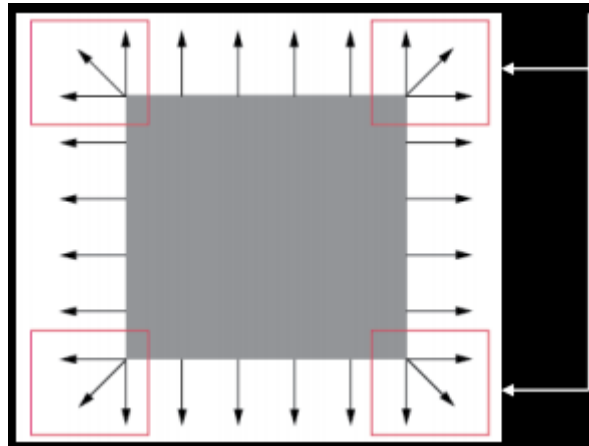


## Détection des contours

Toute l'idée de détection des points isolés et coins repose sur la détection des contours. Là où les contours sont à plusieurs directions, on considère le point comme étant un coin.

➔ Détecter pixels où le gradient est fort dans plus d'une direction – ce sont les pixels situés sur les coins



## Méthode de Harris

### Etape 1 : Calculer les images de gradient $I_x$ et $I_y$

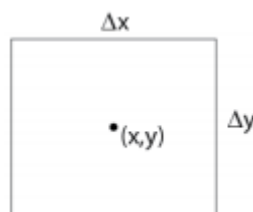
La détection des contours peut se faire facilement à l'aide de l'opérateur Sobel. Il y a d'autres opérateurs comme celui de **Prewitt**, celui que nous avons utilisé pour calculer le gradient de la pente à partir d'un MNT.

Opérateur de Sobel :

$$I_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times A \quad I_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times A$$

### Etape 2 : pour tous les points de l'image de gradient, calculer la matrice de covariance du gradient (dans une fenêtre $2N + 1 \times 2N + 1$ ) et en extraire les valeurs propres $\lambda_1$ et $\lambda_2$

Pour visualiser cette étape, on peut prendre un pixel  $(x, y)$  et un voisinage  $(\Delta X, \Delta Y)$



La fonction d'**autocorrélation** de l'image en  $(x, y)$  est donnée par :

*Intuitivement, la corrélation de deux objets mesure leur dépendance réciproque ; L'autocorrélation d'un signal mesurera donc les dépendances internes de ce signal. Nous l'appliquerons dans le cas d'un signal à une variable entière.*

*Une application sur laquelle nous nous attarderons un peu ici est la détection du bruit dans une image.*

$$c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2$$

Si on approxime  $I(x_i + \Delta x, y_i + \Delta y)$  par sa série de Taylor, en se limitant aux 2 premiers termes, on obtient :

$$I(x_i + \Delta x, y_i + \Delta y) = I(x_i, y_i) + \frac{\partial I}{\partial x} |_{x_i, y_i} \Delta x + \frac{\partial I}{\partial y} |_{x_i, y_i} \Delta y$$

$$I(x_i + \Delta x, y_i + \Delta y) = I(x_i, y_i) + I_x(x_i, y_i) \Delta x + I_y(x_i, y_i) \Delta y$$

$$I(x_i + \Delta x, y_i + \Delta y) = I(x_i, y_i) + \begin{bmatrix} I_x(x_i, y_i) & I_y(x_i, y_i) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

Remplaçons maintenant dans la fonction d'autocorrélation :

$$c(x, y) = \sum_w \left[ I(x_i, y_i) - I(x_i, y_i) - \begin{bmatrix} I_x(x_i, y_i) & I_y(x_i, y_i) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right]^2$$

Ceci on peut le simplifier :

$$c(x, y) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

La matrice :

$$M = \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix}$$

Illustre la structure du gradient de l'image dans un voisinage du pixel  $(x, y)$ .

Ses valeurs propres et décrivent l'étalement des valeurs du gradient dans deux directions orthogonales (et peuvent être associées aux "courbures" principales de la fonction d'autocorrélation).

