

Analyse de la vidéo

Chapitre 1 - Représentation d'une vidéo numérique

Dernière rév.:
5 janvier 2015

Plan de présentation

1 Caractéristiques de la vidéo

- Introduction à la vidéo
- Balayage entrelacé et progressif
- Désentrelacement de la vidéo

2 Représentation de la logique temporelle

- Domaine fréquentiel
- Mouvement 2D et 3D
- Modèle de projection
- Mouvement apparent

Vidéo numérique

On caractérise donc la vidéo numérique par les paramètres suivants :¹

- f_t : Fréquence d'image (*frames per seconds (fps)*);
- f_y : Lignes par frame ;
- f_x : Pixels par lignes ;
- N_b : **Bits à encoder.**

Ce dernier paramètre encode la valeur d'intensité (couleur) d'un pixel.

1. Une vidéo numérique peut être obtenue par la discrétisation et l'encodage de la vidéo analogique ou en utilisant directement une série d'acquisition numérique (capteur CCD)

Encodage

Le flux de données dans la vidéo numérique est alors défini par :

$$\mathfrak{F} = N_b \cdot f_y \cdot f_x \cdot f_t \quad (\text{bits/s}) \quad (1)$$

N_b Nb de **Bits de l'encodage** par pixels

f_x Nb de **Pixels** par ligne

f_y Nb de **Lignes** par image

f_t Nb d'**Images** par seconde

Encodage

Bits de l'encodage

On encode chacun des canaux avec un nombre de bits nécessaire pour représenter des valeurs d'intensité.

Dans le standard RGB, on encode généralement de façon égale, ce qui donnera, pour une image couleur, $N_b = 24$ bits à utiliser (8 par canaux).

Pour le standard YUV, plus près du système visuel humain, on encode selon un critère 4 :1 :1 ($N_b = 6$) ou 4 :1 :0 ($N_b = 5$), par exemple.



Codage YUV 4:4:4



Codage YUV 4:2:0



Codage YUV 4:1:1

Vidéo analogique

La vidéo analogique ne permet pas l'encodage de la vidéo comme la vidéo numérique. C'est un flux continu de données analogiques, destiné à être affiché sur un écran de télévision (basé sur le principe du **balayage**).

- Écran cathodique : balayage point par point, suivant des lignes.
- Résolution en *dpi* (dot per inch) (VGA, S-VGA,...)..
- **Fréquence verticale** (30 à 60 Hz) : Indique le taux de rafraîchissement d'une ligne.
- **Fréquence image** : Indique le taux de rafraîchissement d'affichage d'une image.

Vidéo analogique : Norme

Les principales normes de la vidéo analogique, régies selon les capacités du système électrique et de bande passante du pays :

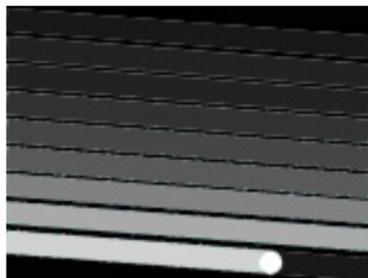
- **PAL (*Phase Alternating Line*)** : Code les vidéos sur 625 lignes (576 seulement sont affichées, car 8% des lignes servent à la synchronisation).
25 frames par seconde à un format 4 3 ;
- **NTSC (*National Television Standards Committee*)** : Code les vidéos sur 525 lignes (480 seulement sont affichées, car 8% des lignes servent à la synchronisation).
30 frames par seconde à un format 4 3 ;

Ces standards utilisent le **balayage entrelacé** pour doubler la fréquence d'image et réduire la bande passante.

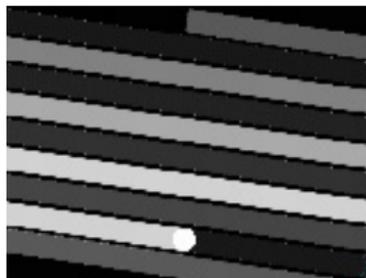
Type de balayage

Le balayage : Une image est reconstituée au moyen d'un faisceau décrivant une suite simple de lignes, qui commence en haut et à gauche de l'écran pour s'achever en bas et à droite avant de revenir à son point de départ.

- **Le balayage entrelacé** : Nombre de lignes impair où le faisceau est deux fois plus rapide ;
- **Le balayage progressif** : Nombre de lignes pair ou impair.



