

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN
INSTITUT FÜR INFORMATIK I



Alexander Thurn

**Projektionen mit
schönen Eigenschaften**

18. April 2007

Seminararbeit im SS 2007

Zusammenfassung

Diese Arbeit befasst sich mit der Berechnung von *schönen* perspektivischen Projektionen von Punkten und Liniensegmenten im dreidimensionalen Raum. *Schön* bedeutet hierbei, dass sie eine Ausgabe von Objekten in allgemeiner Lage erzeugen. Auf der Arbeit von F.Gomez [2] aufbauend werden Projektionen beschrieben die x eindeutige, nicht kollineare und nicht kozyklische Punktmengen erzeugen sowie Projektionen die Liniensegmente so projizieren dass diese nicht parallel sind.

Inhaltsverzeichnis

1	Einführung	2
2	Grundlagen	2
2.1	Perspektivische Projektion	2
2.2	Reduktionsbeweise	4
3	Projektionen von Punktmengen	5
3.1	Projektionen mit eindeutigen x Koordinaten	5
3.1.1	Existenz	5
3.1.2	Berechnung	6
3.2	Projektionen mit eindeutigen x und y Koordinaten	8
3.2.1	Existenz	9
3.2.2	Berechnung	9
3.3	Nicht-Kollineare Projektionen	9
3.3.1	Existenz	12
3.3.2	Berechnung	12
3.4	Nicht kozyklische Projektionen	14
3.4.1	Existenz	16
3.4.2	Berechnung	16
4	Projektionen von Liniensegmenten	17
4.1	Nichtparallele Projektionen von Liniensegmenten	17
4.1.1	Existenz	19
4.1.2	Berechnung	20
5	Fazit	20

1 Einführung

In vielen Algorithmen der Algorithmischen Geometrie wird die Annahme gemacht, dass die beteiligten Punkte oder Liniensegmente sich in „allgemeiner Lage“ befinden. Dies hat den Sinn, dass so bestimmte degenerierte Fälle ausser Acht gelassen werden können und der Algorithmus überhaupt korrekt ausgeführt werden kann. Eine allgemeine Lage kann dabei von Fall zu Fall etwas anderes bedeuten. Z.B. sollen alle x Koordinaten einer Punktmenge eindeutig sein, keine drei Punkte auf einer Geraden liegen oder auch keine vier Punkte auf einem Kreis liegen dürfen. Leider kann es in der Realität vorkommen dass diese Bedingung nicht immer erfüllt ist. Man kann also nicht direkt davon ausgehen, dass die berechnete theoretische Komplexität eines Algorithmus, der eine solche Bedingung inne hat, auch in der Praxis die gleiche Laufzeit besitzt oder überhaupt korrekt terminiert.

Eine Idee zur Sicherstellung der allgemeinen Lage ist es einen PreProcessing Schritt durchzuführen. Dazu wird eine Transformation auf die Objektmenge angewandt, die die Objekteigenschaften insofern verändert, dass der gewünschte Algorithmus durchführbar ist (die Objekte sich also in allgemeiner Lage befinden) und das gewünschte gleiche Ergebnis liefert, das man auch ohne die Transformation erhalten sollte. F.Gomez [2] hat sich mit einer speziellen Transformation, der perspektivischen Projektion von dreidimensionalen Punktmenge und Liniensegmenten auf eine 2 dimensionale Ebene beschäftigt. Die abgebildeten Objekte sollen sich dann auf der Ebene in allgemeiner Lage befinden. In seiner Arbeit erörtert er 5 Projektionen mit schönen Eigenschaften.

1. Projektionen mit eindeutigen x Koordinaten
2. Projektionen mit eindeutigen x und y Koordinaten
3. Nicht-Kollineare Projektionen
4. Nicht-Kozirkulare Projektionen
5. Keine parallelen Liniensegmente

Auf diese soll in dieser Arbeit näher eingegangen werden.

2 Grundlagen

2.1 Perspektivische Projektion

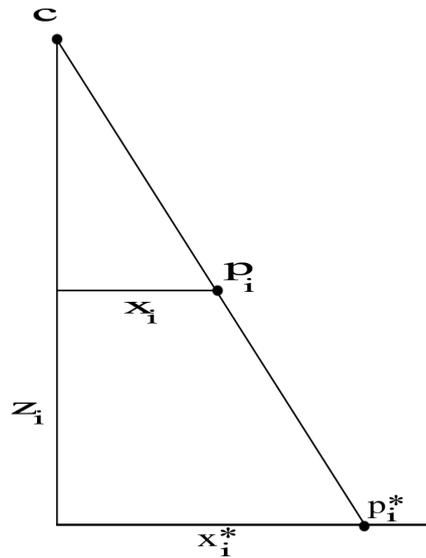
Die in den folgenden Kapiteln vorgestellten Algorithmen basieren alle auf einer gemeinsamen Technik, der perspektivischen Projektion. Im Besonderen ist die Zentralprojektion relevant. Darum machen wir uns als erstes klar, was es damit auf sich hat.

Definition 1

Sei Π die Projektionsebene und $c \in \mathbb{R}^3 \setminus \Pi$, dann ist die **Zentralprojektion** des Punktes $p \in \mathbb{R}^3$ definiert als der eindeutig bestimmte Schnittpunkt der Geraden $c + \lambda \overline{cp}$ mit der Ebene Π

Ein Spezialfall der Zentralprojektion ist die Parallelprojektion, bei der das Zentrum unendlich weit entfernt liegt und die Projektionsstrahlen damit parallel verlaufen.

Sei $c = (x_c, y_c, z_c) \in \mathbb{R}^3$ und Π gleich der xy -Ebene. Gesucht sind nun für alle Punkte p der Menge $P = \{p_1, p_2, \dots, p_n\}$ mit $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ die Koordinaten der projizierten Bildpunkte $P^* = \{p_1^*, p_2^*, \dots, p_n^*\}$ mit $p_i^* = (x_i^*, y_i^*, 0) \in \mathbb{R}^3$



Betrachtet man die x- und y- Abschnitte getrennt, ergibt sich auf Grund der Strahlensätze folgende Beziehung:

$$x_i^* = \frac{z_c * (x_i - x_c)}{z_c - z_i} + x_c \quad (1)$$

$$y_i^* = \frac{z_c * (y_i - y_c)}{z_c - z_i} + y_c \quad (2)$$

OBdA liege das Projektionszentrum auf der negativen z-Achse im Punkt $c = (0, 0, z_c)$. Dann vereinfacht sich das ganze zu:

$$x_i^* = \frac{z_c}{z_c - z_i} x_i \quad (3)$$

$$y_i^* = \frac{z_c}{z_c - z_i} y_i \quad (4)$$

Folgende Abbildung beschreibt die gewünschte Projektion

$$(x_i, y_i, z_i) \mapsto \left(\frac{z_c}{z_c - z_i} x_i, \frac{z_c}{z_c - z_i} y_i, 0 \right) \quad (5)$$

2.2 Reduktionsbeweise

In den folgenden Kapiteln werden des öfteren Reduktionen benutzt. Das Ziel einer Reduktion ist es Relationen verschiedener Probleme aufzuzeigen. Lässt sich z.B. Problem A auf Problem B so *umwandeln*, dass die Lösung von B auch die Lösung zu A aufzeigt, kann man mit Lösen von B auch A lösen. Hat nun der Lösungsalgorithmus von B eine Laufzeit von z.B. $O(n \log n)$ und A lässt sich in $O(n)$ auf B umformen, so kann man A natürlich in $O(n \log n)$ lösen. Kennt man nun die minimale Laufzeit von A sicher und die von B nicht, und lässt sich A in B schnell genug umformen, so kann man mit Sicherheit sagen dass B nicht schneller als A lösbar ist, da man sonst A schneller lösen könnte.