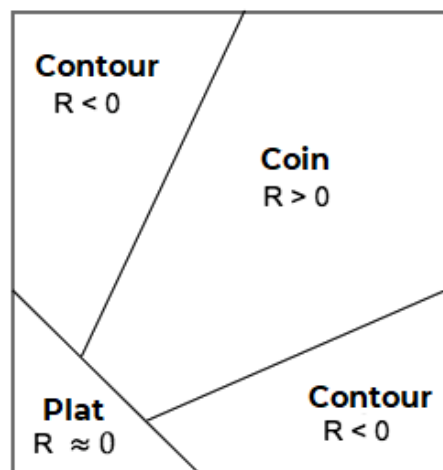
**Etape 3 : On calcule la valeur  $R$  (la réponse de l'opérateur Harris) définie comme :**

$$R = \det(M) - k \times \text{trace}(M)^2$$

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

$k$  : le facteur de sensibilité pour séparer les coins des bords, typiquement une valeur proche de zéro ( $0.4 \leq k \leq 0.6$ )

**Etape 4 : L'analyse des iso-contours de  $R$  fournit l'interprétation suivante :**



## Méthode de Förstner

Förstner identifie également les points d'intérêt à l'aide de la matrice  $M$ .

Or, l'algorithme de Förstner est utilisé pour obtenir une solution approximative avec une précision sous-pixel. Il résout le point le plus proche de toutes les lignes tangentes du coin dans une fenêtre donnée et est une solution par moindres carrés. L'algorithme repose sur le fait que pour un coin idéal, les lignes tangentes se croisent en un seul point.

### Etape 1 : Calcul des images du gradient $I_x$ et $I_y$

Les filtres de Roberts sont une approche discrète de la dérivée de pas 1 d'une fonction : le gradient de cette fonction.

Si  $I(x, y)$  représente un pixel dans une image, alors les amplitudes des gradients en  $x$  et en  $y$  peuvent s'écrire respectivement :

$$I_x = \frac{\partial I}{\partial x} = I(x_{i+1}, y_{j+1}) - I(x, y)$$

$$I_y = \frac{\partial I}{\partial y} = I(x, y_{j+1}) - I(x_{i+1}, y)$$

Cela revient à convoluer l'image avec les deux filtres  $R_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$  et  $R_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$

### Etape 2 : pour tous les points de l'image de gradient, calculer la matrice de covariance du gradient, et en extraire les valeurs propres $\lambda_1$ et $\lambda_2$

$$M = \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix}$$

### Etape 3 : Calcul des valeurs d'intérêts $q$ et $w$ avec $M$

Contrairement à Harris, Förstner prend en compte les deux valeurs propres  $\lambda_1$  et  $\lambda_2$  de l'inverse de la matrice comme valeur d'intérêt. Elles définissent les axes d'une ellipse d'erreur. Par le calcul de leur taille :

$$w = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\det(M)}{\text{trace}(M)}; \quad w > 0$$

Et le facteur de forme relatif à la circularité :

$$q = 1 - \left( \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 = \frac{4 \det(M)}{\text{trace}(M)^2}; \quad 0 \leq q \leq 1$$

Les propriétés suivantes peuvent être déduites :

- Les petites ellipses circulaires définissent un point d'intérêt
- Les ellipses d'erreur allongées suggèrent un bord droit
- Les grandes ellipses marquent une zone homogène

### Etape 4 : Détermination des points candidats avec la condition $q > T_q$ et $w > T_w$

Un point d'intérêt est présent si les valeurs seuils données  $T_w$  et  $T_q$  sont dépassées. Les paramètres appropriés pour cela se situent dans l'intervalle :

$$\begin{cases} T_w = (0.5 \dots 1.5) \bar{w} \\ T_q = (0.5 \dots 0.75) \end{cases}$$

Avec  $\bar{w}$  la moyenne de toutes les valeurs  $w$  de l'image.

### Etape 5 : Déterminer les coins comme points extrêmes qui prennent les plus grandes valeurs des points candidats