

Projet IA54-IN54: Mise en correspondance d'images stéréoscopiques par une approche multi-agents

Béatrice Frey¹ Audrey Cholewka² Cyril Crassin¹

Université de Technologie de Belfort-Montbéliard.
¹UVs IA54 et IN54. ²UV IN54.

Soutenance de projet, Janvier 2006

1 Introduction

- Contexte
- Problématique

2 Méthodes proposées

- Principe
- Traitements graphiques
- Appariement multi-agents
- Modèle ASI
- Modèle ASM

3 L'application

- Contexte de travail, Outils et choix techniques
- Architecture générale
- Accélération matérielle
- Plateforme Multi-Agents
- Architecture objet

4 Conclusion

- Résultats
- Bilan
- Limites et Perspectives

Contexte

Contexte global

- Laboratoire SET de l'UTBM: Projet "Véhicule Intelligent".
- But: concevoir des systèmes embarqués pour l'aide à la navigation d'un véhicule en environnement urbain.



Contexte

Contexte global

- Laboratoire SET de l'UTBM: Projet "Véhicule Intelligent".
- But: concevoir des systèmes embarqués pour l'aide à la navigation d'un véhicule en environnement urbain.

Contexte technique

- Élément important: dispositif de vision artificielle.
- Utilisation d'un dispositif de vision stéréo.
- Besoins en moyens de traitement temps réels des données de vision.
 - Reconstruction 3D de la scène observée.



Problématique

Reconstruction 3D

- Déterminer la profondeur des éléments d'une scène à partir du décalage observé entre leur position dans les images droites et gauches acquises (disparité) et les paramètres du capteur stéréoscopique (longueur focale et distance inter-oculaire).
 - Identifier les points homologues dans les deux images, représentant le même point de la scène.
 - Problème: algorithmes de mise en correspondance extrêmement coûteux (spécialement en 2D)

Problématique

Reconstruction 3D

- Déterminer la profondeur des éléments d'une scène à partir du décalage observé entre leur position dans les images droites et gauches acquises (disparité) et les paramètres du capteur stéréoscopique (longueur focale et distance inter-oculaire).
 - Identifier les points homologues dans les deux images, représentant le même point de la scène.
 - Problème: algorithmes de mise en correspondance extrêmement coûteux (spécialement en 2D)

Idée

- Mettre en oeuvre un système basé sur le paradigme multi-agents pour traiter ce problème.

- 1 Introduction
 - Contexte
 - Problématique
- 2 **Méthodes proposées**
 - Principe
 - Traitements graphiques
 - Appariement multi-agents
 - Modèle ASI
 - Modèle ASM
- 3 L'application
 - Contexte de travail, Outils et choix techniques
 - Architecture générale
 - Accélération matérielle
 - Plateforme Multi-Agents
 - Architecture objet
- 4 Conclusion
 - Résultats
 - Bilan
 - Limites et Perspectives

Principe général

Contraintes du problème

- Traitement d'images matricielles 2D.
- Capteur stéréo idéal.
 - Contraintes épipolaires.

Principe général

Contraintes du problème

- Traitement d'images matricielles 2D.
- Capteur stéréo idéal.
 - Contraintes épipolaires.

3 étapes

- Traitements graphiques:
 - Extraire les éléments caractéristiques (primitives ou *features*) des images.
- Appariement:
 - Associer les primitives de l'image gauche avec celles de l'image droite afin d'en calculer la disparité.
- Reconstruction 3D et Visualisation:
 - En déduire la profondeur de chaque primitive et la visualiser.

Étapes de traitement pour l'extraction de primitives

Chargement des images gauche et droite

- Chargement au format jpeg.



Étapes de traitement pour l'extraction de primitives

Chargement des images gauche et droite

- Chargement au format jpeg.



Lissage gaussien

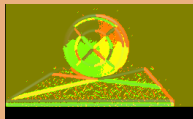
- Filtre de convolution.
- Permet la réduction du bruit parasite.



Étapes de traitement suite...