Definition 2 Gegeben zwei Probleme A und B , A ist $f(n)$ lösbar durch B , wenn jede Problemstellung aus A der Größe n mit einer konstanten Anzahl an Problemstellungen aus B und $O(f(n))$ zusätzlicher Zeit lösbar ist.

Ein Problem das in dieser Arbeit für Reduktionen benutzt wird, ist das sogenannte *3-SUM* Problem:

Gegeben eine Menge S von n ganzen Zahlen.
Gibt es $a, b, c \in S$ mit $a + b + c = 0$?

Der beste bekannte Algorithmus für dieses Problem benötigt $\Theta(n^2)$ Zeit. Lässt sich also *3-SUM* durch das Lösen eines anderen Problems lösen, so ist die Laufzeit von diesem mindestens $\Omega(n^2)$.

Ein weiteres Problem mit diesmal sicherer bekannter unterer Grenze ist das sogenannte *Element-Eindeutigkeits* Problem.

Gegeben eine Menge S von n reellen Zahlen $\{x_1, \dots, x_n\}$.
Gibt es Indizes $i \neq j$ mit $x_i = x_j$?

Dieses Problem lässt sich bewiesenermaßen in $\Theta(n \log n)$ lösen.

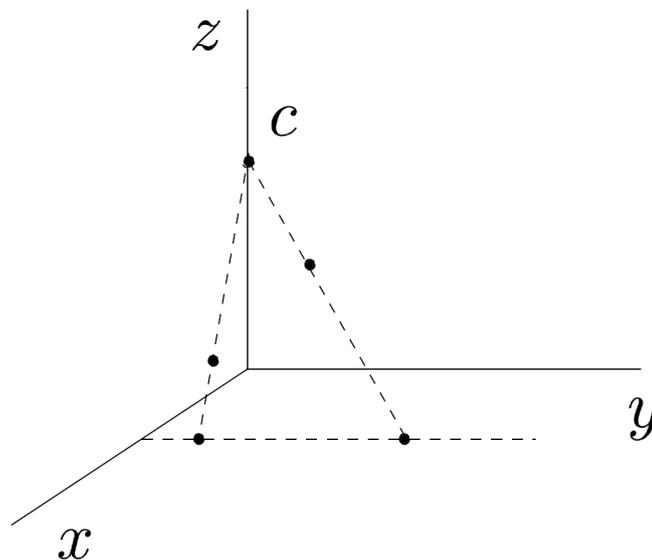
3 Projektionen von Punktmenge

3.1 Projektionen mit eindeutigen x Koordinaten

Wir betrachten die in Abschnitt 2.1 definierte Gleichung 5. Gegeben sei also eine Menge $P = \{p_1, p_2, \dots, p_n\}$ mit $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ und gesucht sei ein Zentrum $c = (0, 0, z_c)$, so dass die auf P angewandte Zentralprojektion eindeutige x -Koordinaten in P^* ergibt.

3.1.1 Existenz

Als erstes stellt sich die Frage, ob es immer eine solche Projektion gibt oder ob es eventuell Punktmenge P gibt, für die sich keine perspektivische Projektion mit eindeutigen x^* Koordinaten finden lässt. Antwort auf diese Frage liefert Gleichung 3. Man sieht, dass x^* lediglich von z_c, z_i und x_i abhängt. Seien p_i und p_j zwei Elemente aus P , die sich nur durch ihren y Wert unterscheiden, also $z_i = z_j, x_i = x_j$ und $y_i \neq y_j$. Dann lässt sich kein $c(z_c)$ finden, dass diesen beiden Punkten unterschiedliche x^* Koordinaten zuweist. Anschaulich handelt es sich um Punkte, die in einer gemeinsamen Ebene liegen, die die xy Ebene parallel zur y -Achse schneidet.



Bei n Punkten kann es $O(n^2)$ viele solcher Fälle geben, es gibt also $O(n^2)$ viele verbotene Ebenen. Da es sich hierbei um eine endliche Anzahl handelt, und Ebenen eine Dicke von 0 haben, lässt sich daraus schließen dass es immer eine perspektivische Projektion mit eindeutigen x^* -Koordinaten geben muss.

3.1.2 Berechnung

Da wir jetzt wissen dass sich immer eine solche Projektion finden lässt, stellt sich als nächstes die Frage, in welcher Laufzeit dies möglich ist.

Theorem 3 *Gegeben sei eine Menge von n verschiedenen Punkten für die gelte: Keine zwei Punkte liegen in einer gemeinsamen Ebene, die die xy Ebene parallel zur y -Achse schneidet. Dann kann eine Projektion mit eindeutigen x -Koordinaten in $O(n \log n)$ berechnet werden.*

Beweis. Man mache sich klar, dass die angewandte Projektion genau dann zwei Punkten die gleiche x -Koordinate zuweist, wenn diese mit dem Zentrum c , projiziert auf die xz -Ebene, auf einer Gerade liegen, diese also in der xz -Ebene kollinear sind. Da es sich um n Punkte handelt, gibt es $O(n^2)$ viele solcher Geraden.

Betrachten wir nun die Schnittpunkte dieser Geraden mit der z -Achse. Diese Schnittpunkte ergeben eine Menge von verbotenen Zentren. Eine erste Idee wäre es nun, diese Menge C_{verb} zu berechnen und alle anderen Punkte als die gesuchte Lösung anzunehmen. Dies würde zu einer quadratischen Laufzeit führen. Wie wir gleich sehen werden, braucht man allerdings nicht alle möglichen verbotenen Zentren zu betrachten.

Sei

$$z_c > \max\{z(p_i) | i = 1, \dots, n\}$$

wobei $z(p_i)$ die z -Koordinate von p_i , und $c(z_c) = (0, 0, z_c)$ ein Punkt auf der z -Achse ist. Ist die Projektion P^* mit Zentrum $c = c(z_c)$ x -eindeutig, haben wir unser gesuchtes Zentrum gefunden.

