

Un dispositif de visualisation 3D immersif pour les documents numériques

Rodrigo Almeida Jérôme Dupire
andrad_r@auditeur.cnam.fr dupire_j@cnam.fr

CNAM/CEDRIC
292 rue St-Martin
F-75003 Paris

Motivations

"A Comparison of Reading Paper and Online Documents" O'Hara et Sellen

Les périphériques de visualisation et d'interaction traditionnels ne sont pas adaptés au travail dans un environnement 3D

Utilisation d'un plus grand champ de vision:
· plus grande immersion = concentration
· plus grand espace de travail pour organiser l'information

Expérimentation

Expérimenter un environnement de consultation de documents numériques visualisé avec une *VisionStation* d'Elumens

Le déplacement dans l'espace sera fait à l'aide d'une souris 3D (SpaceMouse de 3D Connexion)

Obstacles

Faible résolution des images affichées sur l'écran
Perte de lisibilité et confort

API pour la Projection Sphérique



sur l'écran traditionnel



sur la VisionStation

Solution

"Focus Plus Context Screens" Baudisch et al.

Deuxième Ecran:

· Visualisation plus précise
· Ajout de fonctions d'interaction

Conclusion et Travaux Futurs

Repartir les tâches de visualisation et d'interaction associées à chaque écran

Remplacer le deuxième écran par un écran tactile:
procurer une interaction plus fine comme le feuilletage de documents

Faciliter l'appropriation de la topologie de la scène 3D par l'utilisateur pour qu'il puisse optimiser ses tâches

Implémentation

Scène: OpenGL

Fonctions de Fenêtrage: GLUT

Projection Sphérique: API SPIClops

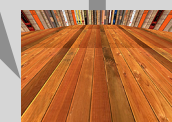
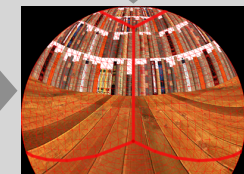
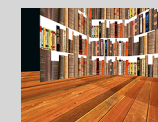
Interaction avec le SpaceMouse par une GLUT modifiée: `glutSpacemouseMotionFunc`

Utilisation d'images de tranches des livres pour les représenter

Distribution cylindrique des documents pour "entourer" l'utilisateur



Redessine la fenêtre: `glutDisplayFunc`
Lors de la projection, l'image finale, avec une vue de 180°, est affichée sur la surface courbe



Définitions de la Scène 3D en OpenGL

Configuration des Canaux `spiInitialize`

4 Canaux =
4 rendus des points de vue latéraux, supérieur et inférieur
Chacun avec une vue de 90°

Les 4 images générées sont stockées comme textures

Les 4 textures sont placées sur un maillage sphérique. `spiFlush` génère l'image finale