

Présentation du projet :

Réalisation d'un encodeur absolu à l'aide d'une webcam

Association VOYAGER 3 Astronomie

- Espace culturel de la Garenne - 44530 Sévérac -

*Association (Loi 1901) pour l'initiation et la pratique de l'astronomie,
Agrément Jeunesse & Education Populaire N°44-08-12*

Commission technique de VOYAGER 3 Astronomie
Sébastien POIRIER

SOMMAIRE : Guide de conception

Algorithmique

I) Calcul préliminaire de la résolution théorique

II) Codage de l'information

- 2.1) Choix du codage
- 2.2) Principe de lecture du code par la webcam
- 2.3) Conversion en valeur numérique
- 2.4) Algorithme de génération du code Gray
- 2.5) Géométrie du fichier d'impression

III) Distorsions dues à l'optique de la webcam

- 3.1) Présentation
- 3.2) Principe de la correction
- 3.3) Principe d'amélioration de l'image à traiter
- 3.4) Algorithme de détection d'un point
- 3.5) Algorithme de détection de la grille
- 3.6) Algorithme de correction des distorsions

IV) Lecture du code Gray par la webcam

- 4.1) Principe d'amélioration de l'image à traiter et optimisation pour implémentation
- 4.2) Détection des lignes guides
- 4.3) Détermination de l'alignement des deux parties de l'image
- 4.4) Inclinaison globale du code
- 4.5) Détermination de l'échelle du code
- 4.6) Bornes de la zone d'information
- 4.7) Fonction de lissage
- 4.8) Détection discontinuité bande
- 4.9) Géométrie de bande de données
- 4.10) Détermination des largeurs adaptatives des lignes de code
- 4.11) Bits de poids faible
- 4.12) Bits de poids fort
- 4.13) Moindres carrés sur un ensemble de bits
 - 4.13.1) Calcul préliminaire
 - 4.13.2) Déterminant de la position du code en haute précision
- 4.14) Détermination « haute précision » de la valeur lue
- 4.15) Exemple de lecture

V) Étalonnage du code Gray

- 5.1) Présentation
- 5.2) Principe
- 5.3) Échantillon d'images du code gray
- 5.4) Réalisation des mesures
- 5.5) Sélection des mesures qui seront utilisées pour l'étalonnage
 - 5.5.1) Détermination de la fiabilité des mesures
 - 5.5.2) Détermination des zones de continuité
 - 5.5.3) Sélection des mesures
- 5.6) Étalonnage
 - 5.6.1) Recherche de la première et la dernière mesures au sens chronologique
 - 5.6.2) Détermination de la vitesse de défilement des graduations
 - 5.6.3) Association d'une valeur angulaire à chaque mesure
 - 5.6.4) Ré-échantillonnage
 - 5.6.5) Détermination des intervalles des valeurs de code gray sans contre-partie angulaire
 - 5.6.6) Détermination de l'origine des mesures et mise en forme des données

Mécanique

VI) Effets de la dilatation

- 6.1) Présentation
- 6.2) Contrainte sur la fixation de la bande imprimée
- 6.3) Effet sur la lecture du code

VII) Effet de la rotation

VIII) Support et montage mécaniques

Hardware

IX) Webcams et contrôleur USB

I) Calcul préliminaire de la résolution théorique

L'idée de la réalisation est issue d'une réflexion sur la détermination du positionnement d'un télescope.

La résolution VGA est de 640x480, ce qui correspond à une valeur de base pour les webcams à l'époque de l'initiation de ce projet. Pour la suite on considérera cette résolution. La plage de réglage de la netteté de l'image des webcams utilisées permet d'observer clairement un objet de l'ordre de 9 mm de largeur

$$\text{Or } \frac{9}{640} \text{ vaut } 0,014 \text{ à } 10^{-3} \text{ près.}$$

Ainsi, la résolution spatiale de la webcam dans ce cas est de 14 μm par pixel.

Afin de réaliser l'encodeur, la webcam sera positionnée de façon à observer des graduations sur un disque. Notre objectif étant de réaliser un encodeur avec le minimum de frais, l'impression sera effectuée sur une simple imprimante.

Compte tenu des dimensions du télescope, l'utilisation de 4 bandes de feuille A4 (297mm de longueur) est retenue.

$$\frac{4 \times 297}{3,14} \text{ vaut } 378 \text{ à l'unité près}$$

Le disque d'encodeur peut donc avoir un diamètre de 378 mm.

Nous utiliserons la valeur de 360 mm afin de tenir compte des marges d'impression. Ainsi le périmètre du disque sera de 1131 mm.

Par ailleurs,

$$\frac{1131}{0,014} \text{ vaut } 80785 \text{ à l'unité près par défaut}$$

$$\text{or } 2^{16} < 80785 < 2^{17}$$

La résolution angulaire théorique de l'encodeur absolu sera donc de 16 bits

Un encodeur absolu de 16 bits pour le prix d'une webcam présente un coût dérisoire par rapport à ce qui est proposé sur le marché.

Naturellement, ce calcul est préliminaire Quelle résolution pouvons nous réellement atteindre avec cette méthode ?

II) Codage de l'information

2.1) Choix du codage

Dans le cadre des encodeurs absolus du commerce, le choix se porte naturellement sur un codage en binaire réfléchi – aussi nommé *code Gray* -. Ce codage permet d'éviter les erreurs de lecture. En effet, ce code est construit de telle façon à ce qu'un seul bit change d'état (0 ou 1) entre deux valeurs.

Nombre en base 10	Nombre en binaire				Nombre binaire réfléchi			
	B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Codage binaire			
B3	B2	B1	B0
0	0	0	0
1	0	0	1
2	0	0	1
3	0	0	1
4	0	1	1
5	0	1	1
6	0	1	1
7	0	1	1
8	1	1	1
9	1	1	1
10	1	1	1
11	1	1	1
12	1	1	1
13	1	1	1
14	1	1	1
15	1	1	1

Codage Gray			
G3	G2	G1	G0
0	0	0	0
1	0	0	1
2	0	0	1
3	0	0	1
4	0	1	1
5	0	1	1
6	0	1	0
7	0	1	0
8	1	1	0
9	1	1	0
10	1	1	1
11	1	1	1
12	1	0	1
13	1	0	1
14	1	0	0
15	1	0	0

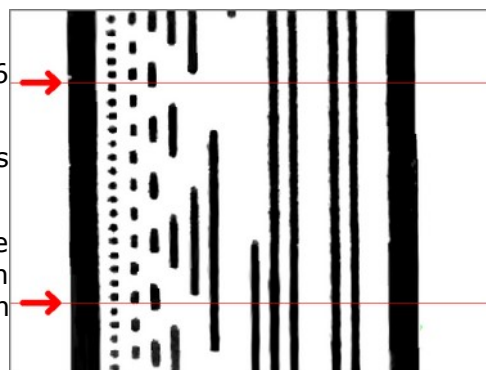
2.2) Principe de lecture du code par la webcam

La webcam est orientée de telle façon à ce que le code Gray apparaisse comme sur l'image à droite.

Le code Gray est lu sur deux lignes de la webcam (ligne 96 et 384 soit un espacement de 288 pixel en verticale).

Les deux valeurs obtenues permettent d'éliminer les éventuelles erreurs de lecture.

Nous verrons par la suite que la double lecture présente d'autres avantages. Il faudra toutefois préciser ce que l'on entend par « lire suivant les lignes n°96 et n°384 » pour un code incliné par rapport à la webcam.



La webcam est placée suffisamment près de la bande pour obtenir une bonne résolution spatiale. Nous verrons par la suite qu'un compromis raisonnable se situe telle que la largeur du code Gray occupe environ 70% de la largeur de la webcam. L'image sera en niveau de gris.

A cette échelle un rectangle du bit de poids faible doit être suffisamment grand pour garantir une bonne lecture. Une dimension raisonnable est choisie de l'ordre de 4 pixels de largeur suivant la perpendiculaire aux lignes de lecture.

Le code imprimé sera donc un code Gray 14 bits.

Finalement, le principe de lecture introduira deux lignes de lecture secondaires afin de réaliser un détrompeur efficace lors des erreurs de lecture.

2.3) Conversion en valeur numérique

$(G_i)_{0 \leq i < n}$ est une suite de 0 et 1 telle que G_0 est le bit de poids faible

$G_0 G_1 G_2 G_3 \dots G_{n-1}$ forme un nombre en binaire réfléchi

On note $B_0 B_1 B_2 B_3 \dots B_{n-1}$ forme le nombre équivalent en binaire

La valeur décimale est donnée par la variable v et vérifiant:

$$\begin{cases} B_{n-1} = G_{n-1} \\ B_i = G_i \oplus B_{i+1} \text{ pour } 0 \leq i \leq n-2 \\ v = \sum_{i=0}^{n-1} B_i 2^i \end{cases}$$

La procédure en C est donnée ci-contre.

```
int GrayCodeToDecimal(int *bit, int n)
{
    int i,v,poids;

    poids=(int)(pow(2,n-1));
    v = bit[n-1]*poids;

    for(i=n-2; i>=0; i--)
    {
        poids /=2;
        bit[i] = (bit[i] + bit[i+1])%2;
        v += bit[i]*poids;
    }

    return(v);
}
```

2.4) Algorithme de génération du code Gray

Le code Gray de l'encodeur sera imprimé. Il nous est donc nécessaire de réaliser un programme créant le fichier à imprimer.

La génération d'un code gray consiste à générer des lignes de bits vérifiant l'algorithme suivant:

On note e_x la largeur d'un bit de poids faible

Pour $0 \leq b < N$ avec N nombre de bit du code

- Pour $n=0$, $x_{0,b} = 0$ $x'_{0,b} = e_x 2^b$

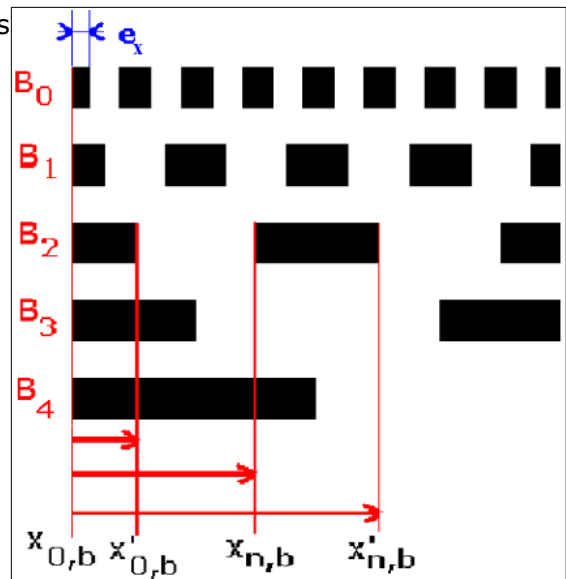
- Si $b < N-2$

Pour $1 \leq n \leq 2^{N-b-3}$, $x_{n,b} = (4n-1) e_x 2^b$
 $x'_{n,b} = (4n+1) e_x 2^b$

Pour $n = 2^{N-b-3} + 1$, $x_{n,b} = (4n-1) e_x 2^b$
 $x'_{n,b} = 4n e_x 2^b$

- Si $b=N-2$

Pour $n = 1$, $x_{n,b} = (4n-1) e_x 2^b$
 $x'_{n,b} = 4n e_x 2^b$



Notations pour le cas $N=5$, $b=3$, $n=1$

2.5) Géométrie du fichier d'impression

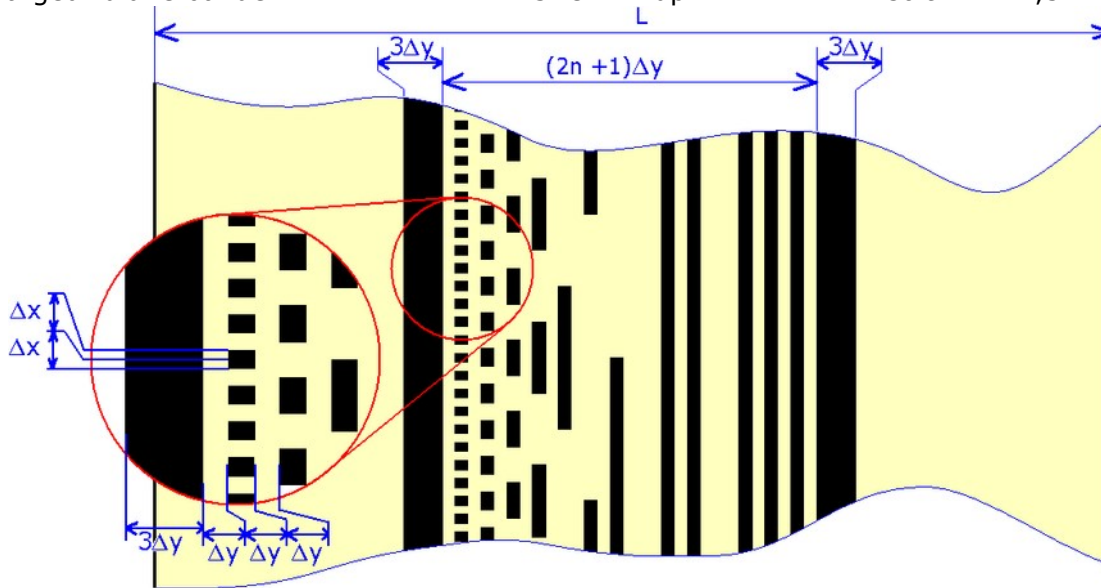
2.5.1) Remarque générale sur l'impression du code

Le programme génère un fichier au format .svg (développé par W3C), présentant l'avantage d'être un format vectoriel.

La résolution de la webcam impliquant une résolution de lecture de l'ordre de $14\mu\text{m}$ cela correspond en résolution d'imprimante à environ 2400 dpi ($25,4/2400=1,05\times 10^{-2}$ mm soit $11\mu\text{m}$). Dans la pratique une impression à 600 dpi est suffisante.

2.5.2) Paramétrage

Nombre de bits de l'encodeur	: $n = 14$	
Longueur du bit faible	: $\Delta x = 0,2$ en 72 dpi	soit $\Delta x \approx 0,07$ mm
Largeur d'un bit	: $\Delta y = 0,58$ en 72 dpi	soit $\Delta y \approx 0,204$ mm
Largeur d'une bande	: $L = 43$ en 72 dpi	soit $L \approx 1,517$ cm



2.5.3) Élargissement du code à l'impression

Lors de l'impression, les bandes noires ont tendance à s'élargir en empiétant sur l'espace des bandes blanches. On définit le facteur d'élargissement relatif α par la relation:

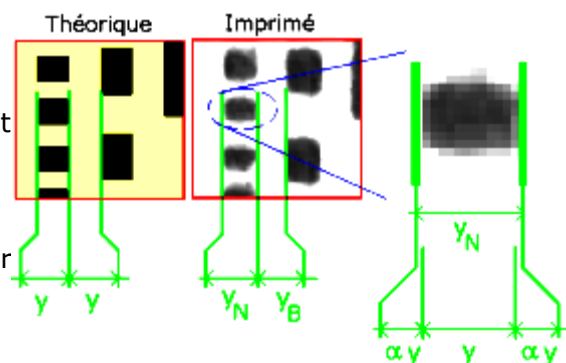
$$\alpha = \frac{1}{2} \frac{y_N - y}{y} \quad \text{ainsi} \quad y_N = y(1 + 2\alpha)$$

De plus, la géométrie du code n'étant globalement pas modifiée nous avons : $y_B + y_N = y + y$

Ainsi, $y_B = y(1 - 2\alpha)$

De cette façon, il suffit de mesurer $y_B + y_N$ pour trouver la valeur de α . Il suffit d'utiliser la relation:

$$\alpha = \frac{1}{2} \frac{y_N - y_B}{y_N + y_B}$$



E: 0.070 (0-4095)	
E: 0.070 (4096-8191)	
E: 0.070 (8192-12287)	
E: 0.070 (12288-16383)	

III) Distorsions dues à l'optique de la webcam

3.1) Présentation

Les webcams possèdent une optique permettant de focaliser l'image sur le plan du capteur photosensible.

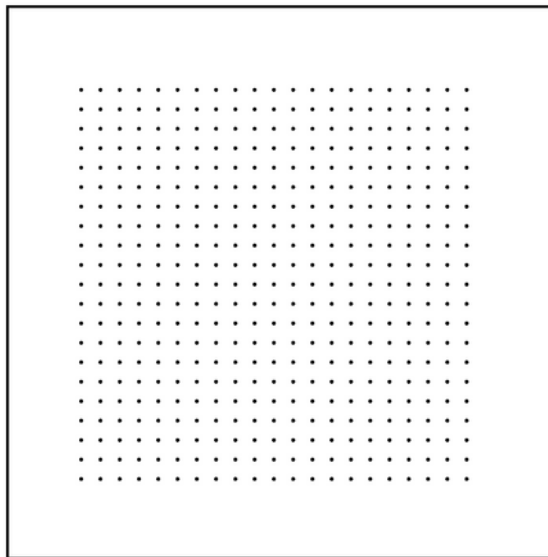
Toute optique présente l'inconvénient de déformer les images, et ceci d'autant plus que l'on s'éloigne de l'axe optique du système.

Afin de réaliser des mesures précises, nous devons nous affranchir le plus possible de ces distorsions du champ. L'étalonnage des graduations du code Gray peut permettre une telle correction. Toutefois la méthode d'étalonnage que nous présenterons par la suite repose sur l'alignement d'images présentant peu ou pas de distorsion.

Ainsi, il faut s'affranchir un maximum des distorsions avant l'étalonnage.

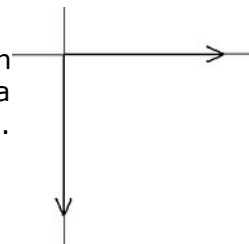
La méthode retenue est l'utilisation d'une mire de calibration. La méthode est classique, mais une mire de calibration permettant une correction optimale est coûteuse. Dans notre cas, la correction des distorsions n'a pas besoin d'être optimale, mais juste raisonnable.

Notre mire de calibration sera donc une simple grille imprimée à l'aide d'une imprimante. Les tests réalisés montrent que cela est suffisant.



La grille est générée par un programme en créant un fichier .svg .
La grille est constituée de 21x21 points placés sur un carré de 1,5 cm de côté.

Pour la suite les coordonnées des points de la grille seront celles d'un repère où l'axe des abscisses est horizontale, orienté de la gauche vers la droite et l'axe des ordonnées est vertical, orienté du haut vers le bas. C'est le choix retenu pour les coordonnées des pixels d'une image.



Le fichier source est : **Generation_Grille.c**

3.2) Principe de la correction

La grille est placée sur le disque portant le code gray.

La webcam est supposée fixe par rapport à ce disque. On suppose que le disque comporte une excentricité négligeable (ce défaut entraînerait que le code Gray soit plus ou moins proche de l'optique de la webcam) introduisant un effet de flou.

La webcam fait l'acquisition d'une image de la grille. Cette grille nous apparaît déformée (image 1). Un algorithme repère automatiquement les points de la grille (image 2). Un second algorithme mémorise les défauts et les corrige (image 3).

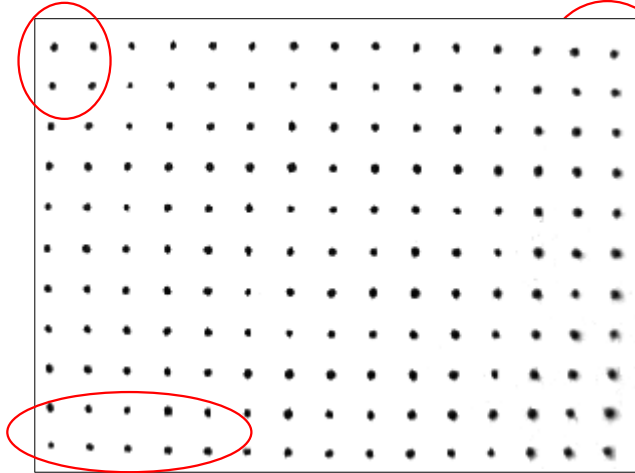


Image 1: Image avec distorsions

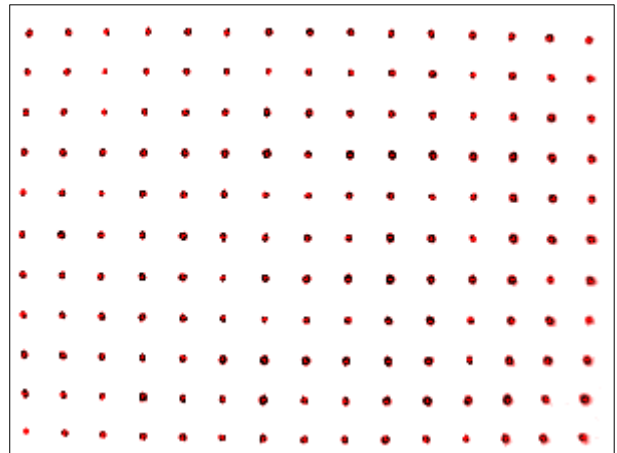


Image 2: Repérage de la grille

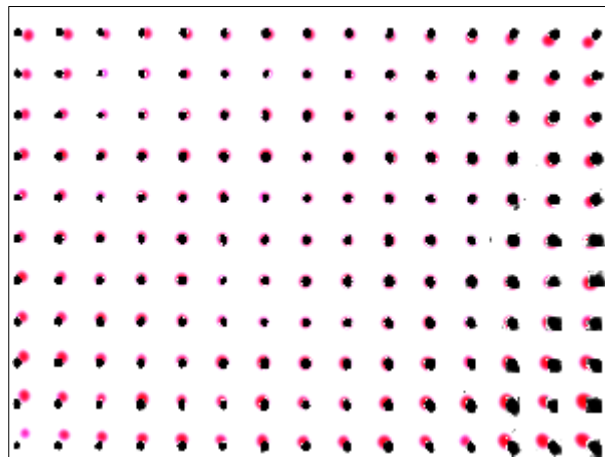


Image 3: Image corrigée (en noir) par rapport à l'image d'origine (en rouge)

3.3) Principe d'amélioration de l'image à traiter

L'éclairage du champ de vue de la webcam n'est pas uniforme. L'objectif est de corriger le gradient de lumière avec des niveaux de référence noirs et de blancs clairement définis sur l'ensemble de l'image.

Ainsi, à partir d'une image à traiter nous allons appliquer le traitement classique suivant :

$$I_c(i; j) = \frac{I_r(i; j) - N(i; j)}{B(i; j) - N(i; j)}$$

$I_c(i; j)$ est la valeur du pixel de coordonnée $(i; j)$ une fois la correction effectuée.

$I_r(i; j)$ est la valeur du pixel de coordonnée $(i; j)$ de l'image réelle initiale.

$N(i; j)$ est la valeur du pixel de coordonnée $(i; j)$ de l'image uniformément noire.

$B(i; j)$ est la valeur du pixel de coordonnée $(i; j)$ de l'image uniformément blanche.

Toutefois la sensibilité des webcams est variable en fonction de la luminosité qu'elles perçoivent. Ainsi, pour une couleur noire donnée, le niveau noir sur une image uniformément noire, n'est pas le même que sur une image comportant du blanc et du noir comme pour une image comportant le codage Grey. On négligera pour la suite la variabilité du niveau en fonction de l'image de code Grey qui pour autant n'est pas constant.

Ainsi $B(i; j)$ et $N(i; j)$ seront pondérés en fonction de la valeur médiane des pixels $(i; j)$ d'une image « code Grey » au niveau d'un ensemble de pixel respectivement noir (médiane noire notée M_N) et blanc (médiane blanche notée M_B).

