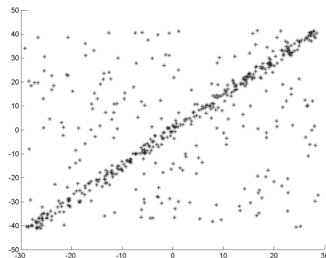


## Bildverarbeitung: RANSAC



Man suche eine Gerade  $ax + by = 1$   
d.h. die unbekannten Parameter  $a$  und  $b$   
anhand einer Lernstichprobe der Punkte  
 $((x^1, y^1), (x^2, y^2) \dots (x^i, y^i))$

---

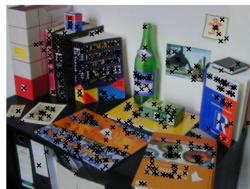
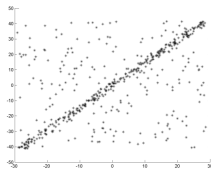
Man schätze die Fundamental Matrix  $F$ , d.h.

$$[x_{l1}, x_{l2}, 1] \cdot \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \cdot \begin{bmatrix} x_{r1} \\ x_{r2} \\ 1 \end{bmatrix} = 0$$

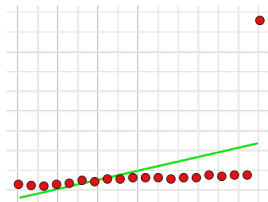
anhand einer Lernstichprobe der Korrespondenzpaare  $((x_l^1, x_r^1), (x_l^2, x_r^2) \dots (x_l^i, x_r^i))$ .

# Outliers (Ausreißer)

Die Daten sind fehlerhaft:



- 1) Rauschen: die Koordinaten weichen von den wahren ab  
(je weiter entfernt, desto weniger wahrscheinlich)
- 2) Outliers: die Daten gehören gar nicht zum betrachteten Modell  
⇒ dürfen **keinen** Einfluss auf das Ergebnis haben



Werden die Outliers nicht explizit berücksichtigt,  
so wird die Lösung verfälscht  
← (siehe das)

Ausweg:

finde die Outliers explizit (die Daten werden klassifiziert),  
schätze das Modell anhand nur “richtiger” Datenpunkte.

Sei  $x \in \mathcal{X}$  der Input-Raum und  $y \in \mathcal{Y}$  der Parameter-Raum.

Die Lernstichprobe besteht aus Datenpunkten:  $L = (x^1, x^2 \dots x^i), x^i \in \mathcal{X}$

Gegeben sei eine Funktion  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ ,

die für jedes Paar  $x$  (Datenpunkt) und  $y$  (Parametersatz, Modellinstanz) angibt, ob der Punkt mit dem Modell konsistent ist.

Beispiele:

- Gerade

$$f(x, y, a, b) = \begin{cases} 1 & \text{wenn } ax + by = 1, \\ 0 & \text{sonst.} \end{cases}$$

- Fundamentalmatrix

$$f(x_l, x_r, F) = \begin{cases} 1 & \text{wenn } x_l F x_r = 0, \\ 0 & \text{sonst.} \end{cases}$$

Gesucht wird das Modell,

das mit der Mehrheit der Punkte in der Lernstichprobe übereinstimmt:

$$y^* = \arg \max_{y \in \mathcal{Y}} \sum_i f(x^i, y)$$

Naiver Ansatz: probiere alle  $y$  durch  $\Rightarrow$  Hough-Transformation.

Problem: Aufwendig, manchmal gar nicht möglich  
(z.B. wenn die Dimension des Parameterraumes groß ist).

---

**Beobachtung:** der Parameterraum ist „dünn besetzt“

– nur sehr wenige Parametersätze stimmen mit der Lernstichprobe halbwegs überein.

**Die Idee:** probiere nicht alles durch, sondern nur das, was „überhaupt Chancen hat“.

Gegeben sei ein **Orakel** – eine Funktion  $g : \mathcal{X}^d \rightarrow \mathcal{Y}$ , die für jedes  $d$ -Tupel von Punkten das Modell schätzt, die mit allen  $d$  Punkten übereinstimmt.

Beispiele: eine Gerade aus 2 Punkten, eine Fundamentalmatrix aus 8 Punkte.

Probiere nicht alle  $y \in \mathcal{Y}$  sondern alle  $d$ -Tupel der Datenpunkte (d.h.  $L' \subset L, |L'| = d$ ):

$$L'^* = \arg \max_{L' \subset L, |L'|=d} \sum_i f(x^i, g(L')), \quad y^* = g(L'^*)$$

Beispiel: eine Gerade zu finden –  $|L|^2$  Versuche, Fundamentalmatrix –  $|L|^8$

Die Maximierung geschieht über eine diskrete Menge!!!

RANSAC (Random Sample Consensus, Fischler und Bolles 1981):

probiere nicht mal alle Teilmengen, sondern würfele manche davon.

**Wiederhole oft**

**Würfele**  $L' \subset L, |L'| = d$

**Schätze**  $y = g(L')$

**Bewerte**  $f(y) = \sum_i f(x^i, y)$

**wenn**  $f(y) > f(y^*)$

setze  $y^* = y$  und merke  $f(y^*)$

Wie oft muss gewürfelt werden?

