld keine Bedingung für die Abbil-

dung des korrespondierenden Tiefenpixels existiert. Hat der Eintrag einen anderen Wert (j), bekommt das Pixel i die gleiche Farbe wie das Pixel j . Verweisen alle Einträge im Array auf sich selbst oder auf Einträge rechts von sich ($j > i$), kann man das Array von rechts abarbeiten und die Indizes zum Erzeugen des SIS verwenden.

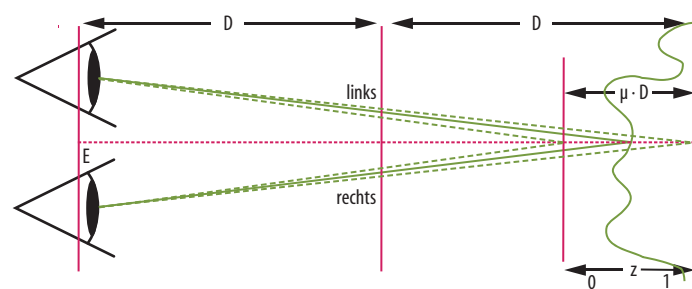
Die Implementierung berücksichtigt darüber hinaus die Sichtbarkeit von Pixeln im Tiefenbild: Verdeckt nämlich ein näher liegender Punkt einen ferner liegenden für ein Auge, muss dieser in der Projektion nicht beachtet werden und kann eine beliebige Farbe bekommen. Gleiches gilt für den Fall, dass ein Abbildungspunkt eines Paares außerhalb des SIS liegt.

Das sollte zum groben Verständnis des Algorithmus zum Erzeugen der Abbildungsvorschriften genügen. Die Details sind ausführlich in [1] beschrieben.

Dass Pixelpaare dieselbe Farbe erhalten müssen, ist nun klar, aber nicht, welche das ist. Im einfachsten Fall genügen zufällig ausgewählte schwarze oder weiße Pixel. Nicht ganz so gut funktionieren Pixel beliebiger Farbe, weil der Kontrast geringer ist. Aber auch Farbwerte aus Texturen sind möglich, wie das Titelbild demonstriert.



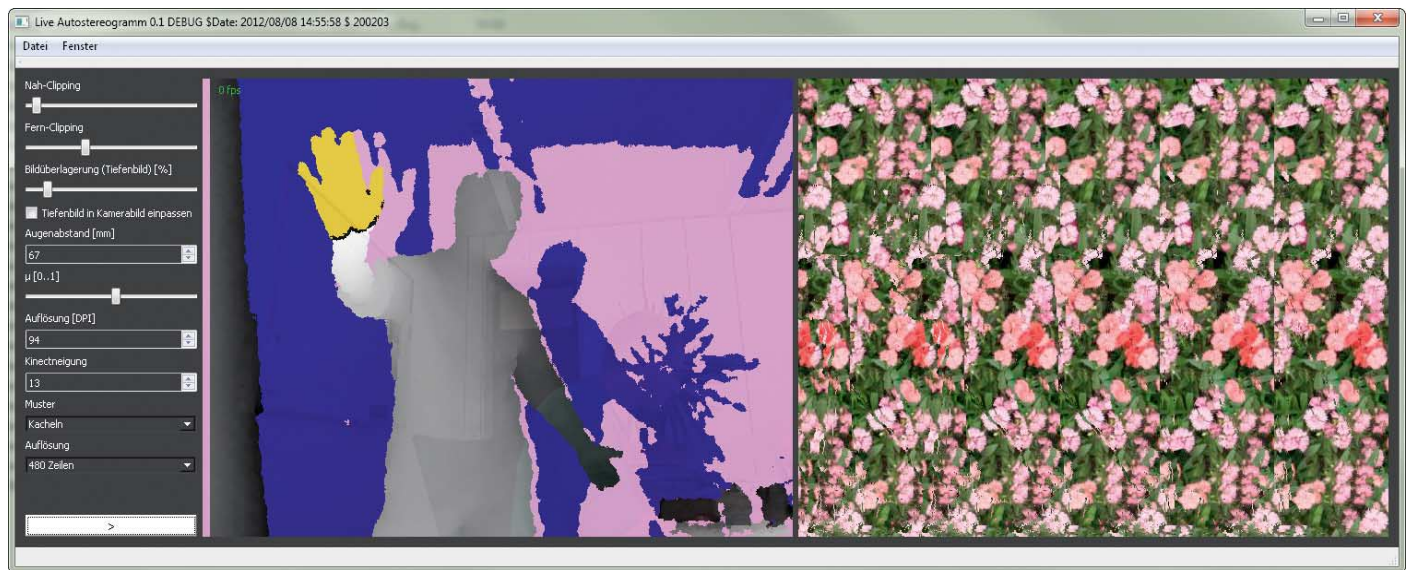
Rein mathematisch betrachtet, ergeben zwei Punkte, die man mit zwei Augen betrachtet, vier Punkte auf einer dazwischenliegenden gedachten Ebene. Das Gehirn fasst sie aber zu zweien zusammen und erzeugt für jeden davon einen Tiefeneindruck.



