



Vision par ordinateur / réalité augmentée

- **Traitement d'image en temps réel**
 - Obtention de zones identifiables de l'image
 - Contour
 - Coins
- **Interprétation 3D**
 - Placement d'un marqueur dans l'espace à partir de la géométrie de l'objet projeté
- **Synthèse d'image**
 - Superposition d'un objet 3D sur l'image

Zone identifiable

- **Choix d'un marqueur**
 - Facilement identifiable
 - Couleurs
 - Variations fortes (hautes fréquences)
 - De taille cohérente avec le contexte de la zone recherchée
 - Géométrie adaptée au placement 3D
 - Soulever les incertitudes liées à la géométrie du marqueur

Deux situations de marqueurs

<ul style="list-style-type: none"> ● Salle de réalité virtuelle <ul style="list-style-type: none"> – Environnement sombre – Utilisation d'éléments lumineux <ul style="list-style-type: none"> • Actifs <ul style="list-style-type: none"> ✓ Led • Passifs <ul style="list-style-type: none"> ✓ Éléments réfléchissants ✓ Domaine Infrarouge 	<ul style="list-style-type: none"> ● Salle de conférence <ul style="list-style-type: none"> – Environnement éclairé – Utilisation d'éléments faciles à extraire de la scène <ul style="list-style-type: none"> • Couleurs, fréquences décelables
---	---

Recherche du marqueur dans une image

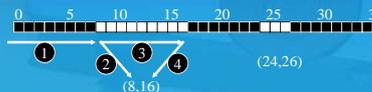
- **Cas des éléments lumineux (LED)**
 - Chaque LED provoque une tache claire sur l'image
 - Rechercher le centre de gravité de la tache
 - Identifier la LED d'origine
 - ✓ Sélection sur la couleur

Un algorithme de recherche du centre

- Algorithme de recherche du centre des zones claires dans une image
- Deux étapes (algorithme de remplissage) :
 - **Traitement ligne à ligne**
 - Détermination d'une série de segments clairs par ligne
 - Utilisation d'un tableau de segments par ligne
 - ✓ Avantage : taille mémoire fixe
 - ✓ Inconvénient : nombre maximum de segments par ligne ?
 - **Traitement des segments**
 - Mise en relation des segments admettant des points connexes
 - Calcul du centre de gravité

Algorithme de recherche du centre

- Recherche des segments d'une ligne
 1. Parcours de gauche à droite en cherchant un point clair
 2. Mémorisation du point clair
 3. Parcours de gauche à droite en cherchant un point foncé
 4. Mémorisation du segment



Algorithme de recherche du centre

```

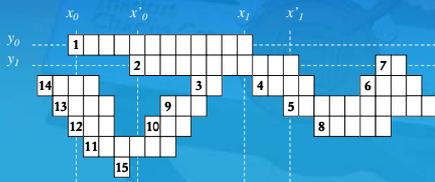
iy = ly; // compteur de ligne
while (iy-->0)
{
  n=0; // nombre de points sur la ligne
  ix = 0; // position courante dans la ligne
  ptrSeg->nbreSegments=0; // nombre de segments
  while (n<nbrePointsMaxParLigne && ix<lx)
  { // recherche d'un pixel blanc (!=0)
    while (ix<lx && !*ptr)
    { ptr+=3; // saute RVB
      ix++; // passage au pixel de gauche
    }
    if (ix<lx) // si on a trouvé un pixel blanc
    { ptrSeg->pos[n++]=ix; // mémorisation du point
      // recherche d'un pixel noir (=0)
      while (ix<lx && *ptr)
      { ptr+=3;
        ix++;
      }
      if (n==nbrePointsMaxParLigne-1) return false;
      if (ix<lx) // si on a trouvé un pixel blanc
      { ptrSeg->pos[n++]=ix; // mémorisation du point
        ptrSeg->nbreSegments++;
      }
    }
  }
  ptrSeg++; // passage à la ligne suivante
}
    
```

