



HAL
open science

Calcul des angles charnières à partir d'images numériques tournées

Yohan Thibault, Yukiko Kenmochi

► **To cite this version:**

Yohan Thibault, Yukiko Kenmochi. Calcul des angles charnières à partir d'images numériques tournées. 20èmes journées de l'association francophone d'informatique graphique, Nov 2007, Marne-la-Vallée, France. pp.156-166. hal-00622236

HAL Id: hal-00622236

<https://hal.science/hal-00622236>

Submitted on 21 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calcul des angles charnières à partir d'images numériques tournées

Yohan Thibault¹ Yukiko Kenmochi^{1,2}

¹Université Paris Est - Marne la Vallée

²Université Paris Est - Marne la Vallée

Abstract

Nouvel et Rémila ont proposés une notion de rotation discrète basée sur les angles charnières pour les images numériques en 2D [NR06]. Les angles charnières sont connus comme étant des angles particuliers déterminés par une image numérique donnée, tel que quelque soit l'angle compris entre deux angles charnières consécutifs le résultat renvoyé par RED soit le même. Ils ont montrés que tout angle charnière peut être représenté de manière unique sous la forme d'un triplet d'entiers, et donc qu'une rotation d'image numérique pouvait être réalisée avec un calcul uniquement en entiers. Dans cet article, nous montrerons, dans un premier temps, comment obtenir l'angle charnière directement inférieur à un angle Euclidien donné. Ensuite nous présenterons un algorithme efficace pour effectuer une rotation discrète à partir de l'angle charnière précédemment trouvé, avec le même résultat que l'angle Euclidien. Nous montrerons enfin comment obtenir, à partir d'une paire d'images numériques, le triplet d'entiers qui définit la relation géométrique entre ces deux images.

1. Introduction

La rotation est l'une des transformations de base pour les images numériques, elle est nécessaire dans de nombreuses applications pour l'imagerie numérique. Le but de cet article est de trouver la relation géométrique existante entre deux images numériques, dont on sait qu'elle représente le même objet, c'est à dire la transformation pour passer de l'une à l'autre. Dans le plan Euclidien \mathbb{R}^2 , la rotation est définie comme suit. La rotation Euclidienne d'angle γ déplace un point (x_1, y_1) de \mathbb{R}^2 en un point (x_2, y_2) de \mathbb{R}^2 tel que

$$x_2 = x_1 \cos \gamma - y_1 \sin \gamma, \quad (1)$$

$$y_2 = x_1 \sin \gamma + y_1 \cos \gamma. \quad (2)$$

Considérons maintenant la rotation dans le plan discret \mathbb{Z}^2 . Du fait que les valeurs x_2 et y_2 dans (1) et (2) ont peu de chance d'être entières même si x_1 et y_1 sont des entiers, nous ne pouvons appliquer (1) et (2) au points de \mathbb{Z}^2 directement. L'approximation la plus simple d'une telle rotation Euclidienne est l'utilisation de la fonction arrondie, telle que

$$P_2 = \lfloor P_1 \cos \gamma - Q_1 \sin \gamma + \frac{1}{2} \rfloor, \quad (3)$$

$$Q_2 = \lfloor P_1 \sin \gamma + Q_1 \cos \gamma + \frac{1}{2} \rfloor, \quad (4)$$

où un point (P_1, Q_1) dans \mathbb{Z}^2 est déplacé en (P_2, Q_2) dans \mathbb{Z}^2 . Une telle rotation entre deux points discrets de \mathbb{Z}^2 est appelé rotation Euclidienne discrétisée d'angle γ , abrégé par la suite RED. Cependant, RED utilise les fonctions sinus et cosinus qui elles-mêmes utilisent des nombres réels ce qui constitue un problème. De plus en pratique, les erreurs d'approximation de ces fonctions peuvent causer des erreurs dans les résultats de RED.

Dans cet article, nous avons pour but d'éviter de tels erreurs. A cette fin, nous avons développé un algorithme de rotation garantissant d'obtenir les mêmes résultats que RED. De plus notre algorithme nous permet d'effectuer la rotation d'images numériques en utilisant uniquement des calculs avec des entiers, i.e. sans utiliser de sinus ou de cosinus. Une rotation n'utilisant que des calculs avec des entiers sera appelée dans cet article une rotation discrète.

Il existe déjà des travaux sur des rotations discrètes. Andrès a étudié quelques rotations discrètes se basant sur les rotations par cercles, par droites, et par droites Pythagoriciennes, respectivement [And92]. Toutes ces rotations utilisent seulement des entiers durant leurs calculs, mais leur résultats sont rarement les mêmes que ceux obtenus par RED. Dans [And96], il a proposé une autre rotation dis-

crête appelée rotation quasi transvection. Bien que les résultats obtenus soient meilleurs que ceux des précédentes méthodes, cette rotation discrète ne garantit toujours pas d'obtenir les mêmes résultats que RED. Dans [NR06], Nouvel et Rémila ont, de leur côté, proposé une nouvelle rotation discrète se basant sur les angles charnières. Les angles charnières sont connus comme étant des angles particuliers déterminés par une image numérique donnée, tel que quelque soit l'angle compris entre deux angles charnières consécutifs le résultat renvoyé par RED soit le même. En d'autres termes, les angles charnières correspondent aux discontinuités de (3) et (4). Nouvel et Rémila ont montré que tout angle charnière peut être représenté de manière unique sous la forme d'un triplet d'entiers, et donc qu'une rotation d'image numérique pouvait être réalisée avec un calcul uniquement en entiers. De plus leur algorithme donne les mêmes résultats que RED. En conséquence, les angles charnières, désignés par les triplets d'entiers leur correspondant, donnent suffisamment d'informations pour effectuer toutes les rotations d'images possibles.