On définit ainsi : $\alpha_N = \frac{255}{M_N}$ et $\alpha_B = \frac{255}{M_B}$. On suppose ici que le niveau blanc est 255.

d'où d'une façon générale :

$$I_c(i; j) = \frac{I_r(i; j) - \alpha_N N_r(i; j)}{\alpha_N B_r(i; j) - \alpha_B N_r(i; j)}$$

Si $\alpha_N B_r(i; j) = \alpha_B N_r(i; j)$ ou si $\frac{I_r(i; j) - \alpha_N N_r(i; j)}{\alpha_N B_r(i; j) - \alpha_B N_r(i; j)} > 255$ alors $I_c(i; j) = 255$

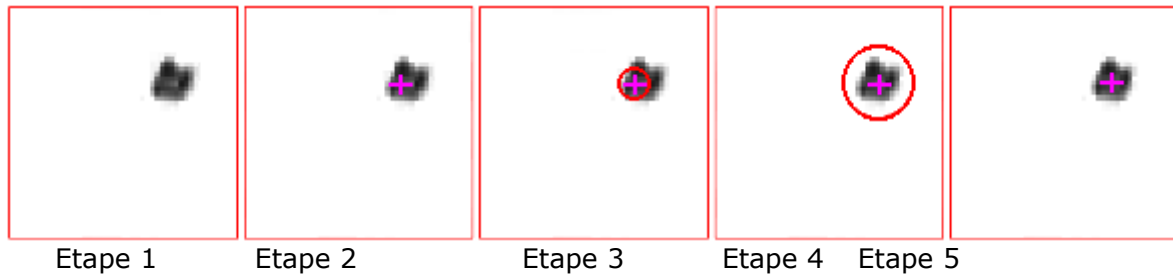
Si $\frac{I_r(i; j) - \alpha_N N_r(i; j)}{\alpha_N B_r(i; j) - \alpha_B N_r(i; j)} < 0$ alors $I_c(i; j) = 0$

Remarque : Pour la détermination de l'image corrigée et des médianes M_N et M_B , nous suivons la méthode suivante :

- A partir de l'image d'une référence correspondant au code grey, on réalise un histogramme. Typiquement, l'image étant noire et blanche, comportant théoriquement 2 niveaux, l'histogramme comporte deux pics : un pour les pixels associés au niveau noir et un pour les pixels associés au niveau blanc. La moyenne sur l'ensemble des pixels va donc se localiser entre les 2 pics. Soit N_s la valeur de ce niveau qui va permettre de séparer les pixels blancs des pixels noirs.
- Sur l'intervalle de niveau $[0 ; N_s[$ on détermine la médiane M_N et sur l'intervalle de niveau $[N_s ; 255]$ on détermine la médiane M_B .
- On détermine les valeurs des pixels de l'image corrigée I_c en suivant la méthode présentée dans ce paragraphe.
- On réitère 2 fois le procédé en considérant l'image corrigée comme nouvelle image de référence.
- Finalement l'image obtenue correspond à l'image qui sera utilisée pour le paragraphe suivant

3.4) Algorithme de détection d'un point

La méthode de détection d'un point peut se décrire suivant 5 étapes:



Préalablement à la détection des points, on réalise un traitement de l'image en déterminant une image moyenne en subdivisant celle-ci en 100 zones. Chaque zone a la couleur correspondant à la moyenne des couleurs des pixels qu'elle contient. On soustrait l'image de la webcam par cette image moyenne, permettant ainsi de retirer le « continu ». Nous divisons l'image obtenu de nouveau par l'image moyenne afin de compenser les effets de variations de lumière (gradient, ombre...). Finalement, les niveaux de l'image ainsi obtenus sont ajustés afin de couvrir la plus grande dynamique.

Pour la détection à proprement parler, on se donne une zone de l'image carrée dont les côtés mesurent $2R_{\max}$

On estime la position du point en déterminant les moyennes des coordonnées des points de la zone image en pondérant en fonction de la couleur des pixels. Pour la fonction de pondération on considère le cube de la valeur du pixel. L'expérience montre que ce choix donne de bons résultats.

$$\begin{cases} \bar{x} = P_T^{-1} \sum_i \sum_j P_{ij}^3 x_{ij} \\ \bar{y} = P_T^{-1} \sum_i \sum_j P_{ij}^3 y_{ij} \end{cases} \quad \text{avec } P_T = \sum_i \sum_j P_{ij}^3$$

Finalement, les coordonnées retenues pour le point de la grille correspondent à $(E(\bar{x}+0,5); E(\bar{y}+0,5))$, E étant la fonction partie entière.

3.5) Algorithme de détection de la grille

3.5.1) Centre de la grille

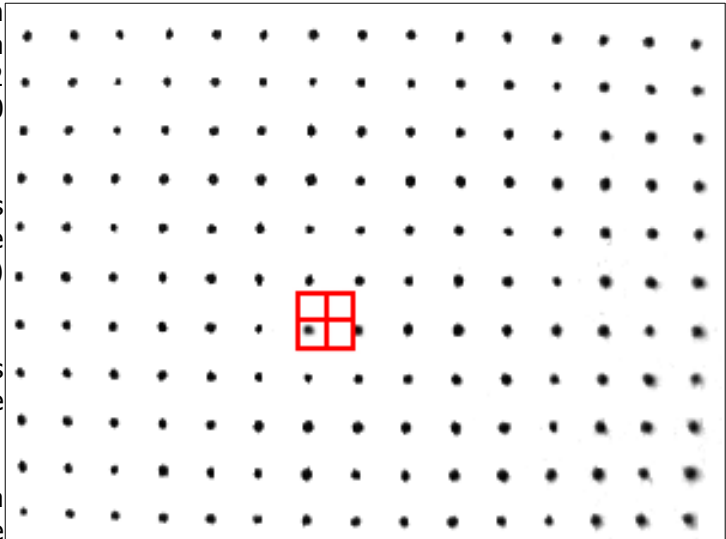
Dans un premier temps on repère un point à proximité du centre de l'image cette recherche se fait en aveugle.

Compte tenu du réglage de la résolution spatiale (voir §2.2) et de l'échelle de la grille (voir §3.1), la distance entre 2 points de la grille est de l'ordre de 50 pixels.

On recherche donc le point le plus proche du centre de l'image (coordonnées du centre (320;240)) dans 4 carrés de 25 pixels par 25 pixels.

L'algorithme de détection des points présenté dans le paragraphe §3.3 donne des coordonnées pour chaque carrés.

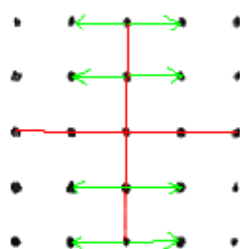
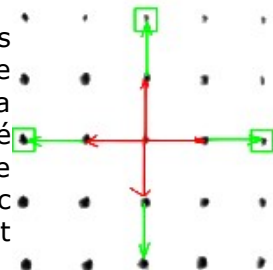
Le point recherché correspond à la solution ayant le plus grand contraste parmi les 4 obtenus.



3.5.2) Détection de l'ensemble de la grille

On suppose qu'au moins un point a été détecté, ne serait-ce que celui correspondant au centre de la grille du paragraphe précédent.

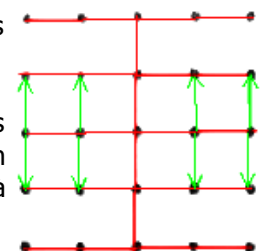
A partir de ce point initial, nous allons rechercher l'ensemble des points situés suivant un axe horizontal et un axe vertical de la grille. Le vecteur servant à déterminer la position du point central de la prochaine zone de recherche a, par défaut, 25 pixels et est orienté suivant les axes naturels de l'image (horizontal ou vertical). Sa norme et sa direction évoluent suivant la position réelle du point détecté avec pour hypothèse que la grille est peut déformée localement, ce qui est le cas pour notre problème.

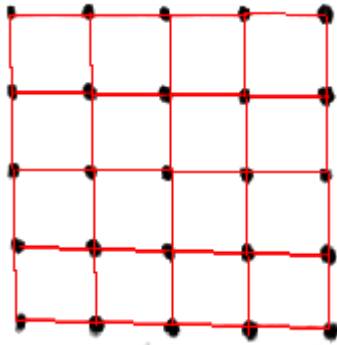


Une fois que l'ensemble des points de la grille de l'image se trouvent sur ce premier axe vertical, une détection des points des axes horizontaux est lancée à partir de chaque point de ce premier axe vertical.

Finalement, nous suivons la même procédure pour les points des axes

Pour chaque point détecté, nous mémorisons ses coordonnées dans l'image (x,y) ainsi que ses coordonnées au sein de la grille (i,j) , en prenant comme point d'origine (coordonnées (0;0) dans la grille) le point à proximité du centre de l'image.





Cette procédure conduit à potentiellement détecter un point deux fois. Elle permet aussi d'éviter de manquer un point de la grille dans le cas d'une grille inclinée ou déformée, où un axe apparaît partiellement ou de façon discontinue.

Une procédure d'identification croisée (*Cross identification*) permet de supprimer un point redondant. En effet chaque point détecté peut avoir des coordonnées sur l'image différentes.

Toutefois, si deux points P_1 et P_2 ont des coordonnées identiques (i,j) sur la grille, il s'agit en réalité d'un seul et unique point. La procédure d'identification croisée les associe et attribue à ce point effectif la moyenne des deux coordonnées (x_1, y_1) et (x_2, y_2) pondérées par les poids $P_{T,1}$ et $P_{T,2}$ (voir §3.3) associée à chacun des points.

Le point effectif a donc pour coordonnées sur la grille: (i,j)
et pour coordonnées sur l'image (x_e, y_e) donc par :

$$x_e = \frac{P_{T,1}x_1 + P_{T,2}x_2}{P_{T,1} + P_{T,2}} \quad y_e = \frac{P_{T,1}y_1 + P_{T,2}y_2}{P_{T,1} + P_{T,2}}$$

Finalement l'ensemble des points détectés est sauvegardé dans un tableau où l'ordre des points est ordonné de la façon suivante:

- Les premiers points de la liste sont ceux ayant la plus petite abscisse sur la grille.
- L'ordre de ces points est celui de l'ordre croissant suivant les ordonnées sur la grille.
- On itère jusqu'à ce que l'ensemble des points détectés soit classé.

3.6) Algorithme de correction des distorsions

3.6.1) Présentation

La procédure de détection de la grille (§3.4) permet d'associer deux types de coordonnées à chaque point observé sur la webcam :

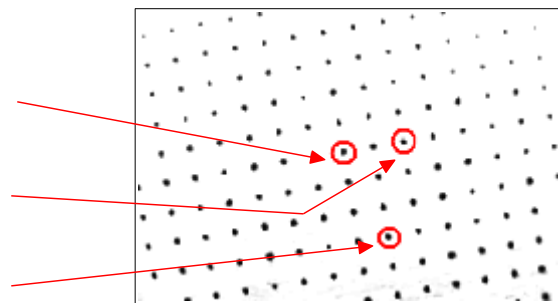
- Des coordonnées correspondent à la position du point sur l'image
- Des coordonnées correspondent à la position du point sur la grille (voir §3.1 et §3.4.2 pour le choix du repère)

Exemples :

Coordonnées sur l'image (318; 242)
Coordonnées sur la grille (0,0)

Coordonnées sur l'image (410;229)
Coordonnées sur la grille (2;0)

Coordonnées sur l'image (382;386)
Coordonnées sur la grille (3;1)



Les coordonnées sur l'image sont des coordonnées que l'on peut qualifier de réelles. C'est celles obtenues en observant la grille via la webcam. Les coordonnées sur la grille sont des

coordonnées que l'on peut qualifier d'idéales.
Le principe de la correction s'effectue en 3 phases:

- 1) Détermination de la rotation et du changement de l'image avant et après la correction.
- 2) Rotation et dilatation appliquées aux coordonnées de la grille des points en fonction des valeurs obtenues à l'étape 2.
- 3) Application du modèle de correction à l'ensemble de l'image.

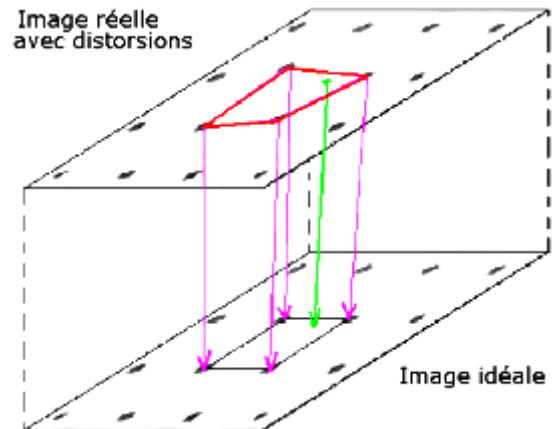
La condition d'arrêt s'effectue lorsque le changement d'échelle et de la rotation de l'image sont négligeables. Dans la majorité des cas, 2 itérations sont suffisantes.

L'association du double jeu de coordonnées pour les pixels correspondant au centre d'un point a été effectuée lors de la détection de la grille.

Sur la figure ci-contre l'association des coordonnées est représentée par les flèches mauves ayant pour origine un des sommets du quadrilatère représenté en rouge sur l'image réelle.

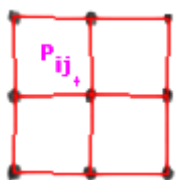
Toutefois, pour les pixels situés à l'intérieur du quadrilatère l'association de coordonnées n'est pas explicitée.

La correction retenue est réalisée par une interpolation bi-linéaire (ou extrapolation le cas échéant pour des pixels des bords de l'image)



L'image réelle est découpée suivant le pavage imposé par le point de la grille. Ainsi chaque pixel de l'image est à l'intérieur d'un quadrilatère ayant pour sommet des points de la grille.

Pour un pixel P_{ij} de l'image réelle donnée, il est nécessaire de connaître le quadrilatère le contenant. Cela revient à trouver les points de la grille les plus proches du pixel P_{ij} .



Afin de tenir compte des différents cas, il faut rechercher les 9 points les plus proches du pixel P_{ij} afin de définir 4 quadrilatères dont un contient nécessairement P_{ij} .

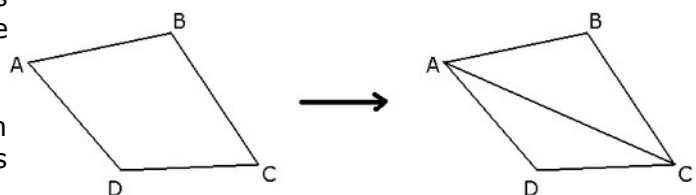
La distance entre les points de la grille et du pixel P_{ij} est calculée à l'aide de la distance euclidienne classique appliquée sur les coordonnées de l'image.

Il faut toutefois créer une procédure permettant de tester si un point se trouve à l'intérieur d'un quadrilatère. Pour simplifier le problème, on ne considère que le cas d'un quadrilatère convexe, ce qui correspond au cas de faibles distorsions comme c'est le cas pour l'optique d'une webcam.

3.6.2) Point à l'intérieur d'un quadrilatère convexe

- Soit M un point donc on connaît les coordonnées $(x_M; y_M)$ dans un repère quelconque.

A, B, C et D sont 4 points définissant un quadrilatère convexe dont on connaît les coordonnées dans le même repère.



Si $ABCD$ est convexe alors la diagonale $[AC]$ est à l'intérieur de ce quadrilatère. Ainsi ACD et ACB , sont deux triangles pavant $ABCD$.

- Si M est un point de la surface de ABCD alors soit M est un point de la surface de ACD, soit M est un point de la surface de A de ACB. Vérifier qu'un point à l'intérieur de ce quadrilatère convexe revient à vérifier si ce point est à l'intérieur de l'un des deux triangles.

- On considère le triangle ABC (Pour le car ADC, il suffira de substituer le point B par D)

Un point du plan peut être défini de façon unique comme étant un barycentre de 3 points avec les données des 3 coefficients de pondération.

Ainsi soit M le barycentre de (A ; α) (B ; β) (C ; γ) avec $\alpha + \beta + \gamma = 1$

Trouver les valeurs de α , β et γ revient à résoudre le système

$$\text{D'où } \begin{cases} \alpha x_A + \beta x_B + \gamma x_C = x_M \\ \alpha y_A + \beta y_B + \gamma y_C = y_M \\ \alpha + \beta + \gamma = 1 \end{cases}$$

Ce système est équivalent à l'égalité matricielle :

$$\begin{pmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x_M \\ y_M \\ 1 \end{pmatrix}$$

Ce système admet donc une solution pour $d \neq 0$, avec $d = (x_B - x_A)(y_C - y_A) - (y_B - y_A)(x_C - x_A)$

Donnée par :

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = d^{-1} \begin{pmatrix} y_B - y_C & x_C - x_B & x_B y_C - x_C y_B \\ y_C - y_A & x_A - x_C & x_C y_A - x_A y_C \\ y_A - y_B & x_B - x_A & x_A y_B - y_A x_B \end{pmatrix} \begin{pmatrix} x_M \\ y_M \\ 1 \end{pmatrix}$$

$$\text{d'où } \begin{cases} \alpha = \frac{1}{d} ((y_B - y_C)x_M + (x_C - x_B)y_M + x_B y_C - x_C y_B) \\ \beta = \frac{1}{d} ((y_C - y_A)x_M + (x_A - x_C)y_M + x_C y_A - x_A y_C) \\ \gamma = \frac{1}{d} ((y_A - y_B)x_M + (x_B - x_A)y_M + x_A y_B - y_A x_B) \end{cases}$$

Ainsi, M est à l'intérieur du triangle ABC si et seulement si: $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ et $0 \leq \gamma \leq 1$

3.6.3) Correction bi-linéaire

Dans cette partie les expressions en gras telles que **AB** définissent des vecteurs.

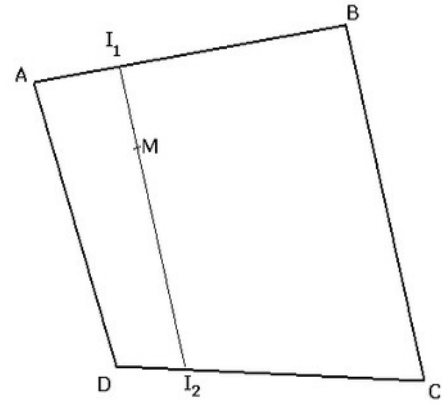
A, B, C et D sont 4 points définissant un quadrilatère convexe dont on connaît les coordonnées dans le repère de l'image.

Soit α et β deux réelles tels que $0 \leq \alpha \leq 1$ et $0 \leq \beta \leq 1$.

On définit le point I_1 par : **$AI_1 = \alpha AB$**

On définit le point I_2 par : **$DI_2 = \alpha DC$**

On définit le point M par : **$I_1M = \beta I_1I_2$**



Remarque: On peut de la même façon définir M en changeant les rôles de **AB** et **AD**. Les images ayant peu de distorsion localement, en d'autre terme ABCD étant proche d'un carré, cela change peu de chose quant à la position du point M ainsi définie. Par ailleurs, une fois le choix défini la position de M est bien déterminé de façon unique.

Ainsi,

$$I_1A + AM = \beta I_1A + \beta AI_2$$

$$AM = (\beta - 1) I_1A + \beta AI_2$$

$$AM = (1 - \beta) \alpha AB + \beta AD + \beta DI_2$$

$$AM = (1 - \beta) \alpha AB + \beta AD + \alpha \beta DC$$

$$AM = \alpha AB + \beta AD + \alpha\beta (DC + BA)$$

Le point M est défini de façon unique par la donnée de $(\alpha ; \beta)$. On peut remarquer que dans le cas d'un carré ces coordonnées correspondent à celles attribuées au point M dans un repère orthonormal.

- Si un point M ($x_M; y_M$) est à l'intérieur d'un quadrilatère ABCD, quelles sont les valeurs de α et β associées à ce point ?

Ce problème consiste à résoudre :

$$AM = \alpha AB + \beta AD + \alpha\beta (DC - AB) \text{ avec } \alpha \text{ et } \beta \text{ pour inconnues}$$

Ce qui équivaut à résoudre :

$$\begin{cases} x_M - x_A = \alpha (x_B - x_A) + \beta (x_D - x_A) + \alpha\beta (x_A - x_B + x_C - x_D) \\ y_M - y_A = \alpha (y_B - y_A) + \beta (y_D - y_A) + \alpha\beta (y_A - y_B + y_C - y_D) \end{cases}$$

On note	$a_{00} = x_A - x_M$	$a_{10} = y_A - y_M$
	$a_{01} = x_B - x_A$	$a_{11} = y_B - y_A$
	$a_{02} = x_D - x_A$	$a_{12} = y_D - y_A$
	$a_{03} = x_A - x_B + x_C - x_D$	$a_{13} = y_A - y_B + y_C - y_D$

Le système devient ainsi :

$$\begin{cases} a_{00} + \alpha a_{01} + \beta a_{02} + \alpha\beta a_{03} = 0 \\ a_{10} + \alpha a_{11} + \beta a_{12} + \alpha\beta a_{13} = 0 \end{cases}$$

$$\begin{cases} a_{00} + \beta a_{02} + \alpha (a_{01} + \beta a_{03}) = 0 \\ a_{10} + \beta a_{12} + \alpha (a_{11} + \beta a_{13}) = 0 \end{cases}$$

d'où $(a_{00} + \beta a_{02})(a_{11} + \beta a_{13}) - (a_{10} + \beta a_{12})(a_{01} + \beta a_{03}) = 0$

$$(a_{02} a_{13} - a_{12} a_{03}) \beta^2 + (a_{00} a_{13} + a_{02} a_{11} - a_{10} a_{03} - a_{12} a_{01}) \beta + a_{00} a_{11} - a_{10} a_{01} = 0$$

On note $A = a_{02} a_{13} - a_{12} a_{03}$ $B = a_{00} a_{13} + a_{02} a_{11} - a_{10} a_{03} - a_{12} a_{01}$ $C = a_{00} a_{11} - a_{10} a_{01}$

d'où $A \beta^2 + B \beta + C = 0$

on note $\Delta = B^2 - 4AC$ Cette équation admet au moins une solution réelle, M étant à l'intérieur du quadrilatère.

donc $\Delta \geq 0$

Si $A \neq 0$	$\beta = \frac{-B - \sqrt{\Delta}}{2A}$	ou	$\beta = \frac{-B + \sqrt{\Delta}}{2A}$
Si $A = 0$	Si $B \neq 0$ alors $\beta = -\frac{C}{B}$		
Si $B = 0$ alors on ne gère pas ce cas puisque M étant à l'intérieur du quadrilatère il existe une unique solution. Ce cas est donc impossible.			

On obtient α de la façon suivante:

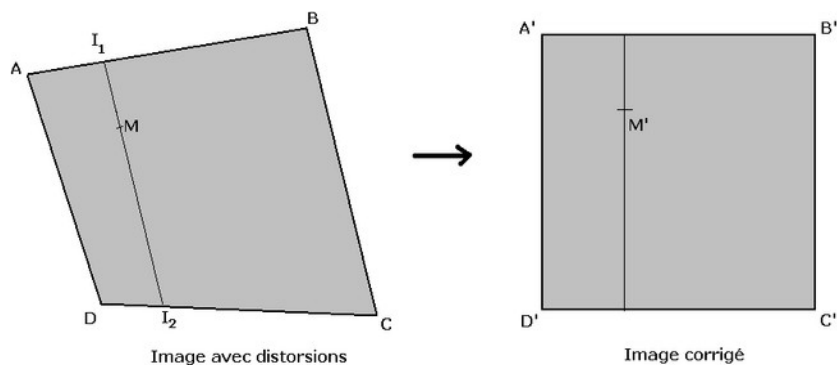
Si $(a_{01} + \beta a_{03}) \neq 0$	alors $\alpha = -\frac{a_{00} + \beta a_{02}}{a_{01} + \beta a_{03}}$
Si $(a_{11} + \beta a_{13}) \neq 0$	alors $\alpha = -\frac{a_{10} + \beta a_{12}}{a_{11} + \beta a_{13}}$
M étant à l'intérieur du quadrilatère il existe une unique solution, α est définie dans l'un des deux cas précédent.	

β admettant deux valeurs possibles on obtient deux couples de solutions vérifiant le système. On note ces couples $(\alpha_1; \beta_1)$ et $(\alpha_2; \beta_2)$. La solution recherchée est telle que $0 \leq \alpha_i \leq 1$ et $0 \leq \beta_i \leq 1$. M étant à l'intérieur du quadrilatère, la solution existe. Soit $i=1$ soit $i=2$.