Algorithme de recherche du centre

- Algorithme de remplissage
 - Traitement des segments connexes
 - Méthode par germes
- Algorithme
 - Recherche d'un premier segment
 - Le mémoriser dans un germe
 - Pour chaque germe
 - On traite le germe pour calculer le centre de la zone
 - Si il existe un segment connexe au-dessus alors on ajoute un germe pour ce nouveau segment
 - Si il existe un segment connexe au-dessous alors on ajoute un germe pour ce nouveau segment

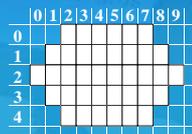
Algorithme de recherche du centre

- Critère de connexité
 - Segment 1 : x_0, x_1, y
 - Segment 2 : x'_0, x'_1, y'
 - Connexe si :
 - Lignes voisines : $|y' - y| = 1$
 - $x'_0 > x_0$ et $x'_1 < x_1$

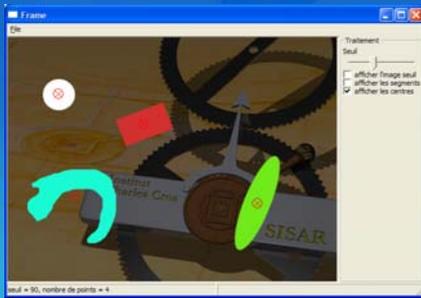


Algorithme de recherche du centre

- Calcul du centre à partir des segments regroupés
 - Centre de gravité
 - Chaque pixel à le même poids
 - ✓ Moyenne des x
 - ✓ Moyenne des y
 - Centre partiel : par ligne
 - ✓ Somme des Moyennes des x * (nombre de points) / nombre de points
 - ✓ Somme des y * (nombre de points) / nombre de points

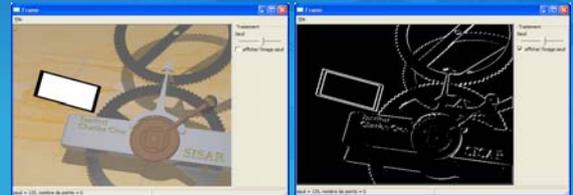


Exemple d'application



Recherche du marqueur dans une image

- Cas des marqueurs couleurs
 - Traitement de l'image pour faire apparaître les contours
 - Filtrage de Sobel
 - Recherche des segments formant les contours
 - Détermination des sommets des marqueurs



Exemple de placement 3D

- Marqueur rectangulaire
 - Propriétés géométriques
 - Coté parallèle 2 à 2
 - Coté orthogonaux 2 à 2
 - Dimensions connues

Recherche d'un marqueur dans une image

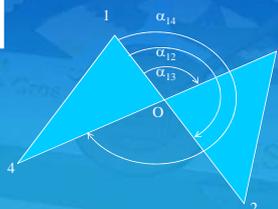
- Ordonner les points
 - Dans la plupart des algorithmes de placement 3D, les points doivent être ordonnés



Ordonner les points

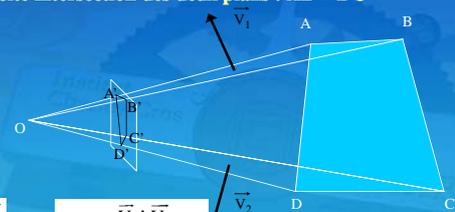
- Chercher le centre des points
- Classifier les points en fonction de l'angle α

$$\alpha_i = \arcsin \left(\frac{(OP_i \wedge OP_j)_z}{|OP_i| |OP_j|} \right)$$



Évaluation des directions des cotés

- L'intersection entre deux plans est une droite unique
- Si $W = V_1 \wedge V_2$, W est orthogonal à V_1 et à V_2
- W dans le plan OAB et dans OCD donc W porte la droite intersection des deux plans : $AB = DC$



$$\begin{aligned} \overrightarrow{OA} \wedge \overrightarrow{OB} &= \vec{V}_1 \\ \overrightarrow{OC} \wedge \overrightarrow{OD} &= \vec{V}_2 \end{aligned} \Rightarrow \overrightarrow{AB} = \frac{\vec{V}_1 \wedge \vec{V}_2}{\|\vec{V}_1\| \times \|\vec{V}_2\|} \times l$$