Progressif



Entrelacé

Balayage entrelacé

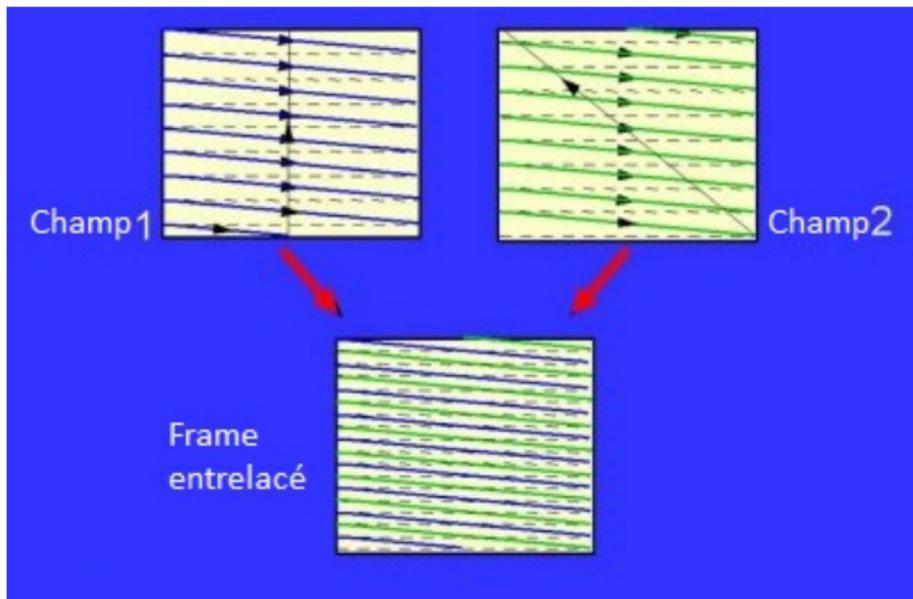
En effectuant un balayage entrelacé, on peut réduire de moitié le nombre de lignes d'une **image** pour les séparer en deux **champs**.

- **Image** : Image complète à afficher.
- **Champs** : Agencement de lignes d'un image.

Si Δt est le temps pour balayer un image, alors le temps de balayage d'un champ est $\Delta t/2$.

Balayage entrelacé

En affichant alternativement un champs et l'autre, on augmente la fréquence d'image !



Avantages du balayage entrelacé

Avantages :

- **Fréquence doublée** : À un instant donné, le système n'affiche qu'un champ plutôt qu'un image complet. Ex. : Notre 576 lignes à 30 Hz est en fait du 288 lignes à 60 Hz.
- **Réduction du scintillement** : Une fréquence trop basse donne un effet de scintillement dans les grandes zones à illumination élevée et avec beaucoup de mouvements.
- Occupe la moitié de la bande passante.

Inconvénients du balayage entrelacé

Inconvénients :

- **Artefacts d'entrelacement** : La fréquence d'image n'étant pas la même que la fréquence verticale, des **artefacts d'entrelacement** peuvent apparaître, surtout dans une scène mouvementée **latéralement**.
- **La complexité** : La compression de donnée s'avère plus complexe, puisque la vidéo se retrouve sur plusieurs champs au lieu d'un seul. Si on visionne un film et qu'on effectue un retour en arrière, on doit soit n'afficher qu'une image sur deux, soit utiliser un filtre de désentrelacement.

Artefact d'entrelacement



Champs comprenant les lignes paires



Champs comprenant les lignes impaires



Image affichée

1. Image sur <http://www.100fps.com/>

Artefact d'entrelacement



Champs comprenant les lignes paires



Champs comprenant les lignes impaires



Image affichée

1. Image sur <http://www.100fps.com/>

Artefact d'entrelacement

Vu l'effet de mélange de deux champs acquis à différents moments, il est impossible de désentrelacer la vidéo **ET** :

- de garder une fréquence d'image constante **en plus**
- de conserver la résolution d'image.

Excepté dans les scènes sans aucun mouvement, il faut altérer un de ces deux points.

Désentrelacement de la vidéo

Les méthodes de désentrelacement que nous allons explorer :

- 1 Ne rien faire (*Weave*) ;
- 2 Moyennage (*Blending, average, combine fields*) ;
- 3 Moyennage local (*Area based blending*) ;
- 4 Rejet (*Discard*) ;
- 5 Redimension (*Resizing*) ;
- 6 Bob (*Line-doubling, Low-quality progressive Scan*) ;
- 7 Adaptatif (*Progressive Scan, Bob+Weave, hybrid*).

Désentrelacement de la vidéo

Ne rien faire (*Weave*)

Avantages :

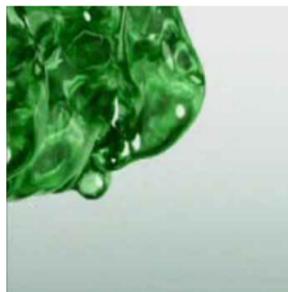
- Conserve la fréquence d'image ;
- Conserve la résolution de l'image.

Désavantage :

- On ne désentrelace pas la vidéo → Effet de peigne dans les scènes dynamiques !

Désentrelacement de la vidéo

Moyennage



Champs 1 et 2 redimensionnés verticalement

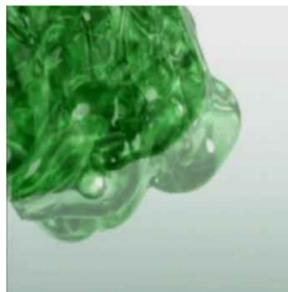
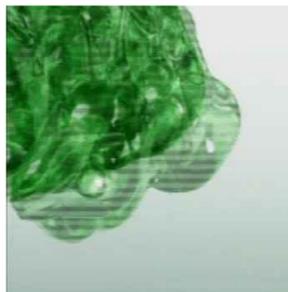


Image affichée avant et après désentrelacement

Désentrelacement de la vidéo

Algorithme du moyennage

Entrées: Champs 1, Champs 2

Sorties: Image

Redimension de Champs 1 de (L, H) à $(L, 2H)$;

Redimension de Champs 2 de (L, H) à $(L, 2H)$;

pour *Tous les pixels (x, y) de Champs 1 et Champs 2* **faire**

$$Image(x, y) \leftarrow \frac{Champs1(x, y) + Champs2(x, y)}{2};$$

fin

Désentrelacement de la vidéo

Moyennage

Avantages :

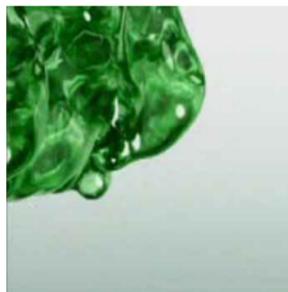
- Enlève l'effet de peigne.