- Finalement afin de réaliser la correction, on applique

$$x_{M'} = x_{A'} + \alpha (x_{B'} - x_{A'}) + \beta (x_{D'} - x_{A'}) + \alpha\beta \underbrace{(x_{A'} - x_{B'} + x_{C'} - x_{D'})}_0$$

$$y_{M'} = y_{A'} + \alpha (y_{B'} - y_{A'}) + \beta (y_{D'} - y_{A'}) + \alpha\beta \underbrace{(y_{A'} - y_{B'} + y_{C'} - y_{D'})}_0$$

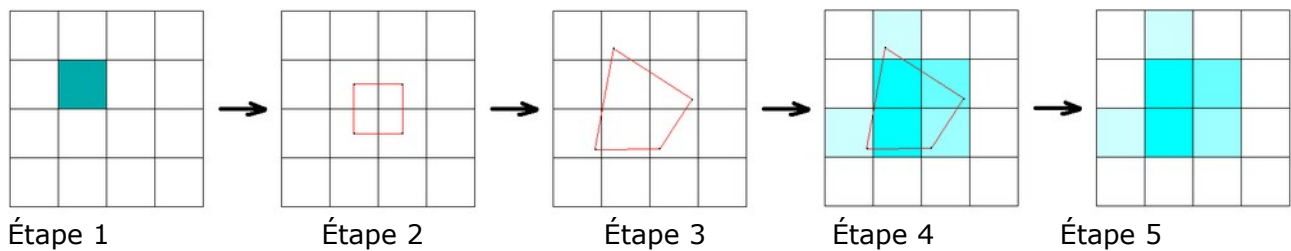


3.6.4) Gestion de la non-bijektivité de la transformation discrétisée

Dans le paragraphe précédent nous avons considéré une correction bilinéaire avec des variables réelles. Compte tenu du pavage des quadrilatères et la transformation employée à l'intérieur de chaque quadrilatère, la transformation considérée est bijective.

Toutefois, une image de webcam est un ensemble discret, les pixels étant les surfaces élémentaires. Un pixel de l'image d'origine ne se transforme pas forcément en un pixel sur l'image corrigée. Ainsi pour le cas discret, la transformation n'est pas bijective.

On peut schématiser ainsi la modification de la transformation afin de tenir compte de la nature discrète des images:



On considère 4 pixels adjacents (1), On applique la transformation du §3.5.3 aux coordonnées de ces pixels (2 et 3). Les coordonnées des 4 points obtenus sont des valeurs à priori réelles (3). Afin de tenir compte de cet effet, on répartit la couleur initiale sur toute la zone obtenue (4). Chaque pixel concerné reçoit une fraction de la couleur initiale (5). La somme des fractions vaut 1. La valeur de la fraction est déterminée à partir des coordonnées réelles du point et non de la valeur de la couleur de ce dernier.

L'effet de la non-bijektivité est minimisé par une répartition des couleurs.

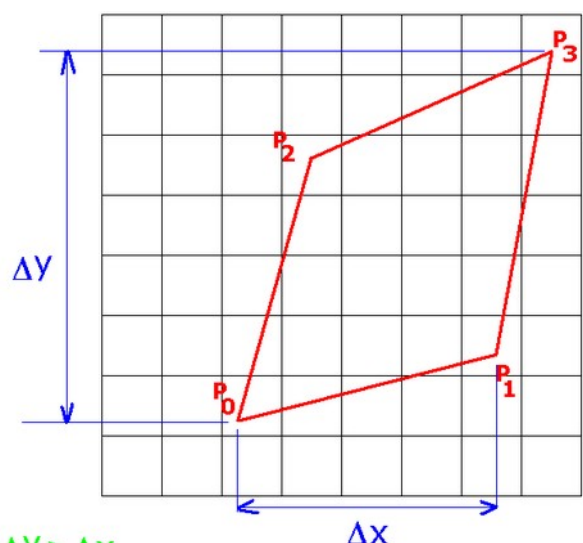
Pour chaque pixel de l'image d'arrivée, on mémorise les coordonnées des pixels qui ont contribué à sa couleur ainsi que les coefficients correspondant à la fraction de couleur apportée par les différents pixels. La fraction ne dépendant pas de la couleur des pixels, le modèle ne dépend que de la distorsion de la webcam. Ainsi, il suffit pour chaque pixel de l'image d'arrivée de mémoriser la liste des pixels de l'image d'origine et les fractions associées.

3.6.5) Calcul de la fraction de couleur à répartir sur chaque pixel

Cette partie n'est utilisée que lorsque l'image d'un pixel est répartie sur plus de 1 pixel, c'est à dire que la différence la plus grande des abscisses des sommets du quadrilatère est supérieure à 2 ou que la différence la plus grande des ordonnées des sommets du quadrilatère est supérieure à 2.

On considère un quadrilatère non convexe. Quitte à réordonner les points et à permuter les rôles des abscisses et des ordonnées, on peut toujours se ramener au cas où:

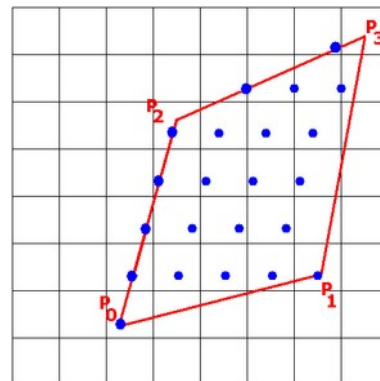
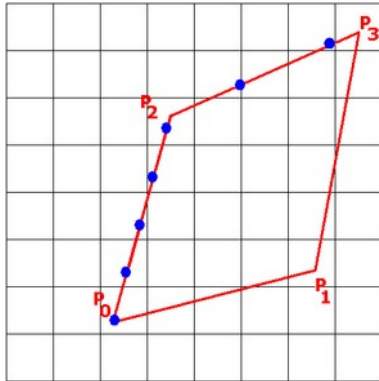
- Le point P_0 est le point de plus petite ordonnée parmi les 4 sommets
- Le point P_3 est le point de plus grande ordonnée parmi les 4 sommets
- Le point P_1 a une abscisse plus petite que P_2
- la différence de la plus grande des abscisses des points est plus petite que la différence la plus grande des ordonnées des points



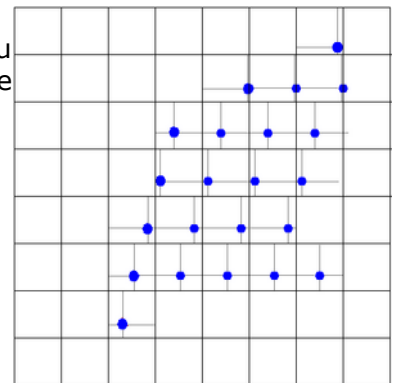
De telles conditions sont vérifiées par la figure ci-contre.

$\Delta Y > \Delta X$

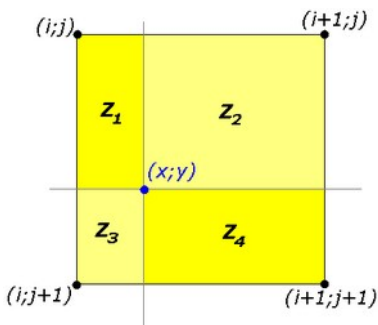
On repère les coordonnées contenues dans le quadrilatère par pas unitaire suivant l'axe des abscisses et des ordonnées en partant de P_0 comme point d'origine. À chaque pas en ordonnées, la première abscisse (valeur la plus petite) est choisie pour que le point associé à cette abscisse se trouve sur un côté du quadrilatère.



Chaque point ainsi définie est un donc un point de la surface du quadrilatère. Chacun d'entre eux définisse 4 parties d'un pixel de l'image, comme représenté sur la figure ci-contre.



Pour chaque point on peut donc définir 4 aires que l'on note Z_1 , Z_2 , Z_3 et Z_4 .



On peut remarquer que $Z_1 + Z_2 + Z_3 + Z_4 = 1$

et $Z_1 = (x-i)(y-j)$ $Z_2 = (i+1-x)(y-j)$

$Z_3 = (x-i)(j+1-y)$ $Z_4 = (i+1-x)(j+1-y)$

On note C_p la couleur du pixel p obtenue par transformation présentée dans la paragraphe §3.5.3 et §3.5.4.

Les fractions de couleur sont prises comme étant Z_1 , Z_2 , Z_3 et Z_4 .

Ainsi la contribution de pixel p à la couleur des pixels $p_{(i,j)}$, $p_{(i+1,j)}$, $p_{(i,j+1)}$ et $p_{(i+1,j+1)}$ est donnée par:

$c_{i,j} = Z_1 C_p$ avec $c_{i,j}$ la contribution à la couleur du pixel $p_{(i,j)}$

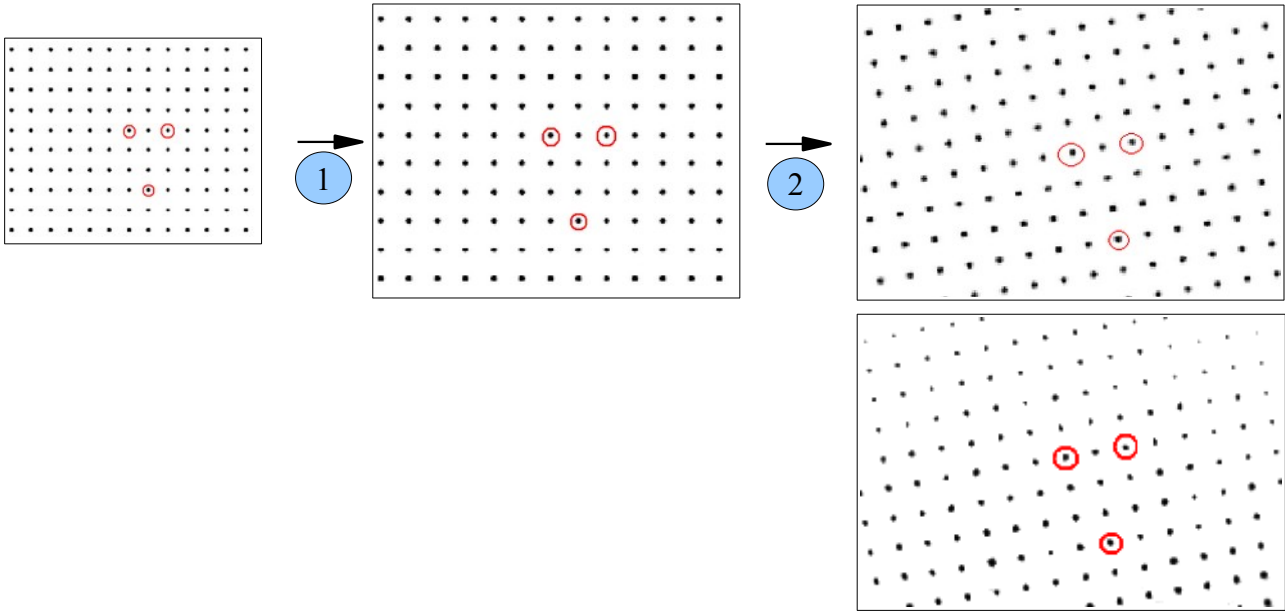
$c_{i+1,j} = Z_2 C_p$ $c_{i+1,j}$ la contribution à la couleur du pixel $p_{(i+1,j)}$

$c_{i,j+1} = Z_3 C_p$ $c_{i,j+1}$ la contribution à la couleur du pixel $p_{(i,j+1)}$

$c_{i+1,j+1} = Z_4 C_p$ $c_{i+1,j+1}$ la contribution à la couleur du pixel $p_{(i+1,j+1)}$

3.6.6) Rotation et dilatation de la grille

Afin de conserver la même échelle (1) et la même orientation du champ de l'image avant et après la correction, les coordonnées de la grille doivent être adaptées.



Afin de déterminer l'échelle et l'angle de rotation, on procède de façon itérative:

- Initialement, on utilise les coordonnées naturelles de la grille (celles associées aux points lors de la détection de la grille (§3.4))
- On a appliqué le modèle de correction au pixel central, P_1 , de l'image de coordonnées $(x_0; y_0)$ et à un autre pixel, P_2 , proche du centre de l'image (par exemple de coordonnées $(x_0+10; y_0)$).

La distance entre les deux points sur l'image d'origine est donc de 10 pixels ($x_0+10-x_0=10$)

On note E le rapport de la distance entre les points pixels sur l'image après la correction et la distance entre les deux pixels avant la correction.

De même, on note θ l'angle entre la direction de la droite passant par les deux points après et avant la correction

Finalement, on note $(x_1; y_1)$ et $(x_2; y_2)$ les coordonnées des points P_1 et P_2 après la correction.

$$\text{Soit } d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\text{Ainsi, } E = \frac{d}{10} \quad \cos \theta = \frac{x_2 - x_1}{E} \quad \text{et} \quad \sin \theta = \frac{y_2 - y_1}{E}$$

On applique la correction suivante aux coordonnées de la grille pour chaque point détecté lors de la procédure de détection de la grille:

On note $(x; y)$ les coordonnées initiales et $(x'; y')$ les coordonnées finales du point.

$$\text{Soit } r_x = x - x_1 \quad \text{et} \quad r_y = y - y_1$$

$$\text{Ainsi, } x' = x_0 + \frac{r_x \cos(\theta) + r_y \sin(\theta)}{E} \quad \text{et} \quad y' = y_0 + \frac{-r_x \sin(\theta) + r_y \cos(\theta)}{E}$$

On réitère 3 fois afin de converger vers une orientation et une échelle correcte de la grille.

Finalement, on applique la procédure d'écrite dans §3.5.3 et §3.5.4 à l'ensemble des points de l'image.

4.2) Détection des lignes guides

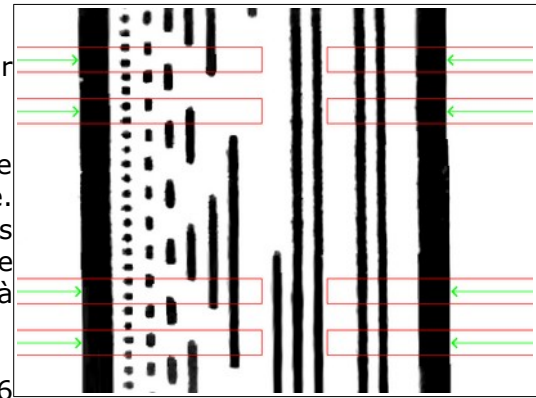
La bande imprimée comporte des bandes plus larges de part et d'autre du code Gray. On les désignera par le terme de lignes guides.

Toute la lecture du code gray repose sur la détection de ces lignes. L'algorithme de détection doit être robuste. Ainsi, nous utilisons une détection mettant en jeu des transformées de Fourier rapide (FFT) permettant de réaliser un filtrage spatial des bruits dus par exemple à la poussière, à la détérioration du code...

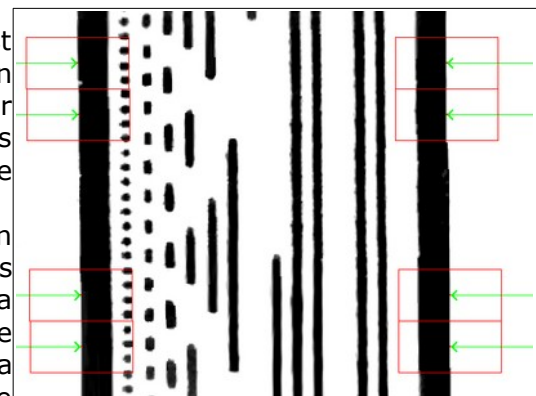
Dans une première étape, on définit 8 cadres de 256 par 32 pixels centrés respectivement sur la ligne 64 et 128, 352 et 416, dont un côté est adjacent au bord de l'image.

Le principe de détection des bords des lignes guides est de partir du bord de ligne jusqu'au moment où l'on observe une transition d'une couleur claire à une couleur sombre. Les trajectoires suivies sont repérées par les flèches bleues. Les pixels correspondant à cette transition sont sur les bords extérieurs.

Lors d'une seconde étape, on réitère la méthode en considérant des cadres de 128 par 64 pixels respectivement centrés sur les pixels trouvés lors de la première étape. Cette seconde étape est nécessaire afin d'éviter de erreurs d'interprétations dues à la création d'artefact due à la non périodicité de l'image lors du filtrage spatial.



Première étape

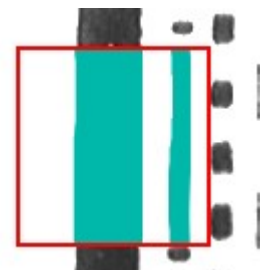
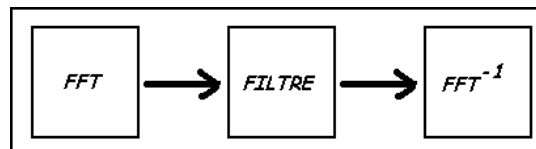
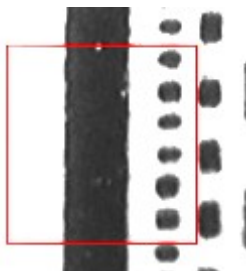


Seconde étape

- Transformées de Fourier et Filtrage du bruit

Afin d'augmenter la qualité de la méthode présentée ci-dessous, nous réalisons un filtrage spatial. Ainsi, la largeur des lignes guides est choisie afin que leurs contours, une fois filtrés, ne sont trop déformés par leur environnement spatial.

Dans un premier temps chaque partie de l'image correspondant aux cadres rouges subit une FFT, on applique ensuite un filtrage dans l'espace des fréquences puis on applique une FFT inverse pour obtenir l'image filtrée.



Le filtre dans l'espace des fréquences a la forme suivante:

Soit r définit par $r = (f_x^2 + f_y^2)^{1/2}$ avec $f_x = i_{\max} - i$ et $f_y = j_{\max} - j$

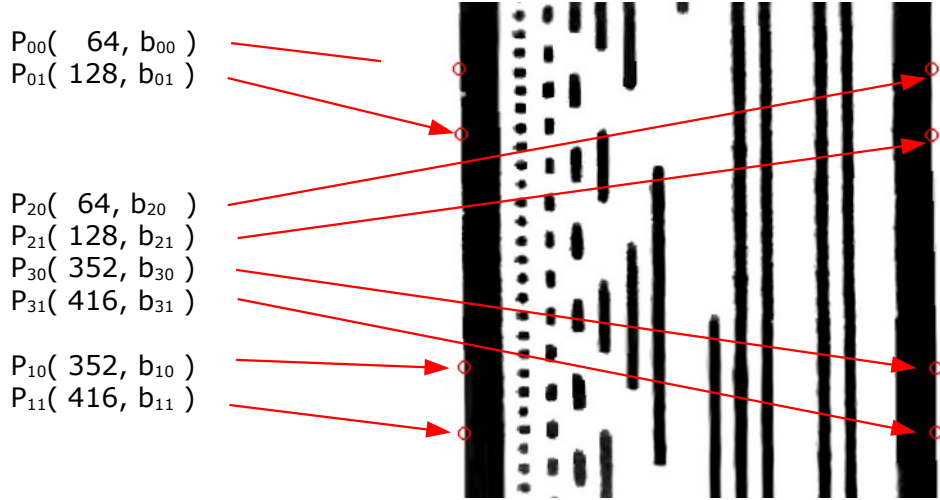
$i_{\max}$ et $j_{\max}$ correspondent aux demies-dimensions de l'image

i et j sont les coordonnées du pixel sur l'image de l'espace des fréquences

Remarque: Le pixel $i=i_{\max}$ et $j = j_{\max}$ correspond à l'amplitude de la valeur moyenne de l'image.

4.3) Détermination de l'alignement des deux parties de l'image

Les 8 points déterminés au paragraphe précédent pour la localisation des lignes guides et représentés ci contre en rouge ont pour coordonnées respectives (abscisses axe vertical, ordonnées axe horizontal):



Il existe nécessairement au moins une discontinuité dans les graduations. Cette discontinuité est constitué par la jonction des deux extrémités de la bande imprimée.

Afin de tenir compte de l'éventuelle présence d'une discontinuité, nous devons analyser l'image avec des bandes guides qui ne sont pas continues entre la partie supérieure de l'image et la partie inférieure.

Afin de déterminer si les bandes guides sont d'un seul tenant ou non, nous vérifions le quasi-alignement d'une part des points P_{00} , P_{01} , P_{10} , P_{11} et d'autre part des points P_{20} , P_{21} , P_{30} , P_{31} .

Les points ne pouvant être alignés horizontalement, nous déterminons une équation réduite de droite par régression linéaire.

La première régression (points à gauche sur l'image ci-dessous)

$$a_1 = \frac{11(b_{11} - b_{00}) + 7(b_{10} - b_{01})}{5440} \quad \text{et} \quad b_1 = \frac{b_{11} + b_{00} + b_{10} + b_{01}}{4} - 240 a_1$$

Si au moins un des 4 points considérés a une ordonnée qui s'écarte de plus de 1,5 pixels par rapport à l'alignement théorique, on considère que les 4 points ne sont pas suffisamment alignés pour considérer la bande comme d'un seul tenant. L'origine de cet non alignement peut venir d'une dégradation locale sur la bande guide ou la présence d'une discontinuité.

$$\begin{aligned} |b_{00} - (a_1 \times 64 + b_1)| &< 1.5 & |b_{01} - (a_1 \times 128 + b_1)| &< 1.5 \\ |b_{10} - (a_1 \times 352 + b_1)| &< 1.5 & |b_{11} - (a_1 \times 416 + b_1)| &< 1.5 \end{aligned}$$

La seconde régression (points à droite sur l'image ci-dessous)

$$a_2 = \frac{11(b_{31} - b_{20}) + 7(b_{30} - b_{21})}{5440} \quad \text{et} \quad b_2 = \frac{b_{31} + b_{20} + b_{30} + b_{21}}{4} - 240 a_2$$

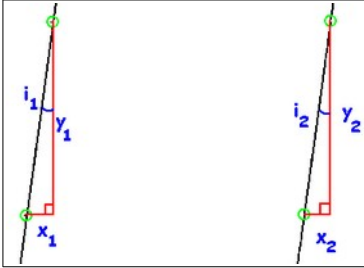
De la même façon, si au moins un des 4 points considérés a une ordonnée qui s'écarte de plus de 1,5 pixels par rapport à l'alignement théorique, on considère que les 4 points ne sont pas suffisamment alignés pour considérer la bande comme d'un seul tenant.

$$\begin{aligned} |b_{20} - (a_2 \times 64 + b_2)| &< 1.5 & |b_{21} - (a_2 \times 128 + b_2)| &< 1.5 \\ |b_{30} - (a_2 \times 352 + b_2)| &< 1.5 & |b_{31} - (a_2 \times 416 + b_2)| &< 1.5 \end{aligned}$$

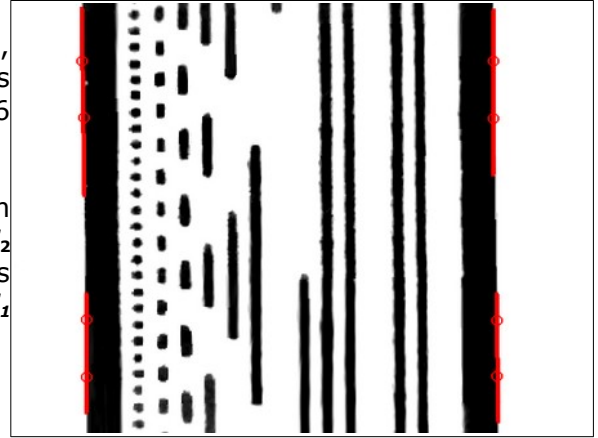
4.4) Inclinaison globale du code

En utilisant la méthode du paragraphe précédent, on connaît la position des bords extérieurs des lignes guide sur la ligne 64 et 128, 352 et 416 (cercles sur l'image de droite).