Évaluation des directions des cotés

- Même calcul pour déterminer BC.

$$\frac{\overline{OB'} \wedge \overline{OC'}}{\overline{OD'} \wedge \overline{OA'}} = \frac{\vec{V}_3}{\vec{V}_4} \Rightarrow \overline{BC} = \frac{\vec{V}_3 \wedge \vec{V}_4}{\|\vec{V}_3\| \times \|\vec{V}_4\|} \times L$$
- On obtient donc l'orientation du plan
 - Sa normale :

$$\overline{N} = \overline{AB} \wedge \overline{BC}$$
- Il manque sa distance à l'observateur
 - Il suffit d'avoir un point !

Placement d'un sommet

- Projection d'un point théorique sur l'écran
 - Application du théorème de Thalès

$$\overline{OA'} = \begin{cases} x_{A'} = \frac{x_A}{z_A} \\ y_{A'} = \frac{y_A}{z_A} \\ z_{A'} = z_p \end{cases}$$

$$\overline{OB} = \overline{OA} + \overline{AB}$$

$$\begin{cases} x_{A'} = \frac{x_A}{z_A} \\ y_{A'} = \frac{y_A}{z_A} \\ z_{A'} = z_p \end{cases}$$

$$\begin{cases} x_B = \frac{x_A + x_{AB}}{z_A + z_{AB}} \\ y_B = \frac{y_A + y_{AB}}{z_A + z_{AB}} \\ z_B = z_p \end{cases}$$

Placement d'un sommet

On cherche une expression de z_A

$$\begin{cases} x_{A'} = \frac{x_A}{z_A} \\ y_{A'} = \frac{y_A}{z_A} \\ z_{A'} = z_p \end{cases}$$

$$\Rightarrow \begin{cases} z_p \times x_{A'} = z_A \times x_{A'} \\ z_p \times y_{A'} = z_A \times y_{A'} \end{cases}$$

$$\begin{cases} x_{B'} = \frac{(x_A \times z_A + x_{AB} \times z_p)}{z_A + z_{AB}} \\ y_{B'} = \frac{(y_A \times z_A + y_{AB} \times z_p)}{z_A + z_{AB}} \\ z_{B'} = z_p \end{cases}$$

$$\Rightarrow \begin{cases} z_A \times (x_{B'} - x_{A'}) - z_p \times x_{AB} = -z_{AB} \times x_{B'} \\ z_A \times (y_{B'} - y_{A'}) - z_p \times y_{AB} = -z_{AB} \times y_{B'} \\ z_{B'} = z_p \end{cases}$$

$$\downarrow z_A$$

Réalité virtuelle et réalité augmentée

Vision stéréoscopique

*Benoît Piranda
Équipe SISAR
Université de Marne La Vallée*

Système de réalité virtuelle

- Ensemble de périphériques, communication avec les sens de l'utilisateur
 - Vision 3D
 - Interaction espace réel / espace virtuel
 - Immersion sonore
 - Autres : odeur, toucher

Vision binoculaire

- Périphérique de séparation des images pour chaque œil
- Principe
 - Système de séparation des sources lumineuse
 - Les deux images
 - Doivent apparaître superposées
 - Doivent être associées à un seul œil

Méthodes utilisées par les périphériques

- **Casque immersif**
 - Deux petits écrans
 - Placés devant chaque œil
- **Avantages**
 - Toujours placé face à l'utilisateur
 - Séparation des sources simple
- **Inconvénients**
 - Système lourd et encombrant
 - Réceptions des signaux vidéo par câble