**Annahme** (trifft nicht immer zu): Sollte man einen  $d$ -Tupel würfeln, der nur aus Inliers besteht, so ist das richtige Modell erwischt.

Ausschlaggebend ist die **Wahrscheinlichkeit**, dass während des Algorithmus das richtige Modell mindestens einmal erwischt wird.

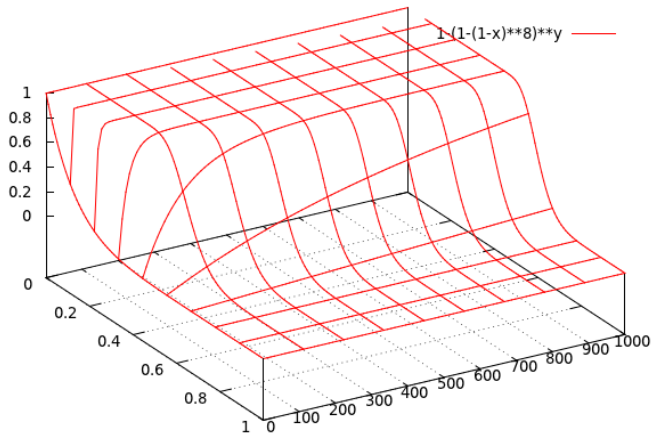
Sei  $\varepsilon$  die Wahrscheinlichkeit der Outliers.

Wahrscheinlichkeit, dass ein zufällig gewürfeltes  $d$ -Tupel nur aus Inliers besteht:  $(1 - \varepsilon)^d$ .

Wahrscheinlichkeit eines falschen  $d$ -Tupels:  $1 - (1 - \varepsilon)^d$ .

Wahrscheinlichkeit, dass  $n$  Würfel alle falsch sind:  $(1 - (1 - \varepsilon)^d)^n$ .

Wahrscheinlichkeit, dass sich darunter mindestens ein richtiges befindet:  $1 - (1 - (1 - \varepsilon)^d)^n$ .



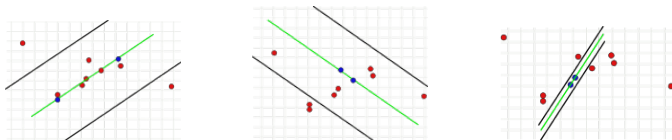
$$1 - (1 - (1 - \varepsilon)^d)^n, \quad d = 8, \varepsilon \in [0, 1], n = 1 \dots 1000$$

Fehler des ersten Typs: Rauschen

**Bewertungsfunktion**  $f(x, y)$ . Wann ist eigentlich ein Datenpunkt „gut“?

Zum Beispiel  $x_l F x_r = 0$  ist so gut wie nie exakt erfüllt.

$\Rightarrow$  Konfidenzintervalle, z.B.  $|x_l F x_r| < \nu$ ,  $|ax + by - 1| < \nu$



Die Wahl des richtigen Konfidenzintervalls ist anwendungsabhängig und oft schwierig

**MSAC**: die Inliers werden auch bewertet,

die **Indikatorfunktion**  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$  wird zur **Straffunktion**  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ , gesucht wird das Modell **minimaler** summarischer Strafe, d.h.  $y^* = \arg \min_y \sum_i f(x^i, y)$

Beispiel für die Fundamentalmatrix:

$$f(x_l, x_r, F) = \begin{cases} 1 & \text{wenn } |x_l F x_r| < \nu, \\ 0 & \text{sonst.} \end{cases}$$

wird zu

$$f(x_l, x_r, F) = \min(|x_l F x_r|, \nu)$$

Bewertung einer Hypothese  $y$ , d.h.  $\sum_i f(x^i, y)$  – oft zu Aufwendig.

$T_{d,d}$ -Test (**Randomized** RANSAC): statt alle  $x^i \in L$  zu prüfen:

- Würfele  $m$  Datenpunkte aus  $L$
- Sind alle gut, teste alle anderen wie früher
- Ist mindestens ein gewürfelten Punkt schlecht, fällt die Hypothese komplett durch

Man darf eine gute Hypothese verpassen, dafür spart man Zeit  
(die schlechten Hypothesen werden schnell erkannt) und kann öfter würfeln.

⇒ Die richtige Hypothese wird irgendwann doch erwischt.

Insgesamt oft (je nach Anwendung) schneller.

---

**Orakel**  $g(L')$ .

Beispiele für Fundamentalmatrix:

8-Punkte Algorithmus: ungenau, schnell, schwieriger zu erwischen ( $d = 8$ ).

7-Punkte Algorithmus: genauer, komplizierter, leichter zu erwischen ( $d = 7$ ).

⇒ Kompromisse, Würfeln+Optimieren, usw.

---

Gas ganze ist gut parallelisierbar.

# Andere Anwendungen

Panoramas – finde die dafür benötigten Homographien



Puzzle zusammensetzen

(S. Winkelbach, M. Rilk, C. Schönfelder, F. Wahl:  
„Fast Random Sample Matching of 3d Fragments“)