Mit den Rechenregeln für ähnliche Dreiecke lässt sich der Abstand der Abbildungspunkte leicht berechnen.

Bedienung von Bineqt

Die Beispielsoftware ist leicht zu bedienen. Die Schieber *Nah-Clipping* und *Fern-Clipping* begrenzen den Tiefenbereich nach vorne und hinten. Damit lassen sich unerwünschte im Vorder- oder Hintergrund befindliche Objekte aus der Szenerie entfernen. Gleichzeitig verbessert sich die Tiefenauflösung.



Die Beispielanwendung „Bineqt“ vereint die Tiefenbilder der Kinect mit einem Muster nach Wahl und berechnet daraus – vorausgesetzt, der Rechner ist schnell genug – in Echtzeit Autostereogramme.

Vordergründig weggeschnittene Punkte werden im *Tiefenbild* (links oben) gelb dargestellt, hintergründig weggeschnittene dunkelblau, die dazwischen in Graustufen, wobei die Tönung mit der Entfernung zunimmt. Punkte, denen das Kinect SDK keine Tiefe zuordnen kann (Reflexionen, zu weit weg), erscheinen in Pastellviolet.

Die Kinect liefert zum Tiefenbild ein Bild der eingebauten Kamera und überlagert beide. Wie stark das Kamerabild durch das Tiefenbild hindurchscheinen soll, bestimmt der Schieber *Bilderüberlagerung*.

Markiert man die Checkbox *Tiefenbild in Kamerabild einpassen*, gleicht das den Parallaxenfehler zwischen Tiefenbild und Kamerabild aus, sodass die Pixel deckungsgleich übereinander liegen.

Im Feld *Augenabstand* steht, welchen Abstand die Sehachsen des Betrachters haben. Der voreingestellte Wert von 67 mm hat sich für die meisten Menschen als brauchbar erwiesen.

Der Schieber μ [0..1] legt in Abhängigkeit vom Betrachtungsabstand den Punkt der 3D-Szene fest, der am nächsten zum Betrachter liegt.

Über *Auflösung* stellt man ein, mit wie viel dpi (Punkte pro Zoll) das *Autostereogramm* (rechts) aufgelöst sein soll. Voreingestellt sind 96 dpi, was zu den meisten Monitoren passt.

Die Kinect hat einen Motor im Fuß, über den man ihre Neigung

verstellen kann. Der im Feld *Kinectneigung* eingetragene Wert (–27 bis +27) gibt sie in Grad vor.

Die Drop-down-Box *Muster* darunter bestimmt, auf welcher Grundlage die Pixelfarben des SIS ausgewählt werden. Zur Auswahl stehen Textur (gekachelt oder aufgespannt), Schwarz-Weiß-Rauschen und Farbrauschen.

In der untersten Auswahlliste lässt sich die *Auflösung* des SIS einstellen. Der Knopf ganz unten startet und stoppt den Datenstrom aus der Kinect.

Zieht man ein Bild im PNG- oder JPG-Format auf das Tiefenbild, ersetzt es das Bild von der Kinect. Bei Drag & Drop eines Bildes auf das Autostereogramm wird dessen Inhalt als Textur verwendet.



Ein Rauschbild sieht zwar unspektakulär aus, aber die darin enthaltene Räumlichkeit ist für die meisten Menschen so am leichtesten stereoskopisch zu erfassen.

Man kann auch vertikale Ausschnitte des Kamerabilds zur Texturierung heranziehen. Der gewünschte Ausschnitt wird einfach mit der Maus im Tiefenbild markiert. Ein Klick auf das Tiefenbild setzt die Auswahl zurück und es kommt wieder die zuvor gewählte Textur zum Einsatz.

Tiefenbild und SIS lassen sich über den Menüpunkt „Datei“ als PNG oder JPG abspeichern.

Wer zwei Bildschirme hat, kann auf das Autostereogramm doppelklicken und es damit flächendeckend auf dem primären Monitor darstellen.

Live auf der IFA

Sie können die Software übrigens nicht nur zu Hause ausprobieren. Auf unserem IFA-Stand (Halle 17,

Stand 119) präsentieren wir eine „Bineqt“-Installation, auf der Sie sich selbst live auf einem Großbildschirm in 3D betrachten können. Einen Ausdruck des Autostereogramms Ihrer Lieblingspose geben wir Ihnen zum Einrahmen kostenlos mit nach Hause. (ola)

Literatur

- [1] Ute Clausen, Josef Pöpsel, Im Rausch der Tiefe, Autostereogramme im Eigenproduktion, c't 7/94, S. 230
- [2] Harald Bögeholz, Norweger Messer, Plattformübergreifend programmieren in C++ mit Qt, c't 15/09, S. 186
- [3] Oliver Lau, Natürlich steuern, Einführung in Microsofts Kinect SDK, c't 24/11, S. 184

www.ct.de/1219164

ct