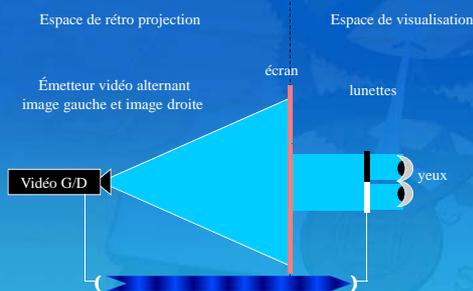


Méthodes utilisées par les périphériques

- **Lunettes actives**
 - Lunettes équipées d'un système d'occultation des yeux
 - Occultation synchronisée avec l'affichage à l'écran
 - Communication nécessaire entre l'émetteur et les lunettes (infrarouge)
- **Avantages**
 - Un ou deux projecteurs
 - Séparation totale des sources
- **Inconvénients**
 - Système encore lourd
 - Un seul utilisateur simultanément regardant l'écran



Lunettes actives



Méthodes utilisées par les périphériques

- **Lunettes passives colorées (anaglyphe)**
 - Décomposition de la lumière en deux couleurs complémentaires
 - Filtres colorés complémentaires
 - Rouge / cyan
 - Vert / violet
 - Jaune / Bleu
- **Avantages**
 - Périphérique très léger
 - Fonctionne avec un simple écran
 - Coût très faible
- **Inconvénients**
 - Altération des couleurs de la scène
 - Un seul utilisateur simultanément regardant l'écran



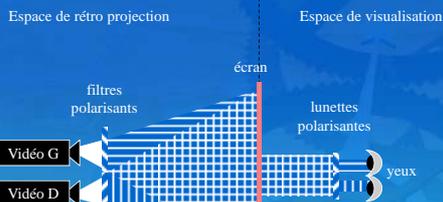
Exemples d'image colorée



Méthodes utilisées par les périphériques

- **Lunettes à filtres polarisants Linéaires**
 - Filtrage de la lumière se propageant dans un plan
 - Plans orientés suivant deux plans orthogonaux (-45° et $+45^\circ$)
- **Avantages**
 - Périphérique très léger
 - Fonctionne avec un simple écran
 - Coût très faible
- **Inconvénients**
 - Un seul utilisateur simultanément regardant l'écran
 - Ne supporte pas la rotation de la tête face à l'écran

Module de projection stéréoscopique



Méthodes utilisées par les périphériques

- **Lunettes à filtres polarisants circulaires**
 - Deux sens différents pour chaque oeil
- **Avantages**
 - Périphérique très léger
 - Fonctionne avec un simple écran
 - Supporte les mouvements de la tête face à l'écran
- **Inconvénients**
 - Un seul utilisateur simultanément regardant l'écran



Méthodes utilisées par les périphériques

- **Alioscopie**
 - Système de lentilles placées sur un écran LCD permettant de construire des zones spatiales devant l'écran permettant la vision stéréoscopique
 - Supporte peu de mouvement de l'utilisateur

Vision stéréoscopique

Principe de création d'une image stéréo

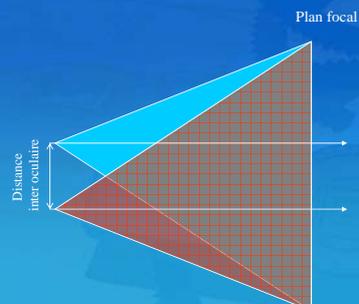


Acquisition des images

- **Deux prises de vue**
 - **Placement des caméras**
 - Décalées de la distance interoculaire
 - ✓ Env. 6.5 cm
 - Dirigées suivant des axes optiques parallèles
 - De même angle optique

Vision binoculaire

- **Placement des caméras**



Construction de l'image 3D

- Différence avec l'image monoscopique :
 - Séparation du plan focal et du support de l'image
- Plan focal :
 - Plan de la scène où les éléments sont vus au même endroit par les deux yeux.
 - Pas de perception du 3D stéréo



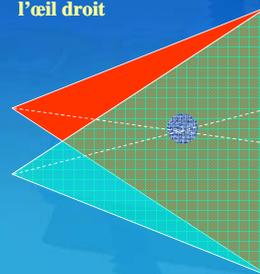
Construction de l'image 3D

- Les deux images produites lors de l'acquisition doivent être placées de façon à coïncider au niveau du plan focal



