

DESCRIPTION DES LOGICIELS IMPLANTES DANS LE PROCESSEUR PMM

Note n°1

Serge BEUCHER

Décembre 1993

Introduction

Cette note est la première d'une série de trois (au minimum). L'ensemble décrit les principaux modules de traitement d'images implantés dans le processeur de Morphologie Mathématique implanté dans le démonstrateur Prolab2. Ces descriptifs sont destinés à faciliter le portage des algorithmes. Les trois principales fonctions portées sur PMM sont:

- La segmentation de la chaussée de type I, c'est-à-dire sans connaissance a priori.
- La segmentation de type II, dans laquelle un modèle de route est injecté afin d'accélérer la segmentation.
- La détection des obstacles présents sur la chaussée.

Cette première note est consacrée à la description de la segmentation de type II. La segmentation de type I a déjà fait l'objet par ailleurs de descriptions partielles. Par contre, l'injection d'un modèle de route, bien qu'ayant été utilisé par X. YU, n'a pas fait l'objet d'une description rigoureuse. De plus, l'algorithme en question n'est pas tout à fait exact. Il suppose en particulier que la position du point de fuite est fixe, ce qui est tout à fait inexact. C'est pourquoi, on décrira dans cette note de façon très détaillée les algorithmes de traitement d'images permettant la segmentation mais aussi les calculs définissant le modèle de route injecté. On verra en particulier qu'il n'est pas nécessaire de disposer d'informations de calibration complexes et qu'une calibration statique très simple permet d'obtenir toutes les informations requises pour la détermination des distances au sol et donc des paramètres collectés par le traitement (position du véhicule, distance des obstacles, etc...).

La segmentation de type II a été implantée sur MorphoPérior. La description de l'algorithme se fera donc principalement à partir du listing commenté du programme TIM.

I. Principe de la segmentation de type II

Dans ce type de segmentation, on est sensé disposer à l'instant n du modèle de voie calculé à l'instant n-1. Ce modèle de voie stocké dans une mémoire binaire sert à générer le (ou les) marqueur(s) pour la segmentation à l'instant n. Cette segmentation effectuée par LPE sur une image (gradient + chapeau haut-de-forme + fonction distance) anamorphosée permet alors l'extraction du bassin versant correspondant à la voie empruntée. La sélection des contours fiables de cette voie (c'est-à-dire ceux qui passent par le marquage au sol) et la séparation des parties droite et gauche dans l'avant-scène conduisent alors au calcul du nouveau modèle de voie. Ce modèle de voie est un modèle en deux parties qui permet de mieux prendre en compte d'éventuelles courbures de la voie.

II. L'algorithme en détails

Nous allons décrire plus en détail chaque étape de l'algorithme jusqu'à l'extraction des contours de la voie à l'instant n. On décrira dans le paragraphe suivant les étapes du calcul du modèle de voie.

L'algorithme décrit ici est celui qui est implanté actuellement sur MorphoPérior. Certaines modifications pourront et devront certainement y être apportées lors du transfert sur le processeur PMM.



Figure 1: Image initiale

II.1. Les étapes du traitement

- Un chapeau haut-de-forme de taille SIZE1 est effectué sur l'image initiale IDN (image à l'instant n).

SIZE1 @ IDN @ WKM WTOPHAT
WKM WKL NMOVE



Figure 2: Chapeau Haut-de-Forme

Ce chapeau haut-de-forme permet d'extraire le marquage au sol. La taille SIZE1 est définie une fois pour toutes en fonction de l'épaisseur du marquage au sol et de la calibration statique (voir plus loin). Le résultat est placé dans l'image de travail WKM (Figure 2).

- Calcul d'une fonction distance à partir de l'image précédente. L'algorithme consiste à itérer la transformée:

$$\sup(\sup((WKL \oplus H) - 1, 0), WKM) \rightarrow WKL$$

WKL est initialisé avec l'image WKM. Ce processus a pour but d'assurer une certaine connexion du marquage au sol lorsqu'il est discontinu et donc de mieux positionner la LPE. Le nombre d'itérations est donné par la variable SIZE2. Ce nombre est également fixé une fois pour toutes et dépend de la distance entre les lignes discontinues (Figure 3).

Remarque: cette étape du traitement est relativement longue et pourrait être allégée voire supprimée si on utilise l'image obtenue par filtre dynamique temporel.

```
SIZE2 @ 1 DO
  WKL WKL NDILATE
  CLEAR_CARRY
  WKL WKL DECR
  OD1 @ SAVE_CARRY
  OD1 @ %INV LOAD_TEMPLATE
  0 WKL NSET
  WKM WKL WKL SUP
LOOP
```

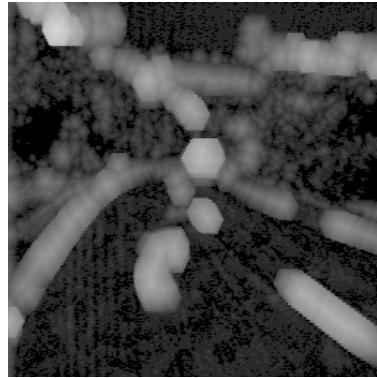


Figure 3: Fonction distance injectée

- Calcul du gradient de l'image de départ et stockage dans WKN (Figure 4). Si le marquage au sol est faible ou inexistant, les contours de la route sont néanmoins détectés. De plus, si des obstacles sont présents, ils sont pris en compte.

```
IDN @ WKN GRADIENT
```



Figure 4: Gradient de l'image

- Sup. des images WKN et WKL. Le résultat contenu dans WKN sera l'image utilisée par la LPE (Figure 5).

```
WKL WKN WKN SUP
```

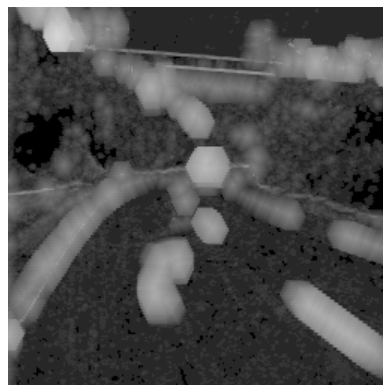


Figure 5: Image initiale pour la LPE

- Seuil automatique de l'image Chapeau haut-de-forme.