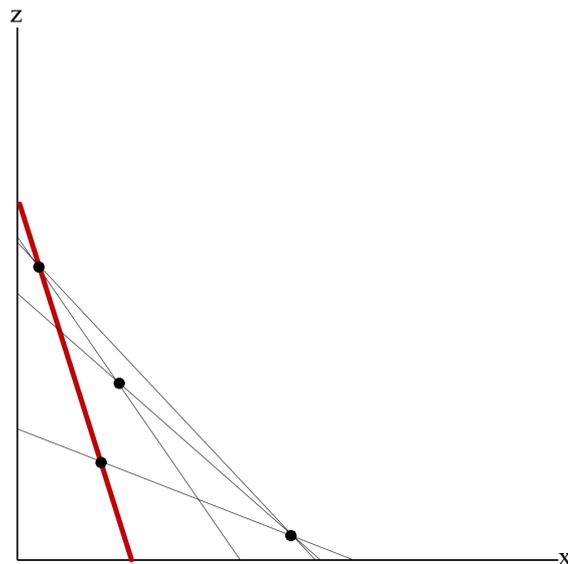
Nehmen wir an, dies wäre nicht so, dann verschieben wir unser Zentrum $c(z)$ mit $z > z_c$ kontinuierlich weiter, bis wir ein gültiges Zentrum gefunden haben. Sei $c(z_d)$ das nächstgelegene verbotene Zentrum zu $c(z_c)$. Da es sich bei z_c um ein verbotenes Zentrum handelt, gibt es zwei nach x -Koordinaten sortierte Punkte $p_{i_j}^*(z_c), p_{l_{j+1}}^*(z_c)$, deren x Werte identisch sind. Betrachtet man nun alle x -sortierten Punkte, stellt man fest dass sich an dieser Stelle die Reihenfolge der sortierten Punkte ändert, nämlich genau bei den von uns betrachteten Punkten mit den Indizes i und l .

Dies tritt natürlich auch ein, wenn wir das verbotene Zentrum bei z_d betrachten. Es gibt zwei sortierte Punkte $p_{n_o}^*(z_d), p_{m_{o+1}}^*(z_d)$, die einen gleichen x Wert besitzen. Die Reihenfolge der nach x -Koordinaten sortierten Punkte ändert sich. Eine Änderung tritt immer dann auf, wenn zwei Punkte den gleichen x -Wert besitzen und dies wiederum, wenn wir mit unserem $c(z)$ eine verbotene Projektion erzeugen. Die beiden Punkte $p_{n_o}^*(z_d), p_{m_{o+1}}^*(z_d)$ müssen schon bei $c(z_c)$ konsekutiv gewesen sein, da es sonst ein anderes $z \in]z_c, z_d[$ mit einer verbotenen Projektion mit Zentrum $c(z)$ geben würde, und $c(z_d)$ damit nicht mehr das nächste zu $c(z_c)$ gelegene verbotene Zentrum

wäre. Dementsprechend muss man lediglich die sortierten Punktepaare betrachten die in $c(z_c)$ konsekutiv sind und das nächstgelegene zu z_c wählen. Als Ergebnis erhält man ein offenes Liniensegment von gültigen Zentren, nämlich $\overline{c(z_c)c(z_d)}$.

Als Laufzeit ergibt sich: Die Projektion benötigt $O(n)$ Laufzeit und Speicherplatz. Anschließend werden die n Punkte nach x Wert sortiert und auf x -Eindeutigkeit überprüft, was $O(n \log n)$ Zeit benötigt. Findet man Duplikate, müssen n viele verbotene Zentren berechnet werden, was wiederum in $O(n)$ erfolgt. Das nächstgelegene Zentrum zu z_c lässt sich auch in $O(n)$ ermitteln. Damit ergibt sich eine Gesamtlaufzeit von $O(n \log n)$. \square

In [2] wird noch ein weiterer Algorithmus vorgestellt, der zwar eine um einen konstanten Faktor höhere Laufzeit hat, jedoch ein unbegrenztes Intervall an gültigen Projektionszentren liefert. Dies kann unter Umständen erwünscht sein.



Beweis. Als erstes ändere das Koordinatensystem insofern ab, dass alle Punkte aus P im ersten Oktanten liegen. Danach rotiere P , sodaß die Orthogonal-Projektion auf die xz Ebene keine Duplikate mehr erhält. Dies kann in $O(n \log n)$ getan werden, siehe [3]. Man betrachte die orthogonale Projektion einer Punktmenge auf die xz Ebene, hier P' genannt. Wie im vorherigen Beweis benutze man die Tatsache, dass zwei Punkte nach der perspektivischen Projektion die gleiche x Koordinate besitzen, wenn diese auf der xz Ebene mit dem Zentrum $c(z_c) = (0, 0, z_c)$ kollinear sind.

Um ein gültiges Zentrum zu finden, führe folgende Schritte durch:

Sei a die Steigung zwischen zwei Punkten aus P'

1. Ermittle die maximale Steigung a_{max} zwischen zwei Punkten aus P'
2. Finde ein $c(z_c) \in (0, 0, z_c)$ so daß folgende Bedingung erfüllt ist:

$$\forall p' \in P' : a(c, p') > a_{max}$$

Genauer gesagt reicht es aus wenn a_{max} eine obere Schranke aller Steigungen ist. Da $a(c, p')$ für alle $p' \in P'$ größer ist als die Steigung zwischen zwei Punkten aus P' , können keine zwei Punkte aus P' mit $c(z_c)$ auf einer Geraden liegen, da sonst für zwei beliebige Punkte $p'_i, p'_j \in P'$ gelten müsste:

$$a(c, p'_i) = a(p'_i, p'_j) = a(c, p'_j)$$

Damit erhält man eine x eindeutige perspektivische Projektion. Schauen wir uns die Schritte etwas genauer an:

1. Um die maximale Steigung zu ermitteln betrachte die nach x Koordinaten sortierten Punkte in P' . Wähle die kleinste x -Differenz (ungleich Null) zweier konsekutiver Punkte, genannt x_{min} . Dies nimmt $O(n \log n)$ viel Zeit in Anspruch. Als nächstes berechne die maximale z -Differenz zweier Punkte, genannt z_{max} . Diese z -Differenz ist im Gegensatz zur x -Differenz von allen Punkten aus zu sehen, nicht nur zwischen direkten Nachbarn ($O(n)$). Jetzt lässt sich eine obere Schranke der Steigungen bestimmen als

$$a_{max} = \frac{z_{max}}{x_{min}}$$

2. Sei p_a der Punkt, der das Maximum aller x -Werte der Punkte als x und das Maximum aller y Koordinaten aus P' als y Koordinate besitzt. Lege nun eine Gerade mit Steigung $-a_{max}$ durch diesen Punkt und berechne den Schnittpunkt mit der z -Achse. Dadurch ist gewährleistet, dass alle Steigungen der Geraden von c durch die Punkte aus P' eine Steigung größer als a_{max} besitzen. Der Schnittpunkt ist das gesuchte Zentrum. $O(1)$

Insgesamt ergibt sich eine Gesamtlaufzeit von $O(n \log n)$. □

3.2 Projektionen mit eindeutigen x und y Koordinaten

Auf der Idee aus Abschnitt 3.1 aufbauend lässt sich leicht die Frage nach einer immer zu findenden Projektion mit eindeutigen x und y Koordinaten beantworten und auch den entsprechenden Algorithmus zur Berechnung eines gültigen Zentrums angeben.