L'inclinaison i de la bande par rapport à la webcam est donnée par la moyenne des inclinaisons i_1 et i_2 des bords des lignes guides. Si la correction des distorsions optiques est correcte, les valeurs de i_1 et i_2 sont proches:



$$\tan i_1 = \frac{x_1}{y_1} \quad \text{et} \quad \tan i_2 = \frac{x_2}{y_2} \quad \text{On pose : } i = \frac{i_1 + i_2}{2}$$



On détermine l'inclinaison du code de l'image par :

Pour la zone du haut de l'image(points $P_{00}(64, b_{00})$, $P_{01}(128, b_{01})$, $P_{20}(64, b_{20})$ et $P_{21}(128, b_{21})$)

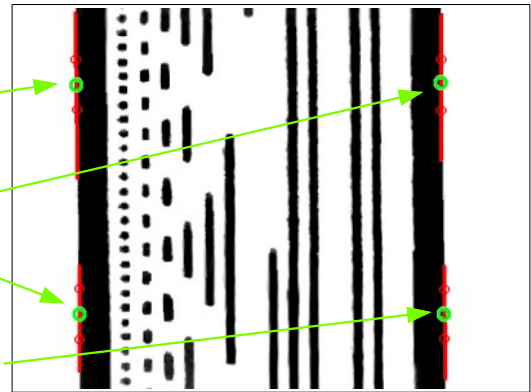
$$i_{haut} = \frac{1}{2} \left(\operatorname{atan} \left(\frac{b_{01} - b_{00}}{64} \right) + \operatorname{atan} \left(\frac{b_{21} - b_{20}}{64} \right) \right)$$

Pour la zone du bas de l'image(points $P_{00}(352, b_{10})$, $P_{11}(352, b_{01})$, $P_{20}(416, b_{30})$ et $P_{21}(416, b_{31})$)

$$i_{bas} = \frac{1}{2} \left(\operatorname{atan} \left(\frac{b_{11} - b_{10}}{64} \right) + \operatorname{atan} \left(\frac{b_{31} - b_{30}}{64} \right) \right)$$

On considère les points :

- D_0 de coordonnées $(96, \frac{b_{00} + b_{01}}{2})$
- D_2 de coordonnées $(96, \frac{b_{20} + b_{21}}{2})$
- D_1 de coordonnées $(384, \frac{b_{10} + b_{11}}{2})$
- D_3 de coordonnées $(384, \frac{b_{30} + b_{31}}{2})$



on pose : $b'_{00} = \frac{b_{00} + b_{01}}{2}$ $b'_{10} = \frac{b_{10} + b_{11}}{2}$ $b'_{20} = \frac{b_{20} + b_{21}}{2}$ $b'_{30} = \frac{b_{30} + b_{31}}{2}$

Si la bande est considérée comme étant d'un seul tenant selon la procédure du paragraphe §4.3, l'inclinaison globale du code est déterminée par :

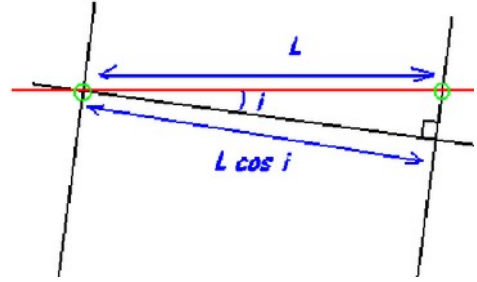
$$i_{global} = i_{haut} = i_{bas} = \frac{1}{2} \left(\operatorname{atan} \left(\frac{b'_{10} - b'_{00}}{64} \right) + \operatorname{atan} \left(\frac{b'_{30} - b'_{20}}{64} \right) \right)$$

4.5) Détermination de l'échelle du code

En utilisant les points B_0 , B_1 , B_2 , et B_3 du paragraphe précédent, on peut déterminer la distance suivant l'axe horizontal des points des lignes guides. On note L cette distance.

Pour la partie du haut : $L = B_1B_0$

Pour la partie du bas : $L = B_3B_2$



La distance L' entre les bords extérieurs des lignes guide est donc donnée par :

$$L' = L \cos i$$

La longueur L' est suivant l'axe perpendiculaire à celui du code gray

On a $L' = \kappa' e_y$ e_y étant la largeur d'un bit de poids faible

$$\text{donc } e_y = \frac{L \cos(i)}{\kappa'}$$

κ est un coefficient de proportionnalité imposé par la géométrie de la bande lors de l'impression de la bande codée. Le fichier d'impression impose aussi le rapport R :

$$R = \frac{e_y}{e_x}$$

On obtient

ainsi e_x

par :

$$e_x = \frac{L \cos(i)}{R \kappa'}$$

Compte tenu du §2.5,

$\kappa =$

n

+

$2 + D$

+

$n+1$

Nombre ligne
de bit

Nombre ligne
guide plus fraction
d'élargissement de
lignes guides

Nombre
d'interlignes

Ainsi, $\kappa = \kappa' + D$

avec $\kappa' = 2(n+2) - 1$

À noter que D se décompose de la façon suivante: $D = D_g + D_d$

avec

$$D_g = T_g + E_g + O_G$$

$$D_d = T_g + E_g - O_D$$

Ligne guide de
gauche

Ligne guide de
droite

Le coefficient T_g provient du fait qu'une ligne guide est 3 fois plus large qu'une ligne de bit, ainsi: $T_g = 3 - 1$. E_g est l'élargissement lors de l'impression de la ligne guide en fraction de largeur d'une ligne de bit. O_G et O_D sont des offsets (pouvant être nuls) des lignes guides dus à l'impression.

Pour la partie haute de l'image ($k=0$) on note l'échelle e_{x_0}

Pour la partie basse de l'image ($k=1$) on note l'échelle e_{x_1}

Remarque : Si la bande est considérée comme dans le paragraphe §4.4, l'échelle globale du code est déterminée par :

$$e = e_{x_0} + e_{x_1} \text{ ce qui revient à définir une échelle moyenne sur l'ensemble de l'image}$$

Dans le cas contraire on pose $e = 2 e_x$

4.6) Bornes de la zone d'information

On définit les points

$$\begin{aligned} B_0 &= D_0 \\ B_1 &= D_1 \\ B_4 &= D_2 \\ B_5 &= D_3 \end{aligned}$$

On pose ainsi les ordonnées des points

$$\begin{aligned} b''_0 &= b'_{00} \\ b''_1 &= b'_{10} \\ b''_4 &= b'_{20} \\ b''_5 &= b'_{30} \end{aligned}$$

$$\begin{aligned} b''_2 &= b''_0 - 32 \tan(i_0) \\ b''_6 &= b''_4 - 32 \tan(i_0) \end{aligned}$$

$$\begin{aligned} b''_3 &= b''_1 + 32 \tan(i_1) \\ b''_7 &= b''_5 + 32 \tan(i_1) \end{aligned}$$

Pour $k \in [[0;3]]$, on définit

$$l_k = \frac{(b''_{4+k} - b''_k) \kappa'}{\kappa + D}$$

on rappelle que $\kappa = \kappa' + D$ et $\kappa' = 2(n+2) - 1$

$$r = \frac{l_k}{\kappa'}$$

r correspond à la largeur d'un bit de donnée

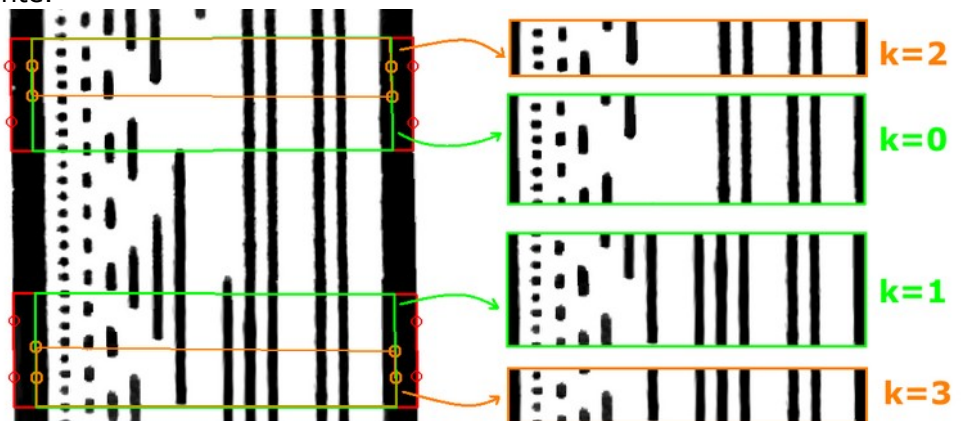
Finalement, on définit:

$$b'''_k = b''_k + D_g r \quad \text{et} \quad b'''_{4+k} = b''_{4+k} - D_d r$$

Ces dernières valeurs correspondent aux ordonnées des 8 points repérés par les cercles oranges sur la figure précédente.

Les 8 points oranges vont permettre de définir les zones d'interprétation du code gray :

k=0 pour la zone de lecture principale du haut, **k=1** pour la zone de lecture principale du bas **k=2** pour la zone de lecture secondaire du haut **k=3** pour la zone de lecture secondaire du bas



Les zones de lectures secondaires seront utilisées au paragraphe sur le détrompeur.

4.7) Fonction de lissage

On définit la fonction de lissage L qui sera utilisée dans la suite.

La fonction L est une fonction de variable x . a et b sont deux constantes.
L'image de x est $L(x,a,b)$. De plus L vérifie les conditions suivantes:

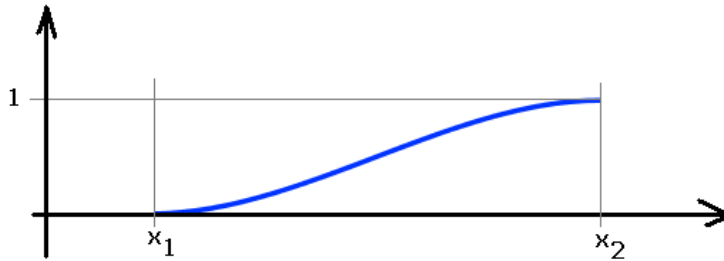
$$\begin{cases} L(x_1, x_1, x_2) = 0 \\ L(x_2, x_1, x_2) = 1 \\ L'(x_1, x_1, x_2) = 0 \\ L'(x_2, x_1, x_2) = 0 \end{cases}$$

En choisissant L comme un polynôme du troisième degré nous obtenons le système:

$$\begin{cases} L'(x, x_1, x_2) = k(x - x_1)(x - x_2) \\ L(x, x_1, x_2) = P(x)(x - x_1)^2 \\ L(x_2, x_1, x_2) = 1 \end{cases}$$

où k est une constante et où $P(x) = ax + b$ avec a et b deux constantes

Ainsi, on obtient l'expression de L , $L(x, x_1, x_2) = -\frac{(2x + x_1 - 3x_2)(x - x_1)^2}{(x_2 - x_1)^3}$



4.8) Détection discontinuité bande

On illustre le principe pour la partie supérieure de l'image $k=0$ (lignes 0 à 239), la stratégie est identique pour la partie inférieure $k=1$ (ligne 240 à 479)

La figure de droite permet de définir les éléments géométriques :

Le vecteur $\vec{u}$ permet de définir la largeur de la bande de données ainsi que l'orientation des bits de données.

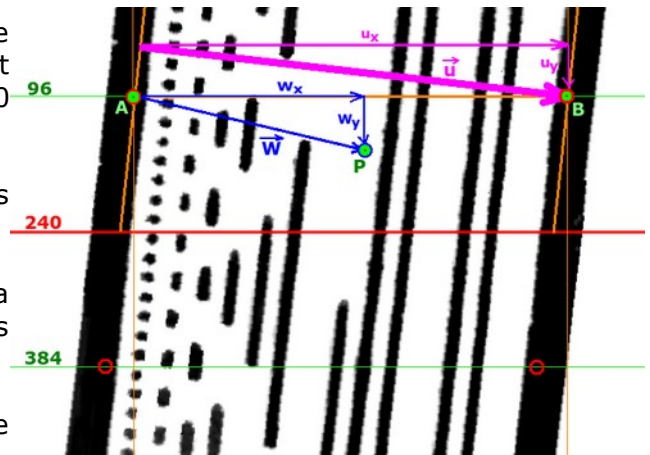
Dans ce paragraphe i est l'inclinaison du code défini au paragraphe §4.5. i vaut donc soit i_{haut} soit i_{bas} .

La norme de $\vec{u}$ vaut $AB \cos(i)$

Les points A et B étant sur la même ligne horizontale $AB = b'''_{k+4} - b'''_k$ (voir §4.5)

Pour une ligne horizontale donnée, on scanne l'ensemble des pixels x , $x \in [240k; 239+240k]$ tel que :

$$b'''_k - (y - y_0) \tan(i) \leq x \leq b'''_{k+4} - (y - y_0) \tan(i) \text{ avec } y_0 = 96 + 288k \text{ et } 0 \leq y \leq 239$$



On définit le point $P(x; y)$ et le vecteur $\vec{w} = \overrightarrow{AP}$

On pose $R = \frac{\vec{u} \cdot \vec{w}}{\|\vec{u}\|}$.

Le point P est sur la zone des bits de données ainsi $0 \leq R \leq \|\vec{u}\|$

Les valeurs des pixels sont moyennées sur l'ensemble des pixels de même valeur de R.

Nous obtenons ainsi un tableau unidimensionnel qui va permettre de localiser les bandes des bits de données

D'après le paragraphe §4.5 la largeur moyenne d'une ligne de bit vaut :

$$r = \frac{l_k}{\kappa'}$$

Pour l'initialisation de la largeur adaptative on pose : $l = r$

$$L_{0,0} = 0 \text{ et } L_{\kappa'-1,1} = l_k$$

On détermine dans un premier temps la largeur des bandes guides sur la zone des bits de données.

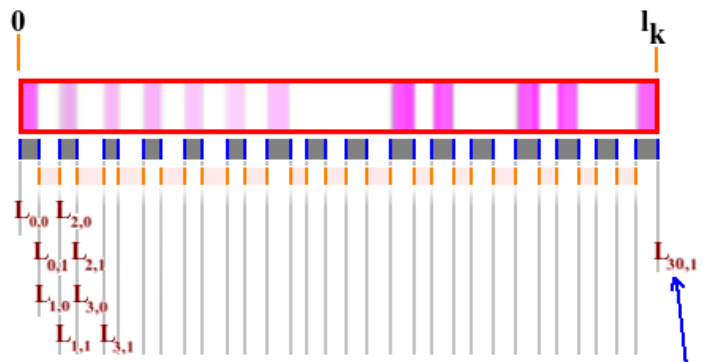
On pose $L_{\kappa'-2,1} = L_{\kappa'-1,0} = j_{mi}$ et $L_{0,1} = L_{1,0} = j_{ma}$

Par pas de 1 pixel on détermine la position de la ligne recherchée en maximisant le contraste entre les parties blanches (zone d'inter-bit) et les parties noires (zone de bit)

A la fin de la recherche j_{mi} et j_{ma} déterminent la position de la ligne :

On pose : $L_{m-1,1} = L_{m,0} = j_{mi}$ et $L_{m,1} = L_{m+1,0} = j_{ma}$

m correspond au numéro de la ligne de donnée considérée. m varie de 2 à $\kappa' - 2$. m est pair pour une ligne noire (guide ou de bit) et m est impair pour une ligne blanche (ligne inter-bit). Le second indice vaut 0 si on considère la limite gauche de la ligne et 1 si on considère la limite droite de la ligne.



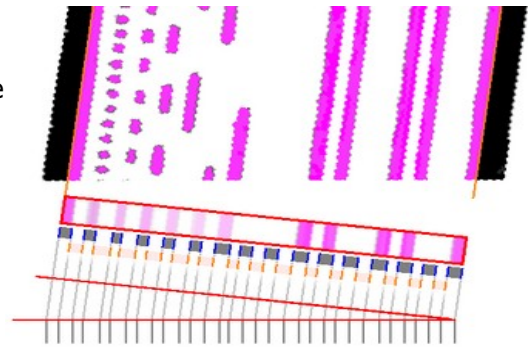
À chaque détermination d'une ligne de bit, m est incrémenté de 2, on recalcule la valeur de la largeur adaptative l par :

$$l = E \left(\frac{L_{\kappa'-1,0} - L_{m-2,1}}{\kappa' - m} - \frac{1}{2} \right) + 1 \quad \text{E est la fonction partie entière}$$

Les valeurs effectives des $L_{m,0}$ et $L_{m,1}$ sont finalement considérées avec l'inclinaison i de l'image.

Pour toutes les valeurs de m , $0 \leq m < \kappa$,

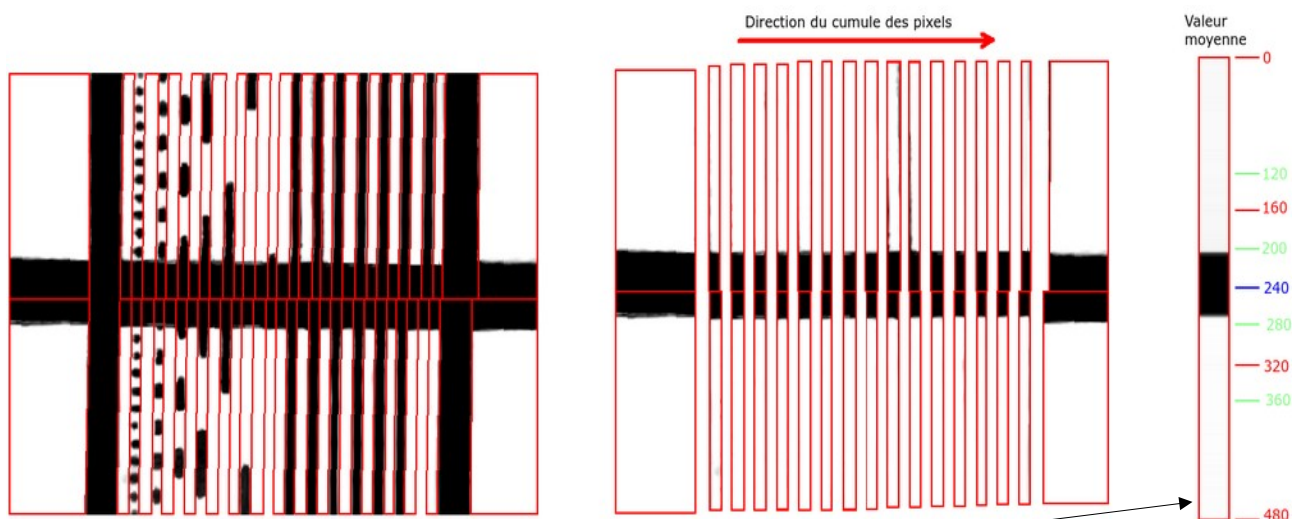
On substitue $L_{m,0}$ par $\frac{L_{m,0}}{\cos(i)}$ et $L_{m,1}$ par $\frac{L_{m,1}}{\cos(i)}$



On procède ensuite, a proprement parler, à la détection et la localisation d'une discontinuité qui se fait en 4 étapes :

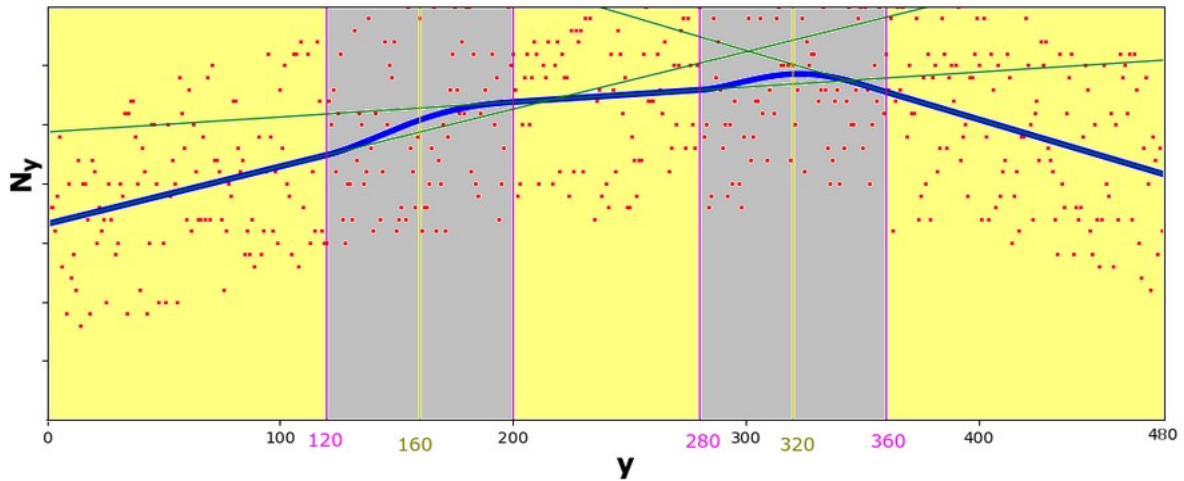
- (1) Localisation des zones blanches de l'image
- (2) Moyennage suivant la direction horizontale de la valeur de bit
- (3) Ajustement d'un modèle de régression linéaire par morceau afin d'éliminer le gradient de luminosité
- (4) Statistique sur la valeur des pixels afin de localiser la position de la discontinuité le cas échéant

(Etape 1&2) L'ensemble de l'image est considéré ici avec l'angle i compensant la rotation du code par rapport à l'image de la webcam. On détermine ensuite les zones blanches ainsi qu'à l'extérieur de la bande guide inter-bit correspondant à des portions blanches en l'absence de discontinuité (en rouge sur l'image)



On obtient un tableau unidimensionnel de 480 pixels, théoriquement de niveau de luminosité uniforme si il n'y pas de discontinuité.

(Etape 3) On modélise le gradient de luminosité par 3 ajustements affines (pondération de la valeur des pixels en assignant un poids basé sur la luminosité) puis on lisse sur 5 secteurs et on soustrait le gradient lissé.



$q=0$: La première régression linéaire est réalisée du pixel n°0 au pixel n°159 $y \in [[0; 159]]$
 $q=1$: La seconde régression linéaire est réalisée du pixel n°160 au pixel n°319 $y \in [[160; 319]]$
 $q=2$: La troisième régression linéaire est réalisée du pixel n°320 au pixel n°479 $y \in [[320; 479]]$

L'expérience montre que le choix du poids $w_y = \frac{1}{(1+N_y)^4}$ est bien adapté. Les niveaux de l'image en mémoire étant inversés par rapport à celle affichée, y_x vaut 0 pour un niveau blanc et 255 pour un niveau noir. Ainsi, une discontinuité aura des niveaux tendant vers le noir et donc un poids w_x tendant vers 0. Les parties hors discontinuité auront un poids w_x tendant vers 1. Ainsi,

$$a_q = \frac{\sum w_y y \sum w_y N_y - \sum w_y \sum w_y y N_y}{(\sum w_y y)^2 - \sum w_y \sum w_y y^2} \quad b_q = \frac{\sum w_y N_y - a_q \sum w_y y}{\sum w_y} \quad \text{avec} \quad w_y = \frac{1}{(1+N_y)^4} \quad 0 \leq N_y \leq 255$$

On définit 2 zones de raccordement afin de définir une continuité dans la courbe modélisant le gradient à partir des 3 régressions linéaires. L est ici la fonction de lissage présentée précédemment.

Pour $y \in [[0; 119]]$	le niveau du gradient vaut	$n_y = a_0 y + b_0$
Pour $y \in [[120; 199]]$	le niveau du gradient vaut	$n_y = (a_0 y + b_0)L(y, 120, 200) + (a_1 y + b_1)L(y, 200, 120)$
Pour $y \in [[200; 279]]$	le niveau du gradient vaut	$n_y = (a_1 y + b_1)$
Pour $y \in [[280; 359]]$	le niveau du gradient vaut	$n_y = (a_1 y + b_1)L(y, 280, 360) + (a_2 y + b_2)L(y, 360, 280)$
Pour $y \in [[360; 479]]$	le niveau du gradient vaut	$n_y = (a_2 y + b_2)$

Puis, pour $y \in [[0; 479]]$ $V_y = N_y - n_y$ si $V_y < 0$ alors on impose $V_y = 0$

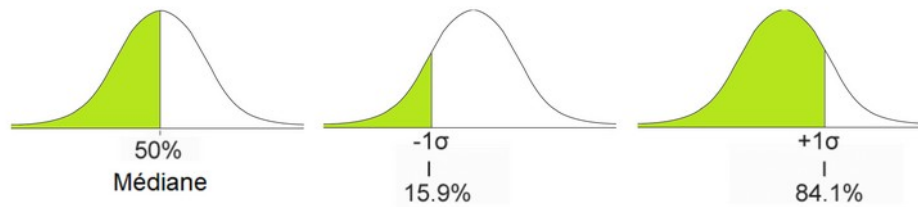
On recherche la valeur minimale et maximale des V_y . On note ces 2 valeurs $V_{\min}$ et $V_{\max}$

Finalement, on impose la dynamique de V_y entre 0 et 255 $V_y = 255 \frac{V_y - V_{\min}}{V_{\max} - V_{\min}}$

Les 480 valeurs V_y obtenues, correspondant à la moyenne des valeurs de pixels sur une ligne originale donnée moins la valeur du gradient, sont utilisées pour calculer une moyenne et un écart type. Pour cela :

On détermine la valeur y , noté y_0 , tel que 16% des valeurs parmi le 480 sont inférieurs à V_y .
 On détermine la valeur y , noté y_1 , tel que 50% des valeurs parmi le 480 sont inférieurs à V_y .
 On détermine la valeur y , noté y_2 , tel que 84% des valeurs parmi le 480 sont inférieurs à V_y .