L'algorithme utilisé n'est pas entièrement finalisé. Le but de ce seuillage étant d'obtenir une image binaire exempte de bruit, la détermination du seuil par calcul de

l'histogramme des niveaux de gris des maxima de l'image chapeau haut-de-forme semble relativement robuste (Figure 6).

WKM OD2 @ AUTOTHRESH



Figure 6: Seuillage de l'image Chapeau Haut-de-Forme

- Calcul de la LPE

Le calcul de la LPE commence par la mise en évidence des marqueurs. Le marqueur est l'intersection de l'érodé du marqueur généré par le modèle de route calculé à partir de l'image précédente (stocké dans ID) et d'un seuil de l'image WKN (Figure 7). La taille de l'érosion est SIZE2 et la valeur de seuil est la même que celle fournie par le seuillage automatique.

ID @ WG INVERT
 SIZE2 WG WG XERODE
 0 THRVAL @ WKN OD1 @ THRESH
 OD1 @ WG %AND WG MOVE



Figure 7: Marqueurs utilisés par la LPE

Ces transformations ont pour but de positionner au mieux le nouveau marqueur sur la portion de chaussée libre devant le véhicule.

La LPE est alors calculée sur l'image anamorphosée. Une anamorphose Log_2 est utilisée (Figure 8).

WKN WG GWTS @ MOD_BWSHED

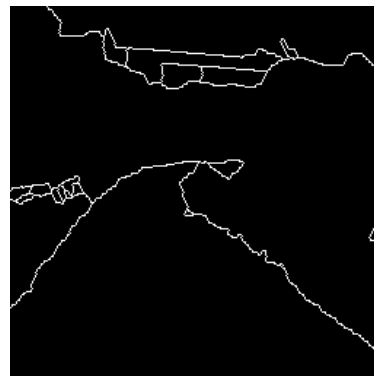


Figure 8: LPE de l'image

Le résultat est stocké dans l'image binaire GWTS.

- Récupération du bassin versant correspondant à la voie

Le bassin versant correspondant à la voie où se trouve le véhicule est extrait par reconstruction à partir d'un marqueur stocké dans MARK (marqueur toujours positionné en bas et au milieu de l'image) et son contour est stocké dans OD3 (Figure 9).

```
GWTS @ OD3 @ INVERT
MARK @ OD1 @ MOVE
OD3 @ OD1 @ BUILD
OD1 @ OD1 @ DILATE
OD1 @ OD3 @ %DIFF OD3 @ MOVE
```



Figure 9: Le bassin versant de la voie

- Extraction des points sûrs du contour

Ce sont les points qui appartiennent également au marquage au sol (image OD2).

```
OD2 @ OD3 @ %AND WK MOVE
3 1 WK OD1 @ LINOPEN
WK OD1 @ %DIFF WK MOVE
```

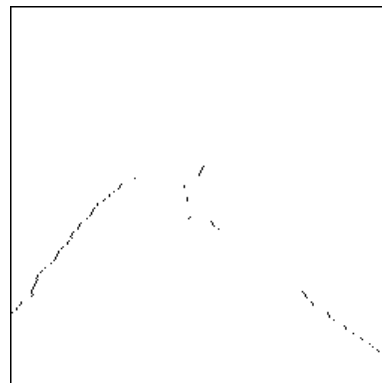


Figure 10: Les points sûrs du contour

L'ouverture linéaire sert à éliminer les portions horizontales du contour. Ces portions horizontales sont dues le plus souvent à des artefacts de construction de la LPE (Figure 10).

- Séparation des frontières situées à droite et à gauche

Cette procédure permet de stocker dans deux images séparées (WG et WD) les points sûrs des bords gauche et droite de la voie (Figure 11).

```
OD3 @ ASMASK @ %AND
OD1 @ MOVE
OD1 @ WG BUILD_FIRST
OD1 @ WG %DIFF WD MOVE
WK WG %AND WG MOVE
WK WD %AND WD MOVE
```

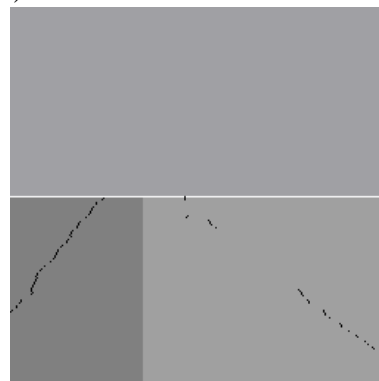


Figure 11: Séparation des points à droite et à gauche

L'image ASMASK contient le masque d'avant-scène (c'est-à-dire la région constituée par la demi-image inférieure).

A la fin de cette procédure, le nombre de points extraits à gauche et à droite est calculé. Si ce nombre est inférieur à une valeur limite (10 sur les images 256 x 256), le marquage au sol est considéré comme non significatif et un message d'erreur est envoyé. Dans la version finale du logiciel, ce message pourra être remplacé par une commutation vers l'algorithme de type I.

II.2. Les routines spécialisées

Les routines spécialisées sont au nombre de deux: le seuillage automatique et la LPE avec anamorphose. La première routine n'a pas été installée dans la version du logiciel présente sur le MorphoPérior. Néanmoins, on peut donner ici des précisions sur leur implantation.

II.2.1. Le seuillage automatique

Ce seuillage est destiné à fournir une image binaire du marquage au sol aussi exempte de bruit que possible. Ce seuillage est effectué sur l'image Chapeau haut-de-forme. Cette image est constituée principalement de pics de différentes hauteurs. Les pics les plus faibles mais également les plus nombreux correspondent au bruit. Les pics d'altitude plus élevée correspondent généralement aux objets significatifs. La valeur de seuil optimale sera donc une valeur située au-dessus de la hauteur médiane des pics. On peut estimer cette hauteur à partir de l'histogramme des valeurs de gris des maxima de l'image Chapeau haut-de-forme. On peut prendre également le mode de cet histogramme, puisque les pics dus au bruit sont les plus nombreux. L'algorithme est le suivant:

- Extraction des maxima de l'image Chapeau haut-de-forme
- Etiquetage avec leurs niveaux de gris (par masquage de l'image Chapeau haut-de-forme)
- Calcul de l'histogramme
- Extraction de la valeur de gris modale
- Seuillage à une valeur double (ou correspondant à la valeur modale + l'écart-type).