3.2.1 Existenz

Bei n Punkten kann es wie in Abschnitt 3.1 gezeigt $O(n^2)$ viele Ebenen geben, die die xy Ebene parallel zur y -Achse schneiden. Die selbe Anzahl gibt es natürlich auch an verbotenen Ebenen, die die xy Ebene parallel zur x -Achse schneiden. Da es sich hierbei wieder um eine endliche Anzahl handelt, und Ebenen eine Dicke von 0 haben, lässt sich daraus schließen dass eine perspektivische Projektion mit eindeutigen x und y Koordinaten immer zu finden ist.

3.2.2 Berechnung

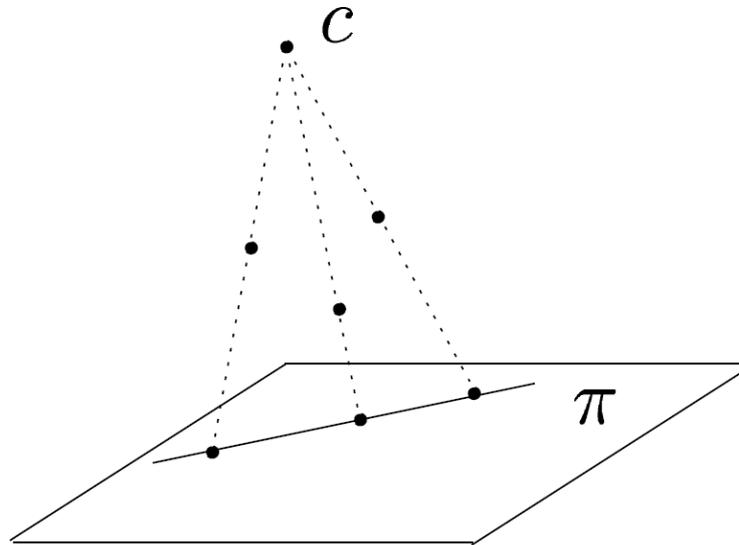
Theorem 4 *Gegeben sei eine Menge n paarweise disjunkter Punkte für die gelte: Keine zwei Punkte liegen auf einer Geraden die parallel zur y Achse ist und auf keiner Geraden, die parallel zur x Achse ist. Dann kann eine Projektion mit eindeutigen x und y Koordinaten in $O(n \log n)$ berechnet werden.*

Beweis. Man führe zuerst den Algorithmus aus Beweis 1 aus Abschnitt 3.1 durch, sodaß man ein offenes Liniensegment $\overline{c(z_c)c(z_d)}$ an gültigen x -eindeutigen Zentren erhält. Dies benötigt wie schon gezeigt $O(n \log n)$ Zeit.

Wähle nun ein z_e mit $z_c < z_e < z_d$. Handelt es sich bei $c(z_e)$ um ein gültiges Zentrum, haben wir unser gesuchtes Zentrum gefunden. Ansonsten suche das nächstgelegene für y verbotene Zentrum $c(z_f)$. Dies kann mit dem gleichen Ansatz wieder in $O(n \log n)$ berechnet werden. Sei $z_g = \min\{z_d, z_f\}$, dann sind alle Punkte aus dem offenen Liniensegment $\overline{c(z_e)c(z_g)}$ gültige Zentren. \square

3.3 Nicht-Kollineare Projektionen

Als nächstes befassen wir uns mit Projektionen die eine gegebene Punktmenge P auf eine Ebene so abbilden, dass die projizierte Menge P^* keine drei Punkte enthält, die auf einer gemeinsamen Geraden liegen, also kollinear sind. Diese Eigenschaft hängt lediglich davon ab, wie das Projektionszentrum gewählt wird, und nicht etwa von der Lage der Ebene im Raum. Daher lässt sich wieder die xy Ebene als Projektionsebene wählen.



Zu beachten ist, dass keine gültige Projektion gefunden werden kann, wenn drei Punkte schon im Raum kollinear sind. In [3] wird ein Algorithmus beschrieben, der dies in $O(n^2)$ feststellen kann.

Theorem 5 *Gegeben sei eine Menge von n unterschiedlichen Punkten. Dann lässt sich in $O(n^2)$ Zeit und Platz entscheiden, ob ein gegebenes Zentrum eine nicht-kollineare Projektion erzeugt.*

Beweis. Wie lässt sich prüfen, ob zwei Punkte aus P^* kollinear sind? Dazu betrachte die dualisierte Menge an Geraden. Der Punkt (a, b) wird dabei abgebildet auf die Gerade $y = ax + b$.

$$(a, b) \mapsto y = ax + b$$

Aus dem Schnittpunkt zweier Geraden dieser dualen Menge lässt sich die Gerade ablesen, die durch a und b verläuft. Schneiden sich also drei beliebige Geraden in diesem dualen Raum in ein und der selben Koordinate, existiert eine Gerade, die durch drei Punkte verläuft, diese also kollinear sind. In [1] findet sich ein Algorithmus, der die Dualisierung und das Überprüfen von n Punkten auf Kollinearität mit dieser Methode in $O(n^2)$ durchführt.

Als erstes ist also in $O(n^2)$ zu prüfen, ob sich im Raum drei kollineare Punkte befinden. Ist dem so, kann keine gültige perspektivische Projektion existieren. Falls nicht, überprüft man wie oben beschrieben in $O(n^2)$ ob die projizierte Punktmenge drei kollineare Punkte besitzt. Insgesamt ergibt sich also eine Laufzeit von $O(n^2)$. \square

Alternativ lässt sich Kollinearität dreier Punkte auf einer gemeinsamen Ebene auch folgendermaßen feststellen.

Lemma 6 Seien $p_i, p_j, p_z \in \mathbb{R}^2$ gegeben. Wenn gilt

$$\begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_z & y_z & 1 \end{vmatrix} = 0,$$

dann sind die 3 Punkte zueinander kollinear.