Détection de contours

- Extraction de gradients (Vecteurs 2D) pour la détection de caractéristiques.
 - Filtres de *Roberts*, *Prewitt* et *Sobel* testés.
 - Compromis qualité/temps de calcul à trouver.
- Résultat: image RGBA avec encodage dans les pixels.
 - Stockage composantes X et Y du vecteur, norme et intensité lumineuse d'origine.



Étapes de traitement suite...

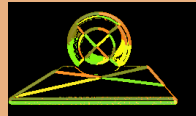
Détection de contours

- Extraction de gradients (Vecteurs 2D) pour la détection de caractéristiques.
 - Filtres de *Roberts*, *Prewitt* et *Sobel* testés.
 - Compromis qualité/temps de calcul à trouver.
- Résultat: image RGBA avec encodage dans les pixels.
 - Stockage composantes X et Y du vecteur, norme et intensité lumineuse d'origine.



Seuillage

- Elimination des gradients non significatifs.
- Les gradients restants constituent des primitives prêtes à être appareillées.



Appariement multi-agents

Idée

- Développer un modèle basé sur des agents réactifs.
- Faire émerger une configuration correcte d'appariement des primitives à partir du comportement individuel d'un ensemble d'agents.

Appariement multi-agents

Idée

- Développer un modèle basé sur des agents réactifs.
- Faire émerger une configuration correcte d'appariement des primitives à partir du comportement individuel d'un ensemble d'agents.

Principe

- Associer un agent à chaque primitive de l'image gauche.
- Chaque agent recherche dans l'image droite la primitive correspondant à celle à laquelle il est associé dans l'image gauche.

Appariement multi-agents

Idée

- Développer un modèle basé sur des agents réactifs.
- Faire émerger une configuration correcte d'appariement des primitives à partir du comportement individuel d'un ensemble d'agents.

Principe

- Associer un agent à chaque primitive de l'image gauche.
- Chaque agent recherche dans l'image droite la primitive correspondant à celle à laquelle il est associé dans l'image gauche.

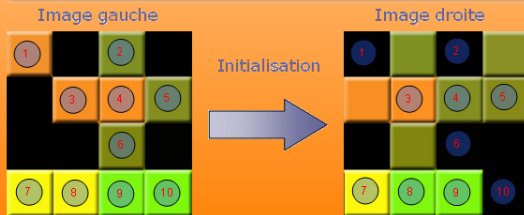
Modèles proposés

- Prise en compte de 5 contraintes locales et globales: contraintes *géométriques, d'orientation, d'unicité, d'ordre* et de *continuité de la disparité*.
- 2 Modèles:
 - Modèle ASI: Agents Sur Image.
 - Modèle ASM: Agents Sur Matrice.

Modèle ASI: Agents Sur Image

Initialisation

- Environnement géographique discret (grille 2D) constitué des primitives de l'image droite.
 - Utilisation directe de l'image contenant les gradients.
 - Chaque case est constituée d'une primitive (éventuellement nulle) et peut contenir un seul agent.
- Un agent est créé par primitive non nulle de l'image gauche qui lui est associée.
- Chaque agent est ensuite placé dans l'environnement (image droite) à la position de sa primitive.

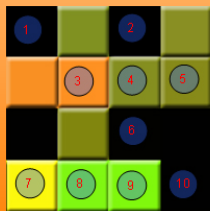


Principe ASI

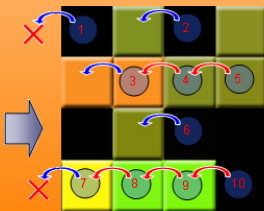
Principe

- Inspiration de l'éco-résolution et du modèle satisfaction-altruisme
- L'agent recherche par déplacement le meilleur appariement avec sa primitive de base.
 - Déplacement par lignes: configuration idéale du capteur stéréo (*contraintes épipolaires*).
 - Calcul de satisfaction: basé sur la distance euclidienne entre les 2 primitives + prise en compte de la différence de disparité.
 - Mécanisme d'attaque: création d'une force globale.