Les valeurs 16 % et 84 % font référence à une loi normale afin de déterminer l'écart-type : y_1 est donc la valeur médiane qui correspond à la ligne localisant la partie centrale de la discontinuité. On note cette valeur Y pour la suite : $Y = y_1$



La largeur de la discontinuité est définie par la valeur de l'écart-type σ .

σ est estimé par :
$$\sigma = \frac{x_2 - x_0}{2}$$

Si l'écart-type est inférieur à 3 pixels ou supérieur à 138 pixels, on impose l'écart-type à 10000, signifiant qu'il n'y a pas de détection d'une discontinuité.

La valeur 138 correspond à un écart-type d'une loi uniforme sur un intervalle d'amplitude 480.

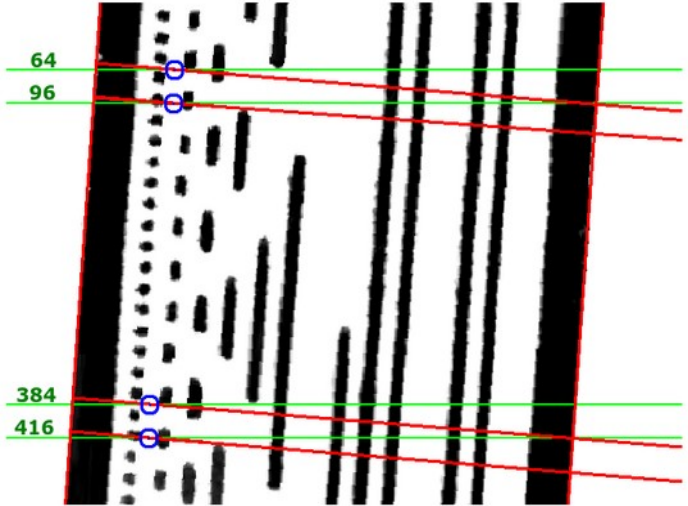
$(\sigma = \frac{480}{\sqrt{12}} \simeq 138,56)$

4.9) Géométrie de bande de données

Notre objectif est de lire les niveaux du code gray associés aux lignes **96** et **384** de la webcam (lecture principale) et associés aux lignes **64** et **416** de la webcam (lecture secondaire).

Les lignes de lecture sont donc inclinées de l'angle i mesuré sur les lignes guides.

Les points de pivotement des lignes guides sont choisis comme rattachés au code Gray et sont localisés sur la bande blanche séparant les deux bits de poids faible (pour la suite on introduit n_e , $n_e = 2$). Les points sont repérés sur l'image ci-contre par les cercles.



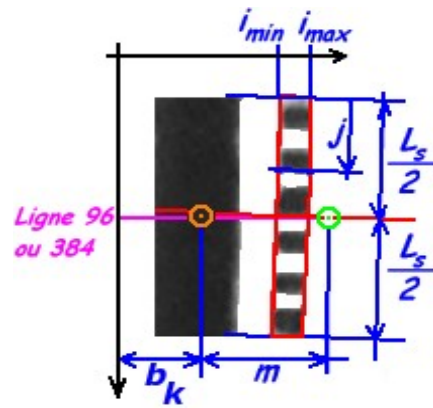
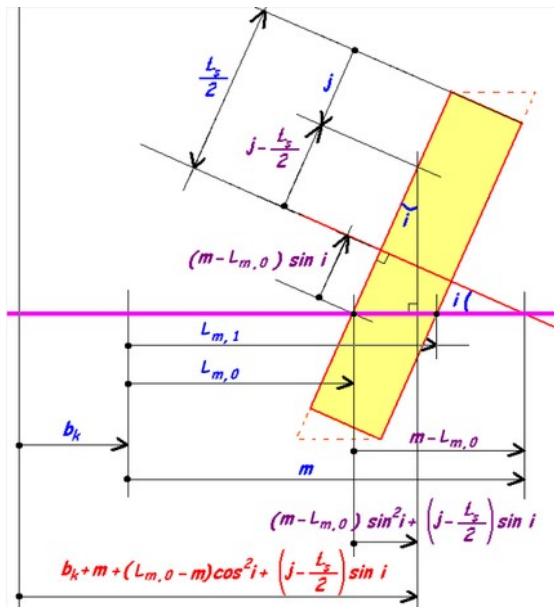
La localisation des points de pivotement est dans la zone des bits de poids faibles. En effet, la méthode de localisation des bits de poids faibles est particulièrement robuste et imposera la valeur finale de la graduation lue par la webcam.

Ainsi, les points de pivotement conservent les ordonnées valant respectivement **96** et **384** en lecture principale (et **64** et **416** en lecture secondaire).

L'ensemble des pixels de chaque zone de lecture sera utilisé par la lecture du code. En effet une simple détection du niveau des pixels situés exactement sur la ligne de lecture conduirait à une interprétation hasardeuse de la valeur du code. En particulier, la méthode serait particulièrement sensible au contraste entre les zones noires et blanches et conduirait à une méthode de lecture peu robuste.

Ainsi, d'une façon similaire à l'œil humain, nous nous servons de toute l'information disponible dans les zones de lectures pour trouver la valeur du niveau (bas ou haut) sur la ligne de lecture.

En utilisant les notations de la figure ci-après, les pixels d'une zone de lecture sont tels que :



$$i_{\min} = b_k + m + (L_{m,0} - m) \cos^2(i) + \left(j - \frac{L_s}{2}\right) \sin(i)$$

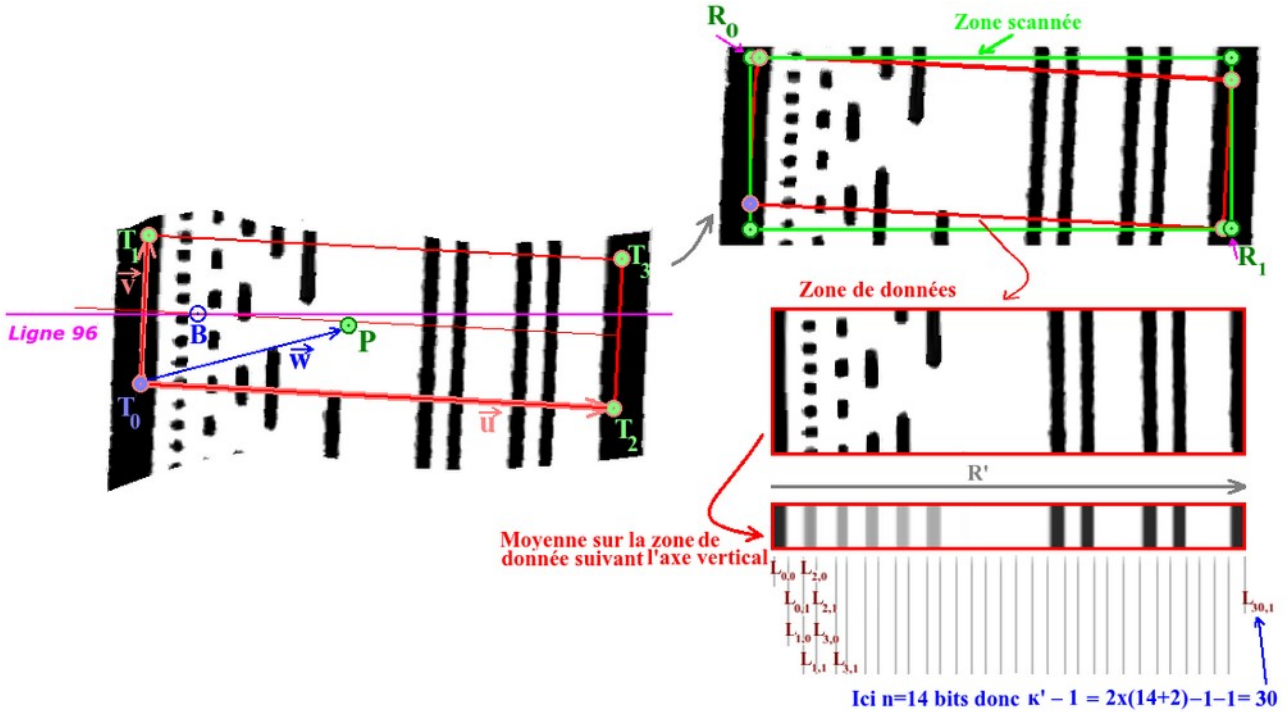
$$i_{\max} = b_k + m + (L_{m,1} - m) \cos^2(i) + \left(j - \frac{L_s}{2}\right) \sin(i)$$

et avec $m = (n_e + 1.5) \frac{l}{K}$, On rappelle que $n_e = 2$

4.10) Détermination des largeurs adaptatives des lignes de code

Suivant la qualité de l'impression, celle ci conduit à une irrégularité sur la largeur de bande codant le code grey ainsi que la largeur des bandes les séparant.

Pour palier à cet éventuel défaut, le programme détermine automatiquement la largeur individuelle des bandes.



Relativement à la figure ci-dessus :

On note B le point de pivotement. $B(x_0; y_0)$

On rappelle que $x_0 = b'''_{k+m}$ et y_0 vaut respectivement 96 et 384 en lecture principale (et 64 et 416 en lecture secondaire).

Relativement au paragraphe § 4.9 sur la géométrie de la bande de données, la zone de lecture de bit de données correspond au cas où $L_{m,0} = 0$ et $L_{m,1} = b'''_{k+4} - b'''_k$. Ainsi,

$$T_0 \left(x_0 - m \cos^2(i) + \frac{1}{2} L_s \sin(i); y_0 - m \cos(i) \sin(i) + \frac{1}{2} L_s \cos(i) \right)$$

$$T_1 \left(x_0 - m \cos^2(i) - \frac{1}{2} L_s \sin(i); y_0 - m \cos(i) \sin(i) - \frac{1}{2} L_s \cos(i) \right)$$

$$T_2 \left(x_0 + (b'''_{k+4} - x_0) \cos^2(i) + \frac{1}{2} L_s \sin(i); y_0 - (b'''_{k+4} - x_0) \cos(i) \sin(i) + \frac{1}{2} L_s \cos(i) \right)$$

$$T_3 \left(x_0 + (b'''_{k+4} - x_0) \cos^2(i) - \frac{1}{2} L_s \sin(i); y_0 - (b'''_{k+4} - x_0) \cos(i) \sin(i) - \frac{1}{2} L_s \cos(i) \right)$$

$$R_0 = \left(\min(x_{T_0}, x_{T_1}); \min(y_{T_1}, y_{T_3}) \right) \quad R_1 = \left(\max(x_{T_2}, x_{T_3}); \max(y_{T_0}, y_{T_2}) \right)$$

Le rectangle rouge $T_0T_1T_2T_3$ est donc inclus dans le rectangle vert de diagonale $[R_0R_1]$

On définit le point $P(x; y)$ et les vecteurs $\vec{u} = \overrightarrow{T_2 T_0}$ $\vec{v} = \overrightarrow{T_1 T_0}$ et $\vec{w} = \overrightarrow{AP}$

On pose $R' = \frac{\vec{u} \cdot \vec{w}}{\|\vec{u}\|}$ et $R'' = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\|}$

Le point P est sur la zone rouge des bits de lecture ainsi $0 \leq R' \leq \|\vec{u}\|$ et $0 \leq R'' \leq \|\vec{v}\|$

Les pixels de l'image scannée sont limités à ceux du rectangle vert. Le pixel sélectionné dans la zone scannée sont ceux du rectangle rouge.

Les valeurs des $L_{m,0}$ et $L_{m,1}$ pour $0 \leq m < \kappa'$ sont déterminées de façon identiques au paragraphe 4.8 (Détection discontinuité bande)

4.11) Bits de poids faible

On conserve les notations introduites au paragraphe précédent.

Les bits de poids faibles que l'on va considérer dans cette partie sont tels que la zone de lecteur est suffisamment longue pour voir au moins une zone noire et une zone blanche entièrement. La sélection des bits considérée ici dépend donc de l'échelle spatiale e_x .

Par ailleurs, L_s est choisi comme étant une puissance entière de 2. Ainsi, on note p_s la valeur :

$$p_s = \frac{\ln(L_s)}{\ln(2)} \text{ avec } \ln \text{ la fonction logarithme népérien.}$$

Ainsi, la somme des longueurs d'une zone noire et d'une zone blanche d'un bit de poids n doit vérifier :

$$(2^n + 2^n)e_x < L_s$$

$$\text{d'où} \quad n \ln(2) + \ln(2e_x) < p_s \ln(2)$$

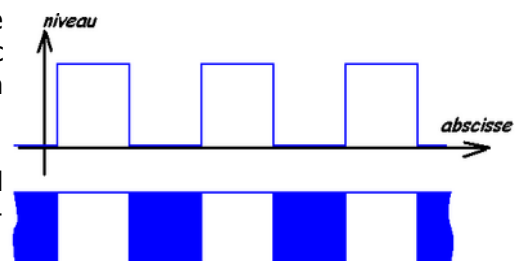
$$\text{Ainsi} \quad n < p_s - \frac{\ln(2e_x)}{\ln(2)}$$

Le nombre de bits considéré $n_{\max}$ est donc $n_{\max} = E\left(p_s - \frac{\ln(2e_x)}{\ln(2)}\right)$, E étant la fonction partie entière.

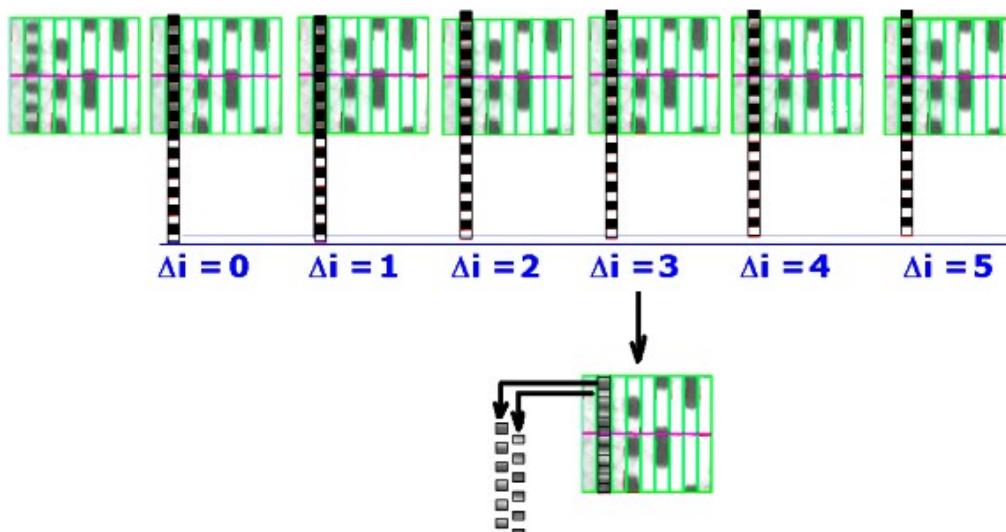
$$\text{Or } e = 2e_x \quad n_{\max} = E\left(p_s - \frac{\ln(e)}{\ln(2)}\right)$$

- Une ligne de bit est une succession de rectangles noirs et de rectangles blancs de même largeur

Les longueurs des intervalles sont connues et ne dépendent que de la valeur de e . On peut donc considérer le modèle idéal et le comparer à l'observation sur l'ensemble de la zone de lecture d'un bit.



Le principe est de tester le modèle en le décalant pixel par pixel par rapport à l'observation (voir figure ci-après).



Pour chaque cas on calcule le contraste entre la partie couverte par le niveau haut du modèle et la partie couverte par les niveaux bas (illustration ci-dessus pour le cas $\Delta i=3$). Entre chaque cas, on déplace par pas de 1 pixel le modèle par rapport à l'image.

On note S_h la somme des niveaux des pixels couverte par un niveau haut
 S_b la somme des niveaux des pixels couverte par un niveau bas

On obtient la valeur C du contraste définie par
$$C = \left| \frac{S_h - S_b}{S_h + S_b} \right|$$

Le contraste minimum (ou maximum, ce qui revient au même à un déphasage d'une demie-période) indique le meilleur alignement entre le modèle et le signal observé.

On réitère la procédure sur chaque bit considéré dans cette partie, indépendamment les uns des autres.

4.12) Bits de poids forts

Le principe précédent ne peut pas être utilisé pour les bits de poids fort. En effet l'algorithme de minimisation du contraste revient à chercher délibérément la position d'un front de passage d'un niveau haut à un niveau bas (ou inversement). Or, pour les bits de poids fort, il existe des zones où le bit reste à zéro ou à un sur la totalité du champs de vue de la webcam.

Il faut donc utiliser un algorithme qui détecte un front si il existe, ou détecte un niveau si il n'y a pas de front.

(Etape 0) Il faut dans un premier temps initialiser la valeur des niveaux. On utilise les lignes guides (noires) et les interlignes (blanches) pour déterminer la valeur des niveaux hauts et bas sur l'image. On considère de plus la valeur des niveaux hauts et bas des bits de poids faible.

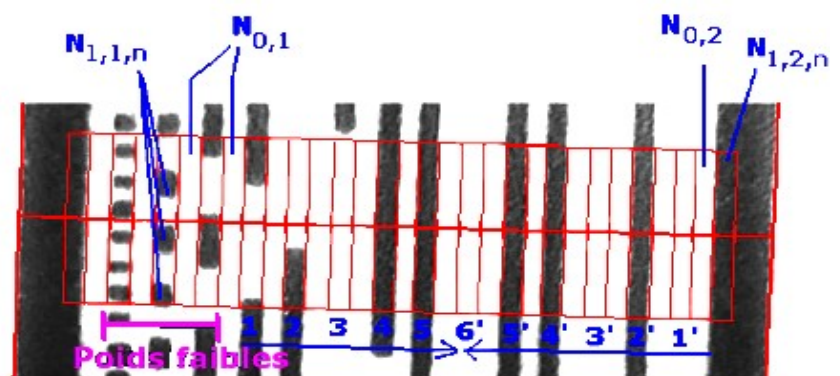
Pour le coté bits de poids fort : le niveau noir est initialisé à l'aide de la couleur moyenne de la ligne guide de droite. On note la valeur de ce niveau $N_{1,2,n}$. Le niveau blanc correspondant est initialisé sur la couleur moyenne de l'interligne précédent cette ligne guide. La valeur de ce niveau est noté $N_{0,2}$.

Pour le coté bits de poids faible : on utilise soit le dernier bit de poids faible ($n_f=1$), soit l'avant dernier bit de poids faible ($n_f=2$). (Le dernier bit de poids faible est utilisé si l'on est certain d'avoir un niveau noir suffisamment grand pour obtenir une bonne estimation de la valeur de ce niveau. Dans le cas contraire on utilise l'avant dernier.)

La couleur noire moyenne du bit de poids faible retenue est notée $N_{1,1,n}$

Le niveau blanc correspondant est initialisé sur la couleur moyenne des deux interlignes encadrant la ligne du dernier bit de poids faible. La valeur de ce niveau est noté $N_{0,1}$

Finalement, afin de tenir compte du gradient de luminosité on interpole la valeur des niveaux noirs et blancs dans la zone du bit considéré.

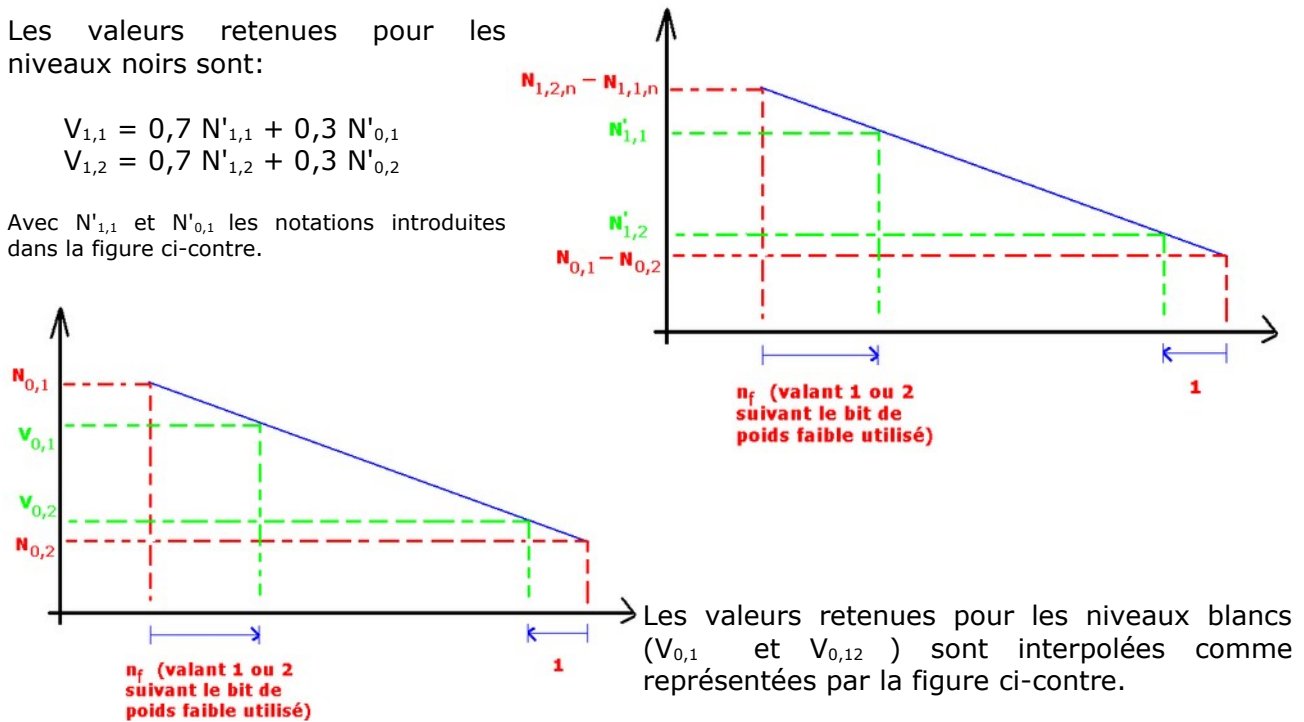


Les valeurs retenues pour les niveaux noirs sont:

$$V_{1,1} = 0,7 N'_{1,1} + 0,3 N'_{0,1}$$

$$V_{1,2} = 0,7 N'_{1,2} + 0,3 N'_{0,2}$$

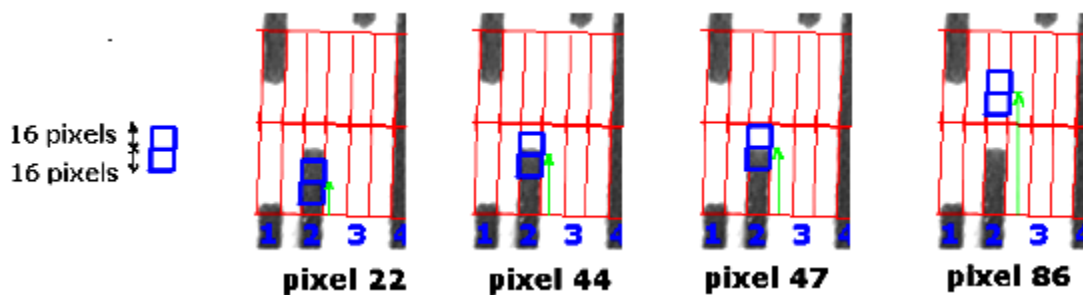
Avec $N'_{1,1}$ et $N'_{0,1}$ les notations introduites dans la figure ci-contre.