Dans cet article, nous aborderons la rotation discrète basée sur les angles charnières. Nous introduirons dans un premier temps la notion d'angle charnière ainsi que certaines de leurs propriétés. Ensuite, nous montrerons comment obtenir un angle charnière à partir d'un angle donné, puis comment à partir du triplet d'entiers calculer de manière efficace la rotation d'une image numérique. Parce que l'intérêt dans [NR06] était plus d'étudier la dynamique de la rotation discrète que les angles charnières en eux-mêmes, leur algorithme pour la rotation d'images numériques nécessite de calculer tous les angles charnières possibles pour une image donnée; la complexité en temps est alors de $O(n^3 \log(n))$ où n est la taille du plus grand côté de l'image. Nous améliorons leur algorithme pour notre objectif en en développant un avec une complexité de $O(n^2 \log(n))$. Nous présenterons enfin comment obtenir, à partir d'une paire d'images numériques, représentant le même objet, le triplet d'entiers qui définit la relation géométrique entre ces deux images.

2. Les angles charnières

Considérons les points de \mathbb{Z}^2 comme les centres des pixels, leurs rotations se feront autour d'un centre à coordonnées entières. Les angles charnières sont des angles particuliers qui déplacent par rotation des points de \mathbb{Z}^2 sur la frontière entre deux pixels. Dans cette section, nous donnons la définition des angles charnières ainsi que leurs propriétés. Nous introduisons aussi les angles Pythagoriciens qui sont fortement liés aux angles charnières.

2.1. Définition des angles charnières

Soit \vec{x} un point dans \mathbb{R}^2 tel que $\vec{x} = (x, y)$. On dit que \vec{x} possède une coordonnée demie entière si $x - \frac{1}{2} \in \mathbb{Z}$ ou/et

$y - \frac{1}{2} \in \mathbb{Z}$. L'ensemble des points qui possèdent au moins une coordonnée demie entière se note \mathcal{H} et s'appelle la demie grille. Plus simplement, \mathcal{H} représente l'ensemble des points se trouvant sur la frontière des pixels dont le centre sont des points de \mathbb{Z}^2 .

Définition 1 Un angle α est un angle charnière si il existe au moins un point de la grille de \mathbb{Z}^2 tel que son image par la rotation Euclidienne d'angle α appartienne à \mathcal{H} .

De même que \mathcal{H} peut être vu comme la discontinuité de la fonction arrondie dans (3) et (4), les angles charnières peuvent être vu comme la discontinuité de RED. Plus simplement, les angles charnières déterminent la transition durant la rotation d'un point de la grille \mathbb{Z}^2 d'un pixel à un pixel adjacent.

La proposition suivante est importante car elle montre que tout angles charnières peut être exprimé à partir d'un triplet d'entiers.

Proposition 1 Un angle α est un angle charnière si il existe un triplet d'entiers (P, Q, K) tel que

$$2Q \cos \alpha + 2P \sin \alpha = 2K + 1. \quad (5)$$

avec $(K + \frac{1}{2})^2 < P^2 + Q^2$

La preuve est donné dans [NR06]. Géométriquement, un angle charnière α est composé de deux segments partants de l'origine et se terminant l'un sur un point de la grille entière (P, Q) et l'autre sur un point de la demi grille $(K + \frac{1}{2}, \lambda)$ comme montré sur la Figure 1(gauche). On peut déduire de cette proposition que tous les calculs liés aux angles charnières peuvent être effectués uniquement avec des entiers. Dans la suite, α indiquera toujours un angle charnière, et nous noterons $\alpha(P, Q, K)$ l'angle charnière généré par le triplet d'entiers (P, Q, K) .

Posons $\lambda = \sqrt{P^2 + Q^2 - (K + \frac{1}{2})^2}$, les équations suivantes peuvent facilement être déduites de (5) et de la Figure 1(gauche).

$$\cos \alpha = \frac{P\lambda + Q(K + \frac{1}{2})}{P^2 + Q^2}, \quad \sin \alpha = \frac{P(K + \frac{1}{2}) - Q\lambda}{P^2 + Q^2}. \quad (6)$$

Notons que l'on peut avoir un point de la demi grille $(\lambda, K + \frac{1}{2})$, à la place de $(K + \frac{1}{2}, \lambda)$. Dans ce cas là, les équations précédentes deviennent $\cos \alpha = \frac{Q\lambda + P(K + \frac{1}{2})}{P^2 + Q^2}$, et $\sin \alpha = \frac{Q(K + \frac{1}{2}) - P\lambda}{P^2 + Q^2}$. Les symétries sur les angles charnières sont importantes car elles permettent de réduire les rotations au premier quadrant du cercle, tel que $\alpha \in [0, \frac{\pi}{2}]$.

Corollaire 1 Chaque triplet d'entiers (P, Q, K) correspond à quatre angles charnières symétriques $\alpha + \frac{\pi k}{2}$ avec $k = 0, 1, 2, 3$.

Dans le but de distinguer le cas $(K + \frac{1}{2}, \lambda)$ du cas $(\lambda, K + \frac{1}{2})$, nous changeons le signe de K ; nous utilisons $\alpha(P, Q, K)$ pour le cas $(K + \frac{1}{2}, \lambda)$, et $\alpha(P, Q, -K)$ pour le cas $(\lambda, K + \frac{1}{2})$.

