Extraction de Structure

Bruno Vallet
bruno.vallet@ign.fr
LASTIG — ENSG - IGN

Master PPMD

Introduction

Introduction

Le problème de l'extraction de structure:

Ensemble de points $\xrightarrow{\text{Extraction de structure}}$ Primitives géométriques

Applications:

- Détection d'objets
- Reconstruction d'objets structurés
- Souvent étape intermédiaire dans une chaine de traitements de données

Introduction

Ensemble de points:

Considérés comme des mesures, des évidences des objets cherchés.

Exemples:

- Points de contours image
  - évidence d'un bord visible
- Points d'un scan laser
  - évidence d'une interface vide/plein)
- Points résultant d'une corrélation
  - évidence d'une surface visible

Introduction

Primitives:

Données continues décrites par

une **équation mathématique:**

En 2D:

- Droite
- Segment
- Cercle
- Ellipse
- Triangle
- Rectangle
- Carré
- Spline

En 3D:

- Plan
- Triangle
- Sphère
- Droite
- Spline
- Quadrique

Elles dépendent d'un nombre limité de **paramètres.**

Introduction

Problème:

Retrouver des primitives dans les points de données.

Compromis:

- Le moins de primitives possible
  - Mesuré par le nombre de primitives

- Expliquant un maximum de points
  - Expliquer= être suffisament proche de.
  - Sous entend une association point/primitive.
  - Mesuré par le nombre de points expliqués et leur distance aux primitives

Plan du cours:

- **Introduction**
- I Moindres carrés
- II Hough
- III RANSAC
- IV Comparaisons
- Conclusion

I – Estimation aux moindres carrés

Moindre carrés

Problème de l'estimation aux moindres carrés:

Nuage de points	→	Primitive
$\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^m\}_{i=1..n}$		$\mathcal{P}(a, b, c, ...)$

- Trouver une primitive dans un nuage de points
- On suppose que tous les points appartiennent à la primitive
- On cherche donc uniquement à combiner les mesures pour réduire le bruit/erreurs de mesures

Moindre carrés

Problème de l'estimation aux moindres carrés:

Nuage de points

$$\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^m\}_{i=1..n}$$

→

Primitive

$$\mathcal{P}(a, b, c, ...)$$

- La primitive dépends de plusieurs paramètre
- On minimise l'erreur:

Energie L_2

$$\mathcal{E}_{L_2}(a, b, c, ...) = \sum_{i=1..n} dist(\mathbf{x}_i, \mathcal{P}(a, b, c, ...))^2$$

Moindre carrés

Problème de la régression linéaire:

Mesures

$$\mathcal{X} = \{(x_i, y_i)^t \in \mathbb{R}^2\}_{i=1..n}$$

Fonction affine

$$y = ax + b$$

- On cherche une relation linéaire entre des mesures
- Les x sont supposés parfaits, les y imparfaits
- On minimise

Energie L_2

$$\mathcal{E}_{L_2}(a,b) = \sum_{i=1..n} (ax_i + b - y_i)^2$$

Moindre carrés

Résolution:

- $\mathcal{E}_{L_2}(a,b)$ est convexe, donc elle a un unique minimum
- Ce minimum vérifie $\nabla\mathcal{E}_{L_2}(a,b) = 0$

$$\frac{\partial\mathcal{E}_{L_2}(a,b)}{2\partial a} = \sum_{i=1..n} x_i(ax_i + b - y_i) = a\sum_{i=1..n} x_i^2 + b\sum_{i=1..n} x_i - \sum_{i=1..n} x_i y_i = 0$$

$$\frac{\partial\mathcal{E}_{L_2}(a,b)}{2\partial b} = \sum_{i=1..n} ax_i + b - y_i = a\sum_{i=1..n} x_i + nb - \sum_{i=1..n} y_i = 0$$

- On résout aisément cette paire d'équations linéaires

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum x_i y_i \\ \sum y_i \end{pmatrix}$$

Moindre carrés

Exemple du plan: $\mathcal{P}(\vec{n}, r) = \{\mathbf{x} | \mathbf{n}.\mathbf{x} = r\} \quad ||\mathbf{n}|| = 1$

$$dist(\mathbf{x}_i, \mathcal{P}(\mathbf{n}, r)) = |\mathbf{n}.\mathbf{x}_i - r|$$

- **Energie L_2 pour un plan**

$$\mathcal{E}_{L_2}(\mathbf{n}, r) = \sum_{i=1}^{n} (\mathbf{n}.\mathbf{x}_i - r)^2$$

- Problème: la normale au plan être normalisée:

$$\mathbf{n}^t \mathbf{n} = 1$$

- Solution: Lagrangien

$$\mathcal{L}(\mathbf{n}, r, \lambda) = \sum_{i=1}^{n} (\mathbf{n}.\mathbf{x}_i - r)^2 + \lambda(\mathbf{n}^t \mathbf{n} - 1)$$

Moindre carrés

Exemple du plan: Résolution

- Résolution:

$$\frac{\partial \mathcal{L}(\mathbf{n},r,\lambda)}{\partial \lambda} = \mathbf{n}^t\mathbf{n} - 1 = 0$$

$$\frac{\partial \mathcal{L}(\mathbf{n},r,\lambda)}{\partial r} = -2\mathbf{n}.\sum_{i=1}^{n} \mathbf{x}_i + 2nr = 0 \quad r = \mathbf{n}.\mathbf{g} \quad \mathbf{g} = \sum_{i=1}^{n} \mathbf{x}_i/n$$

$$\mathcal{L}(\mathbf{n}, r, \lambda) = \sum_{i=1}^{n}(\mathbf{n}.\mathbf{x}_i - \mathbf{n}.\mathbf{g})^2 + \lambda(\mathbf{n}^t\mathbf{n} - 1)$$

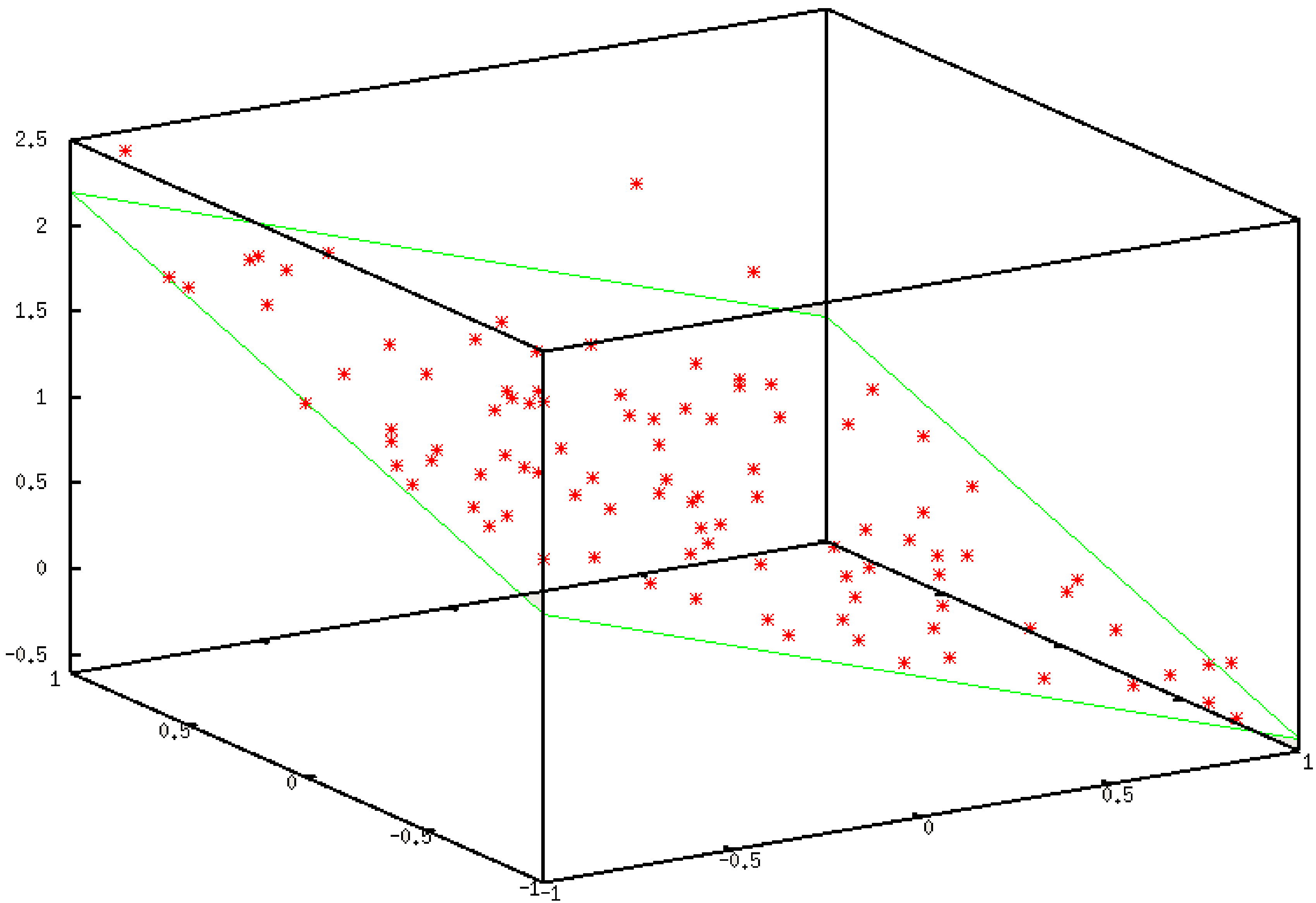
$$\mathcal{L}(\mathbf{n}, r, \lambda) = \mathbf{n}^t E_{L_2}\mathbf{n} + \lambda(\mathbf{n}^t\mathbf{n} - 1) \quad E_{L_2} = \sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{g})^t(\mathbf{x}_i - \mathbf{g})$$

$$\nabla_{\mathbf{n}}\mathcal{L}(\mathbf{n}, \lambda) = (E_{L_2} + \lambda I)\mathbf{n} = 0$$

- C'est un problème de valeurs propres: le minimum est le vecteur propre associé à la plus petite valeur propre de E_{L_2}

Moindre carrés

Exemple du plan: Résultat



Moindre carrés

Problème: détection de primitives multiples:

Nuage de points

$$\mathcal{X} = \{\mathbf{x}_i = (x_i, y_i, z_i)^t \in \mathbb{R}^3\}_{i=1..n}$$

→

Primitives

$$\{\mathcal{P}(a_i, b_i, c_i, ...)\}_{i=1..n}$$

- Comment trouver plusieurs primitives ?
- Comment gérer les points hors primitive (outliers) ?
- Comment définir l'objectif en termes d'optimisation ?