Désavantage :

- Réduction de la résolution verticale ;
- Réduction de moitié de la fréquence d'image ;
- Ajout d'un effet "fantôme" global important.

Désentrelacement de la vidéo

Moyennage localisé



Champs 1 et 2 redimensionnés verticalement

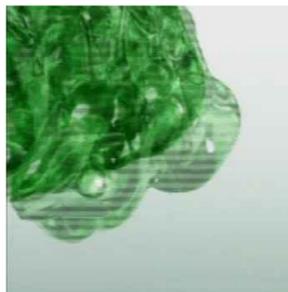


Image affichée avant et après désentrelacement

Désentrelacement de la vidéo

Algorithme du moyennage localisé

Entrées: Image combinant les champs 1 et 2

Sorties: Image moyenné localement

Détection de lignes entrelacées^a;

pour *Tous les pixels (x, y) des lignes entrelacés* **faire**

 | $Image(x, y) \leftarrow \frac{Image(x, y-1) + Image(x, y+1)}{2};$

fin

a. L'algorithme de détection de lignes entrelacées est globalement basé sur la différence entre les images et la différence entre les champs. En utilisant un certain seuil, on peut contrôler la sensibilité du détecteur. Le détecteur, si trop sensible, peut détecter des bords comme étant des lignes entrelacées.

Désentrelacement de la vidéo

Moyennage localisé

Avantages :

- Retire l'effet de peigne.
- Réduction de la résolution verticale **seulement** dans les zones de mouvement ;

Désavantage :

- Réduction de moitié de la fréquence d'image ;
- Ajout d'un effet "fantôme" local aux zones de mouvement.

Désentrelacement de la vidéo

Algorithme du rejet

Entrées: Champs 1 ou 2

Sorties: Image

Image \leftarrow Champs 1 ou 2 redimensionné de (L, H) à $(L, 2H)$.

Désentrelacement de la vidéo

Rejet

Avantages :

- Enlève l'effet de peigne.

Désavantage :

- Réduction de moitié de la résolution d'image ;
- Réduction de moitié de la fréquence d'image ;
- Perte importante d'information du mouvement.

Désentrelacement de la vidéo

Algorithme de redimension

Entrées: Champs 1 et 2

Sorties: Image 1 et 2

Image 1 \leftarrow Champs 1 redimensionné de (L, H) à $(L/2, H)$;

Image 2 \leftarrow Champs 2 redimensionné de (L, H) à $(L/2, H)$;

Désentrelacement de la vidéo

Redimensionnement

Avantages :

- Enlève l'effet de peigne ;
- Pas de création de nouvelle information ;
- Ne réduit pas la fréquence d'image.

Désavantage :

- Réduction du quart de la résolution d'image.

Désentrelacement de la vidéo

Algorithme de Bob

Entrées: Champs 1 et 2

Sorties: Image 1 et 2

pour *Toutes les lignes* $x \in [0 \dots H]$ *de Champs 1 et Champs 2 faire*

$Image1(2x) \leftarrow Champ\ 1(x);$

$Image1(2x + 1) \leftarrow Champ\ 1(x);$

$Image2(2x) \leftarrow Champ\ 2(x);$

$Image2(2x + 1) \leftarrow Champ2(x);$

fin



Désentrelacement de la vidéo

Bob

Avantages :

- Enlève l'effet de peigne ;
- Ne réduit pas la fréquence d'image.

Désavantage :

- Réduction de moitié de la résolution d'image ;
- Introduction d'artefact de Bob (effet d'aliasing) ;

Désentrelacement de la vidéo

Adaptatif (*Bob + Weave*)

On combine les champs en mélangeant Bob et Weave : on double le image combinant les deux champs, mais aux endroits où l'on détecte des artefacts de désentrelacement, on utilise la ligne du dessus pour contrer l'effet de peigne :

Entrées: Champs 1 et 2

Sorties: Image 1 et 2

Image Weave \leftarrow Weave(Champs 1 et Champs 2);

Image 1 \leftarrow Bob(Champs 1);

Image 2 \leftarrow Bob(Champs 1);

Détection des lignes x du Image Weave ayant un effet de peigne;

pour *Tous les lignes $y \neq x$ n'ayant pas un effet de peigne* **faire**

 Image1(x) \leftarrow ImageWeave(x);

 Image2(x) \leftarrow ImageWeave(x);

fin

Désentrelacement de la vidéo

Adaptatif (*Bob + Weave*)



Bob et Weave séparément



Adaptif (Bob + Weave)

Désentrelacement de la vidéo

Adaptatif (*Bob + Weave*)

Avantages :

- Enlève l'effet de peigne (dépend de l'algorithme de détection de peigne) ;
- Ne réduit pas la fréquence d'image ;
- Ne réduit pas la résolution de l'image dans les zones statiques.

Désavantage :

- Introduction (réduite) d'artefact de Bob (effet d'aliasing).