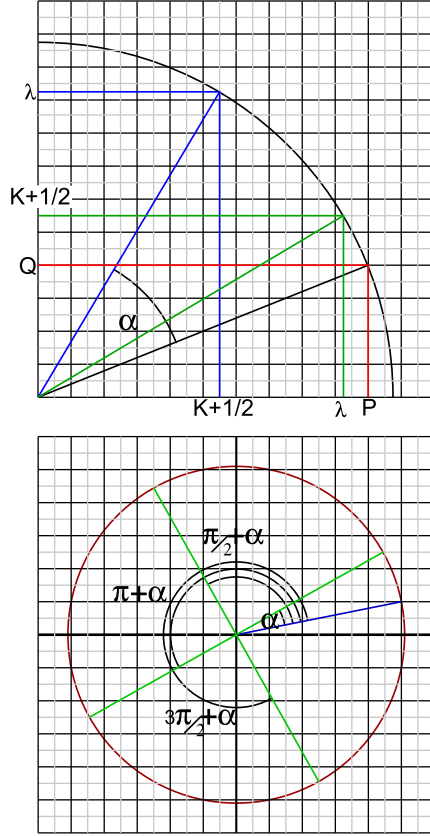


Figure 1: Un angle charnière $\alpha(P, Q, K)$ (gauche) et quatre angles charnières symétriques (droite).

$\frac{1}{2}$). Notons que les symétries nous permettent de restreindre α à l'intervalle $[0, \frac{\pi}{2}]$, nous savons donc que K est toujours positif.

2.2. Propriétés relative aux angles Pythagoricien

Du fait que les angles charnières sont fortement liés aux angles Pythagoriciens, les propriétés des angles Pythagoriciens sont nécessaires pour démontrer certaines propriétés des angles charnières. Par conséquent, nous donnons maintenant la définition des angles Pythagoriciens et leurs propriétés.

Définition 2 Un angle θ est appelé Pythagoricien si $\cos \theta$ et $\sin \theta$ appartiennent à l'ensemble des nombres rationnels \mathbb{Q} .

Nous pouvons déduire de la Définition 2 que tout angle Pythagoricien θ est défini par un triplet d'entier (a, b, c) tel que

$$\cos \theta = \frac{a}{c}, \quad \sin \theta = \frac{b}{c}. \quad (7)$$

Dans la suite, θ indiquera un angle Pythagoricien. Le lemme suivant est nécessaire pour la proposition qui le suit.

Lemme 1 Soit (a, b, c) un triplet d'entiers générant un angle Pythagoricien avec $|a| < |b| < |c|$. Si $\gcd(a, b, c) = 1$, alors c est impair.

Proof Nous supposons que c est pair tel que $c = 2d$ avec d dans \mathbb{Z} . Nous obtenons alors a et b tout deux impair car $a^2 + b^2 = c^2 = (2d)^2$. Sinon nous aurions $\gcd(a, b, c) = 2n$ avec $n \in \mathbb{Z}$, ce qui est contradictoire. Nous pouvons donc Poser $a = 2e + 1$ et $b = 2f + 1$ avec $e, f \in \mathbb{Z}$, en remplaçant nous obtenons $(2e + 1)^2 + (2f + 1)^2 = 4d^2$, ce qui peut être réécrit $e^2 + e + f^2 + f + \frac{1}{2} = d^4$. Cela implique que d ne peut pas appartenir à \mathbb{Z} . Nous pouvons donc conclure que c est impair. \square

Si $\gcd(a, b, c) = i$, alors $\gcd(\frac{a}{i}, \frac{b}{i}, \frac{c}{i}) = 1$ et le triplet d'entiers $(\frac{a}{i}, \frac{b}{i}, \frac{c}{i})$ génère le même angle Pythagoricien que (a, b, c) .

Proposition 2 Soit E_h l'ensemble des angles charnières et E_p l'ensemble des angles Pythagoriciens. Nous avons alors $E_h \cap E_p = \emptyset$.

Proof Supposons qu'il existe un angle α tel que $\alpha \in E_h$ et $\alpha \in E_p$. Du fait que α appartient à E_p , il existe alors un triplet d'entiers (a, b, c) générateur de l'angle α avec $\gcd(a, b, c) = 1$. Par substitution de (7) dans (6), on obtient

$$2 \frac{Qa + Pb}{c} = 2K + 1 \quad (8)$$

d'où on dérive $2 \frac{Qa + Pb}{c} \in \mathbb{Z}$. On sait grâce au lemme 1 que c est impair, on déduit donc que $\frac{Qa + Pb}{c} \in \mathbb{Z}$. Cependant, cela contredit le fait que pour toute pair n, m appartenant à \mathbb{Z} , on ne peut pas avoir $2n = 2m + 1$. On peut en déduire que α ne peut pas appartenir à E_h et E_p simultanément. \square

Cette proposition montre qu'un angle charnière ne peut pas tourner un point de la grille sur un point (x, y) tel que $x = i + \frac{1}{2}$, $y = j + \frac{1}{2}$, avec $(i, j) \in \mathbb{Z}^2$.

3. Calcul d'angles charnières à partir d'un angle Pythagoricien

Dans cette section, nous proposons une méthode pour calculer l'angle charnières correspondant à un angle donné afin d'effectuer la rotation d'une image numérique donnée. Nouvel et Rémila ont présenté une méthode pour calculer tout les angles charnières pour un point de la grille ou un pixel dans une image numérique [NR06]. Leur méthode peut être adaptée pour trouver l'angle charnière qui nous intéresse à savoir la borne inférieure d'un angle donné. La complexité en temps serait alors de $O(n \log(n))$ avec n le nombre d'angle charnière existant pour le point de la grille donné. Notons que n dépend des coordonnées du point de la grille. Dans la sous-section 3.1, nous améliorons leur méthode en utilisant une structure d'arbre sur les angles charnières, la complexité de notre méthode est alors de $O(\log(n))$. Dans la sous-section 3.2, nous présentons une méthode pour trouver l'angle charnière constituant la borne inférieure d'un angle pour une image numérique donnée, c'est à dire pour

l'ensemble des pixels de l'image. Finalement, nous donnons un algorithme de rotation discrète à partir de l'angle charnière précédemment calculé. Cet algorithme donne le même résultat que RED avec l'angle de rotation original. Notons que, dans cet article, l'angle donné en entrée dans l'algorithme est un angle Pythagoricien, comme dans [NR06]. Cependant, il est possible d'utiliser un angle Euclidien car il existe une méthode en temps linéaire $O(m)$ pour approximer un angle Euclidien par un angle Pythagoricien avec une précision de $\frac{1}{10^m}$ [Ang88].