Moindre carrés

Problème: détection de primitives multiples:

Nuage de points		Primitives
$\mathcal{X} = \{\mathbf{x}_i = (x_i, y_i, z_i)^t \in \mathbb{R}^3\}_{i=1..n}$	→	$\{\mathcal{P}(a_i, b_i, c_i, ...)\}_{i=1..n}$

- On cherche:
  - N primitives
  - Une association des points aux primitives (ou non)

$$\mathcal{E} = \sum_{j=1..N} \sum_{i \in \mathcal{P}_j} dist(\mathbf{x}_i, \mathcal{P}_j(a, b, c, ...))^2 + \lambda N + \mu^2 N_{outlier}$$

- Induit un choix d'association trivial:
  - A la primitive la plus proche si $dist^2 < \mu^2$
  - Outlier sinon

III – Transformée de Hough

Transformée de Hough

Problème: détection de primitives multiples:

Nuage de points

$$\mathcal{X} = \{\mathbf{x}_i = (x_i, y_i, z_i)^t \in \mathbb{R}^3\}_{i=1..n}$$

→

Primitives

$$\{\mathcal{P}(a_i, b_i, c_i, ...)\}_{i=1..n}$$

Principe de la transformée de Hough:

- Extraire des objets paramétriques simples en se projetant dans l’espace des paramètres de l’objet
- Accumuler dans l'espace des paramètres
- Les primitives cherchées correspondent aux maxima

Transformée de Hough

Problème: détection de primitives multiples:

Nuage de points

$$\mathcal{X} = \{\mathbf{x}_i = (x_i, y_i, z_i)^t \in \mathbb{R}^3\}_{i=1..n}$$

→

Primitives

$$\{\mathcal{P}(a_i, b_i, c_i, ...)\}_{i=1..n}$$

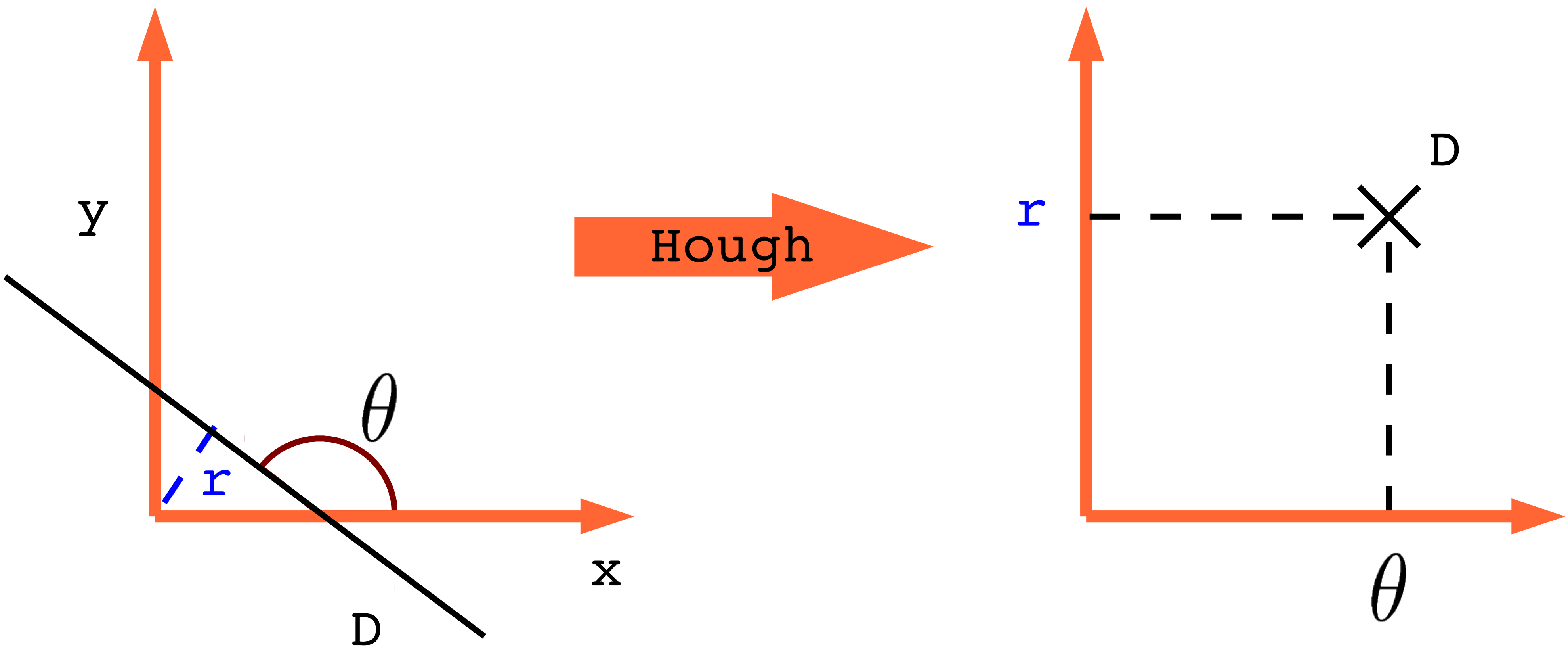
Principe de la transformée de Hough:

- Extraire des objets paramétriques simples en se projetant dans l'espace des paramètres de l'objet
- La primitive dépend de plusieurs paramètre :
  - Droite : 2 $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$
  - Cercle : 3 $(x_0, y_0, r) : (x - x_0)^2 + (y - y_0)^2 - r^2 = 0$
  - Plan : 4 $(r, \theta, \phi) : x\sin(\theta)\cos(\phi) + y\sin(\theta)\sin(\phi) + z\cos(\phi) - r = 0$
  - Sphère : 4 $(x_0, y_0, z_0, r) : (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0$
  - Droite 3D: ???

Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

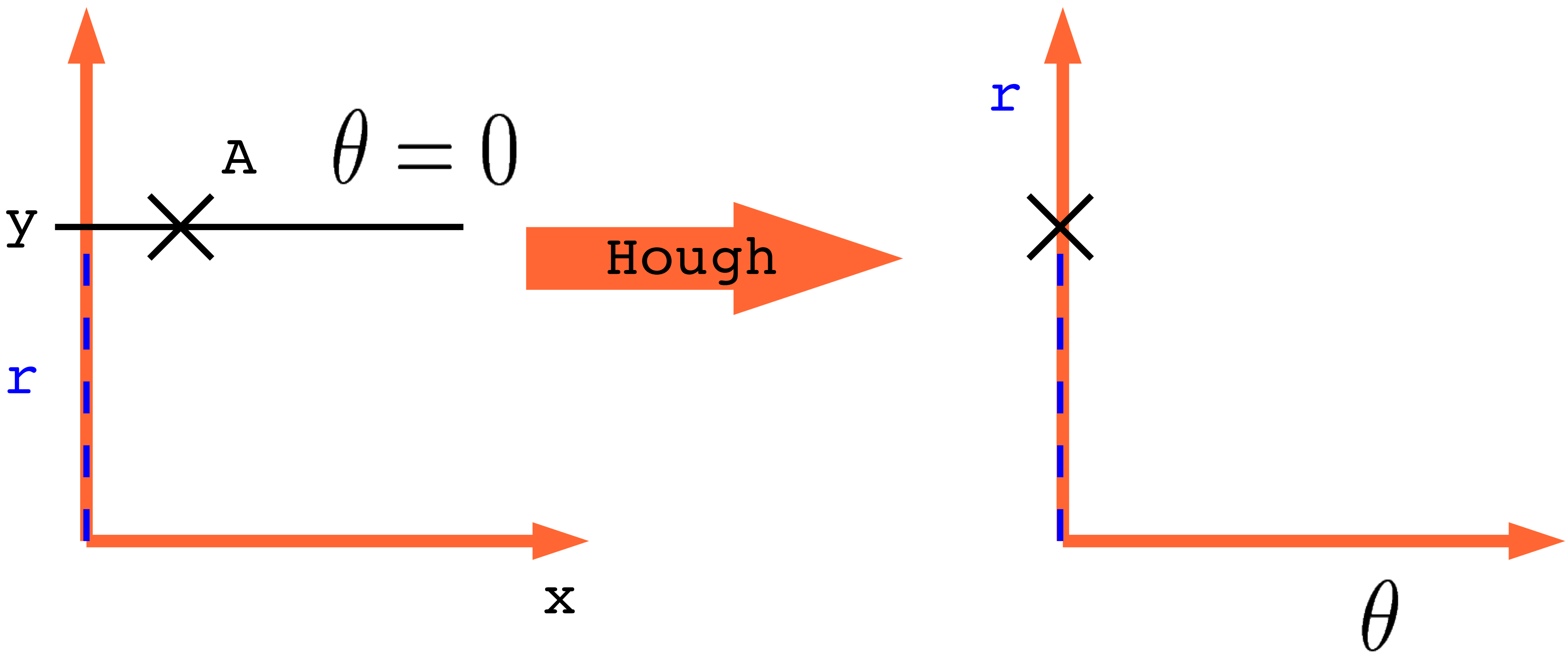
- Transformation d'**une droite** dans l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r,\theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

