Orthogonale Projektion

MAK

25.7.2014

1 Einleitung

Eine orthogonale Projektion ist eine mathematische Funktion, die einen Vektor $\vec{v} \in \mathbb{R}$ auf eine (Hyper-)Ebene X projiziert, also den Vektor aus dem Raum X abbildet, der die kürzeste Distanz zum abzubildenden Vektor hat.

Dieser Artikel beschäftigt sich im speziellen mit der dazugehörigen Projektionsmatrix (für die Eingliederung in CE3D).

2 Die orthogonale Projektion

Eine orthogonale Projektion hat folgende Eigenschaften:

- Der projizierte Vektor $p(\vec{x})$ ist eine Linearkombination aus den Basisvektoren $\vec{v}_1, \vec{v}_2, ..., \vec{v}_m$ der Projektionsebene X.
- Der projizierte Vektor abzüglich des ursprünglich zu projizierenden Vektors muss senkrecht sein zur Projektionsebene ($\langle p(\vec{x}) - \vec{x}, \vec{v} \rangle = 0$).

Es folgt eine Liste von Definitionen im Rahmen dieses Artikels:

- Der übergeordnete Vektorraum, in dem projiziert wird: V. Die Dimension des Raums ist definiert als $dim(V) = n$.
- Die Projektionsebene: $X = span(\vec{v}_1, \vec{v}_2, ..., \vec{v}_m)$. Damit gilt für die Dimension der Ebene: $dim(X) = m$.
- Die Projektionsfunktion: $p(\vec{x})$. $p = \begin{cases} \mathbb{R}^n \to \mathbb{R}^n \\ \vec{x} \mapsto p(\vec{x}) \end{cases}$

3 Die Projektionsmatrix

Um die Projektionsmatrix herzuleiten, ziehen wir die obigen Bedingungen heran und geben ihnen eine verallgemeinerte mathematische Form:

$$\begin{array}{ll} \text{I} & p(\vec{x}) = \sum_{i=1}^{m} \mu_i \vec{v}_i, \quad \mu_i \in \mathbb{R} \\ \text{II} & \langle p(\vec{x}) - \vec{x}, \vec{v}_i \rangle = 0, \quad \forall i \in \{w \in \mathbb{N} | 1 \leq w \leq m\} \end{array} \tag{1}$$

Die erste Bedingung lässt sich in die zweite einsetzen und in ein Gleichungssystem umschreiben:

$$\left\langle \sum_{j=1}^{m} \mu_j \vec{v_j} - \vec{x}, \vec{v_i} \right\rangle = 0, \quad \forall i \in \{w \in \mathbb{N} | 1 \leq w \leq m\} \tag{2}$$

$$\iff \begin{array}{rcl} \langle \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m - \vec{x}, \vec{v}_1 \rangle & = & 0 \\ \langle \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m - \vec{x}, \vec{v}_2 \rangle & = & 0 \\ \vdots & = & \vdots \\ \langle \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m - \vec{x}, \vec{v}_m \rangle & = & 0 \end{array} \tag{3}$$

Alle konstanten Faktoren auf die rechte Seite bringen:

$$\iff \begin{array}{rcl} \langle \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m, \vec{v}_1 \rangle & = & \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m, \vec{v}_2 \rangle & = & \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots & = & \vdots \\ \langle \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m, \vec{v}_m \rangle & = & \langle \vec{x}, \vec{v}_m \rangle \end{array} \tag{4}$$

Ausnutzen der Distributivität und Homogenität ergibt:

$$\iff \begin{array}{rcl} \mu_1 \langle \vec{v}_1, \vec{v}_1 \rangle + \mu_2 \langle \vec{v}_2, \vec{v}_1 \rangle + \cdots + \mu_3 \langle \vec{v_3}, \vec{v}_1 \rangle & = & \langle \vec{x}, \vec{v}_1 \rangle \\ \mu_1 \langle \vec{v}_1, \vec{v}_1 \rangle + \mu_2 \langle \vec{v}_2, \vec{v}_1 \rangle + \cdots + \mu_3 \langle \vec{v_3}, \vec{v}_1 \rangle & = & \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots & = & \vdots \\ \mu_1 \langle \vec{v}_1, \vec{v}_1 \rangle + \mu_2 \langle \vec{v}_2, \vec{v}_1 \rangle + \cdots + \mu_3 \langle \vec{v_3}, \vec{v}_1 \rangle & = & \langle \vec{x}, \vec{v}_m \rangle \end{array} \tag{5}$$