3.1. Calcul de l'angle charnière constituant la borne inférieure pour un point

Pour chaque point de la grille $\vec{p} = (P, Q)$ dans \mathbb{Z}^2 , il y a au plus $n = \lfloor \sqrt{P^2 + Q^2} + \frac{1}{2} \rfloor$ angles charnières différents [NR06]. Cela implique qu'il existe une séquence de $K_i, i = 0, 1, \dots, n-1$ dans \mathbb{Z} , avec $0 < K_i < n$ pour tout \vec{p} . Il est possible de comparer les angles charnières deux à deux, nous pouvons donc obtenir un ensemble croissant totalement ordonné $\{\alpha_1(P, Q, K_1), \alpha_2(P, Q, K_2), \dots, \alpha_{max}(P, Q, K_{max})\}$ tel que $\alpha_1 < \alpha_2 < \dots < \alpha_{max}$. Soit θ un angle Pythagoricien donné, pour trouver l'angle charnière constituant la borne inférieure α_i tel que $\alpha_i < \theta < \alpha_{i+1}$, nous utilisons une structure d'arbre. Par une recherche binaire, nous pouvons trouver α_i en un temps $O(\log(n))$, en supposant que l'on puisse faire la comparaison entre un angle charnière et un angle Pythagoricien en temps constant. Cet algorithme est décrit dans la Figure 1.

```

Data: Point  $\vec{p}(P, Q)$ , angle Pythagoricien  $\theta$ 
Result:  $\alpha(P, Q, K)$ 
var  $K_{max} = \lfloor \sqrt{P^2 + Q^2} - 1 \rfloor$ ;
var  $K_{min} = 0$ ;
var  $K_{min} = 0$ ;
while  $K_{max} + K_{min} \neq 1$  do
  if  $\alpha(P, Q, K) > \theta$  then
    |  $K_{max} = K$ ;
  else
    |  $K_{min} = K$ ;
  end
   $K = \lfloor \frac{K_{max} + K_{min}}{2} \rfloor$ ;
end
return  $\alpha(P, Q, K)$ 
Function Calcul de l'angle charnière

```

La proposition suivante montre que l'on peut faire la comparaison entre un angle charnière et un angle Pythagoricien en temps constant.

Proposition 3 Soit α un angle charnière et θ un angle Pythagoricien. On peut vérifier si $\alpha > \theta$ en temps constant et avec un calcul en entier.

Proof Soit $\alpha(P, Q, K)$ un angle charnière dans $[0, \frac{\pi}{2}]$ et $\theta(a, b, c)$ un angle Pythagoricien dans $[0, \frac{\pi}{2}]$. A partir de (6)

et (7), on obtient

$$\cos \alpha - \cos \theta = \frac{P(K + \frac{1}{2}) + Q\lambda}{P^2 + Q^2} - \frac{a}{c}.$$

Si θ est plus grand que α , $\cos \alpha - \cos \theta > 0$, alors

$$cP(4K + 2) - 4a(P^2 + Q^2) > -4cQ\lambda. \quad (9)$$

Du fait que c, Q, λ sont positifs, la partie droite de (9) est toujours négative. Donc si la partie gauche de (9) est positive ou nulle, alors $\theta > \alpha$. Sinon, on élève (9) au carré, et il ne reste plus qu'à vérifier l'inégalité suivante :

$$[cP(4K + 2) - 4a(P^2 + Q^2)]^2 < 16c^2Q^2\lambda^2. \quad (10)$$

Par définition $\lambda = \sqrt{P^2 + Q^2 - (K + \frac{1}{2})^2}$, donc $16\lambda^2$ est un entier et la partie droite de l'équation (10) n'est composé que de valeurs entières. Nous pouvons donc vérifier (10) avec un calcul uniquement en entier. Si l'inéquation est vraie alors $\theta > \alpha$; sinon $\alpha > \theta$. \square

Notons la particularité des rotations d'angles $\frac{\pi}{2}$ et de ses multiples. Si l'angle de rotation est égal à $\frac{\pi}{2}, \pi, \frac{3\pi}{2}$, il suffit d'inverser et/ou de changer le signe des coordonnées x et/ou y . Pour cette raison nous pouvons restreindre l'angle de rotation en entrée de l'algorithme θ à $0 < \theta < \frac{\pi}{2}$.

3.2. Calcul de l'angle charnière constituant la borne inférieure pour une image

Dans cette sous-section, nous présentons un algorithme, basé sur le précédent, pour calculer l'angle charnière constituant la borne inférieure pour un angle Pythagoricien θ donné et pour une image numérique donnée S se résumant en m points de la grille tel que $S = \{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_m\}$. La sortie est un triplet d'entiers qui représentent l'angle charnière. L'algorithme calcule l'angle charnière correspondant à chacun des m points de la grille, puis les trie pour conserver le plus grand. Plus précisément, il calcule dans un premier temps l'angle charnière inférieur à l'angle θ pour le premier point de S , et il le stocke comme référence. Ensuite il calcule l'angle charnière pour le second point de S , le compare avec la référence et conserve le plus grand des deux. Après avoir répété cette procédure pour l'ensemble des points de S , notre algorithme retourne l'angle charnière α constituant la borne inférieure tel que $\alpha < \theta$. La complexité en temps de notre algorithme est de $O(m \log(n))$ car il appelle m fois la fonction décrite en Figure 1 dont la complexité est de $O(\log(n))$. La Figure 2 illustre notre algorithme. Comme montré dans la proposition suivante, la comparaison entre deux angles charnières est réalisée en temps constant, la complexité globale de notre algorithme n'est donc pas modifiée.