Vision binoculaire

- Cas d'un objet placé devant le plan focal
 - Élément vu par l'œil gauche à droite de l'image de l'œil droit



Vision binoculaire

- Cas d'un objet placé devant le plan focal
 - Élément vu par l'œil gauche à gauche de l'image de l'œil droit



Effets de déplacement de l'observateur

- Mouvements pseudoscopiques
 - Phénomènes optiques
 - Liés aux mouvements de l'observateur
 - Perturbent la perception 3D
 - Déplacement de l'image
 - Changement de dimensions de l'image

Mouvements pseudoscopiques

- Rapprochement de l'image

Mouvements pseudoscopiques

- Rapprochement de l'image

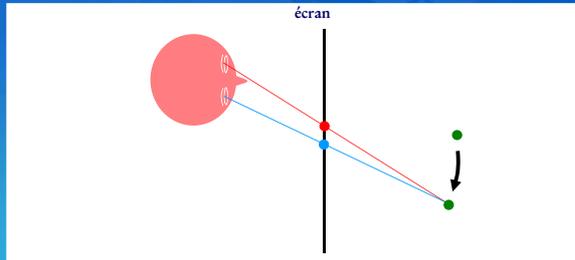
l'observateur se rapproche de l'écran l'objet virtuel se rapproche de l'écran