II.2.2. La LPE avec anamorphose

Cet algorithme consiste à calculer la LPE non pas sur l'image de départ, mais sur une image anamorphosée. L'anamorphose choisie est une anamorphose logarithmique base 2. Cela permet de ramener le nombre de niveaux de gris de 256 à 8. En fait, on peut même se contenter de 6 niveaux, voire 5 sans que la dépréciation soit importante. En effet, l'image sur laquelle on calcule la LPE est une image gradient + Chapeau haut-de-forme. Cette image n'est pas très informative lorsque les niveaux dépassent 32. L'algorithme de LPE utilisé est alors l'algorithme "antique" d'inondation niveau par niveau à l'aide d'un SKIZ géodésique. Le listing du programme sur MorphoPérior est le suivant:

```
: MOD_BWSHED
  OUTPUT OD
  INPUT ID
  NINPUT IDN
  WORK WK
  DINTEGER FIELD
  INTEGER J
  S_WINDOW? 2DROP M* -> FIELD
  ID OD MOVE
```

```

1 -> J
8 1 DO
    0 J 1- IDN WK THRESH
    WK AREA FIELD D= IF R> DROP 8 >R THEN
    WK ID %OR WK MOVE
    OD WK OD GDSKIZ
    J 2* -> J
LOOP
OD OD INVERT ;

```

II.3. L'optimisation des vitesses

Augmenter la vitesse des opérations est un souci primordial, même si cela doit se faire au détriment des performances. Parmi les actions, citons:

- La diminution du nombre de niveaux de l'image anamorphosée utilisée par la LPE, comme déjà indiqué plus haut. Rappelons que la plupart des tests algorithmiques réalisés dans ce projet l'ont été sur des images à 64 niveaux de gris. Cela entraîne que 5 niveaux devraient suffire pour l'image anamorphosée. Remarquons également qu'il n'est pas nécessaire de calculer réellement l'image anamorphosée pour réaliser l'opération. Des seuillages judicieux de l'image de départ suffisent comme le montre le listing de l'algorithme utilisé sur le MorphoPérior.
- L'utilisation de l'image produite par filtrage dynamique temporel permet d'éviter l'étape de connexion du marquage discontinu. Ce filtre dynamique temporel est malheureusement long dans son implantation originale. On peut cependant le simplifier et se contenter de dilatations temporelles plutôt que de filtres alternés séquentiels.
- Le seuillage automatique utilise la dynamique des maxima de l'image Chapeau haut-de-forme. On a vu que cette dynamique correspond en fait à leur hauteur. Cependant le calcul des maxima d'une image n'est pas toujours rapide. On peut là encore simplifier l'algorithme en détectant non pas les maxima mais tous les plateaux à bord descendant de l'image Chapeau haut-de-forme. En effet, étant donné la nature de cette image, il y a fort à parier que les plateaux qui ne seront pas aussi des maxima seront peu représentés. Ces plateaux se mettent en évidence de façon quasi immédiate par fléchage, une transformation réalisable par le processeur PIMM1.

III. Le calcul du modèle de route

L'étape suivante du traitement consiste à calculer le modèle de voie et à l'injecter à nouveau dans l'image ID1 pour qu'il serve de marqueur pour la segmentation suivante (Figure 12). Le modèle de voie est un modèle composé de deux tronçons. Le premier tronçon est calculé à partir des points du marquage au sol situés à l'avant-scène. Les points à droite et à gauche inclus respectivement dans WD et WG permettent de calculer par interpolation linéaire deux droites. Les segments correspondants appartenant à l'avant-scène sont les frontières de la voie ainsi modélisée. L'intersection des deux droites fournit le point de fuite correspondant ainsi que la ligne d'horizon. Le deuxième tronçon est alors modélisé en estimant à partir des points sûrs de la voie situés en arrière-plan les droites correspondantes. On notera qu'il n'est pas nécessaire d'estimer les deux droites. Une seule suffit à partir du moment où on connaît la ligne d'horizon. On choisit généralement d'estimer la droite dont la pente est la plus faible. Dans le cas où le nombre de points à l'arrière-plan est insuffisant, le

tronçon en arrière-plan est dans le prolongement du tronçon en avant-scène. Les schémas suivants donnent les détails du calcul.