Dieses Gleichungssystem kann durch eine Matrix wiedergegeben werden:

$$\iff \begin{pmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_2, \vec{v}_1 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_1 \rangle \\ \langle \vec{v}_1, \vec{v}_2 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_2 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{v}_1, \vec{v}_m \rangle & \langle \vec{v}_2, \vec{v}_m \rangle & \cdots & \langle \vec{v}_m, \vec{v}_m \rangle \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} = \begin{pmatrix} \langle \vec{v}_1, \vec{x} \rangle \\ \langle \vec{v}_2, \vec{x} \rangle \\ \vdots \\ \langle \vec{v}_m, \vec{x} \rangle \end{pmatrix} \tag{6}$$

Wir erhalten also die Gram'sche Matrix. Nun müssen wir diese Gleichung so umformen, sodass wir eine Abbildung der Form

$$P\vec{x} = p(\vec{x}) \tag{7}$$

bekommen mit P als zugehörige Projektionsmatrix. Desweiteren definieren wir zur Vereinfachung die Gramsche Matrix und den Koeffizientenvektor:

$$G := \begin{pmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_2, \vec{v}_1 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_1 \rangle \\ \langle \vec{v}_1, \vec{v}_2 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_2 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{v}_1, \vec{v}_m \rangle & \langle \vec{v}_2, \vec{v}_m \rangle & \cdots & \langle \vec{v}_m, \vec{v}_m \rangle \end{pmatrix} \tag{8}$$

$$\vec{\mu} := \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} \tag{9}$$

Unser Gleichungssystem wird nach $\vec{\mu}$ umgeformt und in unsere erste Bedingung für die orthogonale Projektion eingesetzt (jeder projizierte Vektor muss eine Linearkombination der Basis der Projektionsebene X sein):

$$G\vec{\mu} = \begin{pmatrix} \langle \vec{v}_1, \vec{x} \rangle \\ \langle \vec{v}_2, \vec{x} \rangle \\ \vdots \\ \langle \vec{v}_m, \vec{x} \rangle \end{pmatrix} \iff G^{-1} \begin{pmatrix} \langle \vec{v}_1, \vec{x} \rangle \\ \langle \vec{v}_2, \vec{x} \rangle \\ \vdots \\ \langle \vec{v}_m, \vec{x} \rangle \end{pmatrix} = \vec{\mu} \tag{10}$$

Bevor wir einsetzen wird die erste Bedingung auch in eine Matrixform überführt:

$$\sum_{i=1}^{m} \mu_i \vec{v}_i = p(\vec{x}) = \left\langle \vec{\mu}, \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_m \end{pmatrix} \right\rangle = \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m \end{pmatrix} \vec{\mu} \tag{11}$$

Erneut definieren wir zur Vereinfachung:

$$A := \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m \end{pmatrix} \tag{12}$$

A ist also die Matrix, die die Basis unserer Projektionsebene X enthält.

Jetzt muss nur noch mathematisch umgeformt werden um die Projektionsmatrix P zu erhalten:

$$p(\vec{x}) = A\vec{\mu} = AG^{-1} \begin{pmatrix} \langle \vec{v}_1, \vec{x} \rangle \\ \langle \vec{v}_2, \vec{x} \rangle \\ \vdots \\ \langle \vec{v}_m, \vec{x} \rangle \end{pmatrix} = AG^{-1} \begin{pmatrix} \vec{v}_1^T \vec{x} \\ \vec{v}_2^T \vec{x} \\ \vdots \\ \vec{v}_m^T \vec{x} \end{pmatrix} = AG^{-1} A^T x \tag{13}$$

Die Gram'sche Matrix lässt sich darstellen durch

$$G = A^T A \tag{14}$$

.

Damit haben wir unsere endgültige Projektionsmatrix zusammen:

$$p(\vec{x}) = A(A^T A)^{-1} A^T \vec{x} \tag{15}$$

$$P = A(A^T A)^{-1} A^T \tag{16}$$

mit unserer Basis-Matrix

$$A = \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m \end{pmatrix} \tag{17}$$

.

4 Projektion auf die Koeffizienten $\vec{\mu}$ im Projektionsraum

Für 3D-Anwendungen ist die Projektion auf die Koeffizienten mindestens genauso interessant wie wichtig, z.B. um die Funktion einer Kamera zu implementieren, die einem die x- und y-Koordinaten der Bildobjekte auf dem Sensor bzw. auf dem Display liefert.