Proposition 4 Soit α_1, α_2 deux angles charnières. Nous pouvons vérifier en temps constant et avec un calcul en entier si $\alpha_1 > \alpha_2$.

La preuve est similaire à celle de la proposition 3.

Data: Image numérique S , Angle Pythagoricien θ
Result: $\alpha(P, Q, K)$
var HA, HA_{temp} * angle charnière *\
 $HA =$ Calcul de l'angle charnière (premier point de S, θ);
for $\forall \vec{p} \in S \setminus \{\vec{p}_1\}$ **do**
 $HA_{temp} =$ Calcul de l'angle charnière(\vec{p}, θ);
 if $HA < HA_{temp}$ **then**
 $HA = HA_{temp}$
 end
end
return HA
Function Calculer un ange charnière pour
une image numérique

3.3. Rotation d'images numériques par angles charnières.

Data: image numérique S , angle charnière α
Result: image tournée S'
var HA : angle charnière;
for $\forall \vec{p} \in S$ **do**
 $HA =$ Calcul de l'angle charnière(\vec{p}, α);
 déplacer \vec{p} en $(K, \lfloor \lambda + \frac{1}{2} \rfloor)$ ou $(\lfloor \lambda + \frac{1}{2} \rfloor, K)$, selon
 le signe de K et stocker dans S' ;
end
return S'
Function Rotation discrète

Dans la Figure 3, nous présentons un algorithme de rotation pour une image numérique. Il suppose que le centre de rotation se trouve sur l'origine. Pour l'ensemble des points, il appelle la fonction décrite dans la Figure 1 afin de trouver l'angle charnière correspondant, angle charnière qui désigne la nouvelle position du point. Soit n la plus grande coordonnée pour l'ensemble des points de S , on sait qu'il y a au plus $4n^2$ point dans S . Nous pouvons en conclure que la complexité est de $O(n^2 \log(n))$. Les résultats obtenue par la rotation discrète par angles charnières sont exactement les mêmes que ceux obtenus par RED. L'avantage de notre méthode est de ne pas recourir au calcul flottant, ce qui la désigne particulièrement aux processeurs à arithmétique entière.

4. Obtention d'une paire d'angles charnières à partir d'une paire d'images numériques

Dans cette section, nous montrons comment obtenir, à partir d'une paire d'images numériques, dont la seconde est une rotation de la première, l'encadrement par deux angles charnières de l'angle de rotation pour passer de l'une à l'autre. Soit $A = \{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_l\}$ et $B = \{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_l\}$ deux ensembles de points de la grille dont on connaît les correspondances tel que \vec{p}_i corresponde à \vec{q}_i . Pour A et B donnés, on obtient une paire d'angles charnières α_1, α_2 , tel que $\alpha_1 < \gamma < \alpha_2$ ou γ est un angle de rotation cohérent pour la

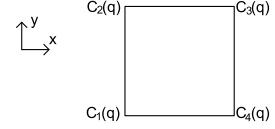


Figure 2: La demi-grille $\mathcal{H}(\vec{q})$, à savoir le pixel autour du point \vec{q} ainsi que ses quatre coins

correspondance des points de A et B . Dans la suite, nous supposons que A est l'ensemble d'origine et B l'ensemble de points tournés par l'angle γ .

4.1. Définition du centre de rotation

Pour toute rotation, il est nécessaire de définir le centre de rotation. Dans cet article, nous prenons un des points de la grille appartenant à l'image numérique comme centre de rotation. En posant les centres de rotations en \vec{p}_1 et \vec{q}_1 , on définit deux fonctions \mathcal{T}_A et \mathcal{T}_B tel que

$$\begin{aligned}\mathcal{T}_A(\vec{p}_i) &= \vec{p}_i - \vec{p}_1 \\ \mathcal{T}_B(\vec{q}_i) &= \vec{q}_i - \vec{q}_1\end{aligned}$$

pour tout $\vec{p}_i \in A, \vec{q}_i \in B$, afin d'obtenir les centres de rotations sur l'origine après translations. Dans la suite, nous utiliserons les ensembles de points $A' = \{\mathcal{T}_A(\vec{p}_1), \mathcal{T}_A(\vec{p}_2), \dots, \mathcal{T}_A(\vec{p}_l)\}$ et $B' = \{\mathcal{T}_B(\vec{q}_1), \mathcal{T}_B(\vec{q}_2), \dots, \mathcal{T}_B(\vec{q}_l)\}$, à la place de A et B . Cependant pour simplifier la lecture, nous les noterons par $A = \{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_l\}$ et $B = \{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_l\}$.