Mouvements pseudoscopiques

- Déplacement de l'image

Mouvements pseudoscopiques

- Déplacement de l'image

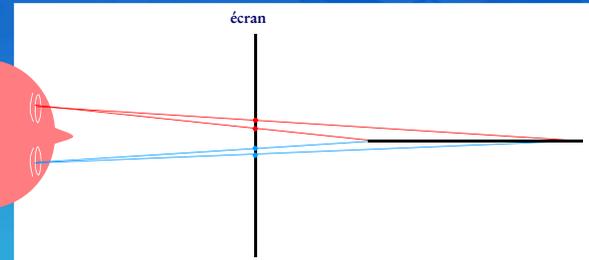


l'observateur se décale par rapport à l'écran

l'objet virtuel se décale par rapport à l'écran

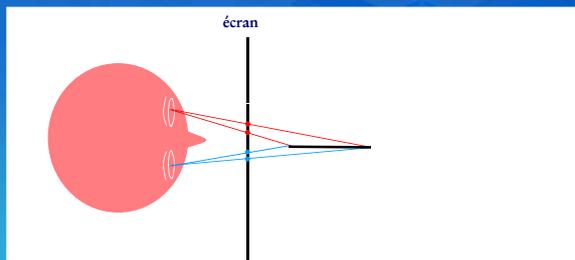
Mouvements pseudoscopiques

- Changement d'échelle



Mouvements pseudoscopiques

- Changement d'échelle

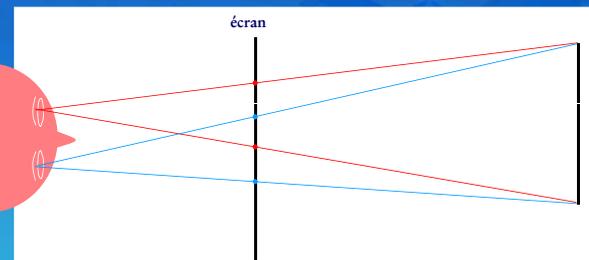


l'observateur se rapproche de l'écran

diminution la taille de l'objet

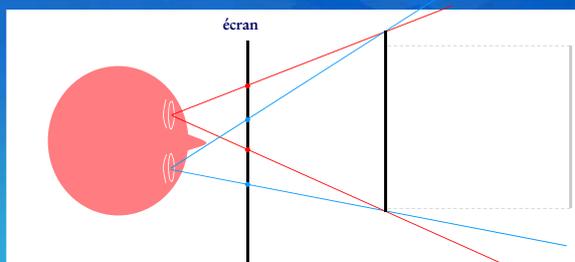
Mouvements pseudoscopiques

- Changement d'échelle



Mouvements pseudoscopiques

- Changement d'échelle



l'observateur se rapproche de l'écran

agrandissement de l'objet

Mouvement pseudoscopique

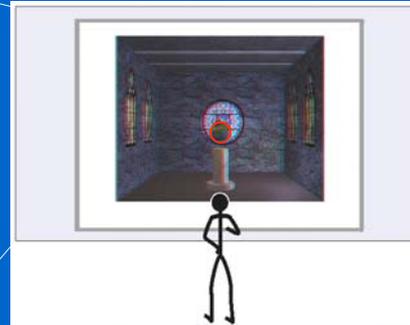
- Déplacement de l'observateur devant l'écran
 - Modifie la position et la dimension visible des objets
- Besoin d'une correction de l'image en fonction de la position de l'observateur
 - Déplacement des caméras virtuelles
- Périphérique d'acquisition de la position de l'observateur devant l'écran
 - Ensemble de caméra permettant le placement de l'observateur

Asservissement

- Déplacement des projections des objets en fonction du déplacement de l'observateur



Asservissement



Asservissement



Asservissement

- Point de vue de l'observateur



Outils de réalité virtuelle

Programmation de la stéréoscopie sous OpenGL

Institut Charles Cros

SISAR

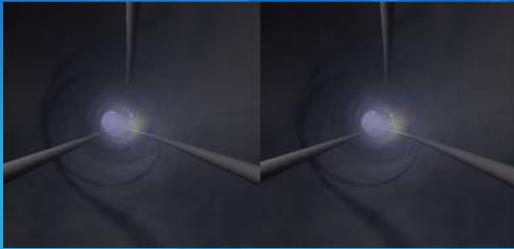
Benoît Piranda
Équipe SISAR
Université de Marne La Vallée

Principe de la synthèse stéréoscopique

- Produire deux images différentes
 - Une pour l'œil gauche, une pour l'œil droit
 - Correspond à deux caméras OpenGL différentes
- Imposer de synthétiser
 - Deux images par 24^{ème} de seconde
 - De la résolution des vidéo-projecteurs
 - Résolution actuelle 2x1024x768

Placement et direction des caméras

- **Caméra**
 - projection centrale
 - décalées de l'espace interoculaire
 - dirigées parallèlement

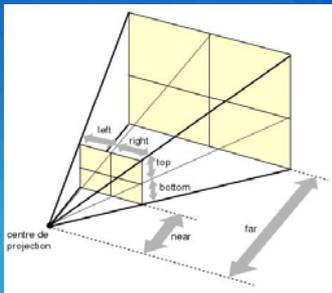


Création d'une paire d'images stéréo.

- Paramètres de la pyramide de projection OpenGL
- Système de 2 caméras :
 - observateur fixe devant l'écran
 - observateur se déplace devant l'écran

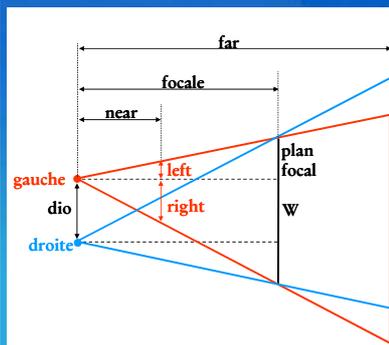
Images stéréoscopiques OpenGL

- Paramètres de projection OpenGL
 - Centre optique
 - Pyramide de projection
 - Left, right
 - Top, bottom
 - Near, far

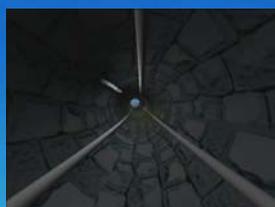


Images stéréoscopiques OpenGL

- Détermination des paramètres de projection pour un système de 2 caméras virtuelles