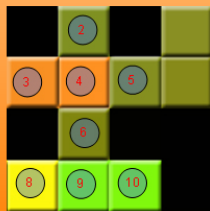
Initialisation



Test de satisfaction /
Attaque



Déplacement



Modèle Agent ASI

Comportement

Initialisation : calcul de satisfaction sur la case courante \rightarrow curSat.

Live():

- Test de la case à gauche de l'agent (*contrainte d'ordre*) \rightarrow testSat.
- Si $(\text{testSat} + \text{altCoef} * \text{oppGain}) \geq \text{curSat}$:
 - Si un agent O est présent sur la case,
 - Si EstAttaqué ET $O.\text{curSat} > \text{SEUIL-SAT}$, Pop();
 - Sinon Attaquer(O, $\text{abs}(\text{curSat} - \text{testSat}) + \text{oppGain}$); (*contrainte d'unicité*).
 - Sinon aller sur case. EstAttaqué=Faux, oppGain=0;
- Sinon,
 - Si EstAttaqué, Pop();

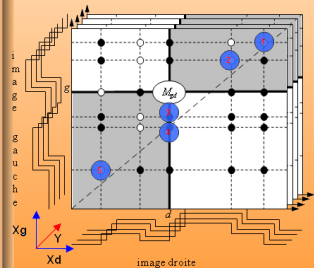
Attaquer(Agent, Gain):

- EstAttaqué=Vrai, oppGain=Gain;

Modèle ASM: Agents Sur Matrice

Principe

- Après obtention du champ de primitives, une étape de pré-traitement est effectuée pour générer une matrice de mise en correspondance permettant d'éliminer les appariements invalides (*contraintes géométrique et d'angle*).
- Expression différente du problème, se ramène à une méthode similaire.
 - Les agents sont positionnés sur le premier appariement valide à droite de chaque ligne de la matrice.
 - Ils se déplacent par lignes de la matrice et attaquent les agents sur leur colonne (*contrainte d'ordre*).
- Permet une optimisation des calculs de satisfaction.



- 1 Introduction
 - Contexte
 - Problématique
- 2 Méthodes proposées
 - Principe
 - Traitements graphiques
 - Appariement multi-agents
 - Modèle ASI
 - Modèle ASM
- 3 **L'application**
 - Contexte de travail, Outils et choix techniques
 - Architecture générale
 - Accélération matérielle
 - Plateforme Multi-Agents
 - Architecture objet
- 4 Conclusion
 - Résultats
 - Bilan
 - Limites et Perspectives

Contexte de travail

Outils

- Développement C++.
- Bibliothèque graphique OpenGL pour la visualisation et l'accélération matérielle.

Contexte de travail

Outils

- Développement C++.
- Bibliothèque graphique OpenGL pour la visualisation et l'accélération matérielle.

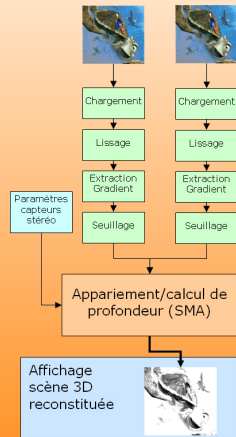
Contraintes

- Portabilité du développement.
- Optimisation maximale des algorithmes pour permettre un traitement temps réel.
- Développements des algorithmes au sein d'une architecture objet générique et extensible afin de garantir leur réutilisabilité.

Architecture de traitement

Pipeline de visualisation

- Mise en oeuvre sous la forme d'un pipeline de traitement graphique.
- Ensemble de modules indépendants inter-connectés.
- Traitement de données de type image 1D, 2D ou 3D et *framework* générique.



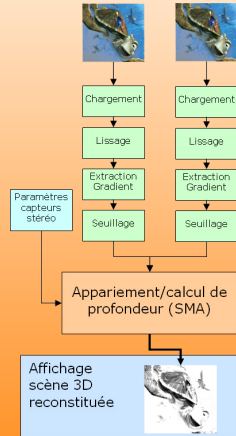
Architecture de traitement

Pipeline de visualisation

- Mise en oeuvre sous la forme d'un pipeline de traitement graphique.
- Ensemble de modules indépendants inter-connectés.
- Traitement de données de type image 1D, 2D ou 3D et *framework* générique.