Beweis. Determinante = 0

⇒ Matrix nicht invertierbar

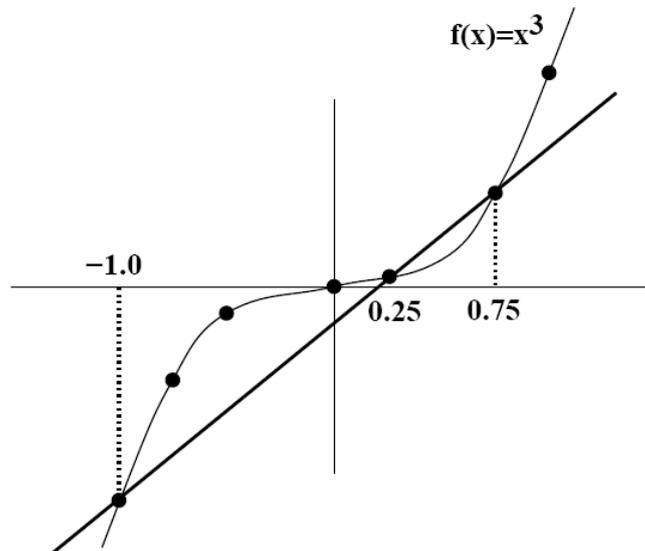
⇒ Zeilen linear abhängig

⇒ Es lässt sich eine Zeile als Linearkombination der anderen Beiden schreiben

⇒ Kollinear □

Weiterhin lässt sich zeigen, dass die Entscheidung, ob ein Zentrum eine nicht kollineare Projektion erzeugt, 3-SUM hart ist.

Theorem 7 Gegeben eine Menge von n paarweise disjunkten Punkten. Zu entscheiden ob ein Zentrum eine gültige nicht-kollineare Projektion erzeugt, ist 3-SUM hart.



Beweis. Beweis durch Reduktion. Sei $\{m_1, \dots, m_n\}$ die Menge ganzer Zahlen aus 3-SUM. Die Entscheidung ob es drei ganze Zahlen m_i, m_j, m_k gibt mit $m_i + m_j + m_k = 0$ lässt sich auf das Kollinearitätsproblem auf der Ebene folgendermaßen reduzieren. Dazu bilden wir die Zahlen auf Punkte in der Ebene folgendermaßen ab.

$$m_i \rightarrow (m_i, m_i^3)$$

Sind nun drei Punkte $(m_i, m_i^3), (m_j, m_j^3), (m_k, m_k^3)$ in der Ebene kollinear, so gilt $m_i + m_j + m_k = 0$. Dies lässt sich durch triviale Umformungen unter Zuhilfenahme von Lemma 6 zeigen. Könnten wir also das Kollinearitätsproblem in der Ebene schneller als in $O(n^2)$ lösen, könnten wir auch β -SUM schneller lösen. Als letztes transformieren wir noch die Punkte in der Ebene so, dass diese dem Ausgangsproblem entsprechen.

$$(m_i, m_i^3) \rightarrow \left(\frac{d-i}{d} m_i, \frac{d-i}{d} m_i^3, i \right)$$

Die perspektivische Projektion mit Zentrum $(0, 0, d)$ liefert damit genau wieder (m_i, m_j) , womit gezeigt ist dass das Problem β -SUM hart ist. □

3.3.1 Existenz

Eine ungültige Projektion erhält man genau dann, wenn das Projektionszentrum in einer gemeinsamen Ebene mit drei beliebigen Punkten aus P liegt. Bei n vielen Punkten gibt es $O(n^3)$ viele solcher Ebenen, und damit wieder eine endliche Anzahl von Ebenen, die alle eine Dicke von Null besitzen. Damit existiert immer eine nicht-kollineare perspektivische Projektion.

3.3.2 Berechnung

Theorem 8 *Gegeben eine Menge von n paarweise disjunkten Punkten. Dann lässt sich eine nicht-kollineare Projektion in $O(n^2)$ Zeit und $O(n^2)$ Platz berechnen.*

Beweis. Sei $P = \{p_1 = (x_1, y_1, z_d), \dots, p_n = (x_n, y_n, z_n)\}$ eine Menge von nicht kollinearen Punkten im Raum. Zunächst berechnet man wie in Abschnitt 3.1 eine x -eindeutige Projektion auf die xy Ebene. Wir erhalten dabei ein offenes Liniensegment an gültigen Zentren. Weiterhin können wir alle Punkte durch eine Translation so verschieben, dass gilt

$$x_i > 0, y_i > 0$$

und

$$z_c > \max\{z_i \mid i = 1, \dots, n\} > 0$$

Als erstes versuchen wir die Projektion mit $c(z_c) = (0, 0, z_c)$ durchzuführen. Handelt es sich um ein gültiges Zentrum, sind wir fertig. Ansonsten sei $c(z) = (0, 0, z)$ ein Punkt auf der z -Achse mit $z > z_c$. Ein Punkt p_i wird dann folgendermaßen abgebildet

$$(x_i, y_i, z_i) \mapsto \left(\frac{z}{z-z_i} x_i, \frac{z}{z-z_i} y_i, 0 \right)$$

Findet man das nächste zu z_c auf der z -Achse gelegene verbotene Zentrum, hätte man wieder ein offenes Liniensegment an gültigen Projektionszentren gefunden. Dazu bemerke man, daß wenn man z gegen unendlich gehen lässt, p_i^* gegen p_i konvergiert.

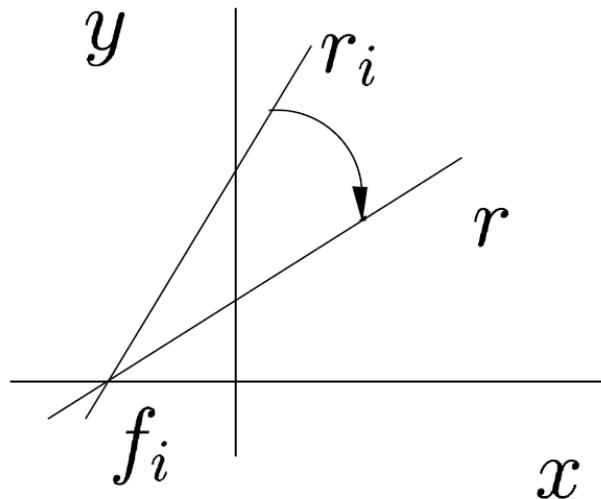
$$z > z_i, \frac{z}{z - z_i} > 1 \Rightarrow \lim_{z \rightarrow \infty} \frac{z}{z - z_i} = 1 \Rightarrow \lim_{z \rightarrow \infty} p_i^* = (x_i, y_i)$$

Für die duale Menge an Geraden $(x_i, y_i) \mapsto y = x_i x + y_i$ gilt ebenso

$$\lim_{z \rightarrow \infty} \left(\frac{z}{z - z_i} x_i x + \frac{z}{z - z_i} y_i \right) = x_i x + y_i$$

Alle Geraden gehen durch einen festen Punkt

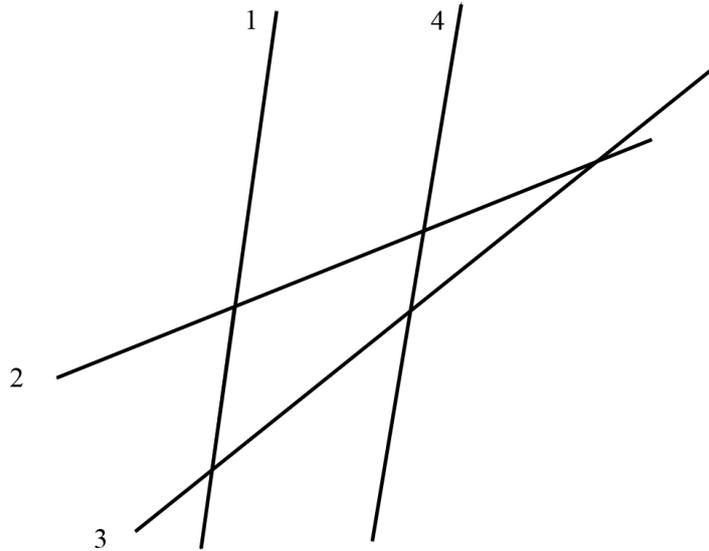
$$f_i = \left(-\frac{y_i}{x_i}, 0 \right)$$