Plan de présentation

1 Caractéristiques de la vidéo

- Introduction à la vidéo
- Balayage entrelacé et progressif
- Désentrelacement de la vidéo

2 Représentation de la logique temporelle

- Domaine fréquentiel
- Mouvement 2D et 3D
- Modèle de projection
- Mouvement apparent

Rappel : Fourier

On peut voir la vidéo comme étant une image variant avec le temps. C'est en quelque sorte une 3e dimension que l'on ajoute à l'image et qui possède des propriétés spéciales, un lien que l'on qualifiera de mouvement.

Tout comme pour l'analyse d'image, la transformée de Fourier (\mathcal{F}) est utilisée pour passer au domaine fréquentiel de la vidéo.

Rappel : Fourier

Soit $I(x, y, t)$ l'intensité d'un pixel (x, y) de la vidéo au temps t . Si nous supposons que la vidéo est une fonction continue, la transformée de Fourier \mathcal{F} de I sera donnée par l'intégrale suivante :

$$\begin{aligned}\hat{I}(u, v, \tau) &= \mathcal{F}(I) \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(x, y, t) \cdot \exp \{-2\pi j(ux + vy + \tau t)\} dx dy dt,\end{aligned}$$

u, v Fréquences spatiales dans le sens horizontal et vertical de la vidéo

τ Fréquence temporelle

Rappel : Fourier

La transformée de Fourier inverse de la vidéo \mathcal{F}^{-1} est obtenue par l'intégrale suivante :

$$\begin{aligned}l(x, y, t) &= \mathcal{F}^{-1}(\hat{l}) \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{l}(u, v, \tau) \cdot \exp\{2\pi j(ux + vy + \tau t)\} du dv d\tau\end{aligned}$$

Fourier pour la vidéo numérique

Dans le cas de la vidéo numérique, on utilise la FFD suivante :

$$\begin{aligned}\hat{l}(u, v, \tau) &= \mathcal{F}(l) \\ &= \frac{1}{MNK} \sum_{t=0}^K \sum_{x=0}^M \sum_{y=0}^N l(x, y, t) \cdot \exp \left\{ -2\pi j \left(u \frac{x}{M} + v \frac{y}{N} + \tau \frac{t}{K} \right) \right\}\end{aligned}$$

Et la transformée inverse :

$$\begin{aligned}l(x, y, t) &= \mathcal{F}^{-1}(\hat{l}) \\ &= \sum_{\tau=0}^K \sum_{u=0}^M \sum_{v=0}^N \hat{l}(u, v, \tau) \cdot \exp \left\{ 2\pi j \left(u \frac{x}{M} + v \frac{y}{N} + \tau \frac{t}{K} \right) \right\}\end{aligned}$$

Mouvement rectiligne en Fourier

Supposons qu'un objet se déplace dans la scène avec un mouvement rectiligne de vitesse constante $\vec{V}(v_x, v_y)$.

Soit $I(x, y, t_0)$ le premier image de la séquence. La transformée de Fourier de la vidéo est alors donnée par :

$$\begin{aligned} \hat{I}(u, v, \tau) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(x, y, t) \cdot \exp\{-2\pi j(ux + vy + \tau t)\} dx dy dt \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(x - v_x t, y - v_y t, t_0) \cdot \\ &\quad \exp\{-2\pi j(u(x - v_x t) + v(y - v_y t))\} \\ &\quad \exp\{-2\pi j(uv_x + vv_y + \tau)t\} dx dy dt \end{aligned}$$

Mouvement rectiligne en Fourier

En posant le changement de variables $x' = x - v_x t$ et $y' = y - v_y t$, l'équation peut être réécrite comme suit :

$$\begin{aligned} \hat{l}(u, v, \tau) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} l(x', y', t_0) \cdot \exp\{-2\pi j(ux' + vy')\} dx' dy' \\ &\quad \cdot \int_{-\infty}^{+\infty} \exp\{-2\pi j(uv_x + vv_y + \tau)t\} dt \\ &= \hat{l}_0(u, v) \cdot \hat{\delta}(uv_x + vv_y + \tau) \end{aligned}$$

$\hat{l}_0(u, v)$ TF du premier image.

$\hat{\delta}$ TF de la fonction de Dirac.

Mouvement rectiligne en Fourier

$$\hat{l}(u, v, \tau) = \hat{l}_0(u, v) \cdot \hat{\delta}(uv_x + vv_y + \tau) \quad (2)$$

Par les propriétés de la *TF* d'une δ , on sait que pour que la *TF* de la vidéo soit nulle, en supposant *TF* du premier image non-nulle, alors **l'expression de la $\delta(uv_x + vv_y + \tau)$ doit être nulle.**

$$\begin{aligned} \mathcal{F}^{-1}(\hat{\delta}(uv_x + vv_y + \tau)) &= 0 \\ u \cdot v_x + v \cdot v_y + \tau &= 0 \end{aligned}$$

Mouvement rectiligne en Fourier

On détermine alors la fréquence temporelle (τ) par l'équation suivante :

$$\tau = -uv_x - vv_y = -\langle (u, v), (v_x, v_y) \rangle$$

Où \langle , \rangle désigne le produit scalaire. Si l'objet en mouvement est caractérisé par une fréquence spatiale (u, v) (texture), alors la fréquence temporelle τ perçue dépendra directement de la vitesse du mouvement.