3 Familles de modules

- Extraction de primitives: chargement et traitement d'images.
- Algorithmes multi-agents pour la mise en correspondance.
- Visualisation et interaction.



Accélération Matérielle par GPU

Idée

- Utiliser la puissance de calcul vectorielle des GPU modernes pour le traitement d'images.
 - Unité programmable de traitement de fragments. Conçu pour les opérations vectorielles.
 - Jusqu'à 40GFlops sur NV40, 2x plus puissant que les derniers modèles de CPU.
 - Mais dédié normalement à des opérations d'affichage.
- Utilisation comme noyau d'application de filtre sur des images 1D, 2D ou 3D.



Implémentation

- Module générique de filtrage GPU
 - Intégration dans le framework graphique générique.
 - Adaptation des types *Images* pour la localisation en mémoire graphique.
- Utilisé pour les opérations de pré-traitement graphique (lissage, extraction de contours, seuillage).

Plateforme Multi-Agents

Plateforme

- Framework générique de plateforme pour agents localisés.
- Ordonnanceur et noyau d'exécution pour l'éco-résolution.

Plateforme Multi-Agents

Plateforme

- Framework générique de plateforme pour agents localisés.
- Ordonnanceur et noyau d'exécution pour l'éco-résolution.

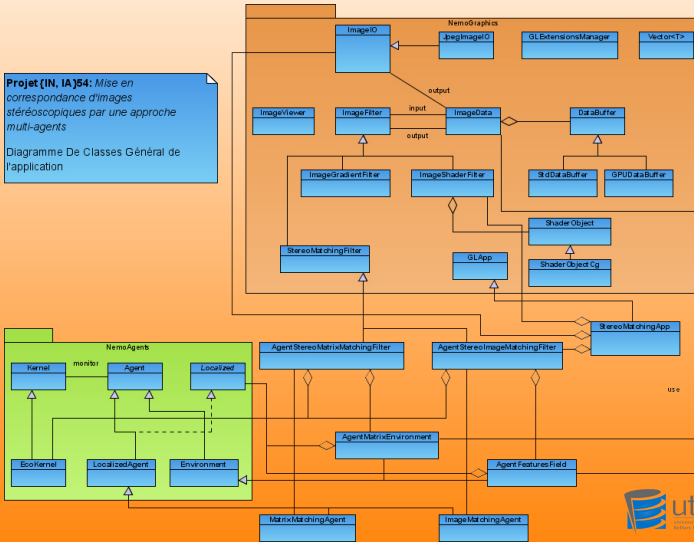
Architecture des Agents

- Mécanismes d'attaque non procéduraux (garantie l'équilibrage de l'ordonnancement).
- Utilisation d'états interne (Recherche, Attaqué) et de valeurs partageables (passage de message trop coûteux).

Diagramme de classes

Projet (IN, IA)54: Mise en correspondance d'images stéréoscopiques par une approche multi-agents

Diagramme De Classes Général de l'application



- 1 Introduction
 - Contexte
 - Problématique
- 2 Méthodes proposées
 - Principe
 - Traitements graphiques
 - Appariement multi-agents
 - Modèle ASI
 - Modèle ASM
- 3 L'application
 - Contexte de travail, Outils et choix techniques
 - Architecture générale
 - Accélération matérielle
 - Plateforme Multi-Agents
 - Architecture objet
- 4 Conclusion
 - Résultats
 - Bilan
 - Limites et Perspectives

Performance comparée des méthodes

Temps moyens, scène réelle, 400x300, 7880 Agents

Méthode	Initialisation	Itération	Temps stabilisation
ASI	68ms	6.5ms	3000ms
ASM	250ms	4.1ms	400ms



Performance comparée des méthodes

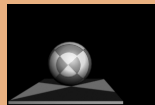
Temps moyens, scène réelle, 400x300, 7880 Agents

Méthode	Initialisation	Itération	Temps stabilisation
ASI	68ms	6.5ms	3000ms
ASM	250ms	4.1ms	400ms



Temps moyens, scène artificielle, 300x200, 1553 Agents

Méthode	Initialisation	Itération	Temps stabilisation
ASI	22ms	2.3ms	400ms
ASM	47ms	1.7ms	180ms



Bilan

Apports

- 2 Méthodes agents pour la résolution du problème de l'appariement stéréo.
- Utilisation du GPU: gain de 50% sur les étapes de traitement graphique.
- Illustration de l'applicabilité du paradigme agent pour les traitements d'images.
- Essai et test de divers critères et fonctions de satisfaction, ouverture sur des développements futurs.
- Base d'un framework ouvert pour le développement d'applications de traitements graphiques orientées agent.