Drei Punkte sind kollinear, wenn ihre dualen Geraden einen gemeinsamen Schnittpunkt haben. Das nächste verbotene Zentrum findet man, in dem man die Ebenen bildet die durch drei Punkte aller Dreieckszellen in $A(z_d)$ und diese auf Schnitt mit der z Achse überprüft. Der Schnittpunkt ist ein ungültiges Zentrum. Da es maximal $O(n^2)$ viele solcher Zellen gibt, kann dies in $O(n^2)$ getan werden. Nun wird noch das nächste zu $c(z_c)$ gelegene verbotene Zentrum $c(z_d)$ in $O(n)$ gefunden und wieder erhalten wir ein offenes Liniensegment an gültigen Zentren $\overline{c(z_c)c(z_d)}$. Insgesamt ergibt sich wegen der $O(n^2)$ vielen Dreieckszellen eine Laufzeit von $O(n^2)$. \square

Es stellt sich die Frage, warum es ausreicht alle Dreieckszellen zu betrachten. Aufgrund der Kontinuität ist klar, dass wenn man z stetig

wachsen lässt, nur zwei Knoten im dualen Arrangement zusammenfallen, die bei der Projektion von $c(z_c)$ auf einer Kante einer beliebig dimensionalen Zelle gelegen haben müssen. Fallen diese zwei Knoten zusammen, haben drei Geraden einen gemeinsamen Schnittpunkt und wir haben das nächste verbotene Zentrum gefunden. Um zu verstehen, warum es reicht die Dreieckszellen zu betrachten, betrachte man folgende Abbildung.

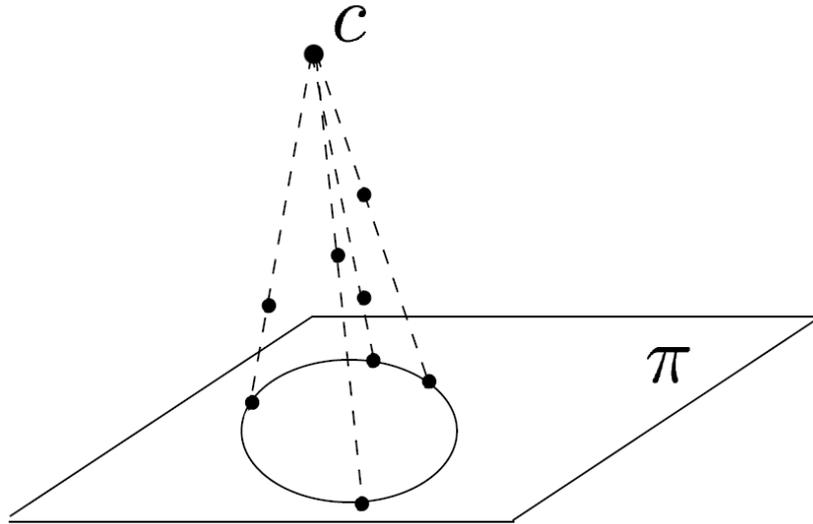


Angenommen es sollen die Schnittpunkte der drei Geraden r_1, r_2, r_3 durch eine stetige Bewegung zu einem Schnittpunkt zusammenfallen OBD. sei der Schnittpunkt (r_2, r_3) durch eine Gerade r_4 von den anderen Beiden abgeschnitten, sodaß (r_1, r_2) und (r_1, r_3) auf dem Rand eines Vierecks liegen. Jetzt gilt folgendes:

Da es sich um eine stetige Bewegung handelt, muss der Punkt (r_2, r_3) , wenn er in Richtung der anderen beiden Schnittpunkte wandert, auf jeden Fall die Gerade r_4 *passieren*. Damit ergibt sich zuerst eine andere kollineare Situation, und zwar in dem Dreieck (r_2, r_3, r_4) und daher reicht es aus sich auf alle Dreiecke zu beschränken.

3.4 Nicht kozirkulare Projektionen

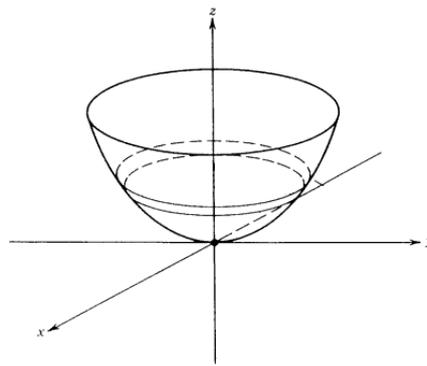
Eine perspektivische Projektion von einer Punktmenge im Raum heisst nicht-kozikular, wenn sie keine vier kozirkulare Punkte beinhaltet, wenn also keine vier Punkte auf einem gemeinsamen Kreis liegen.



Theorem 9 Gegeben eine perspektivische Projektion von n Punkten auf der xy Ebene. Zu entscheiden ob diese kozirkular ist benötigt $O(n^3)$ Zeit und Platz.

Beweis. Um herauszufinden, ob vier Punkte kozirkular sind, betrachten wir folgende duale Abbildung

$$(x, y) \mapsto z = x^2 + y^2$$



Damit werden alle Punkte auf einen Paraboloid angehoben und deren Tangentialebene betrachtet. Jetzt lassen sich folgende Eigenschaften ausnutzen.

1. Drei Ebenen schneiden sich in einem Punkt dann und genau dann wenn die drei korrespondierenden Punkte nicht kollinear sind.
2. Vier Ebenen schneiden sich in einem Punkt dann und genau dann wenn die vier korrespondierenden Punkte nicht kozirkular sind.

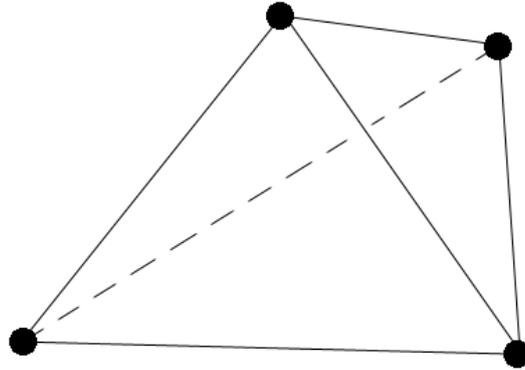
Ob es vier Tangentialebenen gibt die sich in einem Punkt schneiden, lässt sich bei n Ebenen in $O(n^3)$ Zeit und Platz ermitteln ([1]). \square

3.4.1 Existenz

Vier Punkte sind kozirkular, wenn sie auf dem Paraboloid koplanar sind. Vier Punkte sind koplanar, wenn das von ihnen geformte Tetrahedron ein Volumen von Null hat, wenn also

$$\begin{vmatrix} x_i & y_i & x_i^2 + y_i^2 & 1 \\ x_j & y_j & x_j^2 + y_j^2 & 1 \\ x_k & y_k & x_k^2 + y_k^2 & 1 \\ x_l & y_l & x_l^2 + y_l^2 & 1 \end{vmatrix} = 0$$

gilt.