Mouvement rectiligne en Fourier

- Si $u = v = 0$ (objet avec une couleur constante), alors $\tau = 0$. Cela veut dire que peu importe la vitesse de l'objet, on ne percevra aucun changement temporel.
- Si $(u, v) \perp (v_x, v_y)$ alors $\tau = 0$. Cela veut dire que si l'objet se déplace dans la direction perpendiculaire à la fréquence de changement des motifs de sa texture, alors on ne percevra aucun changement temporel. Si, par contre, $(u, v) \parallel (v_x, v_y)$, alors le changement temporel perçu sera maximal.

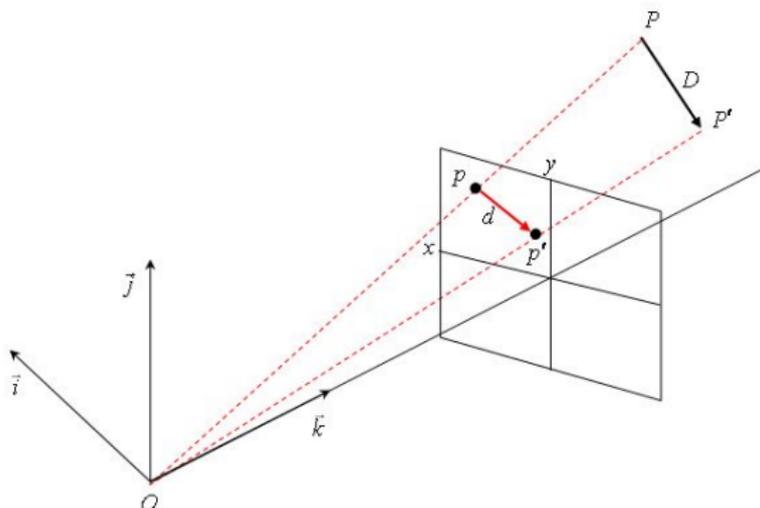
Le modèle du mouvement 2D vs 3D

Quand un objet dans la scène ou la caméra est en mouvement, la projection d'un point d'un objet de la scène subira également un mouvement.

Soit un point $P(X, Y, Z)^T$ sur la scène à l'instant t_1 , qui se déplace vers le point $P'(X', Y', Z')^T = (X + \Delta X, Y + \Delta Y, Z + \Delta Z)^T$, à l'instant $t_2 > t_1$.

La projection de ce point dans l'image se déplacera du point $p(x, y)^T$ vers le point $p'(x', y')^T = (x + d_x, y + d_y)^T$

Le modèle du mouvement 2D vs 3D



On appellera alors $(\Delta X, \Delta Y, \Delta Z)^T$ le vecteur de déplacement 3-D et $(d_x, d_y)^T$ le vecteur de déplacement 2-D.

L'ensemble des vecteurs $(d_x, d_y)^T$ dans l'image s'appelle **le champ de mouvement 2-D**.

Le modèle du mouvement

La vitesse

NOTE :

Dans le domaine de la vidéo, on utilise parfois la vitesse $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ au lieu du déplacement (dx, dy) .

→ Cela dépend entièrement de l'application !

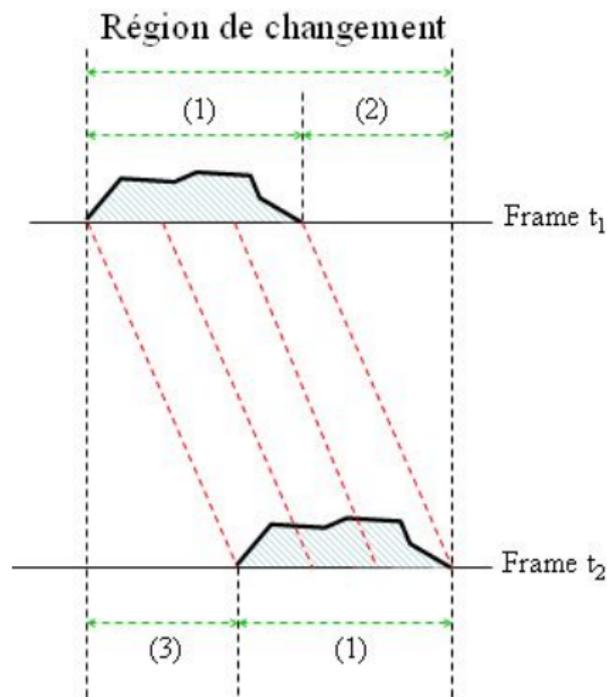
Question :

Dans quel cas le vecteur de déplacement 2D ne peut être défini ?

Cas de vitesse 2-D indéfinie

- 1 **Cas de sortie** : Un point du image 1 qui, au image 2 ne se trouve plus dans le champs de la vidéo.
- 2 **Cas d'entrée** : Un point du image 2 qui ne se trouvait pas au image 1.
- 3 **Cas d'occlusion** : Un point du image 1 qui était caché par un autre objet est maintenant découvert, et vice-versa.
- 4 **Cas d'ouverture** : L'intérieur d'un objet texturé à illumination constante.

Cas d'occlusion



- (1): Objet en mouvement.
- (2): Arrière-plan couvert.
- (3): Arrière-plan découvert.

Correspondance mouvement 2D et 3D

Cas général

Supposons que nous n'aurons toujours que des déplacements rigides (rotation et translation 3D) et que la caméra est fixe.