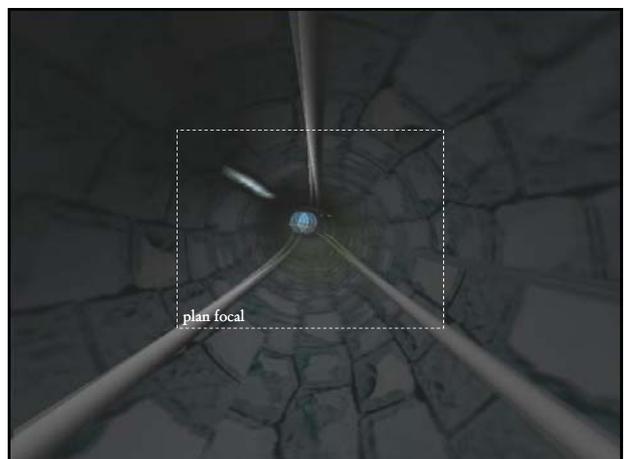
Exemple d'images stéréoscopiques

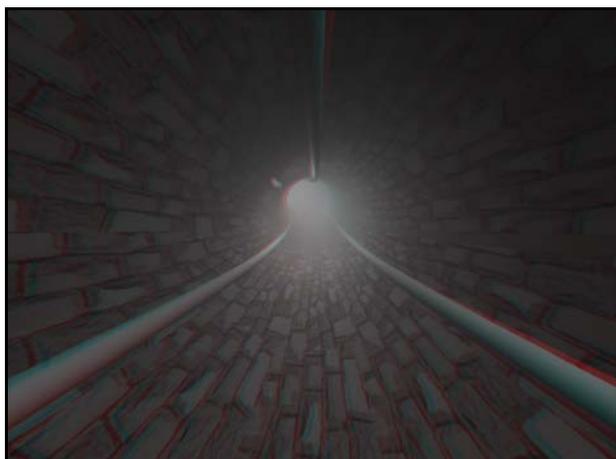


caméra gauche

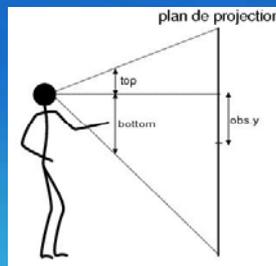


caméra droite





Paramètres verticaux

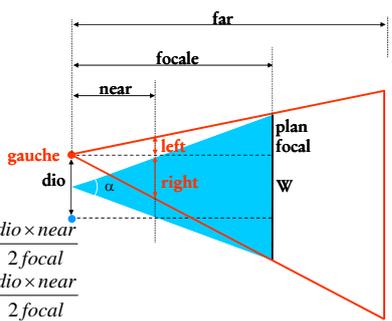


- Détermination des paramètres top et bottom
- Il ne doit pas y avoir de décalage verticaux entre les images

Calcul du décalage des images

$$\frac{left}{near} = \frac{W/2 - dio}{focal}$$

$$W = 2 \times focal \times \tan \frac{\alpha}{2}$$



$$left = near \times \tan \frac{\alpha}{2} - \frac{dio \times near}{2 \times focal}$$

$$right = near \times \tan \frac{\alpha}{2} + \frac{dio \times near}{2 \times focal}$$

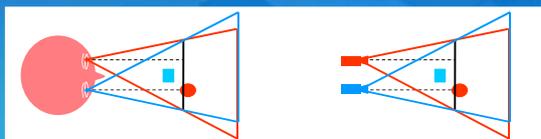
Analogies : scène réelle et scène virtuelle

- Analogies entre la scène réelle et la scène virtuelle
 - observateur / caméras
 - Les caméras représentent les yeux de l'observateur dans la scène virtuelle
 - Distance interoculaire = écartement des caméras
 - écran / plan focal
 - L'écran est le support de l'image
 - Le plan focal représente l'écran dans la scène virtuelle
 - Distance observateur écran = distance focale
 - Largeur de l'écran = largeur du plan focal

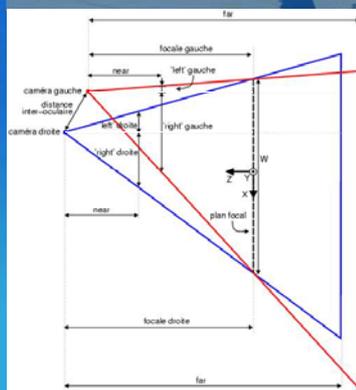
Analogies : scène réelle et scène virtuelle