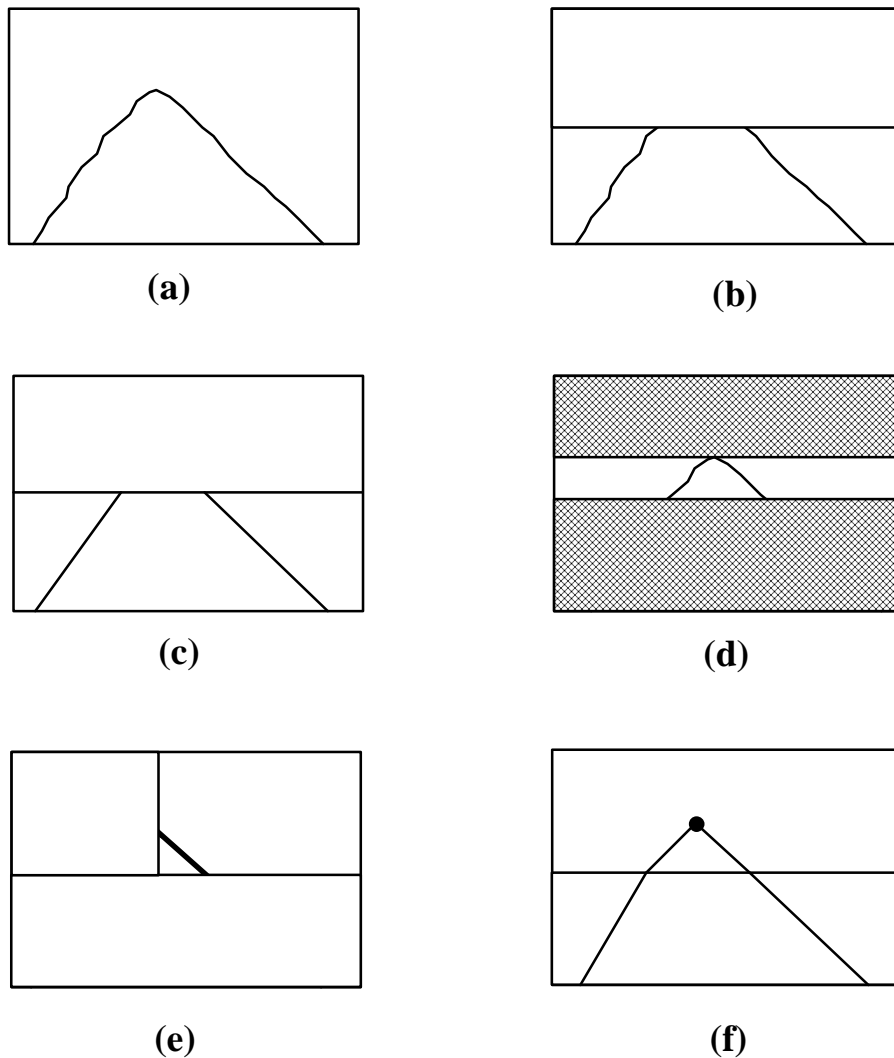


Figure 12: Les étapes du calcul du modèle de voie

- (a) contour initial de la voie
- (b) portion de contour situé à l'avant-scène
- (c) interpolation des droites
- (d) sélection du contour en arrière-plan
- (e) interpolation de la première droite d'arrière-plan
- (f) modèle de voie final

III.1. La régression linéaire

Rappelons les équations permettant le calcul des coefficients d'une droite $y = a_e x + b_e$ à partir d'un ensemble de couples de points (x_i, y_i) .

En posant:

$$\sum_{i=1}^N x_i = S_1 \quad ; \quad \sum_{i=1}^N y_i = S_2 \quad ; \quad \sum_{i=1}^N x_i^2 = S_3 \quad ; \quad \sum_{i=1}^N y_i^2 = S_4 \quad ; \quad \sum_{i=1}^N x_i y_i = S_5$$

on obtient:

$$a_e = \frac{(NS_5 - S_1 S_2)}{NS_3 - S_1^2}$$

$$b_e = \frac{S_2 S_3 - S_1 S_5}{NS_3 - S_1^2}$$

N étant le nombre de points.

Le coefficient de corrélation est donné par:

$$r = \frac{(NS_5 - S_1 S_2)}{(NS_3 - S_1^2)^{\frac{1}{2}} (NS_4 - S_2^2)^{\frac{1}{2}}}$$

III.2. Les voies adjacentes

A partir du moment où on dispose des équations des droites projetées du modèle de voie, on peut tracer facilement la bordure des voies adjacentes (à gauche ou à droite) de la voie interpolée et ainsi générer les marqueurs de ces voies adjacentes (bien que cela ne soit pas obligatoire).

Supposons que la voie adjacente a une largeur L' pas forcément égale à L, la largeur de la voie interpolée (cette voie peut être la bande d'arrêt d'urgence).

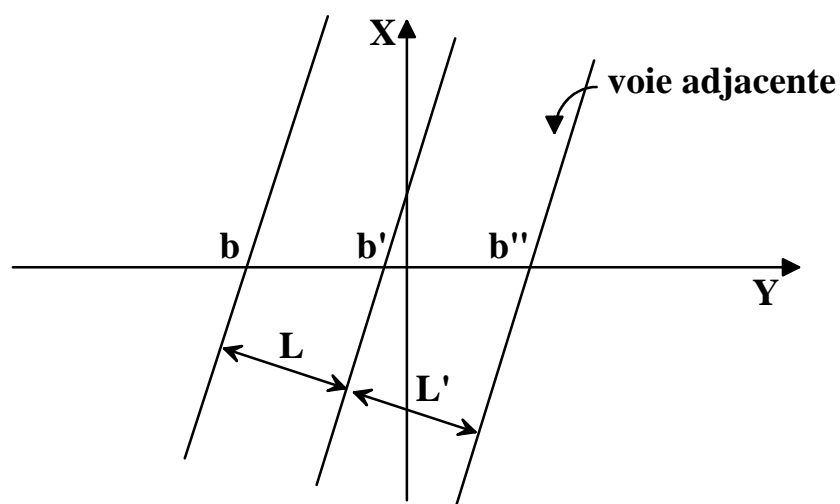


Figure 13: Voies adjacentes (monde réel)

Les voies projetées coupent l'axe X_e en trois points x_e , x'_e et x''_e .

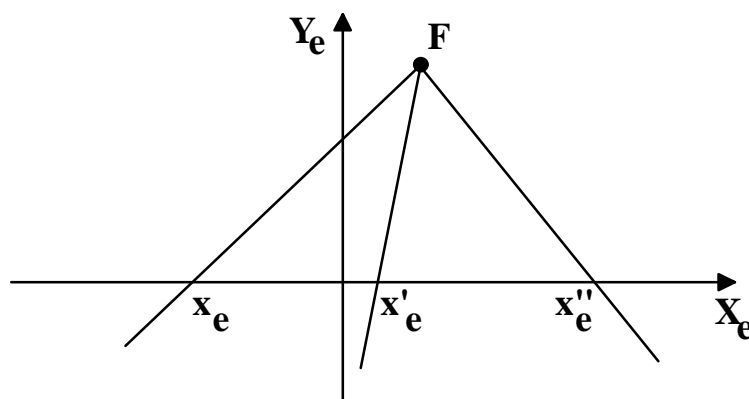


Figure 14: Voies adjacentes en perspective

On a:

$$\frac{b''-b'}{b'-b} = \frac{L'}{L} ; x_e = \frac{bD}{R} ; x'_e = \frac{b'D}{R} ; x''_e = \frac{b''D}{R}$$

soit:

$$x''_e - x'_e = \frac{L'}{L}(x'_e - x_e)$$

La droite projetée passe par le point de fuite et le point d'abscisse x''_e sur l'axe des X_e .