Die Projektionsmatrix für die Koeffizienten lässt sich schnell herleiten aus der ersten Projektionsbedingung (nach der Übersetzung in die Matrixform):

$$p(\vec{x}) = A\vec{\mu} = A(A^T A)^{-1} A^T \vec{x} \tag{18}$$

$$\iff \vec{\mu} = (A^T A)^{-1} A^T \vec{x} \tag{19}$$

Also ist die Projektionsmatrix für die Koeffizienten:

$$P_\mu = (A^T A)^{-1} A^T \tag{20}$$

5 Abstand von $p(\vec{x})$ zu $\vec{x}$

Der Abstand des projizierten Vektors zum ursprünglichen Vektor ist von zentraler Bedeutung in der Computergrafik. Um verschiedene dreidimensionale Objekte auf dem Bildschirm richtig arrangieren zu können, müssen deren Abstände zur Projektionsebene bekannt sein. Ansonsten wäre bei Überschneidungen von Objekten die 3D-Szene nicht richtig wiedergegeben. Deshalb fügt man in die Matrix eine zusätzliche Zeile ein, die die s.g. Tiefe berechnet. Allerdings gibt es dabei ein Problem: Der euklidische Abstand der Vektoren ist eine nicht-lineare Funktion! Damit ist sie auch nicht in einer Matrix darstellbar.

$$||\vec{x} - p(\vec{x})||_2 = \sqrt{\sum_{i=1}^{n} (x_i - p(\vec{x})_i)^2}, \quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{21}$$

Wir benötigen eine Funktion, die den Abstand wiedergibt und eine lineare Funktion ist. Deswegen verwenden wir die 1-Norm (und ignorieren den komponentenweisen Betrag):

$$||\vec{x} - p(\vec{x})||_1 = \sum_{i=1}^{n} x_i - p(\vec{x})_i = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} (\vec{x} - p(\vec{x})) \tag{22}$$

Es existiert eine Matrixform für diese Abbildung, damit ist sie für unsere Zwecke geeignet. Anmerkung: Die 1-Norm gibt nicht die exakte Distanz wieder, ist aber ein guter Ersatz, denn es werden insbesondere Reihenfolgen der Vektoren (also welcher Vektor ist weiter weg als ein anderer) richtig wiedergegeben.

Diese Idee lässt sich allerdings erheblich verbessern, indem wir einen Basiswechsel vollziehen in unseren Projektionsraum plus den restlichen dazugehörigen orthonormalen Vektoren. Diese restlichen orthonormalen Vektoren bilden die Freiheitsgrade unserer Tiefenfunktion, also projizieren wir unser $\vec{x}$ im Endeffekt auf diese und erhalten damit automatisch die 1-Norm in unserer neuen Basis. Die Projektionen müssen dann nur noch aufsummiert werden.

Unsere Tiefenfunktion $d_1(\cdot,\cdot)$ lautet also:

$$d_1(\vec{x}, p(\vec{x})) = \sum_{k=1}^{j} \langle \vec{n}_k, \vec{x} - p(\vec{x}) \rangle, \quad j = n - m \tag{23}$$

mit $\vec{n}$ als restliche zu unserer Projektionsebene X orthonormale Basisvektoren.

Das formen wir erneut um zu einer Matrix:

$$\iff d_1(\vec{x}, p(\vec{x})) = \left\langle \vec{x} - p(\vec{x}), \sum_{k=1}^{j} \vec{n}_k \right\rangle \tag{24}$$

$$\iff d_1(\vec{x}, p(\vec{x})) = (\sum_{k=1}^{j} \vec{n}_k)^T (\vec{x} - p(\vec{x})) \tag{25}$$

Für die Projektionsfunktion setzen wir die Matrixabbildung ein (mit E als Einheitsmatrix):

$$\iff d_1(\vec{x}, p(\vec{x})) = (\sum_{k=1}^{j} \vec{n}_k)^T (\vec{x} - P\vec{x}) \tag{26}$$

$$\iff d_1(\vec{x}, p(\vec{x})) = (\sum_{k=1}^{j} \vec{n}_k)^T (E - P)\vec{x} \tag{27}$$

Und wir erhalten unsere Tiefenmatrix T, die wir als letzte Zeile an unsere Projektionsmatrix P anhängen können:

$$T = (\sum_{k=1}^{j} \vec{n}_k)^T (E - P), \quad j = n - m \tag{28}$$

Also müssen alle orthonormalen Restvektoren unseres Projektionsraums X nur aufsummiert und anschließend transponiert werden.

Diese Tiefenfunktion hat zusätzlich einen sehr praktischen Vorteil: Ist $m = n - 1$, also $j = 1$, so fällt die Tiefenfunktion mit der Hesse'schen Normalform zusammen und gibt dann sogar den euklidischen Abstand wieder! Da nur noch ein orthonormaler Vektor vorhanden ist, fällt die Summe weg:

$$d_1(\vec{x}, p(\vec{x})) = \vec{n}^T \vec{x} = \langle \vec{n}, \vec{x} - P\vec{x} \rangle = d_2(\vec{x}, p(\vec{x})) \tag{29}$$