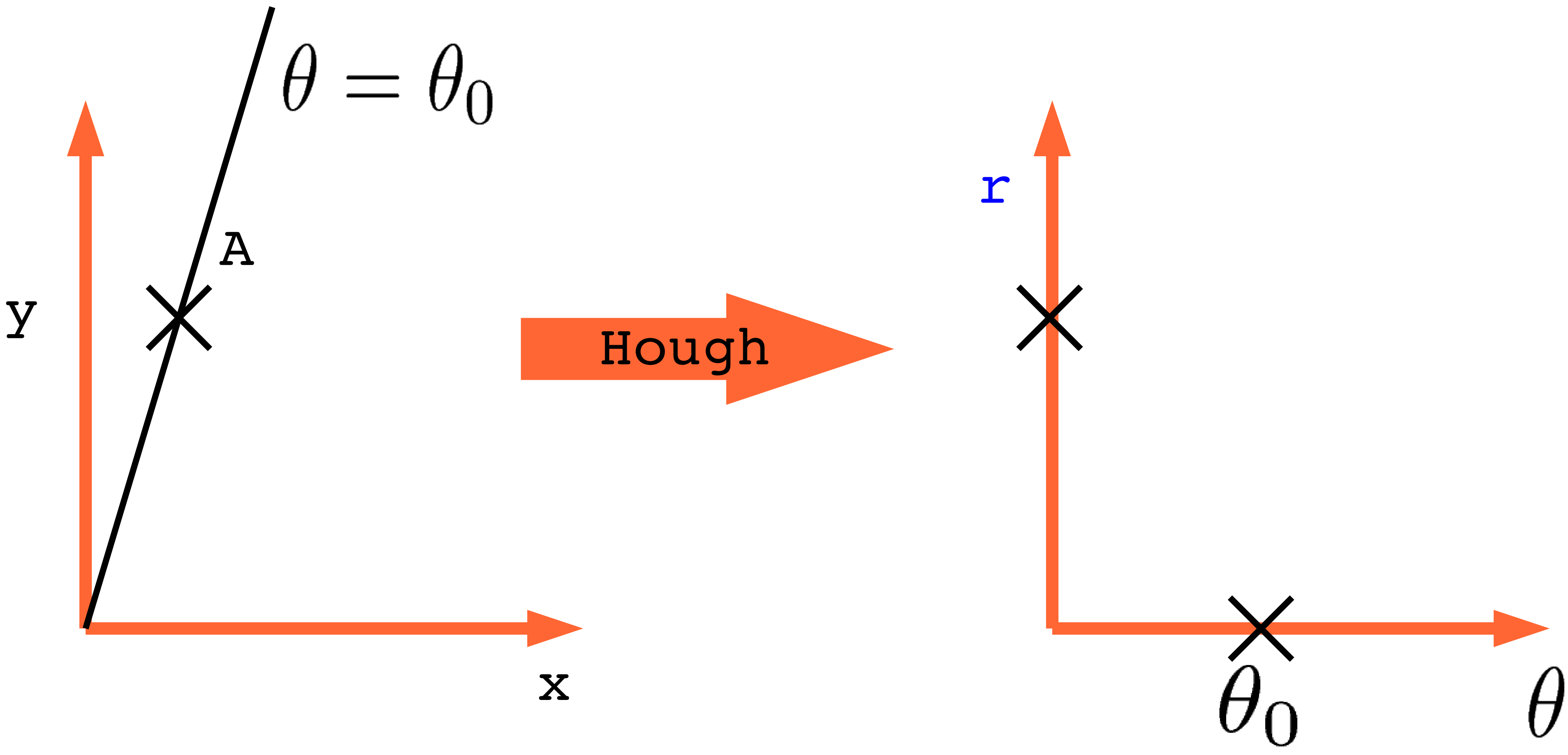
- Transformation d'**un point** dans l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

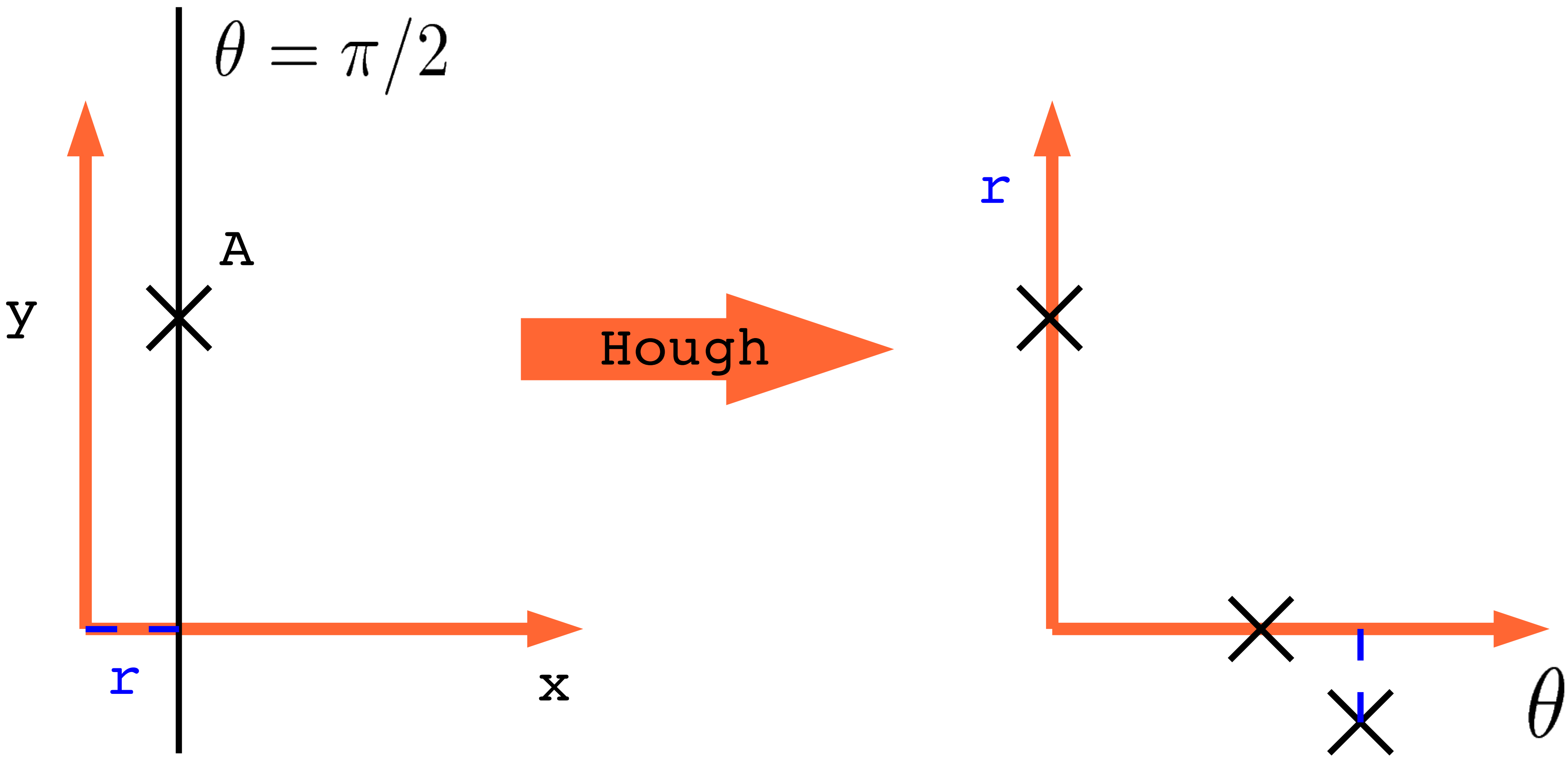
- Transformation d'**un point** dans l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

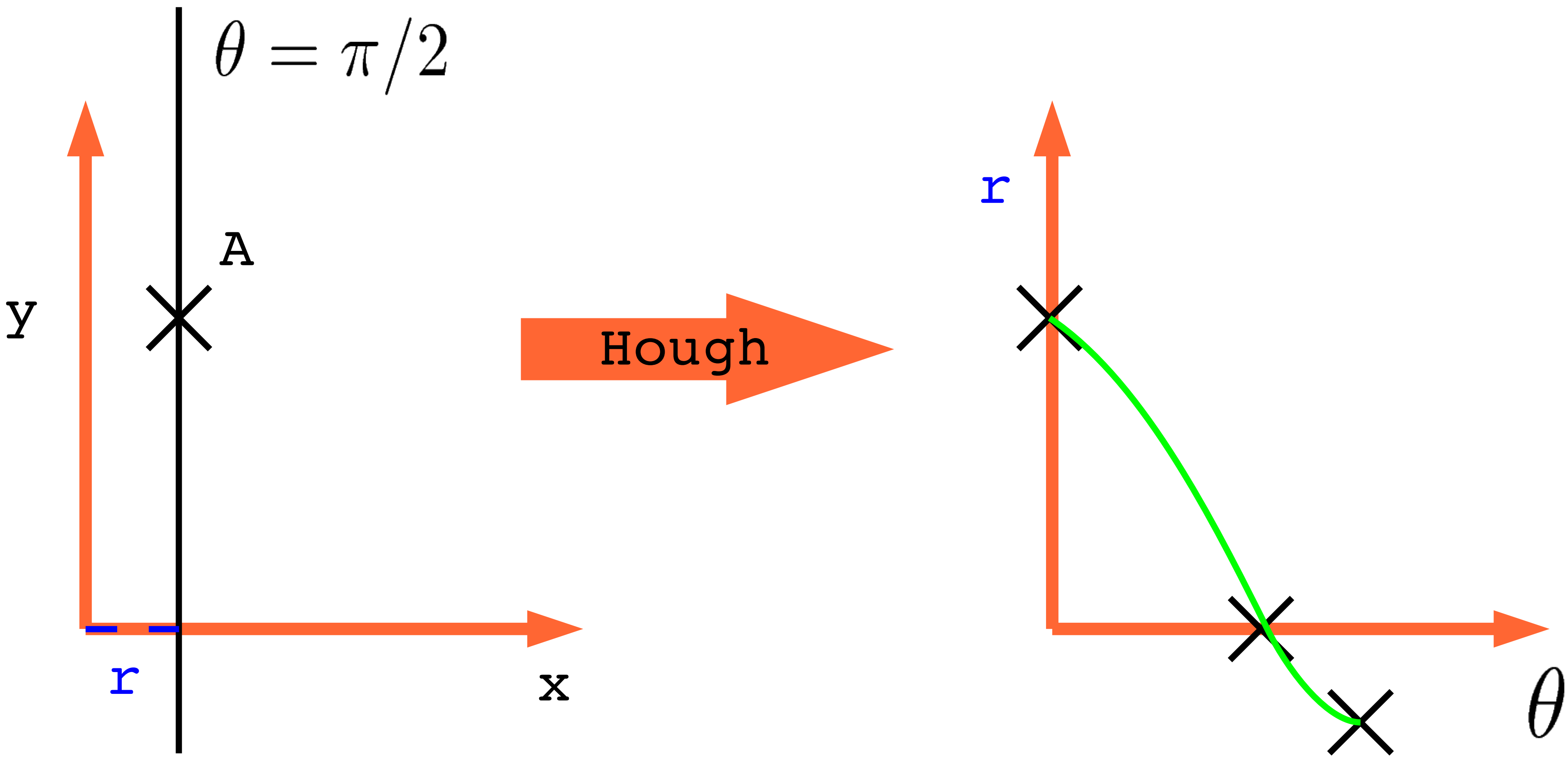
- Transformation d'**un point** dans l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