IV. Les équations de perspective

La segmentation de la chaussée a pour but essentiel de définir où se trouve dans l'image les informations et les événements importants. Elle permet également de calculer à partir du modèle de route les distances et positions réelles des obstacles et du véhicule par rapport à la chaussée, moyennant une calibration préalable. Nous allons détailler ci-dessous l'ensemble des équations qui permettent de passer des coordonnées dans l'image aux coordonnées dans le monde réel. On verra également que la calibration du système se résume à peu de choses.

IV.1. Les systèmes de référence

IV.1.1. Le référentiel route

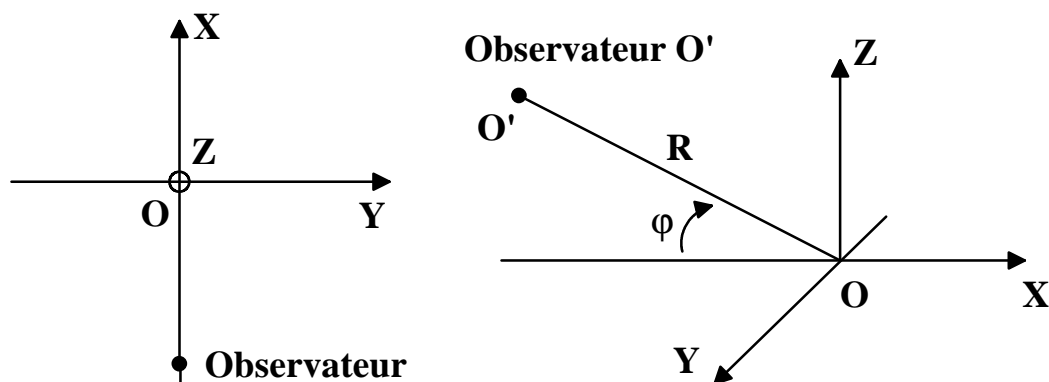


Figure 15: Le référentiel route

Le référentiel route (monde réel) est illustré ci-dessus. L'observateur est à la distance R de l'origine et regarde la scène sous un angle φ (les coordonnées polaires de l'observateur sont donc $[R, 0, \varphi]$). Ce système de référence est donc lié au véhicule se déplaçant sur la chaussée.

IV.1.2. Le référentiel écran

L'image est supposée se former sur un écran (fictif) situé à la distance D de l'observateur et perpendiculaire à l'axe OO' . Les coordonnées de l'image sont indiquées à la figure 16.

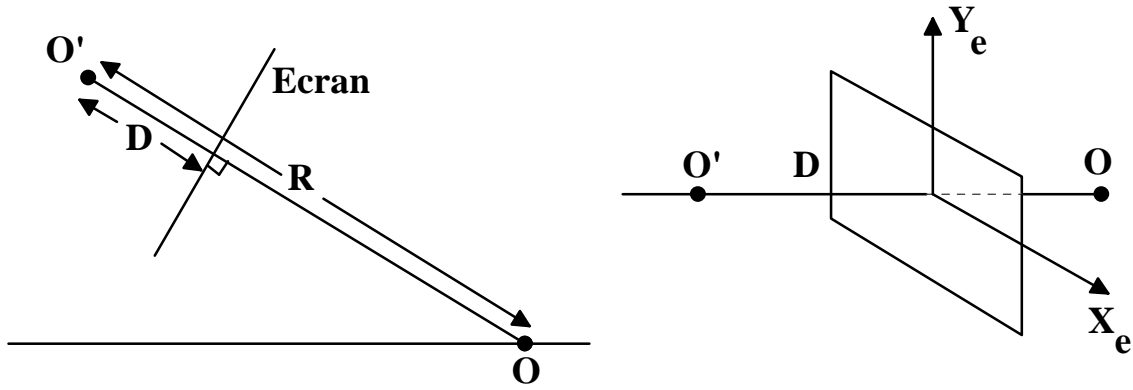


Figure 16: Le référentiel écran

IV.2. Calcul des coordonnées écran d'un point M sur la route

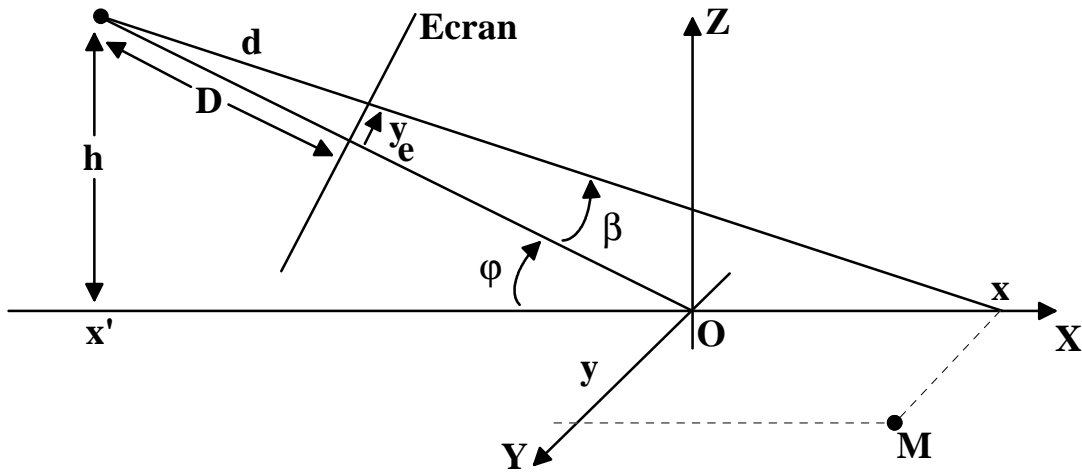


Figure 17: Projection d'un point de la route sur l'image

La route est supposée plane et incluse dans le plan XOY. Un point $M(x, y)$ sur la route se projettera en un point $m(x_e, y_e)$ sur l'écran. On a:

$$y_e = D \tan \beta$$

$$\frac{x-x'}{h} = \tan\left(\frac{\pi}{2} - \varphi + \beta\right) = \frac{1}{\tan(\varphi-\beta)} = \frac{x+R \cos \varphi}{h} = \frac{x+R \cos \varphi}{R \sin \varphi}$$

d'où:

$$x = \frac{R \tan \beta}{\cos \varphi (\tan \varphi - \tan \beta)} = \frac{R y_e}{D \cos \varphi (\tan \varphi - \frac{y_e}{D})}$$

et:

$$\boxed{x = \frac{R y_e}{D \sin \varphi - y_e \cos \varphi}} \quad (1)$$

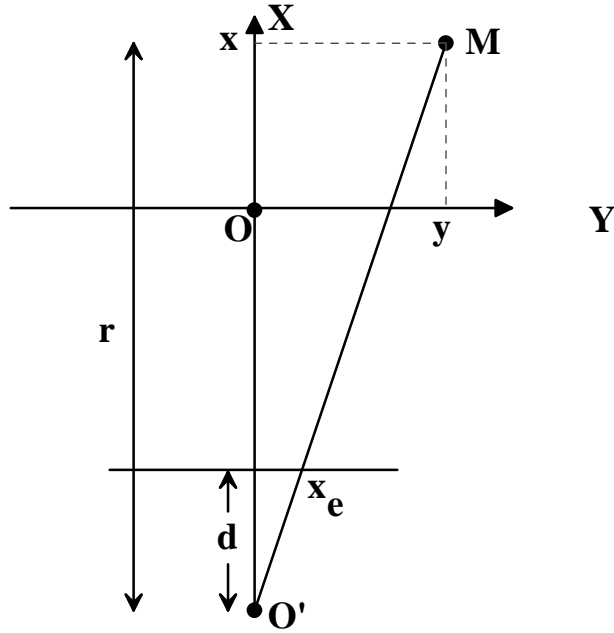


Figure 18: Projection d'un point sur l'image (suite)

De la même façon:

$$\frac{y}{x_e} = \frac{r}{d} \quad ; \quad \frac{D}{d} = \cos \beta \quad ; \quad \frac{x-x'}{r} = \sin\left(\frac{\pi}{2} - \varphi + \beta\right) \quad ; \quad \frac{x-x'}{r} = \cos(\varphi - \beta)$$

$$\frac{y}{x_e} = \frac{r \cos \beta}{D} = \frac{(x-x') \cos \beta}{D \cos(\varphi - \beta)} = \frac{(x+R \cos \varphi) \cos \beta}{D \cos(\varphi - \beta)}$$

$$\frac{y}{x_e} = \frac{x+R \cos \varphi}{D(\cos \varphi + \frac{y_e}{D} \sin \varphi)}$$

$$\frac{y}{x_e} = \frac{x+R \cos \varphi}{D \cos \varphi + y_e \sin \varphi} = \frac{R y_e + R \cos \varphi (D \sin \varphi - y_e \cos \varphi)}{(D \cos \varphi + y_e \sin \varphi)(D \sin \varphi - y_e \cos \varphi)}$$

$$\frac{y}{x_e} = \frac{R y_e \sin^2 \varphi + R D \cos \varphi \sin \varphi}{(D \cos \varphi + y_e \sin \varphi)(D \sin \varphi - y_e \cos \varphi)} = \frac{R \sin \varphi (y_e \sin \varphi + D \cos \varphi)}{(D \cos \varphi + y_e \sin \varphi)(D \sin \varphi - y_e \cos \varphi)}$$

$$\frac{y}{x_e} = \frac{R \sin \varphi}{D \sin \varphi - y_e \cos \varphi}$$

soit:

$$y = \frac{x_e R \sin \varphi}{D \sin \varphi - y_e \cos \varphi} \quad (2)$$

On peut inverser les équations (1) et (2) pour obtenir:

$$\boxed{\begin{aligned} x_e &= \frac{y D}{R + x \cos \varphi} \\ y_e &= \frac{x D \sin \varphi}{R + x \cos \varphi} \end{aligned}} \quad (3)$$

IV.3. Résultats corollaires et remarques

- Lorsque le plan de la route contient une droite d'équation $y = ax + b$, cette droite apparaît sur l'écran comme une droite d'équation $y_e = a_e x_e + b_e$, avec:

$$a_e = \frac{R \sin \varphi}{aR - b \cos \varphi}$$

$$b_e = \frac{bD \sin \varphi}{b \cos \varphi - aR}$$

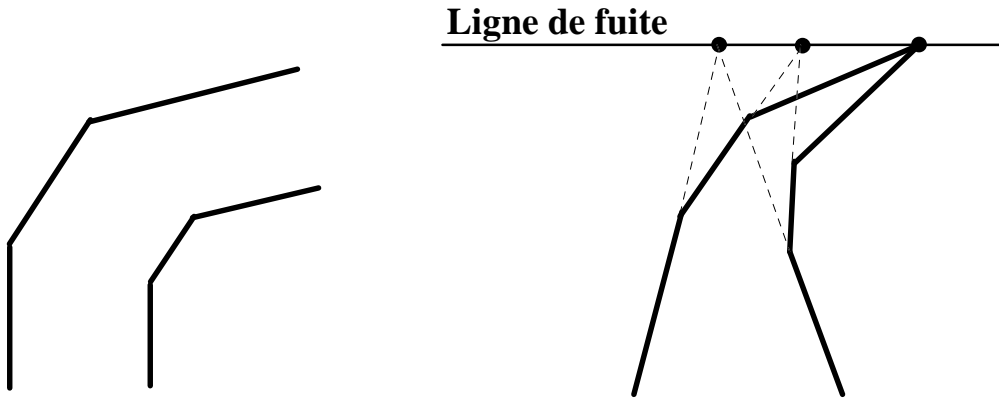


Figure 19: Route en plusieurs tronçons et ligne de fuite

- Un faisceau de droites parallèles dans le plan de la route d'équations générales $y_i = ax + b_i$ produit un faisceau de droites concourantes en un même point F de coordonnées:

$$x_F = \frac{aD}{\cos \varphi}$$

$$y_F = D \tan \varphi$$

Ce point F est le point de fuite. On remarquera que sa coordonnée y_F est une constante qui ne dépend pas de l'équation initiale de la droite. Ainsi, si plusieurs faisceaux existent dans le plan de la route, ils s'intersecteront suivant des points de fuite différents mais tous situés sur la même ligne, la ligne d'horizon (Figure 19). Cette intéressante caractéristique a été utilisée pour la modélisation de la voie en deux tronçons rectilignes (cf. supra).