Bilan

Apports

- 2 Méthodes agents pour la résolution du problème de l'appariement stéréo.
- Utilisation du GPU: gain de 50% sur les étapes de traitement graphique.
- Illustration de l'applicabilité du paradigme agent pour le traitements d'images.
- Essai et test de divers critères et fonctions de satisfaction, ouverture sur des développements futurs.
- Base d'un framework ouvert pour le développement d'applications de traitements graphiques orientées agent.

Bilan humain

- Investissement fort sur un sujet de recherche novateur et encore très ouvert.
- Mise en application des méthodes multi-agent et des principes de traitement d'images abordés en cours.

Limites et Perspectives

Limites de l'approche

- Configuration idéale du capteur stéréo (alignement et parallélisme des droites épipolaires).
- Convergence encore trop lente pour le traitement temps réel d'un flux d'images.
- Résultats perfectibles de la méthode ASM.

Limites et Perspectives

Limites de l'approche

- Configuration idéale du capteur stéréo (alignement et parallélisme des droites épipolaires).
- Convergence encore trop lente pour le traitement temps réel d'un flux d'images.
- Résultats perfectibles de la méthode ASM.

Perspectives d'évolution

- Extension au traitement d'un flux d'images:
 - Pour profiter de l'adaptabilité des agents.
 - Mais problème complexe.
- Extension à un capteur stéréo non idéal (sans contraintes épipolaires).
- Mise en oeuvre en environnement parallèle (cluster de traitement graphique orienté agent ?).

Traitements graphiques

Filtres de convolution

- Traitement d'une image (matrice de pixels) par une matrice (généralement 3×3) appelée Noyau via un produit de convolution.
- Permet d'appliquer divers traitements sur les images de manière extrêmement flexible.

Traitements graphiques

Filtres de convolution

- Traitement d'une image (matrice de pixels) par une matrice (généralement 3x3) appelée Noyau via un produit de convolution.
- Permet d'appliquer divers traitements sur les images de manière extrêmement flexible.

Implémentations

- Élimination du bruit parasite: filtre de lissage *gaussien*.
- Filtres de détection de contour par extraction de gradient.
 - Filtres de *Roberts*, *Prewitt* et *Sobel* testés.
 - Compromis qualité/temps de calcul à trouver.

Traitements graphiques

Filtres de convolution

- Traitement d'une image (matrice de pixels) par une matrice (généralement 3x3) appelée Noyau via un produit de convolution.
- Permet d'appliquer divers traitements sur les images de manière extrêmement flexible.

Implémentations

- Élimination du bruit parasite: filtre de lissage *gaussien*.
- Filtres de détection de contour par extraction de gradient.
 - Filtres de *Roberts*, *Prewitt* et *Sobel* testés.
 - Compromis qualité/temps de calcul à trouver.

$$\begin{array}{|c|c|c|c|c|} \hline 100 & 100 & 100 & 100 & 100 \\ \hline 100 & 100 & 50 & 50 & 100 \\ \hline 100 & 100 & 100 & 100 & 100 \\ \hline 100 & 100 & 100 & 100 & 100 \\ \hline 100 & 100 & 100 & 100 & 100 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline & & \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline 100 & 100 & 100 & 100 & 100 \\ \hline 100 & 100 & 50 & 50 & 100 \\ \hline 100 & 100 & 50 & 100 & 100 \\ \hline 100 & 100 & 100 & 100 & 100 \\ \hline 100 & 100 & 100 & 100 & 100 \\ \hline \end{array}$$