4.2. calcul des angles charnières pour deux paires de points en correspondance

Dans cette sous-section, nous considérons le cas avec $A = \{\vec{p}_1, \vec{p}_2\}$ et $B = \{\vec{q}_1, \vec{q}_2\}$ où $\vec{p}_i = (P_i, Q_i)$ et $\vec{q}_i = (R_i, S_i)$. Définissons d'abord un cercle $\mathcal{C}(\vec{p}_2)$ de centre \vec{p}_1 passant par \vec{p}_2 . Donc le rayon de $\mathcal{C}(\vec{p}_2)$ est $r = d(\vec{p}_1, \vec{p}_2)$ où $d(\vec{p}_1, \vec{p}_2)$ est la distance Euclidienne entre \vec{p}_1 et \vec{p}_2 . Considérons la demi grille autour de \vec{q}_2 telle que

$$\begin{aligned}\mathcal{H}(\vec{q}_2) = \{(x, y) \in \mathcal{H} : S_2 - \frac{1}{2} \leq y \leq S_2 + \frac{1}{2} \text{ quand } x = R_2 + \frac{1}{2}, \\ R_2 - \frac{1}{2} \leq x \leq R_2 + \frac{1}{2} \text{ quand } y = S_2 + \frac{1}{2}\}.\end{aligned}$$

Posons \vec{p}_1 et \vec{q}_1 comme étant les centres de rotations, pour trouver la paire d'angles charnières, nous devons détecter les intersections entre $\mathcal{C}(\vec{p}_2)$ et $\mathcal{H}(\vec{q}_2)$. En d'autre termes, nous étudions les coins de $\mathcal{H}(\vec{q}_2)$ à l'intérieur de $\mathcal{C}(\vec{p}_2)$. Posons les quatre coins de $\mathcal{H}(\vec{q}_2)$ tel que $C_1(\vec{q}_2) = (R_2 - \frac{1}{2}, S_2 - \frac{1}{2})$, $C_2(\vec{q}_2) = (R_2 - \frac{1}{2}, S_2 + \frac{1}{2})$, $C_3(\vec{q}_2) = (R_2 + \frac{1}{2}, S_2 + \frac{1}{2})$, $C_4(\vec{q}_2) = (R_2 + \frac{1}{2}, S_2 - \frac{1}{2})$, comme montré dans la Figure 2. Nous définissons la fonction binaire \mathcal{F} tel

que

$$\mathcal{F}(C_i(\vec{q}_2)) = \begin{cases} 0 & \text{si } C_i(\vec{q}_2) \text{ est en dehors de } \mathcal{C}(\vec{p}_2), \\ 1 & \text{sinon.} \end{cases}$$

Dans le but de calculer $\mathcal{F}(C_i(\vec{q}_2))$ uniquement avec des entiers, nous comparons chacune des équations de la forme $4((R_2 \pm \frac{1}{2})^2 + (S_2 \pm \frac{1}{2})^2)$ avec $4r^2$. Notons que nous pouvons supposer que les intersections entre $\mathcal{C}(\vec{p}_2)$ et $\mathcal{H}(\vec{q}_2)$ existent. Si il n'y a pas d'intersection entre $\mathcal{C}(\vec{p}_2)$ et $\mathcal{H}(\vec{q}_2)$, alors \vec{p}_2 et \vec{q}_2 ne sont pas en correspondance.

Proposition 5 Si deux points \vec{p}_2 et \vec{q}_2 sont en correspondance, un cercle $\mathcal{C}(\vec{p}_2)$ et le bord du pixel $\mathcal{H}(\vec{q}_2)$ possède deux intersections distinctes.

La preuve mathématique rigoureuse est omise dans cet article à cause de la limitation de pages. Par contre, les grandes lignes de la preuve sont donnée dans la suite. Nous distinguons les deux cas suivants.

Le premier cas qui se pose est celui où $\mathcal{C}(\vec{p}_2)$ passe par l'un des coins de $\mathcal{H}(\vec{q}_2)$. Par définition des angles Pythagoriciens, tout les angles entre un point de la grille et le coin d'un pixel sont des angles Pythagoriciens, donc selon la Proposition 2 cela ne peut pas être un angle charnière.

Le second cas est celui où l'intersection entre $\mathcal{C}(\vec{p}_2)$ et $\mathcal{H}(\vec{q}_2)$ est unique et sur une des arêtes de $\mathcal{H}(\vec{q}_2)$. Ce cas pourrait arriver si une des coordonnées de \vec{q}_2 est zéro. Un arc de cercle centré sur l'origine ne peut croiser deux fois une droite parallèle à un des axes que si il coupe lui même l'autre axe. Donc si l'intersection est unique, elle doit être sur l'un des axes, ce qui implique que λ doit être nul. Cependant c'est impossible par définition des angles charnières.

A partir de la Proposition 5, on peut déduire qu'il y a quatre cas différents qui correspondent aux différentes possibilités d'avoir 0,1,2 ou 3 coins à l'intérieur du cercle $\mathcal{C}(\vec{p}_2)$, comme illustré dans la Figure 3.

- Cas A : $\mathcal{C}(\vec{p}_2)$ ne contient aucun coin. Nous avons alors $\mathcal{F}(C_i(\vec{q}_2)) = 0$ pour tout $i = 1, 2, 3, 4$, ce qui est similaire au second cas impossible. Ce cas ne peut arriver que lorsque $R_2 = 0$ ou $S_2 = 0$. Supposons que R_2 et S_2 soient strictement positif. Dans le premier quadrant, la coordonnée y (respectivement la coordonnée x) des points de $\mathcal{C}(\vec{p}_2)$ est strictement décroissante quand x (respectivement y) croît, donc il ne peut croiser deux fois une ligne parallèle à l'axe x (respectivement y). Par conséquent, si par exemple $S_2 = 0$, nous obtenons les deux angles charnières, symétriques autour de l'axe y , décrit par $\alpha(P_2, Q_2, R_2 - 1)$.
- Cas B : $\mathcal{C}(\vec{p}_2)$ contient un seul coin. Par exemple, si $C_1(\vec{q}_2)$ est dans le cercle, nous obtenons les deux angles charnières $\alpha_1(P_2, Q_2, R_2 - 1)$ et $\alpha_2(P_2, Q_2, -S_2 + 1)$.
- Cas C : $\mathcal{C}(\vec{p}_2)$ contient deux coins possédant une coordonnée commune. Par exemple, si $C_1(\vec{q}_2)$ et $C_2(\vec{q}_2)$ sont dans le cercle, nous obtenons les deux angles charnières $\alpha_1(P_2, Q_2, -S_2 + 1)$ et $\alpha_2(P_2, Q_2, -S_2)$.