- Les paramètres R et D ne correspondent pas aux distances réelles. D en particulier n'est pas égal à la focale de l'objectif. Ces paramètres sont des paramètres correcteurs d'échelles. Ils dépendent notamment des échelles et des unités respectives des coordonnées route et écran.

- De manière plus générale, si nous traçons dans le plan route une courbe algébrique polynomiale d'ordre n d'équation:

$$f(x, y) = \sum_{k+l \leq n} a_{kl} x^k y^l = 0$$

cette courbe se projette sur l'écran en une courbe algébrique d'ordre au plus égal à n. En effet, on peut écrire:

$$f_e(x_e, y_e) = \sum_{k+l \leq n} a_{kl} R^{l+k} \sin^l \varphi x_e^k y_e^l (D \sin \varphi - y_e \cos \varphi)^{n-l-k}$$

Pour chaque terme de cette somme, la dimension en x_e est au plus égale à 1 et celle en y au plus égale à $k + n - 1 = n - 1$. La dimension du polynôme est donc au plus n .

L'interpolation de la route par un polynôme de degré supérieur à un n n'a pas été effectuée dans cette implantation de l'algorithme.

IV.4. Recherche des solutions dans le cas de faisceaux de droites

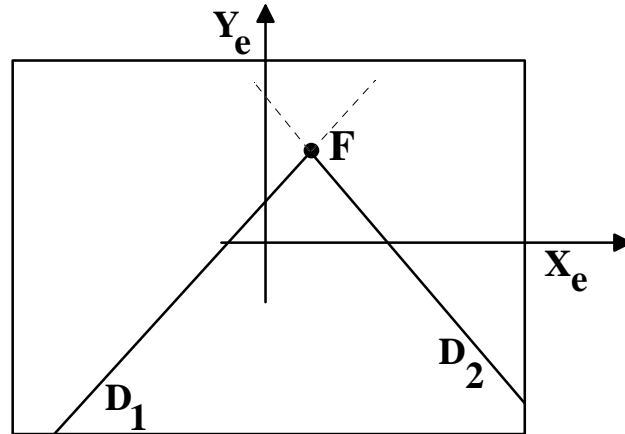


Figure 20: Droites interpolées et point de fuite

Les données de départ sont les deux droites obtenues par régression d'équation:

$$y_e = a_e x_e + b_e$$

$$y_e = a'_e x_e + b'_e$$

Ces deux droites sont supposées être les projections perspective de deux droites parallèles d'équations:

$$y = ax + b$$

$$y' = ax + b'$$

de pente a commune.

Les valeurs connues étant a_e , a'_e , b_e et b'_e , calculons a et b :

$$a = \frac{D \sin \varphi - b_e \cos \varphi}{a_e D}$$

$$b = -\frac{b_e R}{a_e D}$$

Sans connaissance des paramètres R , D et φ , on ne peut pas remonter aux équations 3D de la chaussée. nous avons en effet six inconnues (a , b , b' , R , D et φ) et seulement quatre paramètres. Si on veut résoudre le système, il faut ajouter deux paramètres supplémentaires. On peut le faire en effectuant une calibration du dispositif au repos.

IV.4.1. Calibration statique

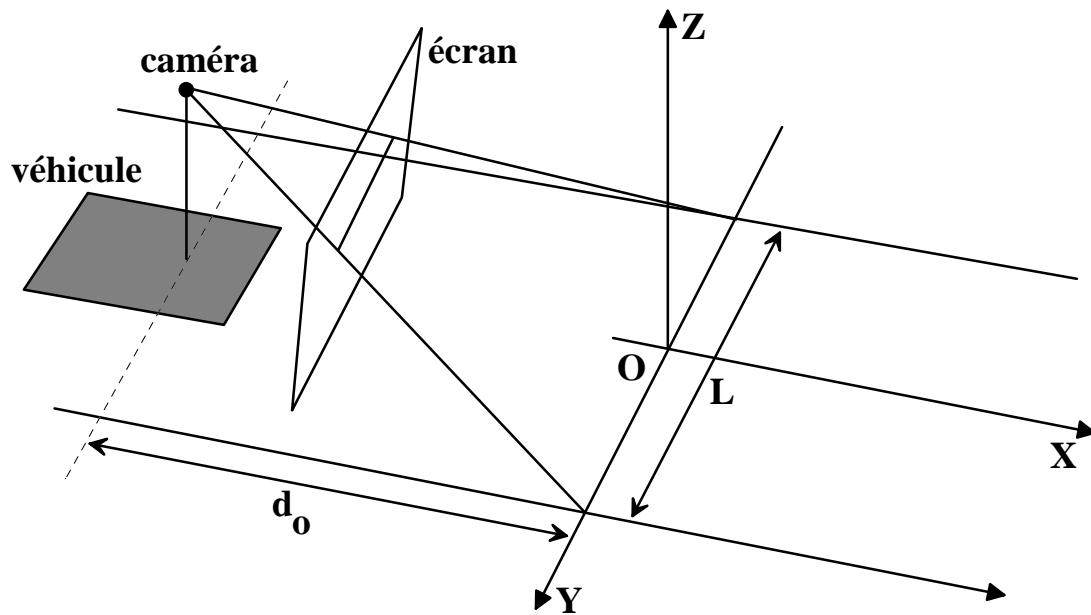


Figure 21: Dispositif pour la calibration statique

Supposons le véhicule au repos (Figure 21). Nous allons introduire deux paramètres supplémentaires: la largeur L de la chaussée et la distance au sol d_0 correspondant à l'écart entre la caméra (l'observateur) et l'origine O du repère $OXYZ$. Ces valeurs peuvent être mesurées facilement une fois pour toutes.

- Calcul de d_0

d_0 peut être mesuré directement. Mais on peut aussi le calculer de la façon suivante si on connaît la distance entre deux points sur la route (c'est par exemple le cas pour le marquage au sol qui est normalisé).