L'image du point $P(X, Y, Z)^T$ est le point $P'(X', Y', Z')^T$, donné par :

$$P' = \mathbf{R}P + \mathbf{T}$$

En ayant $\mathbf{R} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix}$, alors on aura :

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} T_X \\ T_Y \\ T_Z \end{pmatrix}$$

Correspondance mouvement 2D et 3D

Cas général

Supposons un modèle de correspondance suivant une projection perspective, où :

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z} \quad (3)$$

On peut démontrer que le point 2D (x', y') correspondant au point 3D P' est donné par :

$$\begin{aligned} x' &= f \frac{(r_1 X + r_2 Y + r_3 f)Z + T_X f}{(r_7 X + r_8 Y + r_9 f)Z + T_Z f} \\ y' &= f \frac{(r_4 X + r_5 Y + r_6 f)Z + T_Y f}{(r_7 X + r_8 Y + r_9 f)Z + T_Z f} \end{aligned}$$

→ Changement d'échelle des paramètres T_X , T_Y , T_Z et la profondeur Z du même facteur ne changera pas le résultat (x', y') de la correspondance.

→ Points 3D sur la même ligne de vue ont la même projection sur le plan de l'image.

Correspondance mouvement 2D et 3D

Cas planaire

Lorsqu'un objet est planaire, on peut décrire les coordonnées de ses points par une équation du plan, de la forme : $Z = aX + bY + c$.

En remplaçant la valeur de Z dans les équations précédentes, et en utilisant les équations de la projection perspective (Eq.3), on démontre alors que le point 2D (x', y') correspondant au point 3D P' est donné par :

$$x' = \frac{m_0 + m_1x + m_2y}{1 + e_1x + e_2y}$$

$$y' = \frac{n_0 + n_1x + n_2y}{1 + e_1x + e_2y}$$

$(m_0, m_1, m_2, n_0, n_1, n_2, e_1, e_2)$ Coefficients dépendants des paramètres de la matrice de rotation \mathbf{R} , du vecteur de translation $(T_X, T_Y, T_Z)^T$ et des coefficients a, b, c de l'équation du plan de l'objet.

Correspondance mouvement 2D et 3D

Ce système de **mapping par projection** permet de modéliser directement la relation entre les projections d'un point d'un objet après mouvement de ce dernier.

Pour enlever la contrainte d'objet planaire, on subdivise un objet en plusieurs parties planaires et on applique ensuite la projection pour chaque partie.

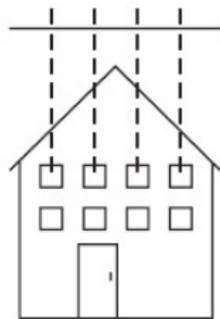
Effet de projection

Le modèle de projection planaire permet de bien modéliser trois effets visuels importants :

- *L'effet du chirping* : Augmentation de la fréquence spatiale de motifs de textures en relation avec la profondeur.
- *L'effet de convergence* : Convergence de lignes parallèles en relation avec la profondeur. Les points de convergence sont appelés **points de fuite**.
- *Linéarité* : On conserve la linéarité des lignes.

Approximation du modèle de projection

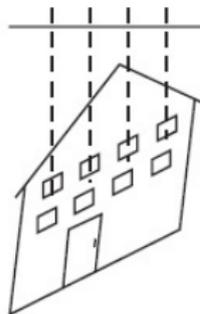
Afin de réduire le nombre de coefficients à estimer et à supprimer les variables à optimiser au dénominateur, on utilise entre autres deux modèles : le modèle **affine** et **bilinéaire**.



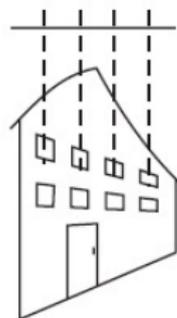
Normal



Projection

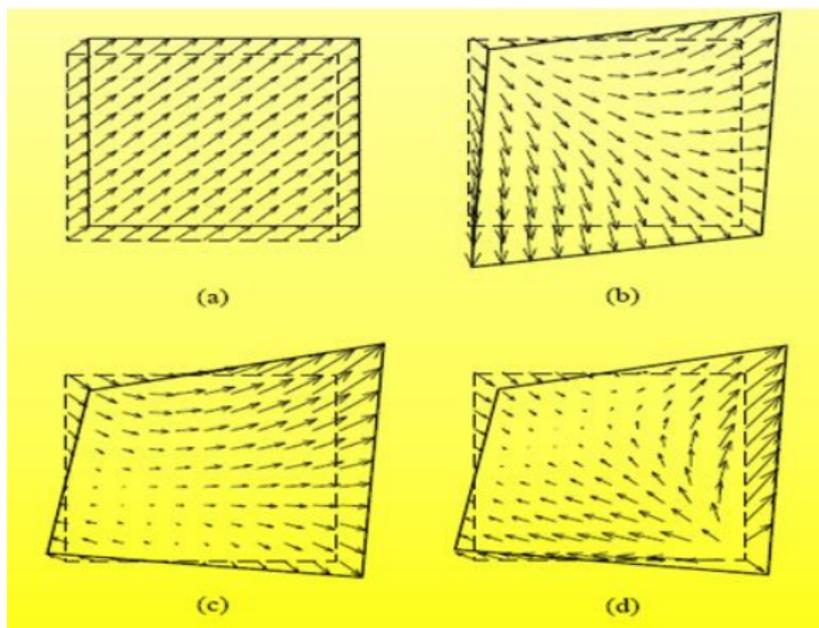


Affine



Bilinéaire

Approximation du modèle de projection



Différents modèles de mouvement et les vecteurs de déplacement qu'ils génèrent : (a) simple translation, (b) modèle affine, (c) modèle bilinéaire et (d) application projective.