Diese Gleichung gibt uns eine endliche Region von verbotenen Zentren. Daher lässt sich immer eine nicht kozirkulare Projektion finden.

3.4.2 Berechnung

Theorem 10 *Ein nicht kozirkulares Zentrum lässt sich in $O(n^3)$ Zeit und Platz berechnen.*

Beweis. Als erstes wird mit dem Algorithmus aus Abschnitt 3.3 ein Zentrum $c(z_c) = (0, 0, z_c)$ berechnet dass eine nicht kollineare Projektion liefert. Wenn $c(z_c)$ auch eine nicht kozirkulare Projektion liefert, haben wir ein gültiges Zentrum gefunden. Ansonsten erinnern wir uns, dass ein Punkt $p_i = (x_i, y_i, z_i)$ vom Zentrum $(0, 0, z)$, $z \geq z_c$ auf

$$p_i^*(z) = \left(\frac{z_c}{z_c - z_i} x_i, \frac{z_c}{z_c - z_i} y_i, 0 \right)$$

abgebildet wird. Lässt man z gegen unendlich gehen, konvergiert $p_i^*(z)$ gegen

$$(x_i, y_i, 0)$$

Betrachten wir die oben beschriebenen dualen Tangentialebenen π_i . Deren Normale ist $(2x_i, 2y_i, -1)$

$$\pi_i : z = 2 * \frac{z}{z - z_i} x_i x + 2 * \frac{z}{z - z_i} y_i y - 1 * \left(\left(2 * \frac{z}{z - z_i} x_i \right)^2 + \left(2 * \frac{z}{z - z_i} y_i \right)^2 \right)$$

Diese wird diese bei steigendem z kontinuierlich in folgende Ebene transformiert

$$\lim_{z \rightarrow \infty} \pi_i : z = 2 * x_i x + 2 * y_i y - 1 * (x_i^2 + y_i^2)$$

und gehen alle bei $z = 0$ durch die fixe Gerade

$$r_i : 2 * x_i x + 2 * y_i y - (x_i^2 + y_i^2) = 0$$

dessen Steigung negativ und gleich $-x_i/y_i$ ist.

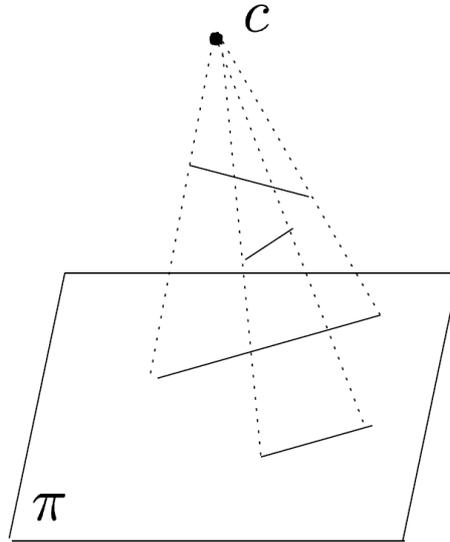
Da die Koeffizienten der Ebenen kontinuierlich von z abhängen, reicht es aus die $O(n^3)$ verbotenen Regionen zu betrachten, die eine tetraedische Zelle in $c(z_c)$ bilden. Diese müssen dann nur noch mit der z -Achse geschnitten werden und der nächste zu z_c gelegene Schnittpunkt z_d berechnet werden. Das offene Liniensegment $\overline{c(z_c)c(z_d)}$ ist dann die gesuchte Lösung. Die Berechnung einer nicht-kollinearen Projektion kann in $O(n^2)$ erfolgen. Zu prüfen, ob diese auch kozirkular ist, benötigt $O(n^3)$. Die Berechnung von $c(z_d)$ kann auch in kubischer Zeit erfolgen, woraus sich eine Gesamtlaufzeit von $O(n^3)$ ergibt.

□

4 Projektionen von Liniensegmenten

4.1 Nichtparallele Projektionen von Liniensegmenten

In vielen Algorithmen dürfen keine zwei Liniensegmente parallel zueinander sein. In diesem Abschnitt werden Projektionen beschrieben, die eine Menge S paarweiser disjunkter Liniensegmente so auf S^* projiziert, dass keine von diesen parallel zueinander sind. Sind in der Menge S Liniensegmente enthalten, die sowohl zueinander, also auch zur xy Ebene parallel sind, lässt sich keine geeignete Projektion finden. Als erstes befassen wir uns damit, wie man solche Fälle feststellen kann.



Lemma 11 Gegeben sei eine Menge von n paarweise disjunkten Liniensegmenten. Dann lässt sich in $O(n \log n)$ Zeit und $O(n)$ Platz feststellen, ob sich in dieser Menge zwei Segmente befinden die zueinander und zur xy Ebene parallel sind.

Beweis. $S = \{s_1, \dots, s_n\}$ sei eine Menge von Liniensegmenten. $a_i = (x_{a_i}, y_{a_i}, z_{a_i})$ und $b_i = (x_{b_i}, y_{b_i}, z_{b_i})$ bezeichne Start bzw. Endpunkt eines Liniensegments s_i . Betrachtet werden von nun an nur noch solche Geraden, die parallel zur xy Ebene sind. Dies können $O(n)$ viele sein. Um herauszufinden ob zwei Liniensegmente parallel sind, schaut man sich anschaulich die *positiven* Steigungen dieser an und vergleicht sie miteinander. Dazu verschiebt man die Segmente zunächst, so dass sich der Startpunkt im Ursprung des Koordinatensystems befindet. Als nächstes normiert man die Liniensegmente, und zwar so, daß diese einen Schnitt mit dem für x positiven Einheitskreis aufweisen. Dieser Schnittpunkt wird im folgenden als γ_i bezeichnet. Zwei Segmente s_i, s_j sind parallel, wenn gilt: $\gamma_i = \gamma_j$. Also müssen lediglich die n vielen γ Werte auf Gleichheit überprüft werden. Dafür benötigt man $O(n \log n)$ aufgrund der Sortierung. \square

Von nun an gehen wir davon aus dass sich keine zu sich und zur xy parallelen Liniensegmente in der Menge S befinden.