(Etape 1) Les niveaux de chaque bit sont déterminés deux à deux en partant de l'extérieur vers l'intérieur, comme présenté sur la figure en haut de cette page.

On recherche un front en moyennant les valeurs sur 16 pixels en amont et en aval du front supposé. On obtient ainsi deux couleurs qui, si elles sont suffisamment différentes, marquent la présence d'un front. La différence de couleur doit d'au-moins 90% de différence de $V_{1,i}$ et $V_{0,i}$, i valant 1 pour le côté poids faible, 2 pour le côté poids fort. La plus grande différence de valeur est la position effective du front.

Si on ne trouve pas une différence de couleur suffisante, alors il n'y a pas de front et le niveau du bit est considéré bas ou haut en fonction de la couleur moyenne de ce niveau relativement à $V_{1,i}$ et $V_{0,i}$.



L'image ci-dessus, illustre le principe. Dans cet exemple, le pixel 22 (ou 86) ne présente pas de différence de couleur importante dans les deux zones. Au contraire, le pixel 44 présente une différence mais comme le pixel 47 présente une différence encore plus importante, le front sera localisé au niveau du pixel 47.

(Etape 2) Finalement, on re-détermine la valeur correspondant au noir et blanc. On utilise les valeurs des interlignes blancs encadrant les bits considérés à l'étape précédente. Pour la valeur du noir, on considère le niveau noir des bits précédents si il a été détecté sinon on conserve la valeur précédente. Pour les deux couleurs, noire et blanche, on tient compte d'un éventuelle gradient de luminosité en utilisant la même méthode que dans l'étape d'initialisation des valeurs (Etape 0).

4.13) Moindres carrés sur un ensemble de bits

4.13.1) Calcul préliminaire

Les paragraphes précédents ont permis de localiser la position des différents bits du code Gray. Cela revient à déterminer les valeurs des $x'_{n,b}$ du paragraphe §2.4., on prenant un référence quelconque.

Si l'on considère deux bits successifs, on a: $x'_{n,j-1} = (4n+1) e_x 2^{j-1}$
et $x'_{n,j} = (4n+1) e_x 2^j$

L'orientation de l'axe des ordonnées sur le fichier d'impression du code gray est opposé à celui de l'axe des ordonnées sur l'image d'une webcam. Ainsi pour la suite on prendra la notation correspondant à l'orientation de la webcam soit:

$$x'_{n,j-1} = -(4n+1) e_x 2^{j-1}$$
$$\text{et } x'_{n,j} = -(4n+1) e_x 2^j$$

et, en particulier, pour $n=0$ pour le bit j et $n=k$ pour le bit $j-1$, on obtient:

$$x'_{k,j-1} = -(4k+1) e_x 2^{j-1}$$
$$\text{et } x'_{0,j} = -e_x 2^j$$

$$\text{d'où } x'_{0,j-1} - x'_{k,j} = e_x 2^j - (4k+1) e_x 2^{j-1}$$

$$\text{et ainsi } k = \frac{x'_{0,j-1} - x'_{k,j}}{2^{j+1} e_x} + \frac{1}{4} \quad (\text{on rappelle que } k \text{ est un entier})$$

$$\text{or } e = 2 e_x . \text{ D'où } k = \frac{x'_{0,j-1} - x'_{k,j}}{2^j e} + \frac{1}{4}$$

4.13.2) Déterminant de la position du code en haute précision

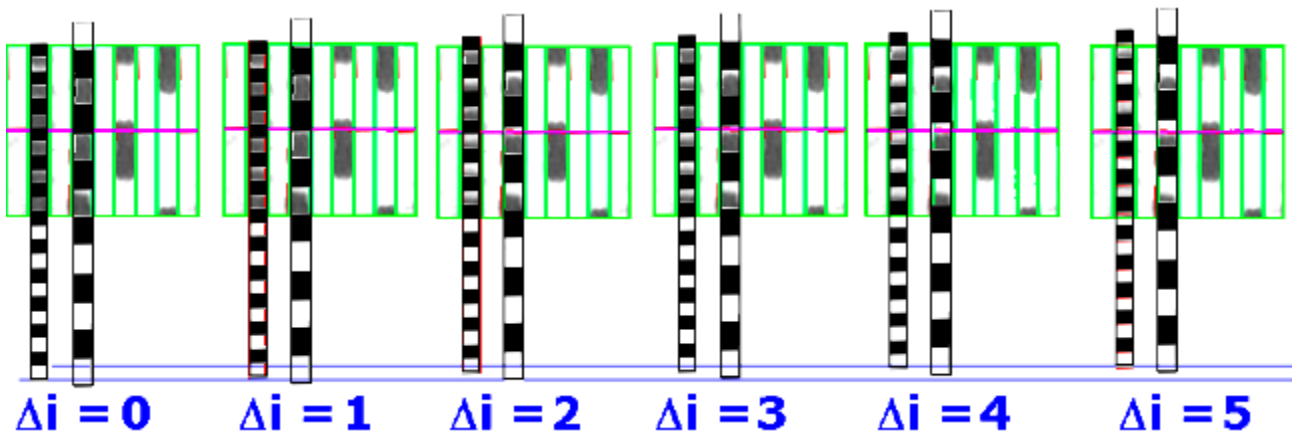
Bits de poids faible

D'après le calcul du paragraphe précédent, les positions des bits de poids faible sont liées par une relation faisant apparaître une valeur entière noté k .

Les valeurs correspondent aux $\mathbf{x'}$ déterminés par la méthode du §4.5 ont été trouvées indépendamment les une des autres.

Dans un premier temps on recalcule par récurrence la valeur de $\mathbf{x'}$ en utilisant les valeurs trouvées au §4.5 tout en imposant une valeur de k entière la plus proche de celle trouvée. Seul le bit de poids le plus faible conserve la valeur de $\mathbf{x'}$ trouvée par la méthode §4.5. Dans la pratique les autres $\mathbf{x'}$ changent peu, la méthode §4.5 étant suffisamment fiable.

On utilise ensuite la méthode décrite au paragraphe §4.5, cette fois-ci en minimisant le contraste sur l'ensemble de bits de poids faible en un seul bloc. De plus, le modèle se déplace par pas de 0,1 pixel par rapport à l'image.



La détermination de la position du code avec une précision de 0,1 pixel est équivalent au principe du vernier sur un pieds à coulisse. En effet, la zone observée couvre 128 pixels. L'ordre de grandeur de l'échelle étant de 5 pixels, on couvre de l'ordre d'une dizaine de période du bits de poids faible. On peut donc mesurer la position du code en dessous d'une échelle de 1 pixel.

Bits de poids fort

On reprend le calcul préliminaire, mais en considérant cette fois-ci des bits non successifs notés i et $i-n$.

$$x'_{k,j-n} = -(4k+1)e \cdot 2^{j-n}$$

$$\text{et } x'_{0,j} = -e_x \cdot 2^j$$

$$\text{d'où } x'_{0,j-n} - x'_{k,j} = e_x \cdot 2^j - (4k+1) e_x \cdot 2^{j-n}$$

$$\text{et ainsi } k = \frac{x'_{0,j-n} - x'_{k,j}}{2^{j-n+2} e_x} + 2^{n-2} - \frac{1}{4} \quad (\text{on rappelle que } k \text{ est un entier})$$

$$\text{or } e = 2 e_x . \text{ D'où } k = \frac{x'_{0,j-n} - x'_{k,j}}{2^{j-n+1} e} + 2^{n-2} - \frac{1}{4}$$

On re-détermine la valeur de k des bits de poids fort en utilisant la relation ci-dessus et en considérant la valeur des x' des bits de poids comme référence. Par défaut, la valeur de n est 1, c'est à dire que l'on recalcule le x' d'un bit en utilisant la valeur de x' du bit qui le précède. Toutefois, si une valeur de k est trop grande, compte tenu de la grande période des bits de poids fort, on risque d'introduire des erreurs. On incrémente n autant de fois que nécessaire afin d'avoir une valeur pour k suffisamment petite.

4.14) Détermination « haute précision » de la valeur lue

Les paragraphes précédents ont permis de déterminer les valeurs du code Gray sur les deux lignes de lecture.

Lire à la précision du code Gray signifie que nous arrivons à déterminer la position de ce code par rapport à la webcam avec une précision valant l'échelle e introduite au §4.5. Toutefois, nous avons vu que nous étions capable de déterminer la position du code à une précision plus grande (cf. §4.7.2). Nous pouvons donc interpoler la valeur lue entre deux valeurs entières de graduation codées par le code gray.

Ainsi, la valeur entière de la graduation sera déterminée par la conversion du code gray en valeur décimale. Le code gray considéré étant un code 14 bits, cette valeur sera comprise entre 0 et $2^{14} - 1 = 16383$.

La valeur décimale est réalisée grâce à une interpolation sur la valeur de $x'_{0,0}$ du bit de poids faible.

La valeur décimale lue V_r est donc décimale de la graduation ($0 \leq f < 1$)

$$V_r = V + f$$

où V est la valeur entière et f la partie

4.13) Détrompeur

La méthode présentée comporte un défaut dans le cas où l'interprétation de l'image de la webcam conduit à une valeur erronée sur l'une des deux lignes de lecture.

Si l'on considère uniquement deux lignes de lecture et que l'une d'entre elle donne une valeur fautive, l'autre aura peut-être une valeur correcte mais nous serons incapable de dire de façon systématique qui des deux donnent une valeur raisonnable.

Ce cas doit être considéré car il existe nécessairement au moins une discontinuité dans les graduations. Cette discontinuité est constituée par la jonction des deux extrémités de la bande imprimée.

Pour s'affranchir de la discontinuité, il faut que celle-ci ait une longueur maximale G_{max} plus faible que l'espacement entre les deux zones de lecture centrées sur les lignes de lecture soit :

$$G_{max} = 288 - 64 - 64$$

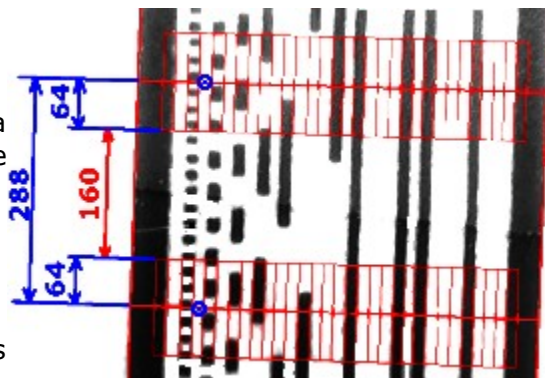
$$G_{max} = 160$$

Toutefois pour éviter des effets de bords, la valeur effective G_{eff} , sera prise à F_d , $F_d = 90\%$ de cette dernière soit :

$$G_{eff} = 0,9 \times 160$$

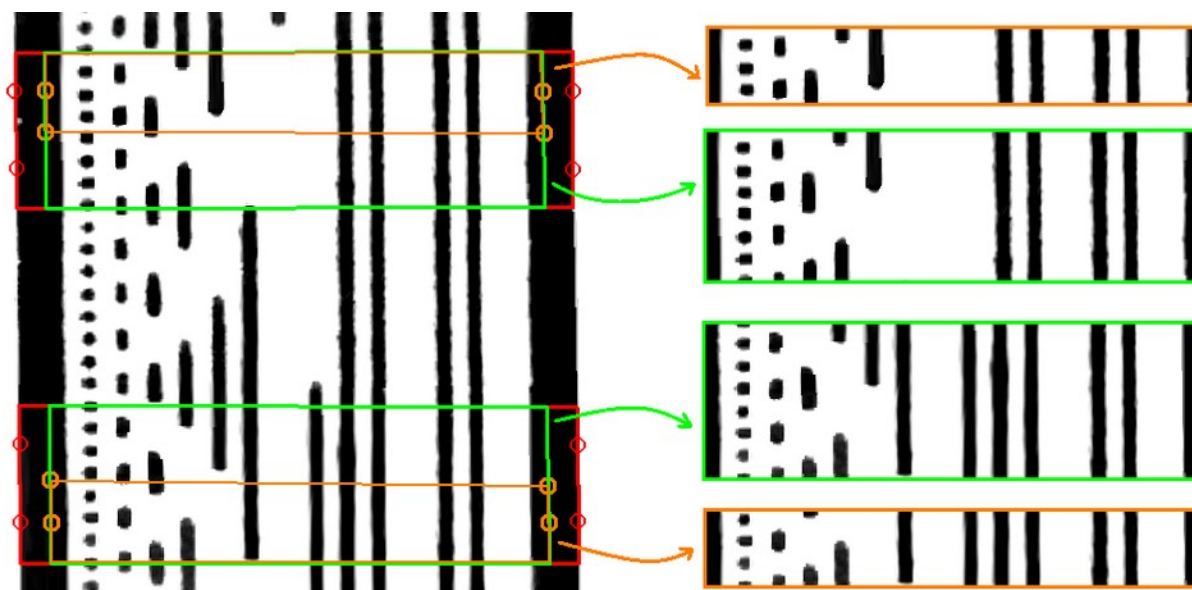
$$G_{eff} = 144$$

Cette zone est repérée par les deux lignes vertes de l'image suivante.



Afin de réaliser le détrompeur, on définit deux lignes de lecture secondaires localisées suivant les lignes de la webcam de 32 pixels avant la ligne de lecture n°96 et de 32 pixels après la ligne de lecture n°384.

La lecture de ces lignes secondaires repose sur le même principe que la lecture des lignes de lecture initiales à ceci près qu'au lieu de centrer des zones de lecture de 128 pixels, on considère des zones de 64 pixels.



Elles correspondent, pour les notations du §4.3 à :

k=2 pour la ligne guide de gauche et ligne de lecture secondaire du haut

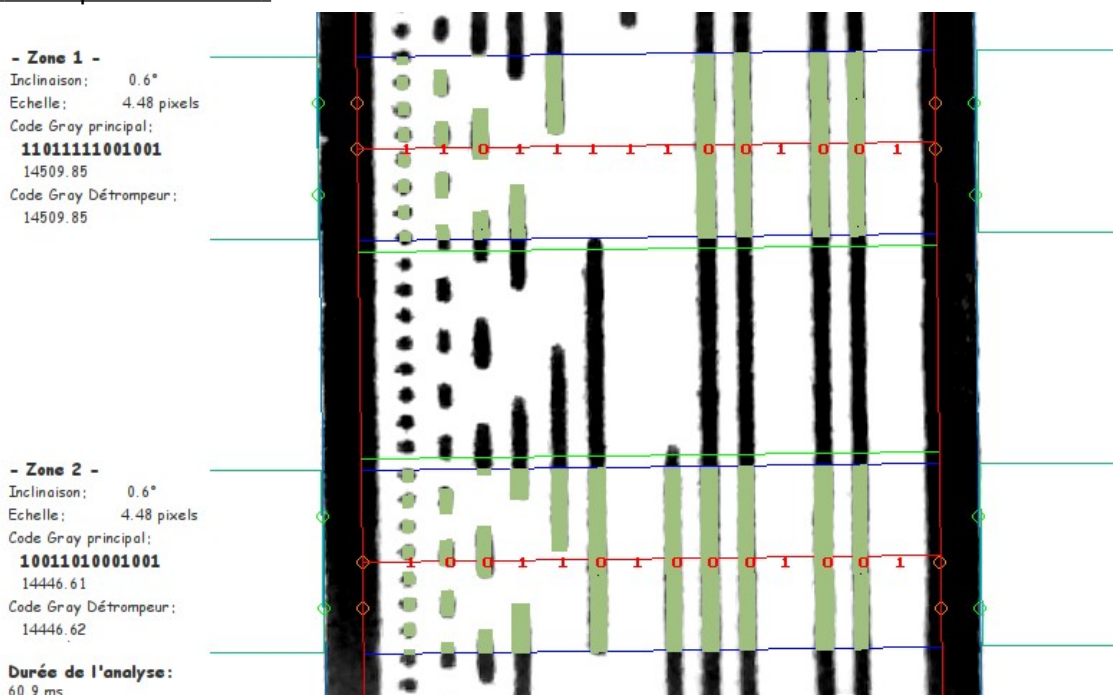
k=6 pour la ligne guide de droite et ligne de lecture secondaire du haut

k=3 pour la ligne guide de gauche et ligne de lecture secondaire du bas

k=7 pour la ligne guide de droite et ligne de lecture secondaire du bas

Ainsi, ce choix permet de garantir quelle est la ligne de lecture qui donne une valeur correcte. La ligne de lecture donnant la bonne indication est celle dont la ligne de lecture secondaire donne la même valeur (modulo le décalage de plus ou moins 32 pixels).

4.15) Exemple de lecture



V) Étalonnage du code gray

5.1) Présentation

Il ne faut pas se faire d'illusion, imprimer un code Gray sur une imprimante ne permettra pas d'avoir des graduations régulières. La mise en place de la bande imprimée apportera des contraintes de régularité, de tension. Les distorsions optiques de la webcam ne seront pas parfaitement corrigées...

Pour un ensemble de raison, il est nécessaire d'étalonner le code du disque afin de compenser les irrégularités des graduations.

5.2) Principe

Afin de réaliser l'étalonnage nous devons utiliser une référence de longueur la plus fiable possible. L'inertie du télescope en rotation, la régulation de la vitesse de rotation du moteur pas à pas et la démultiplication importante nous offre une telle référence. On note v la vitesse de défilement du code Gray devant la webcam, supposée parfaitement constante. On note t le temps écoulé. La longueur L en pixel de la bande supportant le code Grey ayant défilé devant la webcam à un instant t vaut :

$$L = v t$$

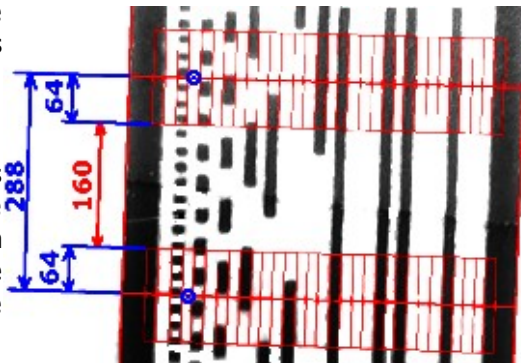
Lors du défilement du code Gray, nous mémorisons régulièrement les valeurs du code Gray lues par l'algorithme aux lignes 96 et 384 et nous leur associons l'instant t où la mesure a été réalisée.

L'objectif est de mesurer la durée d'un tour complet puis par proportion associée à chaque mesure un angle.

5.3) Échantillon d'images du code gray

Il existe des discontinuités dans le code, comme c'est le cas au niveau de la jonction entre les différentes extrémités de la bande imprimée.

Afin de pouvoir s'affranchir de ces discontinuités, pour un champ d'image donné, on fait 2 lectures (ligne 96 et 384). Comme expliqué au §4.13, une discontinuité doit au plus faire 144 pixels. En respectant cette contrainte on pourra construire une continuité de graduation lors de l'étalonnage de la bande.



5.4) Réalisation des mesures

L'échelle du code est supposée théoriquement constante. En effet le disque tourne autour d'un axe et la distance entre la webcam et le code varie d'une façon négligeable.

$$\Delta_i = v_{0,i} - v_{1,i} \approx \text{constante}$$

On suppose ici que le code Gray est orienté de telle façon à ce que la valeur de la mesure du haut $v_{0,i}$ soit supérieure à celle de la mesure du bas $v_{1,i}$.

5.5) Sélection des mesures qui seront utilisées pour l'étalonnage

D'une façon préliminaire on réalise une étude statistique sur l'ensemble des mesures réalisées lors d'une rotation complète du code Gray.

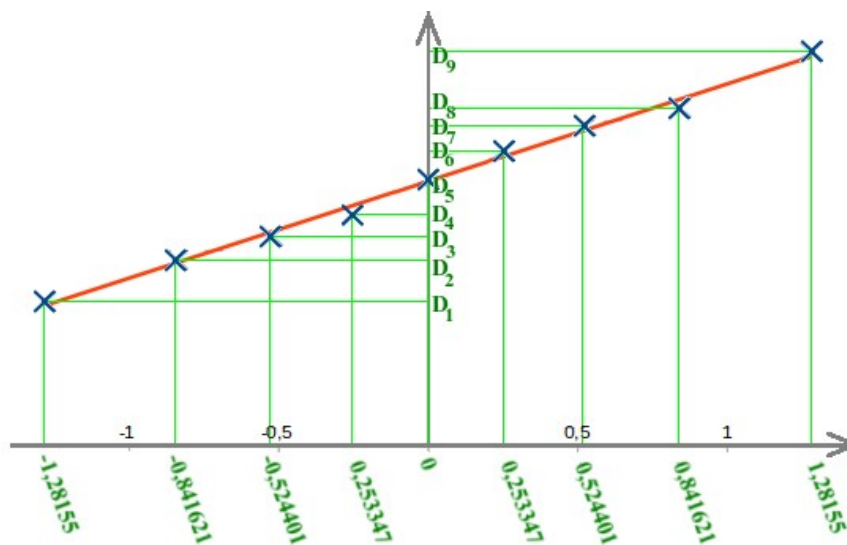
On détermine les déciles de la série notés D_1 à D_9 des valeurs des Δ_i

Il y a suffisamment de mesures pour que la série Δ_i converge vers une loi normale.

Les déciles d'une loi normale centrée réduite est donnée par la table.

Déciles	X
D_1	-1,28155
D_2	-0,841621
D_3	-0,524401
D_4	-0,253347
D_5	0
D_6	0,253347
D_7	0,524401
D_8	0,841621
D_9	1,28155

On réalise une régression linéaire en prenant les déciles d'une loi normale centrée réduite comme abscisse et les déciles obtenue sur la série Δ_n en ordonnée.



$$a = \frac{\sum x \sum y - 9 \sum xy}{(\sum x)^2 - 9 \sum x^2}$$

$$b = \frac{\sum y - a \sum x}{9}$$

Dans ce cas, le coefficient directeur correspond à l'écart-type $\sigma_\Delta = a$ sur les valeurs des Δ_i l'ordonnée à l'origine correspond à la valeur médiane $\bar{\Delta} = b$ des Δ_i .

5.5.1) Détermination de la fiabilité des mesures

Une valeur Δ_i de la série de mesure est considérée non fiable si $|\Delta_i - \bar{\Delta}| \geq 5\sigma_\Delta$ et fiable dans le cas contraire.

5.5.2) Détermination des zones de continuité

Le nombre de discontinuité correspond au nombre découpe du code, dans notre cas il vaut donc $n_D = 4$.

On note m le nombre d'entier représenté par le code Gray de n bits.

Ainsi $m = 2^N$ pour $N=14$ on obtient donc $m=16384$

Le code étant découpé en portion égale on a donc une discontinuité pour les valeurs multiples de $S = \frac{m}{n_D}$.

Les valeurs théoriques des discontinuités sont : $d_n \equiv S \times (n+1) \pmod{m}$ avec $n \in [[0, n_D-1]]$

Ainsi, pour $n \in [[0, n_D-2]]$, $d_n \equiv S \times (n+1)$ et $d_{n_D-1} = 0$

Méthode :

a) On passe en revue l'ensemble de mesure Δ_i . On rappelle que $\Delta_i = v_{0,i} - v_{1,i}$. Si une mesure est fiable il est improbable que $v_{0,i}$ et $v_{1,i}$ ne soient pas fiables. On suppose donc que si Δ_i est fiable, $v_{0,i}$ l'est aussi.

Pour la mesure fiable, on détermine le secteur sur lequel la mesure $v_{0,i}$ se trouve par $n = E\left(\frac{v_{0,i}}{S}\right)$ où E est la fonction partie entière. Ainsi $n \in [[0, n_D-1]]$

Pour chaque secteur n :

On détermine la valeur de i de la dernière mesure fiable $v_{0,i}$ se trouvant avant d_n .