Approximation du modèle de projection

Modèle affine

Le modèle affine est souvent utilisé pour décrire le mouvement 3D sous la projection orthographique. Ce modèle est adapté aux objets lointains.

$$x' = m_0 + m_1x + m_2y$$

$$y' = n_0 + n_1x + n_2y$$

- 6 paramètres à estimer ($m_0, m_1, m_2, n_0, n_1, n_2$).
- Préserve les proportions de longueurs et le parallélisme de segments.
- Ne capte pas l'effet du *chirping* et de *convergence*, mais conserve la linéarité.

Approximation du modèle de projection

Modèle bilinéaire

Le modèle bilinéaire peut être vu comme une transformation d'un carré en un quadrangle. Il approxime bien le modèle de projection pour les objets peu texturés, les déformations étant moins drastiques.

$$x' = m_0 + m_1x + m_2y + m_3xy$$

$$y' = n_0 + n_1x + n_2y + n_3xy$$

- 8 paramètres à estimer ($a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$).
- Capte l'effet de *convergence*.
- Ne préserve pas la linéarité des contours et le *chirping*.

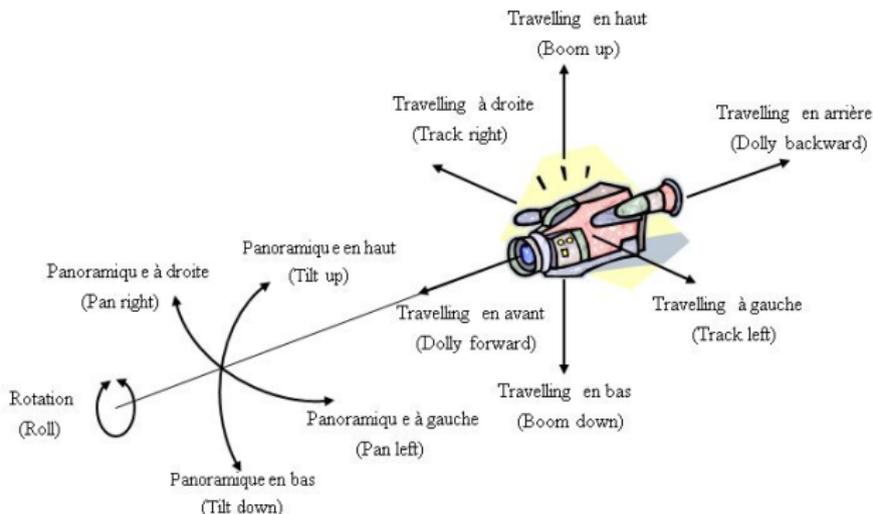
Mouvement apparent

En analyse de la vidéo, le mouvement réel ne correspond pas toujours au mouvement que l'on perçoit. Ce dernier est cependant le plus important, car c'est la perception qui influence le traitement de l'image et la vidéo.

Les deux grandes sources de différences entre le mouvement réel et le mouvement apparent sont :

- Les mouvements de la caméra
- L'illumination

Le modèle de la caméra



- 1 **Boom, track, dolly** ⇒ Translation
- 2 **Pan, tilt, roll** ⇒ Rotation
- 3 **Zoom** ⇒ Changement de focale

Le mouvement apparent

Le modèle de la caméra

On réfère les mouvements de la caméra à des **mouvements globaux** de la scène.

On note aussi que le *zoom* et le *dolly* ont le même effet en terme de mouvement apparent, c'est-à-dire un changement d'échelle.

Voyons les formes paramétriques de ces mouvements de caméra.

Le modèle de la caméra

Translation de la caméra

Les mouvements *Track* et *Boom* correspondent à des translations horizontales et verticales de la caméra, respectivement : $(T_X, T_Y, T_Z)^T$, où $T_Z = 0$.

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} T_X \\ T_Y \\ 0 \end{pmatrix},$$

Par les équations de projection perspective (Eq.3) et sachant que $Z = Z'$, alors le point 2D (x', y') correspondant au point (X, Y, Z) est donné par :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \underbrace{\begin{pmatrix} f \frac{T_X}{Z} \\ f \frac{T_Y}{Z} \end{pmatrix}}_{(d_x, d_y)^T}$$

Le modèle de la caméra

Rotation de la caméra

Les mouvements *Tilt* et *Pan* correspondent à des rotations autour des axes X et des Y , respectivement : R_X et R_Y

$$\mathbf{R}_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_X) & -\sin(\theta_X) \\ 0 & \sin(\theta_X) & \cos(\theta_X) \end{pmatrix},$$

$$\mathbf{R}_Y = \begin{pmatrix} \cos(\theta_Y) & 0 & \sin(\theta_Y) \\ 0 & 1 & 0 \\ -\sin(\theta_Y) & 0 & \cos(\theta_Y) \end{pmatrix},$$

où θ_X et θ_Y sont les angles de rotations autour des axes X et des Y , respectivement.