- Tout est défini à un facteur d'échelle près :

$$\text{facteur d'échelle} = \frac{dio_{obs}}{dio_{caméras}} = \frac{focale_{obs}}{focale_{caméras}} = \frac{largeur_{écran}}{largeur_{plan\ focal}}$$



Paramètres des caméras



Paramètres des caméras

$$\text{focale}_{\text{droite}} = \frac{\text{obsPos}_z}{\text{facteur d'échelle}}$$

$$\text{left}_{\text{droite}} = \frac{\text{near}}{2 \cdot \text{focale}_{\text{droite}}} \left(W + \frac{2 \cdot \text{obsPos}_x}{\text{facteur d'échelle}} \right)$$

$$\text{right}_{\text{droite}} = \frac{\text{near}}{2 \cdot \text{focale}_{\text{droite}}} \left(W - \frac{2 \cdot \text{obsPos}_x}{\text{facteur d'échelle}} \right)$$

Paramètres des caméras

$$\text{top}_{\text{droite}} = \frac{\text{near}}{2 \cdot \text{focale}_{\text{droite}}} \left(\frac{W}{\text{ratio}} - \frac{2 \cdot \text{obsPos}_y}{\text{facteur d'échelle}} \right)$$

$$\text{bottom}_{\text{droite}} = \frac{\text{near}}{2 \cdot \text{focale}_{\text{droite}}} \left(\frac{W}{\text{ratio}} + \frac{2 \cdot \text{obsPos}_y}{\text{facteur d'échelle}} \right)$$

ratio = largeur / hauteur

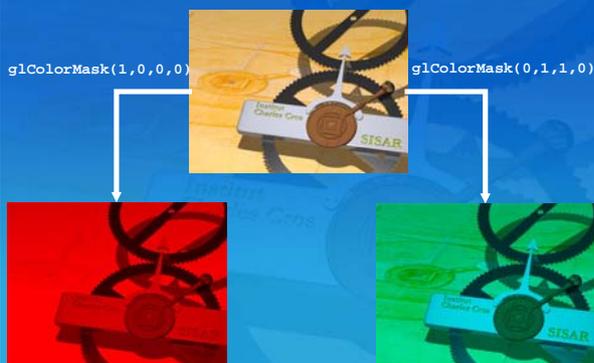
Principe

- Utilisation des complémentaires de couleurs
- Réalisation de deux images superposées
 1. Sélection d'un filtre de couleur
 2. Définition de la caméra pour le premier œil
 3. Dessin de la scène
 4. Sélection du filtre de couleur complémentaire
 5. Définition de la caméra pour le deuxième œil
 6. Dessin de la scène

Outils OpenGL

- Fonctions de sélection du buffer à traiter
 - `glDrawBuffer(GL_BACK);`
 - `glReadBuffer(GL_BACK);`
- Fonction de filtrage des couleurs
 - `glColorMask(r, v, b, a);`
 - Coefficients de masquage des composantes de couleur de l'image
 - r, v, b & a : booléen
 - `GL_FALSE` = composante non affichée
 - `GL_TRUE` = affichage de la composante
- Attention, le fond apparaît sous les couches à `GL_FALSE`

Exemple de projection



Exemple de programmation

- Faux 3D pas simple décalage des deux images
 - Aspect brillant (effet type Bump mapping)

```
// effacement du buffer de couleur
glClear(GL_COLOR_BUFFER_BIT);
// masque rouge
glColorMask(GL_TRUE, GL_FALSE, GL_FALSE, GL_FALSE);
// image à l'origine
glRasterPos2d(-1, -1);
glDrawPixels(lx, ly, GL_RGB, GL_UNSIGNED_BYTE, image);
// masque cyan
glColorMask(GL_FALSE, GL_TRUE, GL_TRUE, GL_FALSE);
// image décalée
glRasterPos2d(-1+8.0/lx, -1);
glDrawPixels(lx, ly, GL_RGB, GL_UNSIGNED_BYTE, image);
```