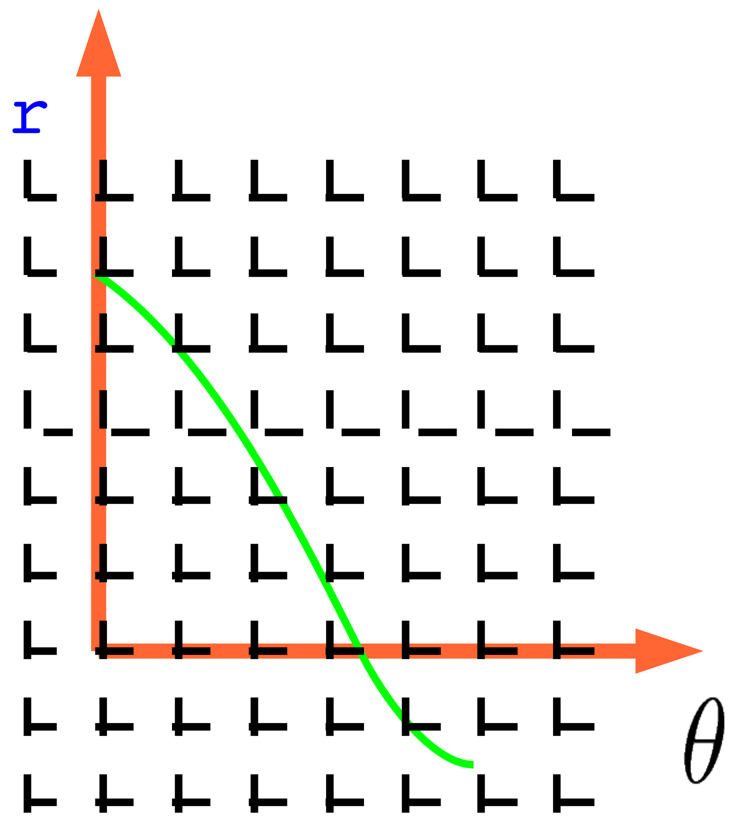
- Transformation d'**un point** dans l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

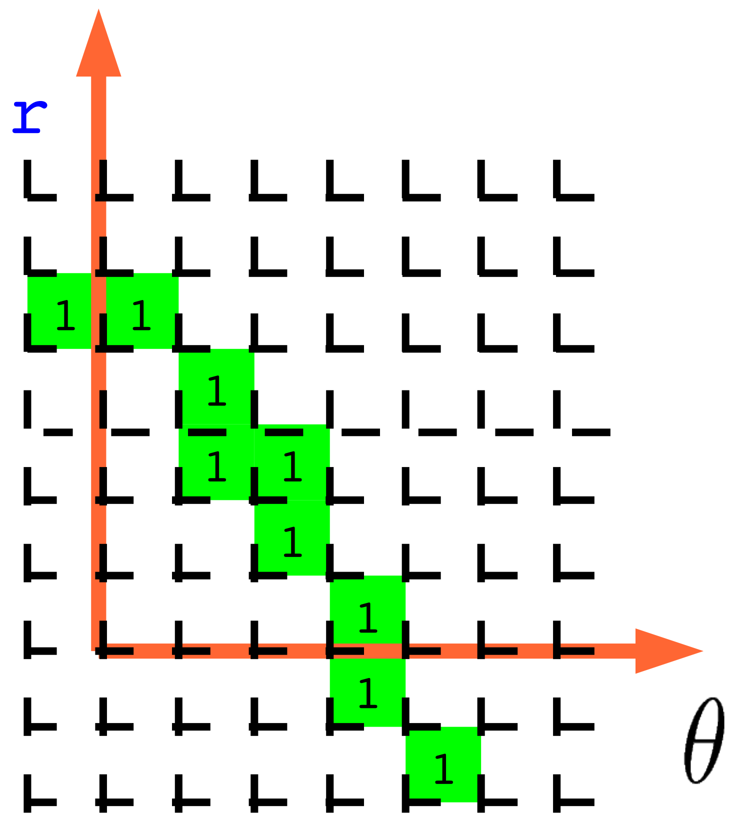
- Discrétisation de l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

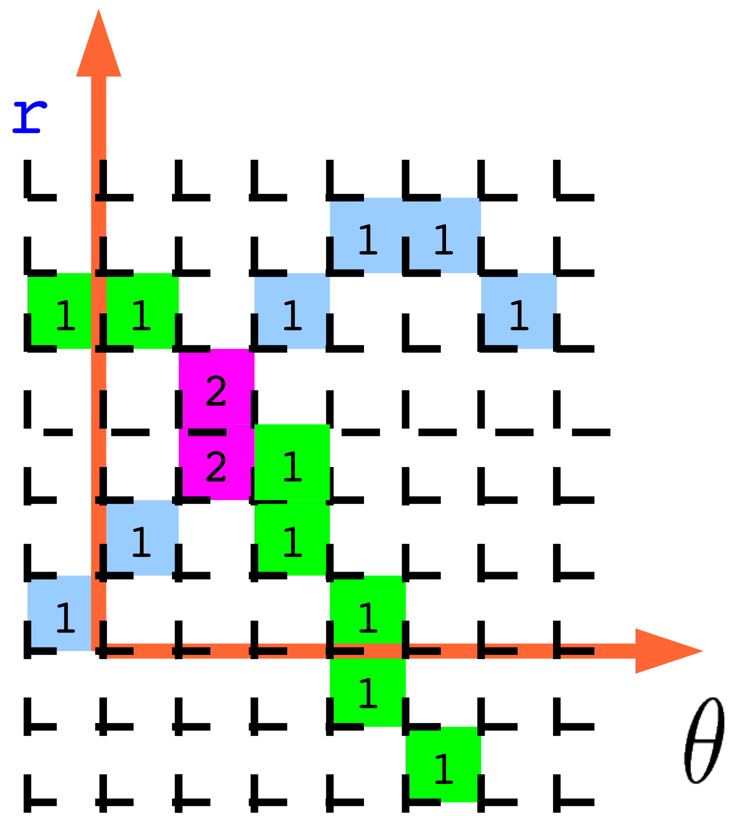
- Discrétisation de l'espace des paramètres:



Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

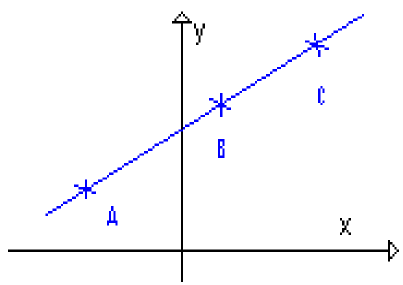
- Accumulation dans l'espace des paramètres:

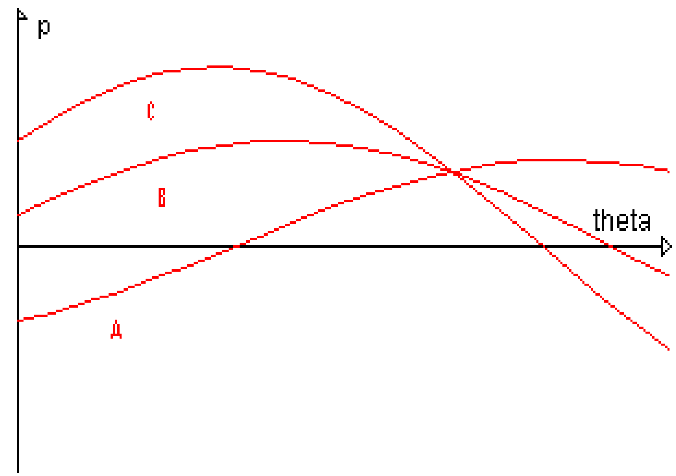


Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

- Les maxima locaux correspondent à des paramètres de doites expliquant le maximum de points





- On les retire tant que leur valeur est supérieure à un seuil :
  - Nombre minimum de points expliqués par une primitive

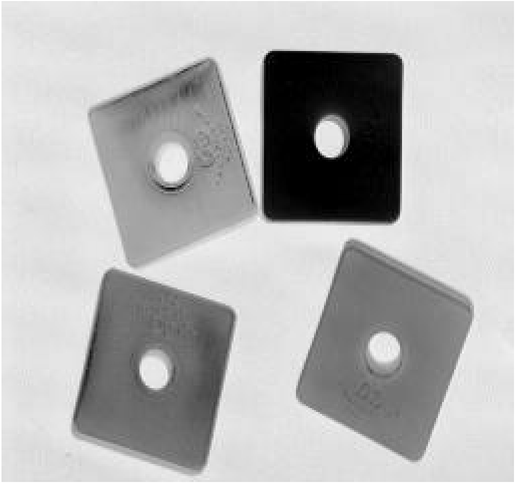
Transformée de Hough

2 paramètres critiques :

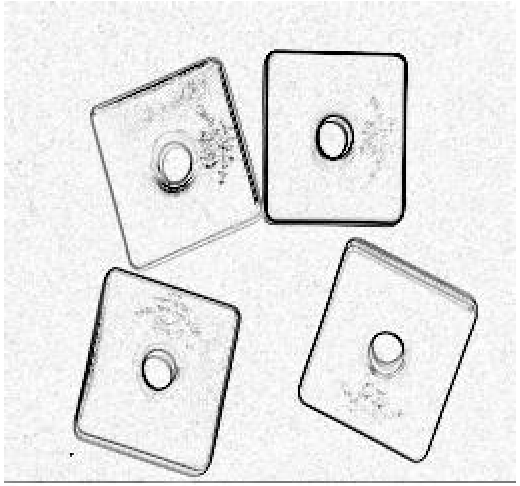
- Pas de discrétisation:
  - Donne le pouvoir de séparation de l'algorithme
  - Compromis entre moins de primitives et moins d'erreur
- Seuil sur l'arrêt:
  - Définit un nombre de points minimum pour justifier l'existence d'une primitive
  - Compromis entre moins de primitives et plus de points expliqués