Le modèle de la caméra

Rotation de la caméra

Supposons un petit angle de rotation (i.e. $\sin(\theta_X) \simeq \theta_X$, $\sin(\theta_Y) \simeq \theta_Y$, $\cos(\theta_X) = \cos(\theta_Y) \simeq 1$), alors le mouvement 3-D d'un point $(X, Y, T)^T$ sera :

$$\begin{aligned} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} &= \mathbf{R}_X \mathbf{R}_Y \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & -\theta_Y \\ 0 & 1 & -\theta_X \\ \theta_Y & \theta_X & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \end{aligned}$$

Finalement, si $X\theta_Y \ll Z$ et $Y\theta_X \ll Z$, et si $Z \approx Z'$, alors :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \underbrace{\begin{pmatrix} -\theta_Y f \\ -\theta_X f \end{pmatrix}}_{(d_v, d_v)^T}$$

Le modèle de la caméra

Rotation de la caméra

Le mouvement *Roll* correspond à la rotation autour de l'axe des Z . Soit \mathbf{R}_Z la matrice de rotation autour de cet axe, qui est donnée par :

$$\mathbf{R}_Z = \begin{pmatrix} \cos(\theta_Z) & -\sin(\theta_Z) & 0 \\ \sin(\theta_Z) & \cos(\theta_Z) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Après un mouvement de *Roll*, en supposant un angle petit, on aura :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \underbrace{\begin{pmatrix} -\theta_Z y \\ \theta_Z x \end{pmatrix}}_{(d_x, d_y)^T}$$

Le modèle de la caméra

Changement d'échelle de la caméra

Les mouvement *Zoom* et *Dolly* correspondent à des changements d'échelle, où le facteur varie selon le mouvement :

$$\begin{aligned} \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} \rho x \\ \rho y \end{pmatrix} \\ &= \begin{pmatrix} x \\ y \end{pmatrix} + \underbrace{\begin{pmatrix} \rho x - x \\ \rho y - y \end{pmatrix}}_{(d_x, d_y)^T} \end{aligned}$$

Zoom → Changement de focale : $\rho = \frac{f}{f'}$

Dolly → Changement de profondeur : $\rho = \frac{Z}{Z+T_Z}$

Modèle de la caméra

Si la scène se déplace selon le même facteur que la caméra, aucun mouvement ne sera détecté, même si en vrai il y en a. Le **mouvement apparent** est nul.

Le même raisonnement s'applique inversement, c'est-à-dire que si la scène est fixe et que la caméra bouge, un mouvement global important sera détecté, alors qu'en réalité il n'y avait aucun mouvement 3D de la scène.

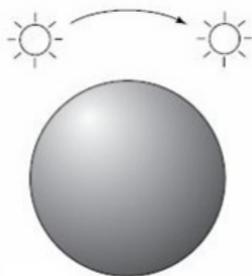
Ces mouvements globaux de caméra pourront être estimés, puis supprimés.

Mouvement apparent

Illumination

Les changements d'illumination sont problématiques pour la détection du mouvement.

Le mouvement apparent est grandement affecté par la réflectivité, la transparence, l'irradiance, lorsque la source lumineuse bouge ou modifie son intensité.



L'intensité de la source modifie la texture apparente d'un objet.

Mouvement apparent

Illumination

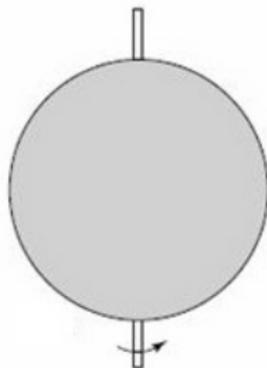
Les algorithmes de détection de mouvement en général ne prennent pas en compte l'illumination. Le mouvement apparent résultant de l'illumination est considéré comme un déplacement réel.

Dans la plupart des cas, la contribution de l'illumination est minime et est considérée comme étant constante.

Mouvement apparent

Texture

Il arrive parfois qu'un objet subit une transformation qui n'est pas apparente. Prenons exemple sur la rotation d'une sphère sans texture :

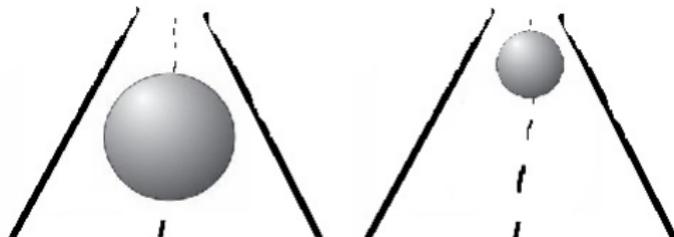


La rotation, bien que réelle, n'a en apparence aucun mouvement, l'absence de la texture n'apportant pas l'information nécessaire.

Mouvement apparent

Déplacement dans la ligne de vue

Imaginons également une sphère qui s'éloigne sur une route.



Le mouvement apparent d'un changement d'échelle en x et y combiné à une translation en y est perçu.

Mouvement apparent

Objet

Les cas de texture et de déplacement dans la ligne de vue peuvent être ignorés.

Cependant, avec une connaissance *a priori* de la modélisation de la scène, de l'objet à observer et des contraintes de mouvement de ce dernier, on peut connaître le déplacement réel.

Dans un cas de déplacement d'un véhicule dans le trafic, par exemple, on peut savoir que s'il y a un changement d'échelle, c'est dû à l'éloignement en Z.