- Cas D : $\mathcal{C}(\vec{p}_2)$ contient trois coins. Par exemple, si $C_1(\vec{q}_2)$, $C_2(\vec{q}_2)$ et $C_4(\vec{q}_2)$ sont à l'intérieur de $\mathcal{C}(\vec{q}_2)$, nous obtenons les deux angles charnières $\alpha_1(P_2, Q_2, R_2)$ et $\alpha_2(P_2, Q_2, -S_2)$.

La fonction principale de notre algorithme pour trouver les deux angles charnières, illustré dans la Figure 4, peut être divisée en trois parties. La première étape définit le centre de rotation sur \vec{p}_1 et \vec{q}_1 et applique les fonctions définies en 4.1. La seconde étape cherche quels sont les coins se trouvant à l'intérieur de $\mathcal{C}(\vec{q}_2)$ et conserve le résultat sous la forme d'un index. L'index est calculé comme la somme des index associés à chacun des coins tel que l'index du coin $C_i(\vec{q}_2)$ est égal à 2^i . Il est donc facile d'identifier quels coins se trouvent à l'intérieur de $C_i(\vec{q}_2)$ à partir de l'index. La troisième étape appelle la fonction illustré dans la Figure 5 qui renvoie les angles charnières correspondant à l'index. Il existe quatorze valeurs possibles pour l'index. Notons que l'index ne peut jamais prendre les valeurs 5 ou 10. La valeur d'index 15 révèle une erreur car elle implique que tout les points se trouvent à l'intérieur de $\mathcal{C}(\vec{q}_2)$. Du fait que la valeur 0 corresponde au Cas A, nous devons vérifier si $\mathcal{H}(\vec{q}_2)$ possède vraiment deux intersections avec $\mathcal{C}(\vec{q}_2)$. Les autres valeurs de l'index, peuvent être classées par paires. Deux valeurs de l'index (d, e) appartiennent à la même paire si et seulement si $d + e = 15$. Deux valeurs d'une même paire désignent les mêmes angles charnières. Du fait que cet algorithme travaille sur un nombre constant de points et que la quantité de calculs est indépendante des points, sa complexité en temps est constante en $O(1)$.

Data: Points

$$\vec{p}_1(P_1, Q_1), \vec{p}_2(P_2, Q_2), \vec{q}_1(R_1, S_1), \vec{q}_2(R_2, S_2)$$

Result: angles charnières $\alpha_1(P, Q, K_1), \alpha_2(P, Q, K_2)$

Var Entier $Rayon_1, Rayon_2, Index_{coin}, K_1, K_2$;

$$\vec{p}_1 = \mathcal{T}_p(\vec{p}_1); \vec{p}_2 = \mathcal{T}_p(\vec{p}_2); \vec{q}_1 = \mathcal{T}_q(\vec{q}_1);$$

$$\vec{q}_2 = \mathcal{T}_q(\vec{q}_2);$$

$$Rayon_1 = 4 * (P_2^2 + Q_2^2); Rayon_2 = 4 * (R_2^2 + S_2^2);$$

$$Index_{coin} = \sum_{i=1}^4 \mathcal{F}(C_i(\vec{q}_2)) * 2^{i-1};$$

$(K_1, K_2) =$ Trouver la paire d'angles charnières

$(q_2(R_2, S_2), Index_{coin});$

return $(\alpha_1(R_2, S_2, K_1), \alpha_2(R_2, S_2, K_2))$

Function Trouver l'encadrement de l'angle

S

4.3. Calcul des angles charnière de manière incrémentale

D'une manière générale, les ensembles de points A et B contiennent plus de deux points. C'est pourquoi dans cette section, nous étendons notre algorithme présenté dans la section précédente pour toute paires d'ensembles de points A et B qui contiennent m points avec $m > 2$. Un nouvel algorithme illustré dans la Figure 6 effectue les calculs pour l'ensemble des points de manière incrémentale. Cet algorithme est divisé en deux parties. La première partie consiste à initialiser