On note I_n la valeur de i. $I_n = i$

De même, on détermine la valeur de i de la première mesure fiable $v_{0,i}$ se trouvant après d_n . On note F_n la valeur de i. $F_n = i$

b) On passe en revue l'ensemble des mesures Δ_i afin d'associer à chaque mesure i un numéro de secteur effectif, que la mesure soit fiable ou non fiable.

Le mesure étant effectuée dans un ordre chronologique, on peut associer un numéro de secteur pour mesurer qui n'est pas fiable. En effet, globalement, il suffit de vérifier si $I_n \leq i \leq F_n$. Il faut naturellement tenir compte aussi de la périodicité des valeurs du code Gray.

c) On passe en revue l'ensemble des mesures Δ_i dans l'ordre chronologique afin de localiser la valeur i des mesures correspondant effectivement à une discontinuité.

Si la mesure Δ_i est fiable, on cherche la valeur n telle que $v_{0,i} \leq d_n$

Afin de gérer la cyclicité du code Gray, on définit 2 variables d' et d'' de la façon suivante :

Si $n \neq m$ alors $d' = d_n$ sinon $d' = 0$

On pose $d'' = d' + S$.

Si $d' \leq v_{0,i} < d''$ et $d_n < v_{1,i}$ alors

Compte tenu du fait que pour chaque mesure i, on a à disposition la valeur Y_i et σ_i localisant la discontinuité. Afin de choisir la localisation de la discontinuité la plus proche de la ligne 240 (centre de l'image) et avec une largeur de discontinuité faible (discontinuité bien localisée), on minimise la valeur $|Y_i - 240|$ pour des mesures i où $\sigma_i < 56$. La valeur 56 correspond à 70 % de la demi-largeur de l'espace entre les 2 zones de lecture (voir §4.13 Détrompeur) $\frac{160}{2} \times \frac{70}{100} \approx 56$.

d) La mesure n°i retenue pour chaque discontinuité est marquée par une variable passant à 1 sinon cette variable a une valeur nulle.

5.5.3) Sélection des mesures

Pas d'échantillonnage :

L'effectif **n** des mesures réalisées lors d'une rotation du disque supportant le code gray peut être grand. Afin de limiter le nombre de mesure utilisées pour l'étalonnage, on réalise une sélection de mesure fiable espacée d'un pas minimal.

L'effectif **n** correspond à 360° soit 21600 minutes d'arc. Ainsi par proportionnalité on peut définir le pas d'échantillonnage de la façon suivante :

$$s = E \left(\frac{R}{60} \times \frac{1}{21600} \times 20 \times n \right) \quad \text{où } E \text{ est la fonction partie entière}$$

R est la résolution angulaire de l'encodeur donnée en seconde d'arc. Le pas minimal d'échantillonnage est donc de 20 fois la résolution angulaire de l'encodeur. La valeur minimale de **s** est imposée à 1.

On sélectionne donc environ 1 mesure sur **s** mesures pour l'étalonnage.

Sélection :

Pour la sélection on débute le passage en revue des mesures sur la mesure i correspondant à la discontinuité centrée sur 0 (Discontinuité numérotée $n_D - 1$). Pour cette mesure $v_{0,i}$ est minimal et $v_{1,i}$ est maximal. On passe en revue l'ensemble des mesures jusqu'à réaliser un tour complet :

Si la mesure a été retenue comme discontinuité alors la mesure est sélectionnée
sinon

On passe au moins 20 mesures avant de potentiellement sélectionner une mesure après la précédente

Si la mesure est fiable alors

Si cette mesure n'est pas dans les zones de discontinuité
et si cette mesure a des valeurs de graduations supérieures à celles de la mesure précédente sélectionnée alors elle est effectivement sélectionnée

dans tous les autres cas la mesure n'est pas sélectionnée.

Remarque : On note d le numéro de la discontinuité $0 \leq d < n_D$. $k(d)$ est la mesure associée à la discontinuité d .

Une mesure i est dans une zone de discontinuité dans les 2 situations suivantes :

Pour $0 \leq d < n_D - 1$: Une mesure i est dans une zone de discontinuité

$$\text{Si } v_{1,k} < v_{0,i} < v_{0,k} \quad \text{OU} \quad v_{1,k} < v_{1,i} < v_{0,k}$$

Pour $d = n_D - 1$: Une mesure i est dans une zone de discontinuité

$$\text{Si } v_{0,i} < v_{0,k} \quad \text{OU} \quad v_{0,i} > v_{1,k} \quad \text{OU} \quad v_{1,i} < v_{0,k} \quad \text{OU} \quad v_{1,i} > v_{1,k}$$

sinon la mesure n'est pas sélectionnée.

5.6) Étalonnage

On utilise les mesures sélectionnées pour l'étalonnage selon la procédure du paragraphe précédent. On note n , l'effectif de mesures sélectionnées. Ainsi, $0 \leq i < n$. Dans cette partie, pour chaque mesure i , on a à disposition :

- Les 2 valeurs des graduation du code gray de l'image $v_{0,i}$ et $v_{1,i}$
- Un temps t_i associée à la mesure (mesuré en milliseconde).
- Une variable d_i valant 0 ou 1 permettant de tester si la mesure est associée ($d_i=1$) ou non ($d_i=0$) à une discontinuité.

Les mesures sont classées dans l'ordre croissant dans valeurs $v_{0,i}$.

5.6.1) Recherche de la première et la dernière mesures au sens chronologique

On passe en revue l'ensemble des mesures : On note i_{min} l'indice de la mesure comportant une valeur t_i minimale et on note i_{max} l'indice de la mesure comportant une valeur t_i maximale

On redéfinit l'instant de mesure en prenant comme origine temporelle la première mesure d'indice i_{min} :

$$\text{On pose ainsi } T_i = t_i - t_{min} \quad \text{Ainsi } T_{min} = 0 \text{ et } T_{max} = t_{max} - t_{min}$$

5.6.2) Détermination de la vitesse de défilement des graduations

La vitesse de défilement de la bande est supposée constante. Dans le cas idéal, la vitesse de graduation devrait donc aussi être constante. Toutefois, l'impression du code Gray peut connaître des distorsions locales. Afin de déterminer la durée pour faire exactement un tour, il nous faut connaître la vitesse de défilement à proximité des premières (et donc dernières mesures en raison de la cyclicité)

On détermine la vitesse de défilement sur les 30 premières mesures, la première mesure ayant l'index i_{min} . On détermine 2 vitesses, $\bar{V}_0$ et $\bar{V}_1$ respectivement associées aux mesures $v_{0,i}$ et $v_{1,i}$

$$\text{Ainsi, } \bar{V}_0 = \frac{1}{30} \sum_{i=i_{min}}^{i_{min}+29} \frac{v_{0,i+1} - v_{0,i}}{T_{0,i+1} - T_{0,i}} \quad \text{et} \quad \bar{V}_1 = \frac{1}{30} \sum_{i=i_{min}}^{i_{min}+29} \frac{v_{1,i+1} - v_{1,i}}{T_{1,i+1} - T_{1,i}}$$

5.6.3) Association d'une valeur angulaire à chaque mesure

On pose $\delta_0 = v_{0,i_{min}} - v_{0,i_{max}}$ et $\delta_1 = v_{1,i_{min}} - v_{1,i_{max}}$, respectivement la distance entre les graduations des mesures i_{min} et i_{max} . Le temps pour parcourir δ_0 vaut donc $\frac{\delta_0}{\bar{V}_0}$. Le temps

pour parcourir δ_1 vaut $\frac{\delta_1}{\bar{V}_1}$. Dans le cas idéal les 2 durées sont égales. On définit donc le temps τ pour faire un tour complet en moyennant ces 2 valeurs et en sommant cette moyenne à T_{max} :

$$\tau = T_{max} + \frac{1}{2} \left(\frac{\delta_0}{\bar{V}_0} + \frac{\delta_1}{\bar{V}_1} \right)$$

Pour chaque mesure i , on associe une valeur angulaire θ_i en degré par :

$$\theta_i = 360 \times \frac{T_i}{\tau}$$

On réalise ainsi la base de l'étalonnage en créant une réalisation entre θ_i et $v_{0,i}$ ainsi qu'entre θ_i et $v_{1,i}$

5.6.4) Ré-échantillonnage

Dans notre cas, la résolution angulaire théorique est de l'ordre de 10 fois celle du code gray utilisé. Le code gray permet de coder m valeur entière. On ré-échantillonne afin d'associer à chaque nombre entier k de **0** à **10m-1** des valeurs angulaires notées respectivement $\alpha_{0,k}$ et $\alpha_{1,k}$ pour la mesure du haut et du bas de l'image.

Pour $i=0$, on définit G_{min} et G_{max} par $G_{min}=E(10v_{0,i})$ et $G_{max}=E(10v_{1,i})$ où E est la fonction partie entière.

Hors extrapolation, les valeurs entières $k \in [[0; G_{min}-1]]$ et $k \in [[G_{max}+1; 10m-1]]$ ne sont pas associées à des valeurs angulaires. Pour marquer cette non association, nous posons pour ces valeurs de k : $\alpha_{0,k}=-1$ et $\alpha_{1,k}=-1$

Pour les valeurs $k \in [[G_{min}; G_{max}]]$ nous calculons les valeurs $\alpha_{0,k}$ et $\alpha_{1,k}$ par interpolation à l'aide des valeurs v_0 et v_1 .

- Méthode d'interpolation:

Pour chaque valeur entière k , $k \in [[G_{min}; G_{max}]]$, on pose $g = \frac{k}{10}$. Ainsi, g est un nombre ayant au plus une décimale après la virgule.

Respectivement pour $q=0$ et pour $q=1$:

On recherche le premier i tel que $g - v_{q,i} \geq 0$

Si $i \neq 0$, on pose $j = i-1$ et si $i=0$, on pose $j=n-1$ n étant le nombre de mesures sélectionnées.

Si $q=0$ et sa mesure i est associée à une discontinuité ($d_i=1$) alors la mesure g est dans une zone de discontinuité et on ne peut lui associer une valeur angulaire. On pose donc $\alpha_{0,k}=-1$

Si $q=1$ et sa mesure j est associée à une discontinuité ($d_j=1$) alors la mesure g est dans une zone de discontinuité et on ne peut lui associer une valeur angulaire. On pose donc $\alpha_{1,k}=-1$

Pour toutes les autres situations, on pose : $\alpha_{q,k} = \frac{g - v_{q,j}}{v_{q,i} - v_{q,j}} (\theta_i - \theta_j + \Delta) + v_{q,j}$
avec $\Delta = 0$ si $\theta_i < \theta_j$
et $\Delta = 360$ dans le cas contraire.

Remarque 1 : Pour gérer la congruence à 360° des angles, on impose que si $\alpha_{q,k} > 360$ alors on retranche 360 à $\alpha_{q,k}$

Remarque 2 : On souligne le fait que les $\alpha_{q,k}$ calculés dans cette partie sont positifs ou nul.

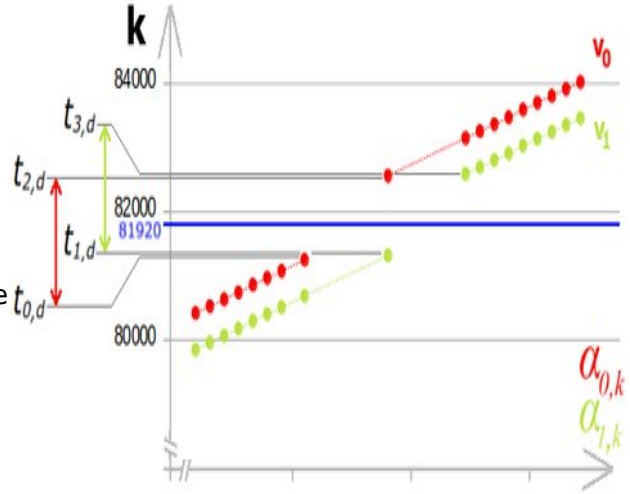
Dans les zones à proximité des discontinuités, seule une des 2 graduations (v_0 ou v_1) ne peut pas être lue en raison de la discontinuité. Ces graduations ont été exclues dans l'étape d'interpolation précédente en raison de la discontinuité des valeurs v_0 ou des valeurs v_1 . Pour ces zones il faut donc extrapoler sur les valeurs précédentes (ou suivantes pour associée une valeur α à la valeur g concernée.

- Méthode d'extrapolation:

Pour chaque discontinuité numéroté d , $0 \leq d < n_D - 1$,
on définit,

pour les graduations de la zone inférieure
de l'image,
 $t_{0,d}$ comme l'index k de la première valeur
 $\alpha_{0,k}$ négative avant la discontinuité et
 $t_{2,d}$ comme l'index k de la première valeur
 $\alpha_{0,k}$ d'indice k , positive ou nulle après
la discontinuité.

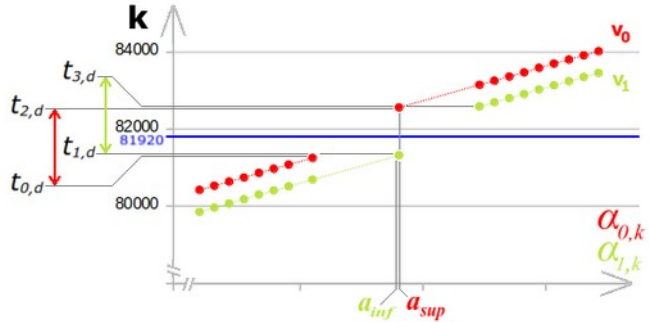
pour les graduations de la zone supérieure
de l'image,
 $t_{1,d}$ comme l'index k de la première valeur
 $\alpha_{1,k}$ négative avant la discontinuité et
 $t_{3,d}$ comme l'index k de la première valeur
 $\alpha_{1,k}$ d'indice k , positive ou nulle après
la discontinuité.



Pour chaque discontinuité numérotée d , $0 \leq d < n_D - 1$,

On définit :

et $a_{inf} = \alpha_{1,k-1}$ pour l'indice $k=t_{1,d}$
 $a_{sup} = \alpha_{0,k}$ pour l'indice $k=t_{2,d}$



• Dans un premier temps on réalise l'extrapolation sur la mesure associée à la partie inférieure de l'image (indice 1)

On détermine le pas angulaire moyen sur les 30 valeurs de α_1 précédant la mesure
d'indice $j=t_{1,d}$ par : $\bar{p}_\alpha = \frac{\alpha_{1,j-1} - \alpha_{1,j-31}}{30}$

On note δ_k la différence angulaire des angles $\alpha_{0,k}$ et $\alpha_{1,k}$ pour la mesure d'indice k .
Afin que $-180^\circ \leq \delta_k \leq 180^\circ$ on définit la valeur de δ_k par :

Si $\alpha_{0,k} - \alpha_{1,k} > 180^\circ$ alors $\delta_k = \alpha_{0,k} - \alpha_{1,k} - 360$

Si $\alpha_{0,k} - \alpha_{1,k} < -180^\circ$ alors $\delta_k = \alpha_{0,k} - \alpha_{1,k} + 360$

La différence angulaire moyenne $\bar{\delta}$ est définie sur les 30 dernières mesures

d'indice k précédant la mesure d'indice $j=t_{0,d}$ et donnée par $\bar{\delta} = \frac{1}{30} \sum_{k=j-30}^{j-1} \delta_k$

Finalement, on définit la longueur angulaire de la discontinuité à extrapoler en
amont et en aval de cette dernière par :

$$L_d = |\bar{\delta}| \frac{1 - F_d}{2} \quad F_d = 90\% \text{ voir §4.13}$$

Pour l'extrapolation des valeurs $\alpha_{1,k}$, on initialise l'indice k à $t_{1,d} - 1$

On incrémente k de 1
On pose $D = \alpha_{1,k-1} + \overline{p}_\alpha$

On calcule $\alpha_{1,k}$ par :

Si $0 \leq D < 360$ alors $\alpha_{1,k} = D$
Si $D < 0$ alors $\alpha_{1,k} = D + 360$
Si $D \geq 360$ alors $\alpha_{1,k} = D - 360$

Si $\alpha_{1,k} < a_{inf} + L_d$ alors on poursuit le processus d'extrapolation.

On note k_e l'indice de la dernière $\alpha_{1,k}$ valeur extrapolée.

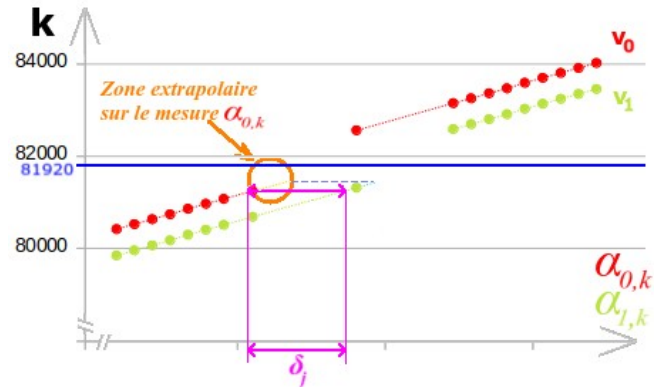
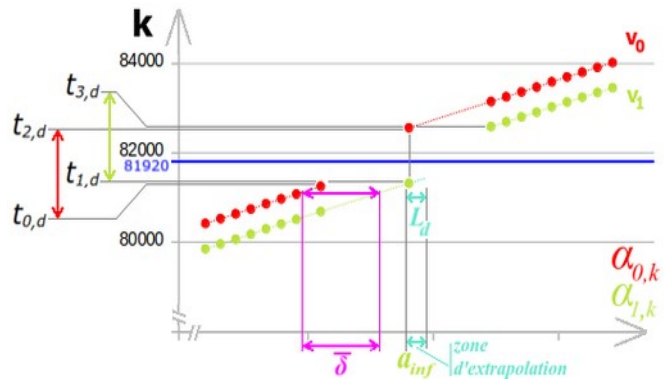
Pour $j = t_{0,d} - 1$, on calcule δ_j la différence angulaire des angles $\alpha_{0,j}$ et $\alpha_{1,j}$. Afin que $-180^\circ \leq \delta_j \leq 180^\circ$ on définit la valeur de δ_j par :

Si $\alpha_{0,j} - \alpha_{1,j} > 180^\circ$ alors $\delta_j = \alpha_{0,j} - \alpha_{1,j} - 360$

Si $\alpha_{0,j} - \alpha_{1,j} < -180^\circ$ alors $\delta_j = \alpha_{0,j} - \alpha_{1,j} + 360$

Pour chaque indice k , $t_{0,d} \leq k \leq k_e$, on pose $D = \alpha_{1,k} + \delta_k$ et on détermine la valeur de $\alpha_{0,k}$ par :

Si $0 \leq D < 360$ alors $\alpha_{0,k} = D$
Si $D < 0$ alors $\alpha_{0,k} = D + 360$
Si $D > 360$ alors $\alpha_{0,k} = D - 360$



• Dans un second temps, on réalise l'extrapolation sur le mesure associée à la partie supérieure de l'image (indice 0) .

On détermine le pas angulaire moyen sur les 30 valeurs de α_1 précédant la mesure d'indice $j = t_{2,d}$ par : $\overline{p}_\alpha = \frac{\alpha_{0,j+30} - \alpha_{0,j}}{30}$

On note δ_k la différence angulaire des angles $\alpha_{0,k}$ et $\alpha_{1,k}$ pour la mesure d'indice k . Afin que $-180^\circ \leq \delta_k \leq 180^\circ$ on définit la valeur de δ_k par :

Si $\alpha_{0,k} - \alpha_{1,k} > 180^\circ$ alors $\delta_k = \alpha_{0,k} - \alpha_{1,k} - 360$

Si $\alpha_{0,k} - \alpha_{1,k} < -180^\circ$ alors $\delta_k = \alpha_{0,k} - \alpha_{1,k} + 360$

La différence angulaire moyenne $\overline{\delta}$ est définie sur les 30 dernières mesures

d'indice k précédant la mesure d'indice $j = t_{3,d}$ et donnée par $\overline{\delta} = \frac{1}{30} \sum_{k=j}^{j+29} \delta_k$

Finalement, on définit la longueur angulaire de la discontinuité à extrapoler en amont et en aval de cette dernière par :

$$L_d = |\overline{\delta}| \frac{1 - F_d}{2} \quad F_d = 90\% \text{ voir §4.13}$$

Pour l'extrapolation des valeurs

$\alpha_{0,k}$, on initialise l'indice k à $t_{2,d}$

On décrémente k de 1

On pose $D = \alpha_{0,k+1} - \overline{p}_\alpha$

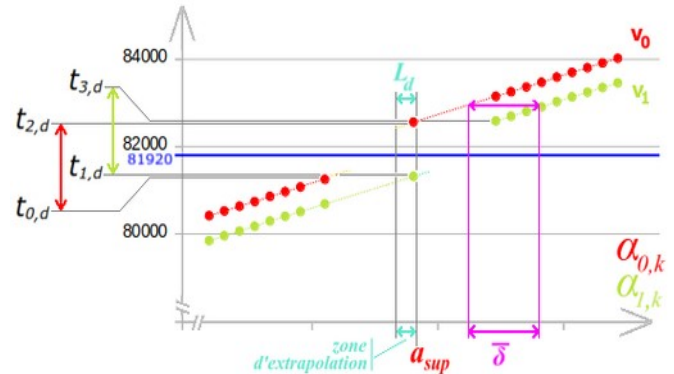
On calcule $\alpha_{0,k}$ par :

Si $0 \leq D < 360$ alors $\alpha_{0,k} = D$

Si $D < 0$ alors $\alpha_{0,k} = D + 360$

Si $D \geq 360$ alors $\alpha_{0,k} = D - 360$

Si $\alpha_{0,k} > a_{\text{sup}} - L_d$ alors on poursuit le processus d'extrapolation.



On note k_e l'indice de la dernière $\alpha_{0,k}$ valeur extrapolée.

Pour $j = t_{3,d}$, on calcule δ_j la différence angulaire des angles $\alpha_{0,j}$ et $\alpha_{1,j}$. Afin que $-180^\circ \leq \delta_j \leq 180^\circ$ on définit la valeur de δ_j par :

Si $\alpha_{0,j} - \alpha_{1,j} > 180^\circ$ alors $\delta_j = \alpha_{0,j} - \alpha_{1,j} - 360$

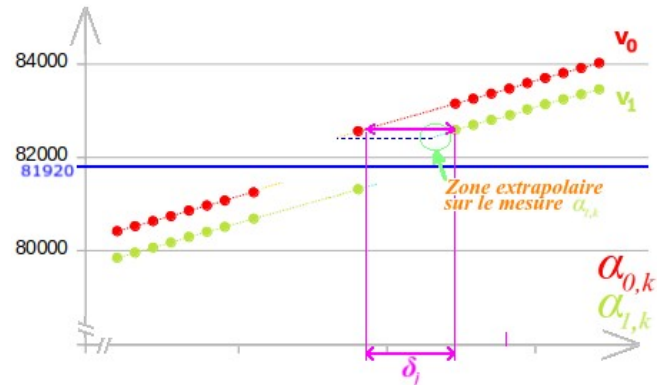
Si $\alpha_{0,j} - \alpha_{1,j} < -180^\circ$ alors $\delta_j = \alpha_{0,j} - \alpha_{1,j} + 360$

Pour chaque indice k , $k_e \leq k < t_{3,d}$, on pose $D = \alpha_{0,k} - \delta_k$ et on détermine la valeur de $\alpha_{0,k}$ par :

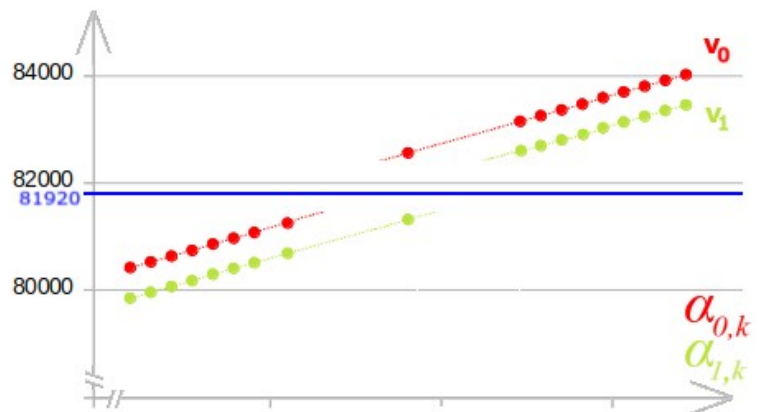
Si $0 \leq D < 360$ alors $\alpha_{0,k} = D$

Si $D < 0$ alors $\alpha_{0,k} = D + 360$

Si $D > 360$ alors $\alpha_{0,k} = D - 360$



À la fin de cette procédure d'extrapolation les valeurs $\alpha_{0,k}$ et $\alpha_{1,k}$ positives ou nulles sont telles que représentation par la figure ci-contre :



5.6.5) Détermination des intervalles des valeurs de code gray sans contre-partie angulaire

Compte tenu du montage du code Gray sur les supports mécaniques, une partie des entiers représentée par le code gray sur l'intervalle $[0 ; m[$, (avec m définie au paragraphe §5.5.2) n'est pas visible par l'encodeur. Ainsi certains entiers sont exclus du domaine de définition de la fonction permettant de passer des entiers du code Gray à leur contre partie angulaire définie sur l'intervalle $[0 ; 360^\circ[$.