Transformée de Hough

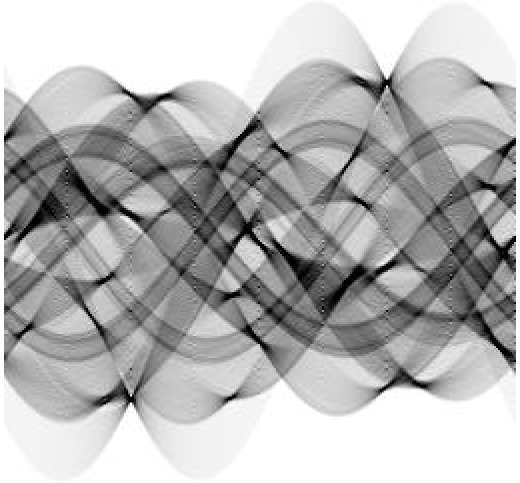
Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$



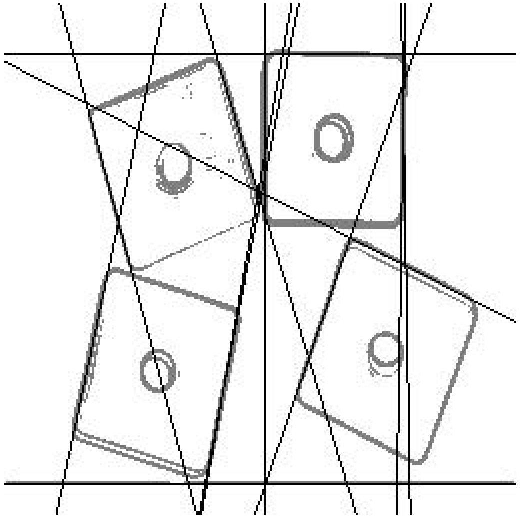
Image



Contours



Accumulateur

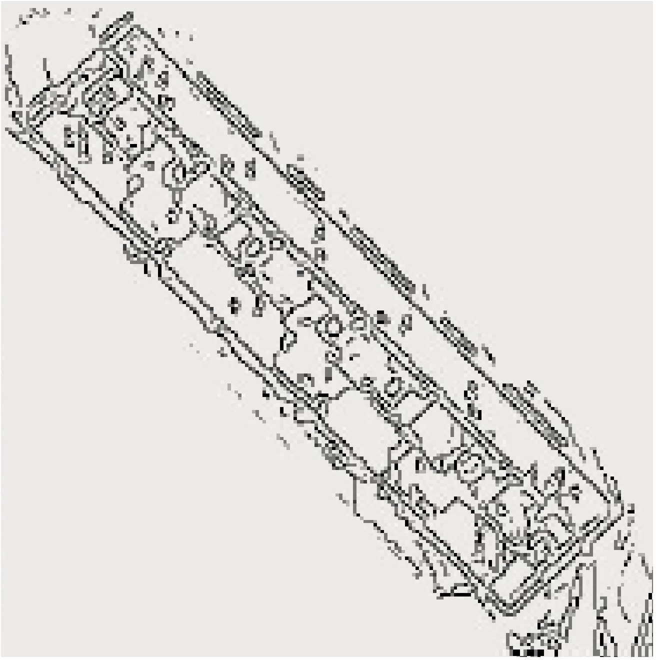


Droites

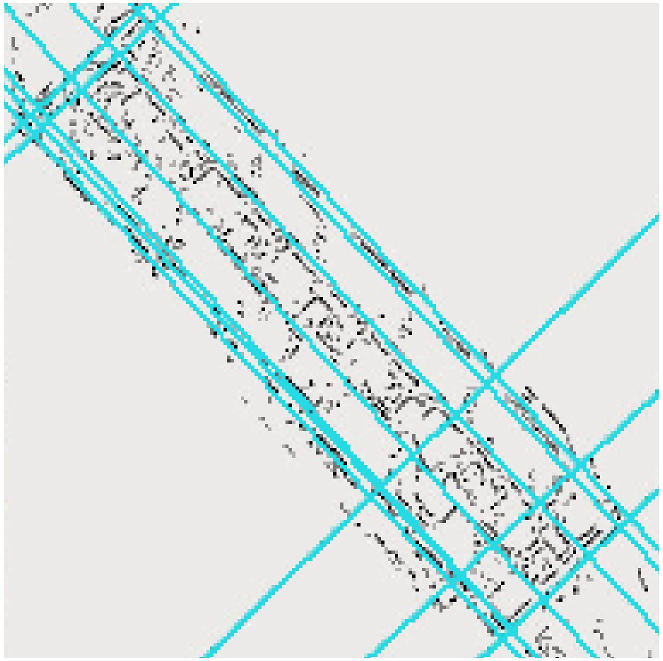
Transformée de Hough

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$

Image



Contours



Droites

Transformée de Hough

Exemple du cercle:

III — RANSAC

RANdom SAmple Consensus

Problème: détection de primitives multiples:

Nuage de points	→	Primitives
$\mathcal{X} = \{\mathbf{x}_i = (x_i, y_i, z_i)^t \in \mathbb{R}^3\}_{i=1..n}$		$\{\mathcal{P}(a_i, b_i, c_i, ...)\}_{i=1..n}$

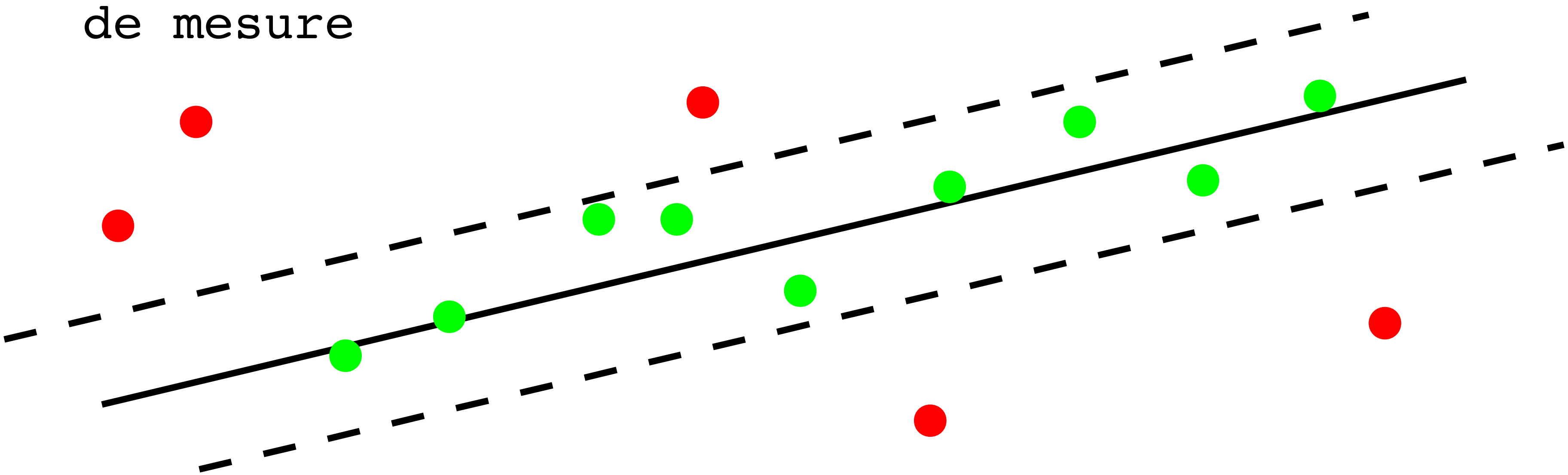
Principe de RANSAC:

- Sélectionner des échantillons de points qui définissent une primitive unique
- Garder la primitive qui explique le plus de points
- Supprimer les points expliqués
- Itérer

RANSAC

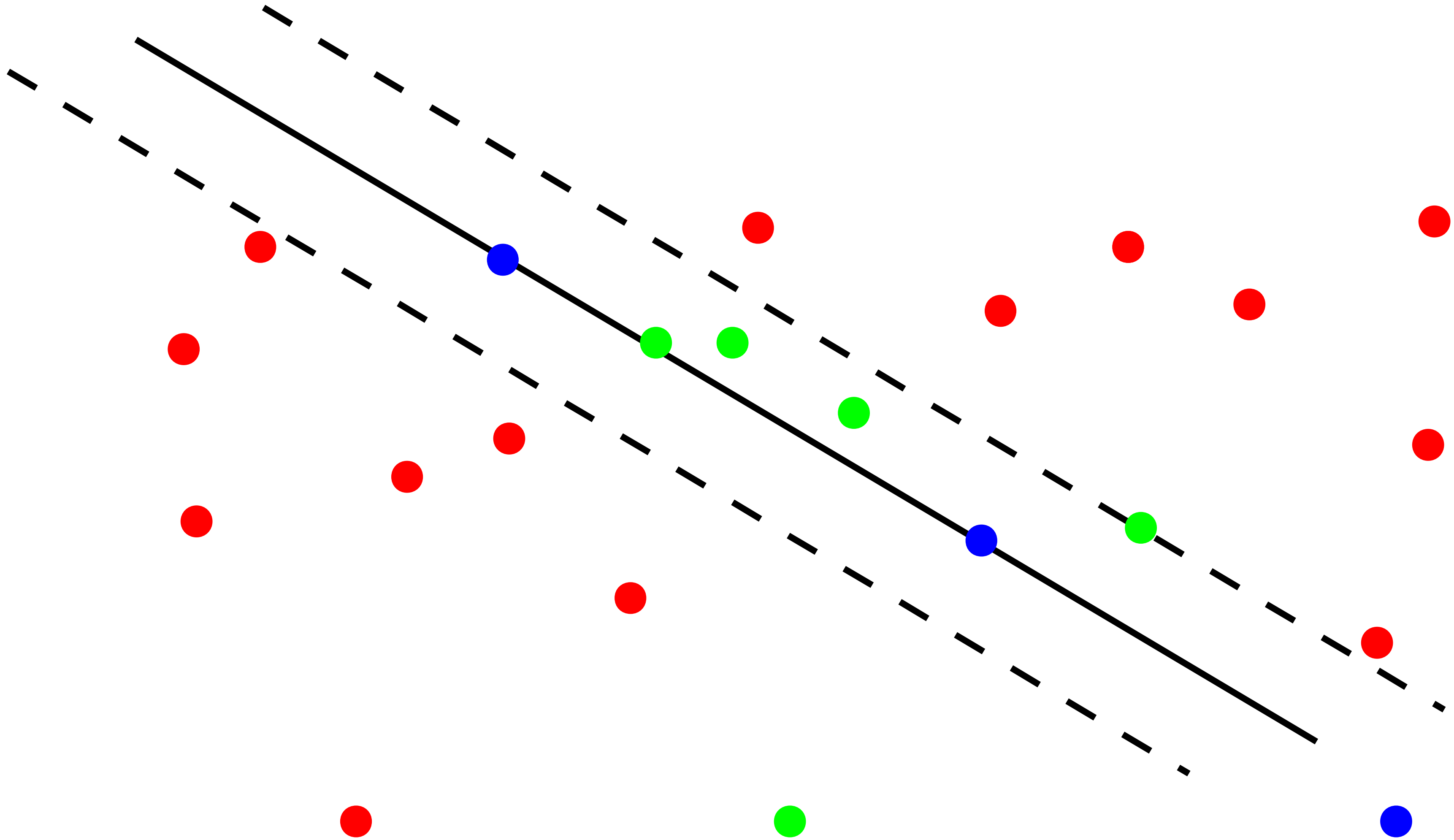
Notion d'inliers/outliers:

- On choisit un seuil de distance
- On appelle **inlier** un point dont la distance à la primitive est inérieure au seuil. On considère que ce point **appartient** à la primitive.
- On appelle **outlier** un point dont la distance à la primitive est supérieure au seuil. On considère que ce point **n'appartient pas** à la primitive. Il peut appartenir à une autre primitive ou venir d'une erreur de mesure



RANSAC

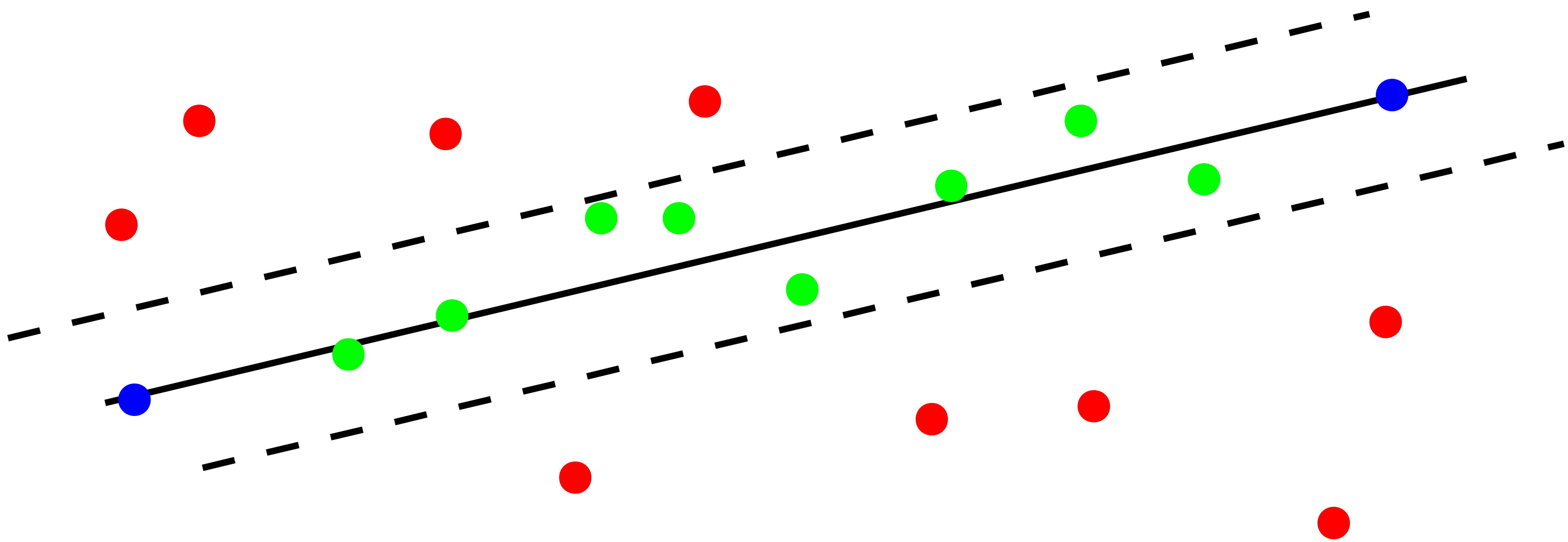
Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$