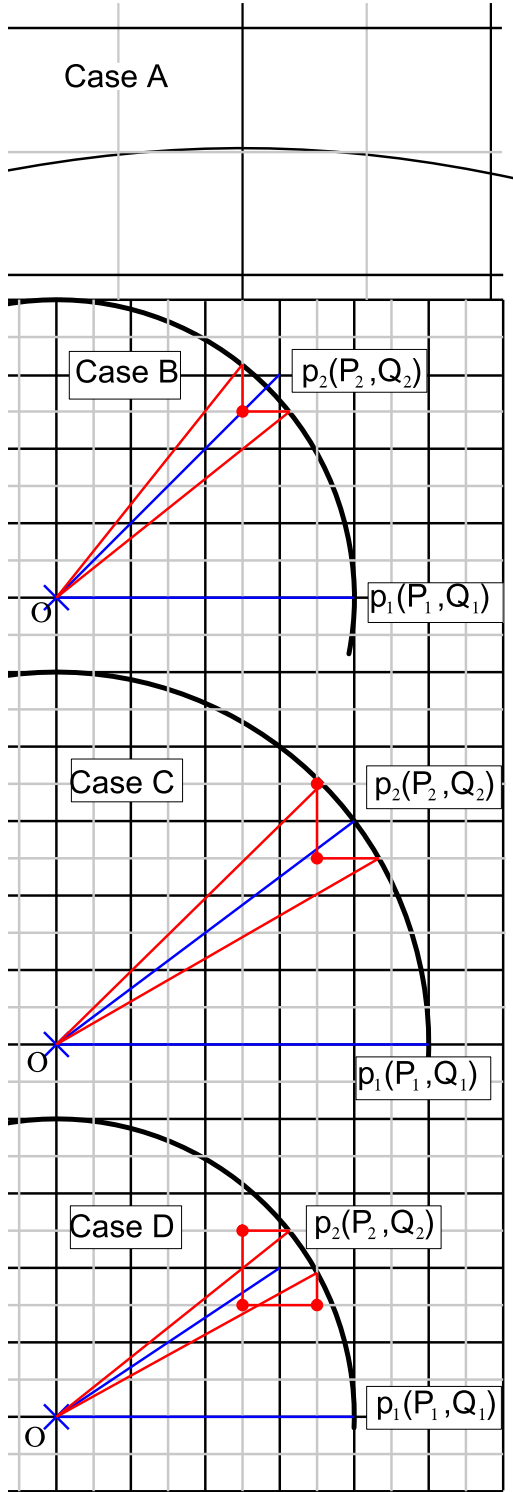


Figure 3: Illustration des cas 1, 2, 3 et 4.

Data: point $\vec{q}_2(R_2, S_2)$, index $Index_{coin}$

Result: Deux entiers pour définir les angles charnières

if $Index_{coin} = 0$ **then**

if $|\sqrt{Rayon_1} - \sqrt{Rayon_2}| > 2$ **then**

return erreur point non compatible

end

if $R_2 = 0$ **then**

return $-S_2 + 1, -S_2 + 1$

else

return $R_2 - 1, R_2 - 1$

end

end

if $Index_{coin} = 1$ or $Index_{coin} = 14$ **then**

return $R_2 - 1, -S_2 + 1$

end

if $Index_{coin} = 2$ or $Index_{coin} = 13$ **then**

return $R_2 - 1, -S_2$

end

if $Index_{coin} = 4$ or $Index_{coin} = 11$ **then**

return $R_2, -S_2$

end

if $Index_{coin} = 8$ or $Index_{coin} = 7$ **then**

return $R_2 - 1, -S_2 + 1$

end

if $Index_{coin} = 3$ or $Index_{coin} = 12$ **then**

return $-R_2 + 1, -S_2$

end

if $Index_{coin} = 6$ or $Index_{coin} = 9$ **then**

return $R_2 - 1, R_2$

end

if $Index_{coin} = 15$ **then**

return erreur point non compatible

end

Function Trouver la paire d'angles charnières

la première paire d'angles charnières en appelant la fonction de la Figure 5 sur les deux paires de points (\vec{p}_i, \vec{q}_i) pour $i = 1, 2$. Cette paire d'angles charnières sert de référence pour la suite de l'algorithme. La seconde partie boucle sur l'ensemble des paires de points afin de trouver de nouveaux angles charnières qui améliorent l'encadrement de γ . Pour toutes les paires de points, on appelle la fonction de la Figure 5 et on compare le résultat avec la référence afin de la modifier si nécessaire.

La complexité en temps de cet algorithme est $O(m)$. Comme expliqué dans la sous-section précédente, la fonction de la Figure 5 est calculé en un temps $O(1)$. De plus, comme expliqué dans la Section 3, nous pouvons comparer deux angles charnières en un temps constant $O(1)$. Nous pouvons en conclure que l'ensemble des calculs de cet algorithme pour m points prennent un temps $m \times (O(1) + O(1)) = O(m)$.

Data: Ensemble de points
 $A = \{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_n\}, B = \{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_n\}$

Result: une paire d'angle charnière α_1, α_2

Var angle charnière $\alpha_1, \alpha_2, Temps\alpha_1, Temps\alpha_2$;

$(\alpha_1, \alpha_2) =$ Trouver l'encadrement de l'angle
 $(\vec{p}_1, \vec{p}_2, \vec{q}_1, \vec{q}_2)$;

for $i = 3; i < n, i++$ **do**
 $(Temp\alpha_1, Temp\alpha_2) =$ Trouver l'encadrement de
 l'angle $(\vec{p}_1, \vec{p}_i, \vec{q}_1, \vec{q}_i)$;
if $Temp\alpha_1 > \alpha_1$ **then**
 | $\alpha_1 = Temp\alpha_1$;
end
if $Temp\alpha_2 < \alpha_2$ **then**
 | $\alpha_2 = Temp\alpha_2$;
end
if $\alpha_2 < \alpha_1$ **then**
 | Retourner Erreur
end
end
 Retourner (α_1, α_2) ;

Function Calcul des angles charnière in-
 crémentale

5. Conclusion

Dans cet article, nous avons montré comment obtenir un angle charnière directement inférieur à un angle Euclidien donné. Nous avons présenté ensuite un algorithme efficace pour effectuer une rotation discrète à partir de l'angle charnière précédemment trouvé, et ceux avec le même résultat que l'angle Euclidien. Nous avons enfin montrés comment obtenir, à partir d'une paire d'images numériques, le triplet d'entiers qui définit la relation géométrique entre ces deux images.

References

- [And92] ANDRES E.: *Incremental and transitive discretized rotations*. PhD thesis, Université Louis Pasteur Strasbourg, 1992.
- [And96] ANDRES E.: The quasi-shear rotation. 307–314.
- [Ang88] ANGLIN W. S.: Using pythagoricien triangles to approximate angles. *American Mathematical Monthly*, 95 (1988), 540–541.
- [NR06] NOUVEL B., RÉMILA E.: Incremental and transitive discretized rotations. *IWCIA* (2006), 199–213.