Afin d'optimiser la taille du tableau permettant l'association rapide d'une valeur du code Gray à sa contrepartie angulaire, nous allons mémoriser uniquement un intervalle de valeur entière k tel que cet intervalle débute à une valeur entière k qui n'est pas associée à des mesures de $\alpha_{0,k}$ et $\alpha_{1,k}$ valant -1 . Idem pour la fin de cette intervalle.

Ainsi on défini $G'_{\min}$ et $G'_{\max}$ de la façon suivante :

$G'_{\min}$ est le premier entier naturel k tel que $\alpha_{0,k} \geq 0$ ou $\alpha_{1,k} \geq 0$

En reprenant la notation $G_{\max}$ du paragraphe §5.6.4 et m du paragraphe §5.5.2,

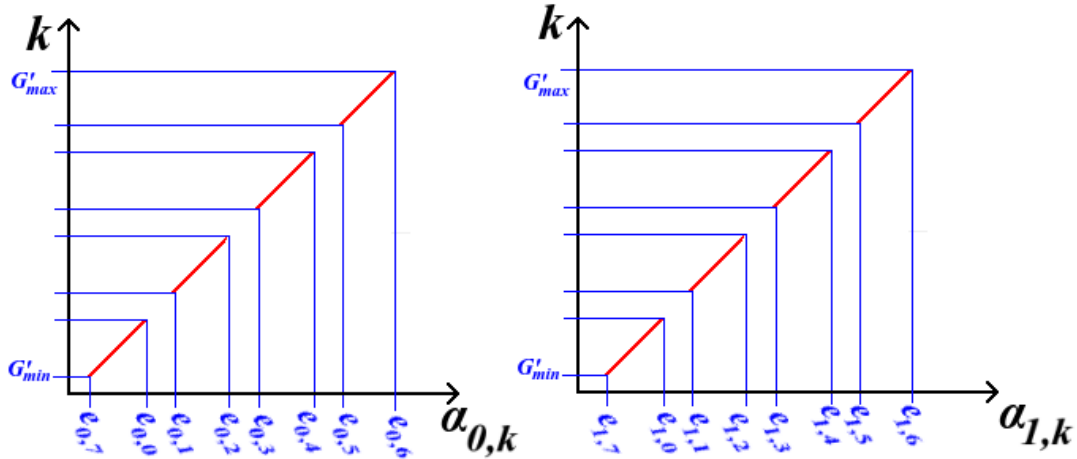
$G'_{\max}$ est le dernier entier naturel $k \in \llbracket G_{\max} ; m-1 \rrbracket$ tel que $\alpha_{0,k} \geq 0$ ou $\alpha_{1,k} \geq 0$

On rappelle que le nombre de discontinuité vaut n_D comme défini au §5.5.2

On recherche les limites angulaires de zone de discontinuité, définie pour chaque zone par un intervalle borné. Ainsi, nous avons $2 n_D$ bornes à déterminer pour les mesures de graduations associées à la partie supérieure de l'image, idem pour les mesures de graduations associées à la partie inférieure de l'image.

On note $e_{q,n}$ ces différentes valeurs avec $q=0$ pour la partie supérieure de l'image et $q=1$ pour la partie inférieure et avec $n \in \llbracket 0 ; 2 n_D - 1 \rrbracket$

L'image ci-dessous illustre la situation pour 4 discontinuités :



Les valeurs $e_{q,n}$ sont telles que :

Pour $n \in \llbracket 0 ; 2 n_D - 2 \rrbracket$ n pair on repère la borne supérieure d'un intervalle ou il n'y pas d'association entre un entier du code Gray et une mesure d'angle. De la même façon, pour n impaire, on repère la borne inférieure de chaque intervalle.

Pour $n = 2 n_D - 1$ on a $e_{0,n} = \alpha_{0,k}$ et $e_{1,n} = \alpha_{1,k}$ avec $k = G'_{\min}$

5.6.6) Détermination de l'origine des mesures et mise en forme des données

Les valeurs effectives des mesures angulaires notées $A_{0,k}$ et $A_{1,k}$ sont translatées par rapport aux valeurs $\alpha_{0,k}$ et $\alpha_{1,k}$ afin que l'origine des valeurs angulaires mesurées correspondent à celle de la discontinuité associée à la valeur zéro du code gray.

On définit ainsi une valeur d'offset Δ_α par:

$$\Delta_\alpha = \alpha_{0,k} \text{ avec } k = E\left(10v_{0,0} + \frac{1}{2}\right) \text{ où } E \text{ est la fonction partie entière.}$$

Ainsi, pour toutes les valeurs nulles ou positives des $\alpha_{q,k}$,
 $q \in \llbracket 0;1 \rrbracket$ et $k \in \llbracket G'_{\min}; G'_{\max} \rrbracket$:

$$\begin{array}{ll} \text{Si } 0 \leq \alpha_{q,k} - \Delta_\alpha < 360 & \text{alors } A_{q,k} = \alpha_{q,k} - \Delta_\alpha \\ \text{Si } \alpha_{q,k} - \Delta_\alpha \geq 360 & \text{alors } A_{q,k} = \alpha_{q,k} - \Delta_\alpha - 360 \\ \text{Si } \alpha_{q,k} - \Delta_\alpha < 0 & \text{alors } A_{q,k} = \alpha_{q,k} - \Delta_\alpha + 360 \end{array}$$

Pour les valeurs négatives des $\alpha_{q,k}$, $A_{q,k}$ prend la valeur de $A_{q,j}$ j étant le plus grand entier précédent k tel que $\alpha_{q,j}$ ne soit pas négatif.

Finalement, pour chaque valeur de $q \in \llbracket 0;1 \rrbracket$ et pour chaque valeur k $k \in \llbracket G'_{\min}; G'_{\max} \rrbracket$,
 On considère k', $k' \notin \llbracket e_{1-q,n}; e_{1-q,n+1} \rrbracket$ et $k' \notin \llbracket e_{1-q,7}; e_{1-q,0} \rrbracket$ avec $n \in \llbracket 0; 2n_D - 1 \rrbracket$, n pair
 si $k' = G'_{\max}$ alors on pose $k'' = G'_{\min}$ sinon $k'' = k + 1$
 si $\alpha_{1-q,k'} > \alpha_{1-q,k''}$ alors $\Delta = 360$ sinon $\Delta = 0$

On recherche k' tel que $\alpha_{1-q,k'} \leq \alpha_{q,k} \leq \alpha_{1-q,k''} + \Delta$ ou $\alpha_{1-q,k'} \leq \alpha_{q,k} + \Delta \leq \alpha_{1-q,k''} + \Delta$

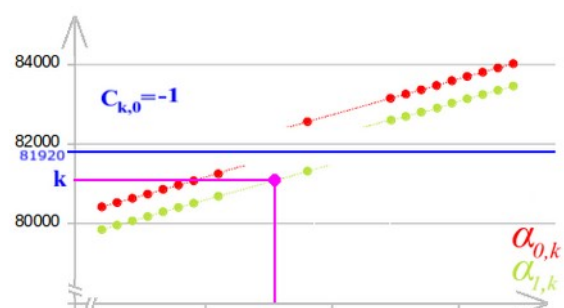
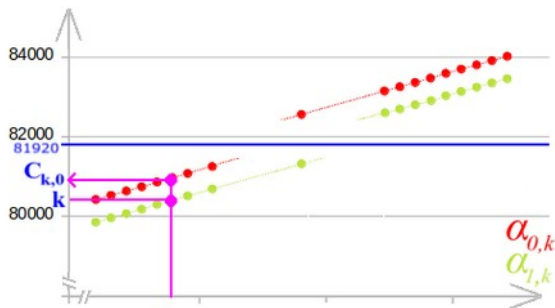
Si aucun k' ne correspond cela signifie que l'on se trouve à proximité d'une discontinuité.
 Dans ce cas, $C_{k,1-q} = -1$ afin de signifier qu'il n'y a pas de correspondance.

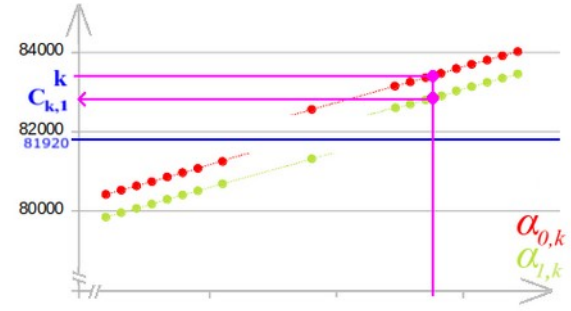
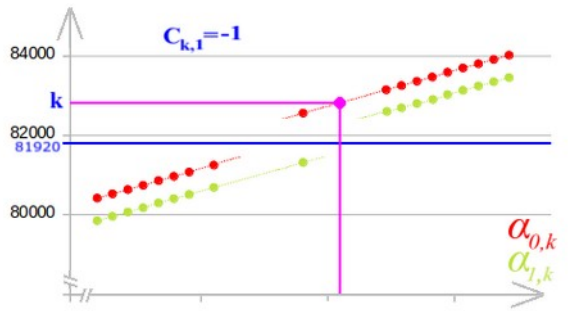
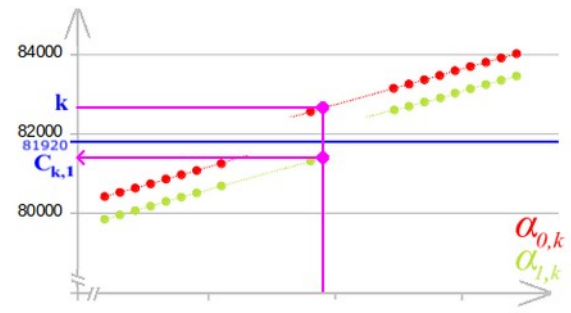
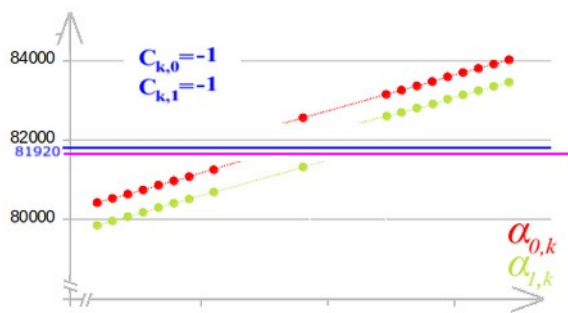
Dans le cas contraire,

$$\begin{array}{ll} \text{Si } |\alpha_{q,k} - \alpha_{1-q,k'}| < |\alpha_{q,k} - \alpha_{1-q,k''}| & \text{alors on pose } \epsilon_{k,q} = |\alpha_{q,k} - \alpha_{1-q,k'}| \text{ et } C_{k,1-q} = k' \\ \text{sinon on pose } \epsilon_{k,q} = |\alpha_{q,k} - \alpha_{1-q,k''}| & \text{et } C_{k,1-q} = k'' \end{array}$$

Pour chaque valeur k $k \in \llbracket G'_{\min}; G'_{\max} \rrbracket$, si $C_{k,0} \geq 0$ et $C_{k,1} \geq 0$ alors on pose $\epsilon_k = \frac{\epsilon_{k,0} + \epsilon_{k,1}}{2}$
 si $C_{k,0} < 0$ et $C_{k,1} \geq 0$ alors on pose $\epsilon_k = \epsilon_{k,1}$
 si $C_{k,0} \geq 0$ et $C_{k,1} < 0$ alors on pose $\epsilon_k = \epsilon_{k,0}$
 si $C_{k,0} < 0$ et $C_{k,1} < 0$ alors on pose $\epsilon_k = 0$

Les images ci-après illustrent les différentes situations en particulier à proximité d'une discontinuité.





Pour finir,

pour la mise en forme et la sauvegarde de l'étalonnage,

Pour chaque valeur $k \in \llbracket G'_{min}; G'_{max} \rrbracket$, on tabule pour chaque ligne du fichier de sauvegarde

$$k ; A_{0,k} \times \frac{\pi}{180} ; A_{1,k} \times \frac{\pi}{180} ; C_{k,1} ; C_{k,0} ; \epsilon_k \times \frac{\pi}{180}$$

VI) Effets de la dilatation

7.1) Présentation

La dilatation des supports ne peut être évitée. Le disque supportant la bande imprimée est à priori composé d'une matière différente de celle de la bande. Ceci crée une dilatation différentielle entre les deux matériaux.

Si on ne prend pas garde à l'effet de dilatation, l'étalonnage du code gray, devra être fait à chaque changement conséquent de température.

Dans cette partie nous allons voir comment limiter l'effet de la dilatation.

À noter que la dilatation modifie les longueurs mais pas les angles. Ainsi, les portions angulaires repérées par les graduations du code gray ne changent pas dans la mesure que la dilatation se produit de façon homogène sur l'ensemble du code.

7.2) Contrainte sur la fixation de la bande imprimée

Pour compenser la dilatation différentielle entre le disque et la bande imprimée, il ne faut pas coller la bande sur le disque. La localisation des jonctions créées par la colle entre les deux matériaux n'est pas maîtrisable. On ne peut garantir l'homogénéité du collage et les effets élastiques localisés du collage.

Ainsi, la bande sera montée tendue sur le disque. Le code gray sera découpé en 4 secteurs afin d'assurer une tension homogène sur chaque secteur.

De plus, si la bande imprimée se dilate plus rapidement que le disque la supportant, la bande risque de friser. À l'inverse, si le disque se dilate plus rapidement que la bande, celle dernière peut potentiellement se rompre. Toutefois, on peut considérer un support relativement élastique pour l'impression du code gray.

L'élasticité d'un matériau est proportionnelle à la force de traction appliquée dans la limite élastique.

On choisi d'imprimer le code gray sur du papier calque indéchirable en polyester. Afin d'alléger le support, le disque est réalisé en aluminium. Le code sera imprimé sur une feuille A3 en polyester afin de s'attribuer de la réserve sur la bande afin d'avoir des prises pour tendre celle-ci.

Nous avons réalisé un essai en traction afin de mesurer le module d'élasticité de Young du film polyester utilisé. Nous avons obtenu une valeur E de 2500 Mpa. On considère un coefficient de dilatation α_p de $100 \mu\text{m.m}^{-1}.\text{°C}^{-1}$ pour le polyester et α_a de $24 \mu\text{m.m}^{-1}.\text{°C}^{-1}$ soit une valeur différentielle de $76 \mu\text{m.m}^{-1}.\text{°C}^{-1}$.

Afin d'éviter que la bande ne soit détendue pour la plus grande température T_{max} , celle-ci sera montée suffisamment tendue à la température de montage T_0 .

L'allongement différentiel entre la bande et le support ΔL_d est donné par :

$$\frac{\Delta L_d}{L} = \alpha (T_{max} - T_0) \quad \begin{array}{l} \text{avec} \\ \text{et} \end{array} \quad \begin{array}{l} \alpha = \alpha_p - \alpha_a \\ L \text{ est la longueur de la bande} \end{array}$$

L'allongement ΔL_e dû à l'élasticité de la bande est donné par :

$$\frac{\Delta L_e}{L} = \frac{F}{SE} \quad \begin{array}{l} \text{avec} \\ \text{et} \end{array} \quad \begin{array}{l} F \text{ la tension de la bande} \\ S \text{ la section de la bande} \end{array}$$

Ainsi, si $\Delta L_d = \Delta L_e$ nous avons:

$$\alpha(T_{max} - t_0) = \frac{F}{SE}$$

$$\text{d'où } T_{max} = T_0 + \frac{F}{\alpha SE}$$

Exemple:

	Dilatation ($\mu\text{m}/\text{m}/^\circ\text{C}$)	Module de Young (Mpa)
Polyester	100	2500
Aluminium	24	69000
Différence	76	

Température de montage	20	$^\circ\text{C}$
Tension de montage	16	N
Largeur du ruban	15	mm
Épaisseur du ruban	80	μm
Section du ruban	1,20E-006	m^2
$T_{max_Théorique}$	90,18	$^\circ\text{C}$

Ainsi, pour un température de montage de 20°C , la bande doit être tendue à 16N pour qu'elle puisse rester tendue jusqu'à 90°C .

Remarque: On retrouve ce tableau de calcul dans le fichier « calculs_encodeur_polyester.xls »

7.3) Effet sur la lecture du code

Le paragraphe précédent garanti la tension du support du code. Toutefois cette tension n'est que longitudinale. Ainsi, du point de vue de la webcam, la dilatation suivant la circonférence du disque n'est pas la même que suivant la direction perpendiculaire.

La lecture du code repose sur la connaissance du rapport e_x et e_y défini au §4.2.

e_x va subir la dilatation du au disque d'aluminium alors que e_y va suivre la dilatation du au film en polyester.

$$\text{Or } \frac{e_x}{e_y} \frac{e_y + \Delta e_y}{e_x + \Delta e_x} = \frac{1 + \alpha_p \Delta T}{1 + \alpha_a \Delta T}$$

Soit avec $\Delta T = 30^\circ$ $\alpha_p = 100 \mu\text{m}.\text{m}^{-1}.\text{C}^{-1}$ et $\alpha_a = 24 \mu\text{m}.\text{m}^{-1}.\text{C}^{-1}$

On obtient $\frac{e_x}{e_y} \frac{e_y + \Delta e_y}{e_x + \Delta e_x} = 1,0022$ soit 0,22% dévolution du rapport

La lecture d'une valeur du code gray est effectuée en observant une zone de 128 pixels, e_x ayant un ordre de grandeur de 4 pixels, on peut observer 32 fois l'échelle de e_x sur cette zone .

Or $(1,0022 - 1) \times 128 \approx 0,3$ pixel

Une erreur de lecture de 0,3 pixel est de l'ordre de 7% d'une graduation soit un ordre de grandeur en-dessous de l'échelle e_x . Cet effet dû à la dilatation ne sera donc pas problématique.

VII) Effet de la vitesse de rotation

La vitesse de rotation du code gray devant l'objectif de la webcam, compte tenu de la vitesse d'acquisition d'une image, est équivalent à l'application d'un filtre faisant disparaître la basse fréquence. Ainsi, on perd la pertinence de la valeur lue par webcam sur les bits de poids faibles, mais l'algorithme présenté précédemment arrivera à lire le bit de poids forts, donnant donc une valeur moins précise mais cohérente de la position angulaire.

Ceci n'est pas gênant car pour un déplacement rapide nous pouvons nous contenter d'une précision sur le positionnement moindre.

À mesure que l'on se rapproche de la position désirée la vitesse de déplacement diminue et donc la précision de la lecture s'améliore.

Ainsi, l'effet d'une vitesse de rotation importante n'est pas problématique.

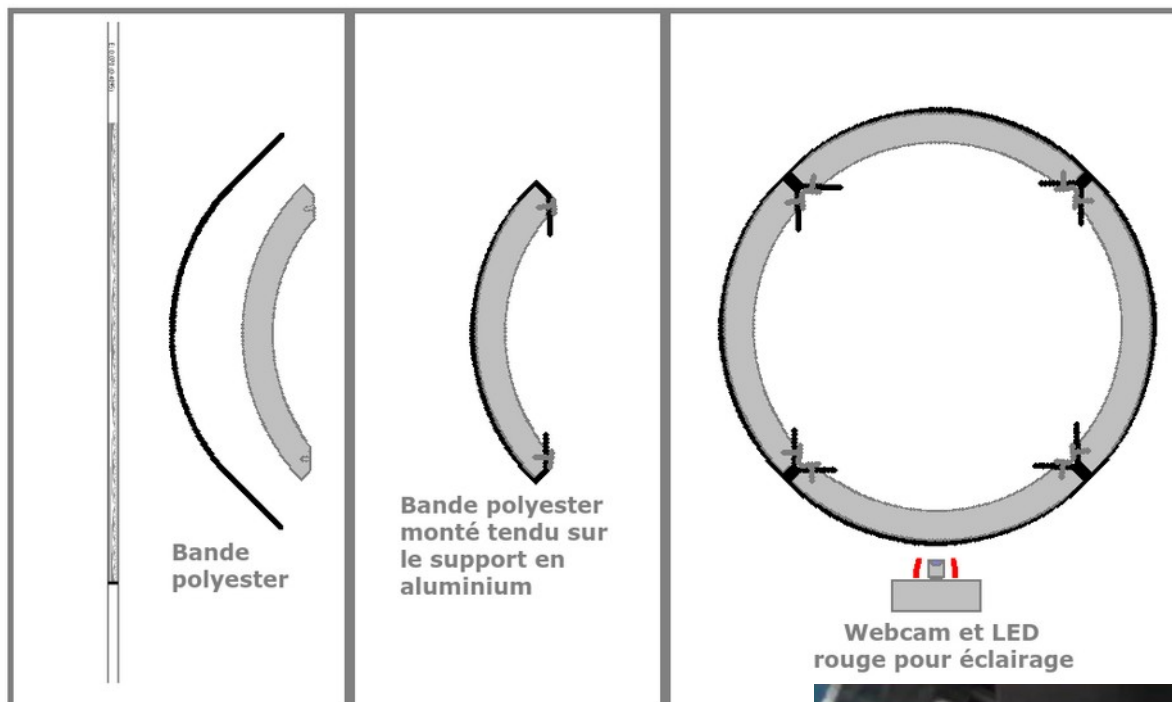
Finalement, pour une application astronomique, la vitesse de rotation de l'axe d'ascension droite est suffisamment faible pour ne pas être perceptible pendant l'acquisition d'une image, il n'y aura donc pas de perte de résolution à cette vitesse.

VIII) Support et montage mécanique

Les supports mécaniques sont réalisés en aluminium en 4 quarts de support circulaire.

Le code Gray est imprimé sur feuille polyester au format A3 et monté tendu à 16N à 20°

La webcam filme le code Gray afin que la largeur de la bande occupe la totalité du champ de la caméra. La webcam est équipée d'un éclairage rouge réalisé par 2 LED permettant d'éclairer discrètement le code Gray pendant son utilisation nocturne.



IX) Hardware

Dans le cadre d'une utilisation de plusieurs webcams à la fois, il faut préciser un point.

Pour une application en astronomie, nous avons besoin de mesurer la position angulaire de deux axes: l'axe horaire et l'axe des déclinaisons. Il nous faut donc deux webcams.

Usuellement, en USB1, une webcam occupe 70% des capacités d'un contrôleur USB. Il faut donc un contrôleur USB par webcam, quel que soit le système d'exploitation (Windows, Linux etc...)

Un contrôleur USB peut gérer plusieurs ports USB. Typiquement, pour une carte USB il y a un contrôleur. Sur certaine carte mère, il peut y avoir plusieurs contrôleurs USB mais ce n'est pas toujours le cas.

Ainsi, pour brancher une seconde webcam, il se peut qu'il faille acheter un carte USB supplémentaire bien qu'il y ait des ports USB encore disponibles sur la machine.

La problématique peut être différente suivant la norme USB (1, 2 ou 3...)