Outlier 13 | Inlier 4 | Echantillon

RANSAC

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$



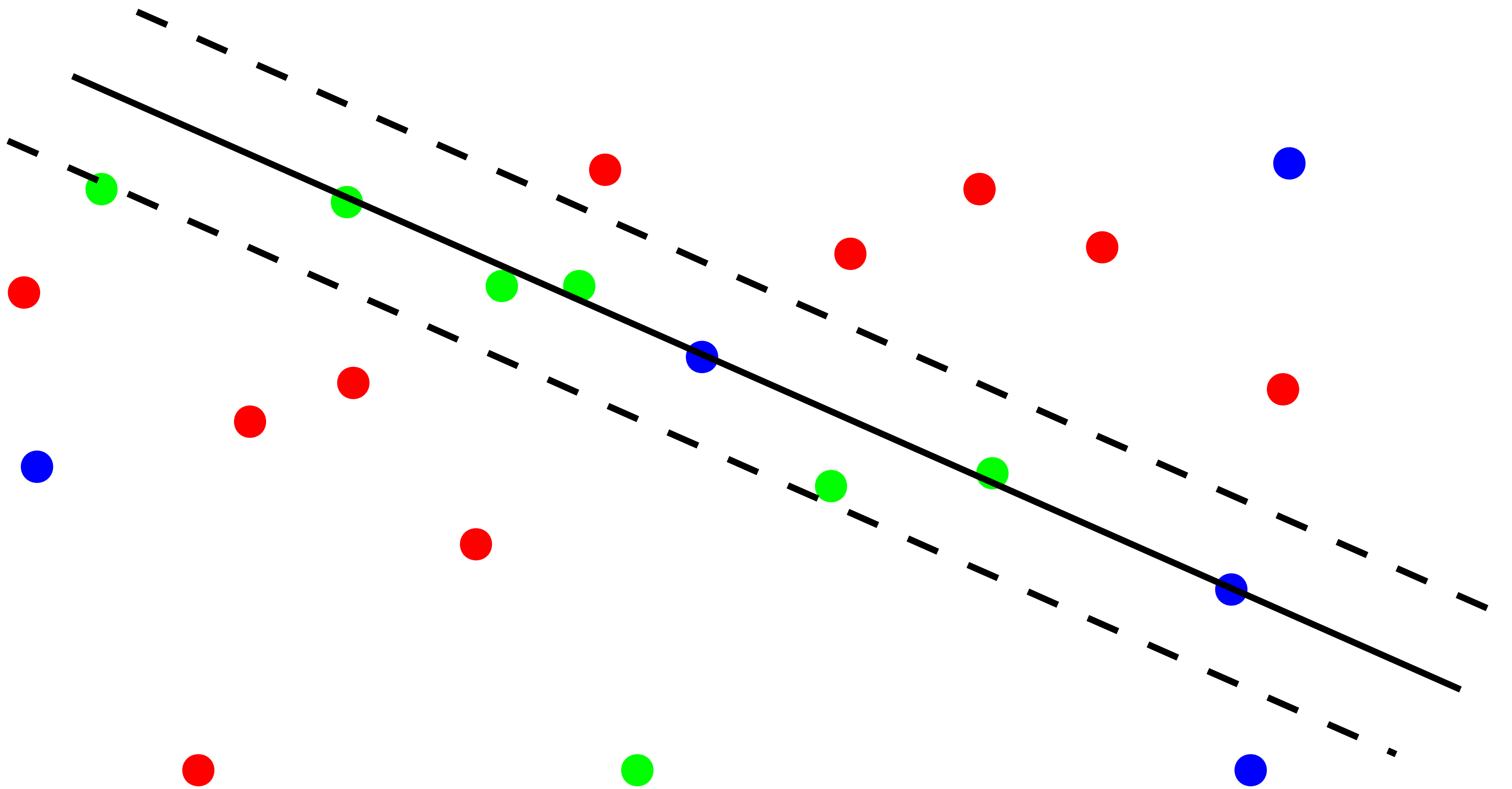
Outlier
9

Inlier
8

Echantillon

RANSAC

Exemple de la droite: $(r, \theta) : x\cos(\theta) + y\sin(\theta) - r = 0$



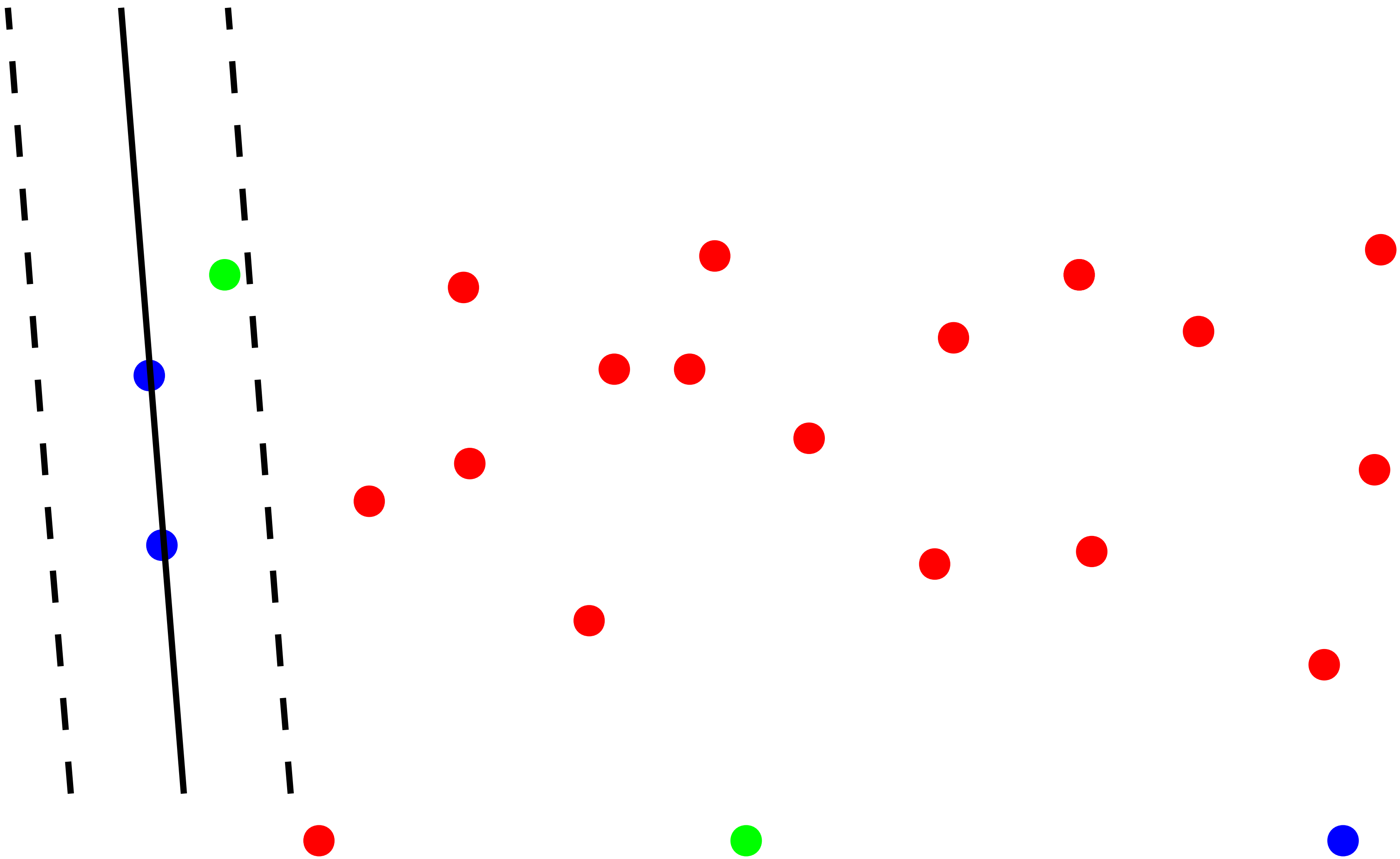
Outlier 11

Inlier 6

Echantillon

RANSAC

Exemple de la droite: $(r,\theta) : x\cos(\theta) + y\sin(\theta) - r = 0$



Outlier
16

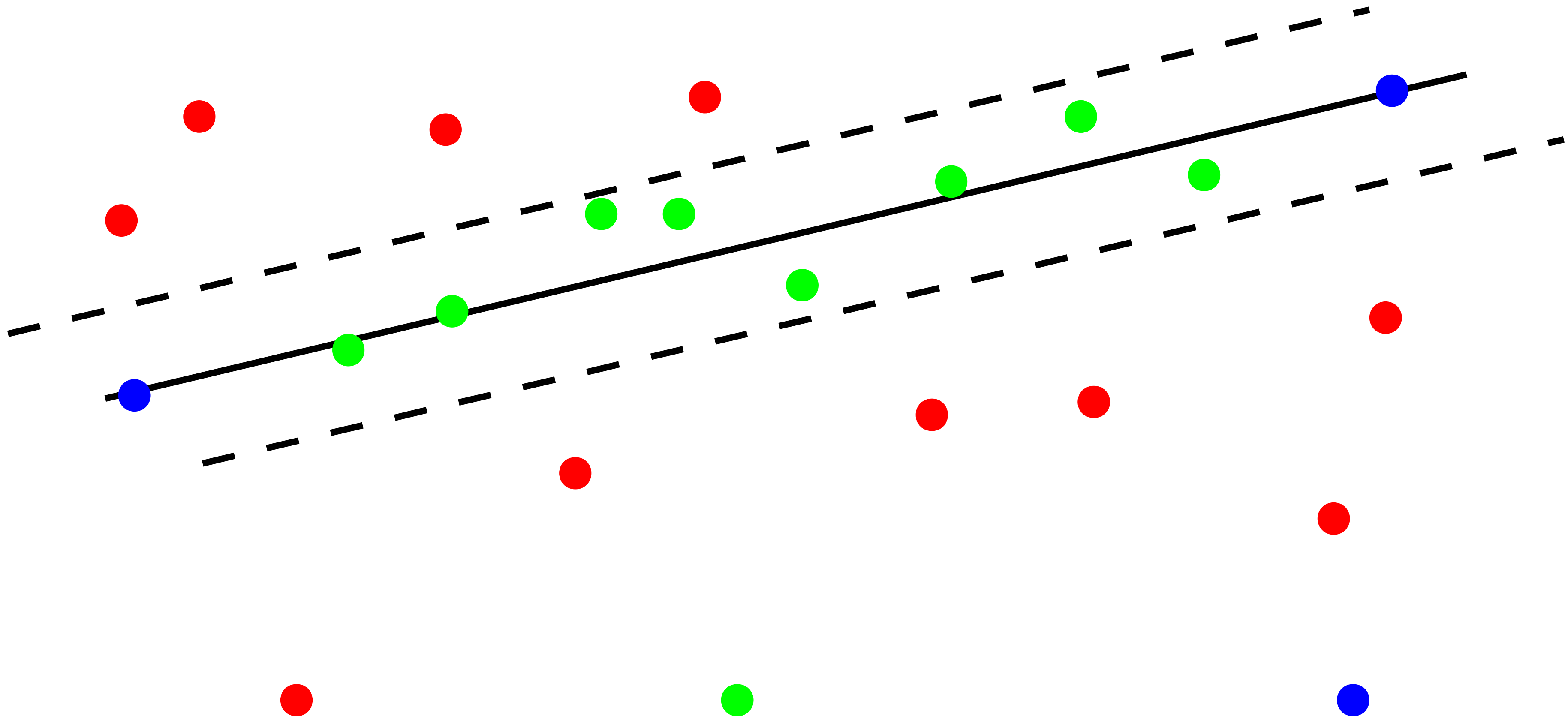
Inlier
1

Echantillon

RANSAC

Maximum d'inliers trouvés:

- On ajoute cette primitive
- On retire ses inliers

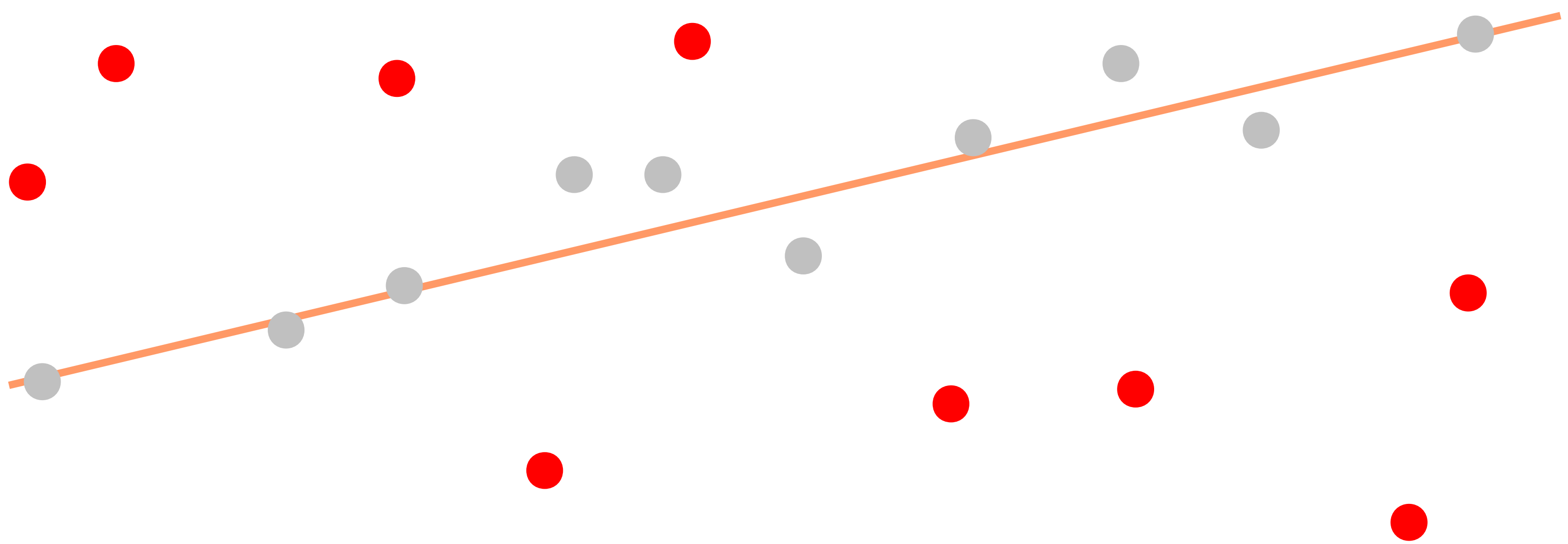


Outlier 9 Inlier 8 Echantillon

RANSAC

Maximum d'inliers trouvés:

- On ajoute cette primitive
- On retire ses inliers



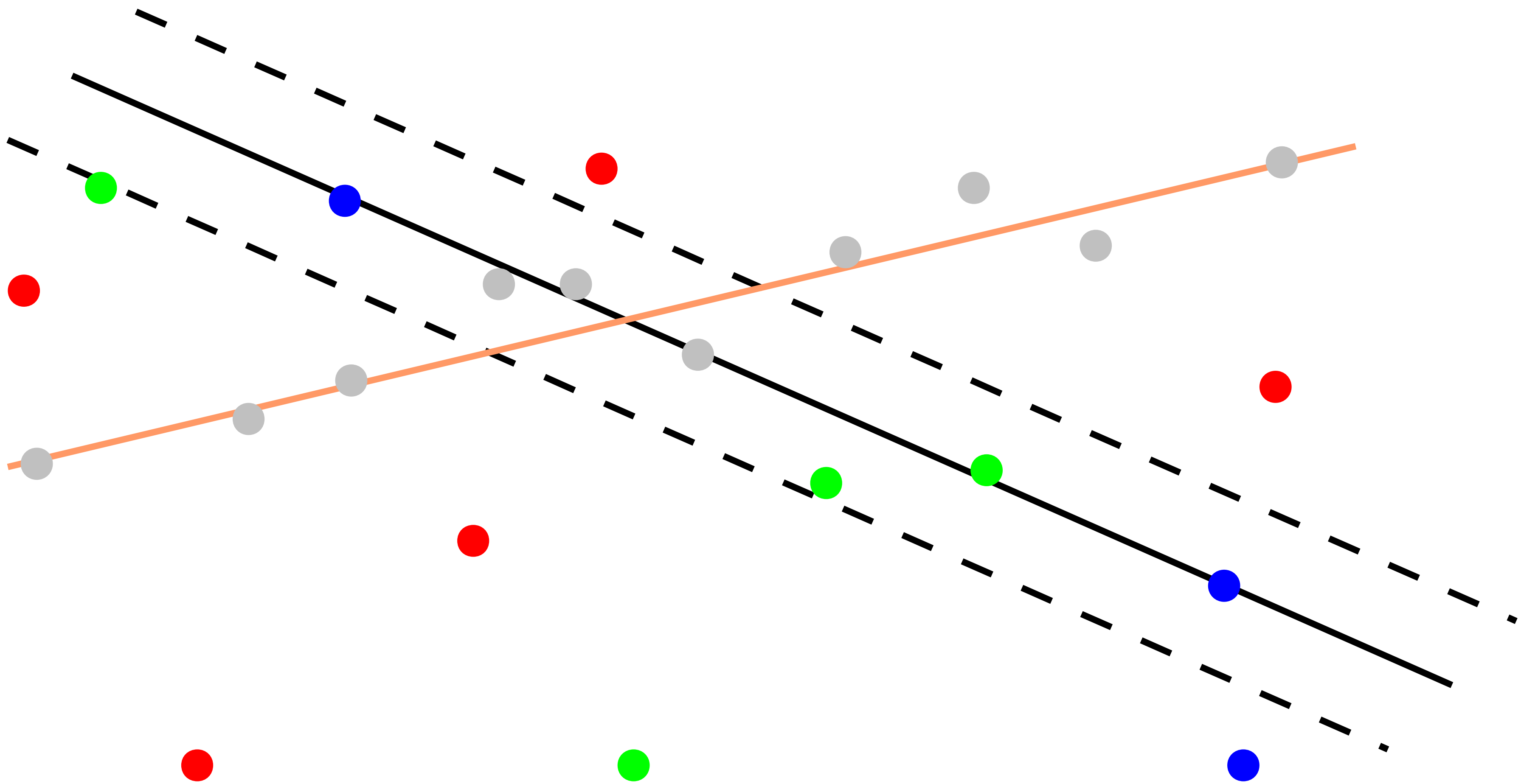
Outlier 9

Inlier 8

Echantillon

RANSAC

Deuxième itération:



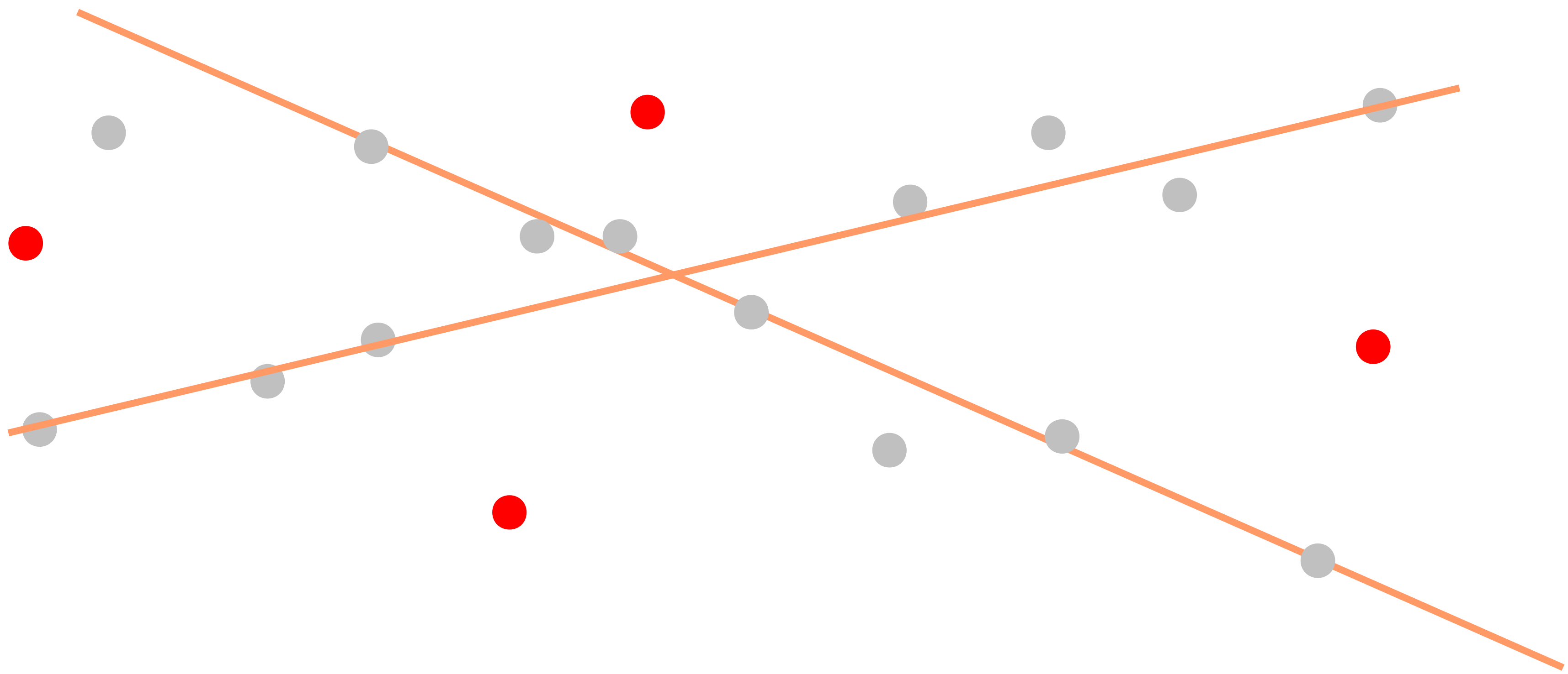
Outlier
4

Inlier
3

Echantillon

RANSAC

Arret quand il n'y a plus assez de points sur la meilleure primitive:



On a trouvé les 2 droites qui expliquent le mieux le nuage de points

RANSAC

Comment choisir le nombre d'itérations ?

- Le nombre minimum de points par primitive est déjà un paramètre, on peut s'en servir:
  - On calcule la probabilité de tirer un point sur une primitive « minimum »: $p_{min} = n_{min}/n_{rest}$
  - On peut donc calculer la probabilité de tirer tous les points définissant la primitive minimum sur celle-ci: $p_{all} = p_{min}^{n}$
  - La probabilité de ne pas trouver la primitive au bout de m tirages est alors: $p_{not} = (1 - p_{all})^{m}$
  - On peut donc se fixer cette probabilité en choisissant:

$$m = \frac{\log(p_{not})}{\log(1 - p_{all})}$$

RANSAC

3 paramètres critiques :

- Distance de séparation inliers/outliers
  - Donne le pouvoir de séparation de l'algorithme
  - Compromis entre moins de primitives et moins d'erreur
- Seuil sur l'arrêt:
  - Définit un nombre de points minimum pour justifier l'existence d'une primitive
  - Compromis entre moins de primitives et plus de points expliqués
- Probabilité de rater une primitive minimale: compromis entre temps de calcul et guaranties sur l'optimalité du résultat

IV Comparaison

Comparaisons

Rappel: critères pour l'extraction de structure

- Minimum de primitives
- Maximum de points expliqués
- Distance minimum des points expliqués aux primitives

Comparaisons

Moindres carrés

- Ne traite qu'une primitive
- N'apparie pas les points aux primitives
- Minimise bien la distance des points expliqués aux primitives

=> appliquer à la fin, quand on a défini un certain nombre de primitives et leurs points appariés

Comparaisons

Hough: 3 critères

- Le nombre de primitives est ajusté en fonction du critère d'arrêt
- Les points peuvent être appariés à la primitives la plus proche, à condition qu'il vote pour elle dans l'espace de Hough
- Pas de minimisation explicite de la distance des points aux primitives:
  - On garantit seulement que la primitive a des paramètres proche des paramètres d'une primitive qui passe par chaque point associé
  - On a un seuil ferme sur l'appariement donné par la discrétisation de l'espace de Hough
  - On peut réestimer aux moindres carrés les primitives trouvées en se basant sur les appariements

Hough: inconvénients

- Problèmes de temps de calcul si trop de points
- Problèmes de mémoire si trop de cases:
  - dimention de l'espace de Hough
  - taille des cases
- Pas toujours facile de paramétrer une primitive de façon à ce que la distance dans l'espace de Hough soit significative
- Amélioration: supprimer les maxima dès qu'ils apparaissent et arrêter quand ils n'apparaissent plus assez vite

Comparaisons

Ransac: 3 critères

- Le nombre de primitives est ajusté en fonction du critère d'arrêt
- Définit explicitement l'appariement point/primitive (inliers/outliers), donc le nombre de points expliqués
- Pas de minimisation explicite de la distance des points aux primitives, mais garantie d'une distance maximale
  - On peut là aussi réestimer aux moindres carrés les primitives trouvées en se basant sur les appariements

Ransac: inconvénients

- Pas de garantie de trouver la primitive
  - Mais la proba peut être rendue arbitrairement basse au détriment du temps de calcul
- Pas toujours facile de définir une primitive à partir de n points
- Seuil sur la distance peut poser problème:
  - Remplacer le score d'une primitive par une somme de gaussiennes en la distance des points (adoucit le seuil)
- Pour les primitives bornées (segment, triangle,...) les grandes sont toujours favorisées
  - Retirer sa mesure au score d'une primitive

Conclusion

Conclusion

- L'extraction de structure est un problème bien posé (cf les 3 objectifs) mais difficile :
  - Inconnues continues (paramètres des primitives)
  - Inconnues discrètes (nombre de primitives, appariements points/primitives)
- Pas de solution pour le problème général, donc une combinaison:
  - D'une méthode optimale (moindres carrés) pour trouver les inconnues continues une fois les inconnues discrètes déterminées
  - De méthodes heuristiques (Hough, Ransac) pour déterminer les inconnues discrètes
- Donc pas de garantie d'optimalité globale