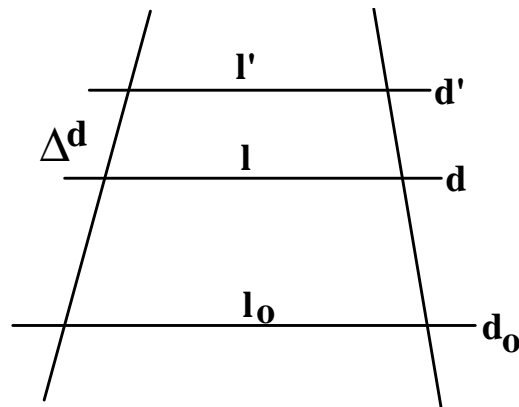


Figure 22: Distances apparentes

Il suffit alors d'écrire que:

$$d_0 = \frac{\Delta d l'}{l_0(l-l')}$$

Rappelons que l_0 est la largeur apparente de la chaussée au centre de l'image.

- Calcul des autres paramètres

Lorsque le véhicule est parallèle à la chaussée, on a:

$$a_e = -\frac{R_0}{b} \tan \varphi$$

R_0 désignant la valeur statique de R .

D'autre part:

$$|b' - b| = L = R_0 \tan \varphi_0 \frac{(a_e - a'_e)}{a'_e a_e}$$

avec L , largeur de la chaussée, φ_0 valeur statique de φ .

D'où:

$$R_0 = \frac{L a'_e a_e}{\tan \varphi_0 (a_e - a'_e)}$$

D'autre part:

$$\tan \varphi_0 = \frac{(R_0 - d_0)^{\frac{1}{2}}}{d_0}$$

En éliminant $\tan \varphi_0$ entre les deux équations, on trouve:

$$R = \left[\frac{d_0^2 + \sqrt{\Delta}}{2} \right]^{\frac{1}{2}}$$

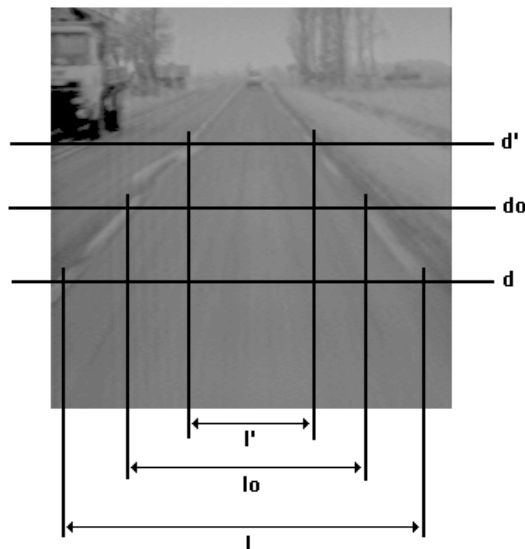


Figure 23: Exemple de calibration

avec $\Delta = d_0^4 + \frac{4L^2 d_0^2 a'_e{}^2 a_e^2}{(a_e - a'_e)^2}$

Pour l'exemple précédent:

$a_e = 1,1135$; $a'_e = -1,224$; $L = 3,5$ m ; $\Delta d = 13$ m (séparation entre lignes de rive)

Ce qui donne:

$$d_0 = 10,31$$

$$R_0 = 10,5$$

On en déduit immédiatement l'inclinaison de la caméra (pour cette image):

$$\varphi = 2,42^\circ$$

Enfin, on obtient D par la formule suivante:

$$D = \frac{y_F R_0 (a_e - a'_e)}{L a_e a'_e}$$

Le véhicule est parallèle à la route et y_F étant l'ordonnée du point de fuite.

Pour notre exemple, $y_F = -89$, ce qui donne $D = 453,38$.

Ce paramètre ne dépend pas en fait de la position de la caméra. On peut donc considérer que ce paramètre est une constante du système de visualisation qui ne variera pas lorsque la caméra sera amenée à bouger pendant le déplacement du véhicule.

On pourra donc estimer les paramètres R et φ en fonctionnement dynamique à partir des ordonnées (x_F , y_F) du point de fuite de chaque image. En effet:

$$\tan \varphi = \frac{y_F}{D}$$

$$R = \frac{L D a_e a'_e}{(a'_e b_e - a_e b'_e)} \sqrt{\frac{D^2 + y_F^2 + x_F^2}{D^2 + y_F^2}}$$

De la même façon, la distance d d'un point de la chaussée de coordonnées (x_e , y_e) est donnée par:

$$d = R \sin \varphi \frac{(D \cos \varphi - y_e \sin \varphi)}{(D \sin \varphi - y_e \cos \varphi)}$$

avec:

$$\sin \varphi = \frac{y_F}{(D^2 + y_F^2)^{\frac{1}{2}}}; \quad \cos \varphi = \frac{D}{(D^2 + y_F^2)^{\frac{1}{2}}}$$

ce qui donne:

$$d = \frac{R y_F (D^2 - y_e y_F)}{D (D^2 + y_F^2)^{\frac{1}{2}} (y_F - y_e)}$$

Dans l'exemple ci-dessus (Figure 23), le véhicule visible au fond de l'image a sa coordonnée y_e égale à -77. Il se trouve donc à la distance d égale à 77 m approximativement.

Conclusion

Récapitulons, en guise de conclusion, l'ensemble des images qui doivent être conservées tout le long du déroulement de cette tâche de fond. Ces images sont les suivantes:

- L'image courante stockée dans la mémoire IDN. Cette image peut être le résultat du filtrage dynamique temporel.
- Le marqueur précédent dans la mémoire binaire ID.
- Le marqueur calculé à l'instant n, stocké dans la mémoire OD1.
- Le seuil de l'image Chapeau Haut-de-Forme contenu dans la mémoire OD2.
- La LPE de la voie dans la mémoire OD3.
- La LPE générale située dans la mémoire GWTS.

A ces images s'ajoutent le masque de l'avant-plan dans la mémoire ASMASK ainsi que le marqueur de la voie dans la mémoire ROADMARK.

Les différentes procédures écrites sur le MorphoPéricolor requièrent de plus un certain nombre de mémoires de travail qui, du fait que cette tâche est une tâche de fond risquent d'être allouées en permanence. Ces mémoires sont au nombre de quatre pour les mémoires

numériques et de quatre pour les mémoires binaires. L'occupation mémoire totale est alors de cinq pour les images numériques et de neuf pour les images binaires.