Theorem 12 Gegeben ein Zentrum $c \in \mathbb{R}^3$, dann lässt sich in $\Theta(n \log n)$ Zeit und $\Theta(n)$ Platz feststellen, ob die perspektivische Projektion eine Menge von nichtparallelen Liniensegmenten erzeugt.

Beweis. Dazu führe man die gleichen Schritte wie im Beweis zu Lemma 11 durch. Dies benötigt $O(n \log n)$ Zeit. Dass dies auch eine untere Grenze ist

lässt sich durch einen Reduktionsbeweis zeigen. Dazu reduziere das Element-Eindeutigkeitsproblem auf unser Entscheidungsproblem in linearer Zeit. Bilde die Menge $M = \{m_1, \dots, m_n\}$ folgendermaßen ab:

$$a_i = \left(i, \frac{i^2}{n}, \frac{n-i}{n}\right)$$

$$b_i = \left(i, \frac{i(i+m_i)}{n}, \frac{n+1-i}{n+1}\right)$$

Wendet man die perspektivische Projektion darauf an, ergibt sich :

$$a_i^* = (n, i)$$

$$b_i^* = (n+1, i+m_i)$$

Und damit ist die Steigung der Geraden:

$$\gamma_i = m_i$$

Daher sind zwei Elemente parallel, wenn zwei Elemente aus M identisch sind. □

4.1.1 Existenz

Sei $c = (x, y, z)$ Zentrum. Zwei Liniensegmente s_i, s_j mit $z_i = z_j$ sind genau dann parallel, wenn gilt:

$$\frac{\Delta y_i^*}{\Delta x_i^*} = \frac{\Delta y_j^*}{\Delta x_j^*}$$

$$\frac{\Delta y_i^*}{\Delta x_i^*} = \frac{y_{b_i}^* - y_{a_i}^*}{x_{b_i}^* - x_{a_i}^*} = \left(\frac{y_{b_i} - y}{z - z_{b_i}} + y\right) - \left(\frac{y_{a_i} - y}{z - z_{a_i}} + y\right) = \frac{y_{b_i} - y}{z - z_{b_i}} - \frac{y_{a_i} - y}{z - z_{a_i}}$$

$$\frac{\left(\frac{y_{b_i} - y}{z - z_{b_i}} - \frac{y_{a_i} - y}{z - z_{a_i}}\right)}{\left(\frac{x_{b_i} - x}{z - z_{b_i}} - \frac{x_{a_i} - x}{z - z_{a_i}}\right)} = \frac{\left(\frac{y_{b_j} - y}{z - z_{b_j}} - \frac{y_{a_j} - y}{z - z_{a_j}}\right)}{\left(\frac{x_{b_j} - x}{z - z_{b_j}} - \frac{x_{a_j} - x}{z - z_{a_j}}\right)}$$

Dies lässt sich auf in folgende quadratische Form bringen.

$$Az^2 + Bxz + Cyz + Dx + Ey + Fz + G = 0$$

Auflösen nach z ergibt in dieser Gleichung maximal zwei Lösungen. Da wir $O(n^2)$ viele solcher Gleichungen haben, kann es auf der z Achse nur $O(n^2)$ viele verbotene Punkte geben. Daher kann immer eine gültige Projektion gefunden werden.

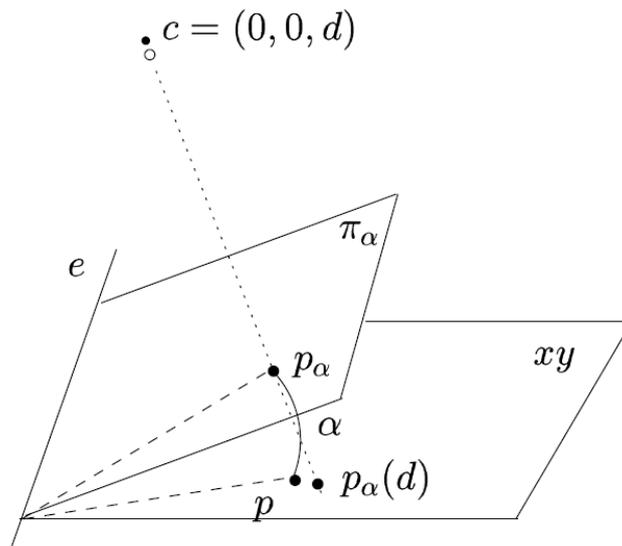
4.1.2 Berechnung

Theorem 13 *Gegeben seien n paarweise disjunkte Liniensegmente, dann lässt sich ein gültiges Zentrum einer nicht-parallelen Projektion in $O(n \log n)$ Zeit und $O(n)$ Platz berechnen.*

Beweis. Ein naiver Ansatz wäre, alle $O(n^2)$ Schnittpunkte der Quadriken mit der z -Achse zu berechnen und die restlichen Punkte als Lösung anzunehmen. Dies würde $O(n^2)$ Zeit in Anspruch nehmen. Stattdessen berechne die $\gamma_i(z)$ für ein erstes Zentrum $c(z) = (0, 0, z)$ mit $z > \max(z_i, i = 1, \dots, n)$. Falls alle $\gamma_i(z)$ paarweise verschieden sind, sind wir fertig. Ansonsten müssen wir ein neues gültiges Zentrum finden. Wir suchen das nächste zu z gelegene verbotene Zentrum $c(z_d)$ auf der z -Achse. Haben wir dies gefunden, lässt sich ein offenes Liniensegment von gültigen Zentren angeben. Aufgrund der Kontinuirlichkeit von γ_i reicht es die Quadriken der $O(n)$ vielen direkt benachbarten $\gamma_i(z)$ auf Schnitt mit der z -Achse zu prüfen. Da hierzu n viele Elemente sortiert werden müssen, ergibt sich eine Gesamtkomplexität von $O(n \log n)$. \square

5 Fazit

Die in dieser Arbeit vorgestellten Algorithmen lassen sich alle in polynomieller Zeit ausführen und sind ebenso anwendbar auf Punktmenge die sich in einer gemeinsamen Ebene befinden. Hierfür hebt man die Punkte durch eine Rotation um eine Achse innerhalb der Ebene in den drei-dimensionalen Raum und führt dann die oben beschriebenen Algorithmen durch.



Dabei ist darauf zu achten, dass man den Rotationswinkel möglichst klein hält, um die Punktmenge P^* nicht zu stark von P abweichen zu lassen. In [2] wird darauf noch näher eingegangen. Auch für höhere Dimensionen sollten die Algorithmen funktionieren, wobei dies noch in Zukunft genauer ausgearbeitet werden muss.

Literatur

- [1] H. Edelsbrunner, J. O'Rourke, and R. Seiel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15(2):341–363, 1986.
- [2] F. Gómez, F. Hurtado, J. Sellarès, and G. Toussaint. On degeneracies removable by perspective projection. *International Journal of Mathematical Algorithms.*, 2000.
- [3] F. Gomez, S. Ramaswami, and G. T. Toussaint. On removing non-degeneracy assumptions in computational geometry. In *CIAC*, pages 86–99, 1997.