



Interprétation d'images

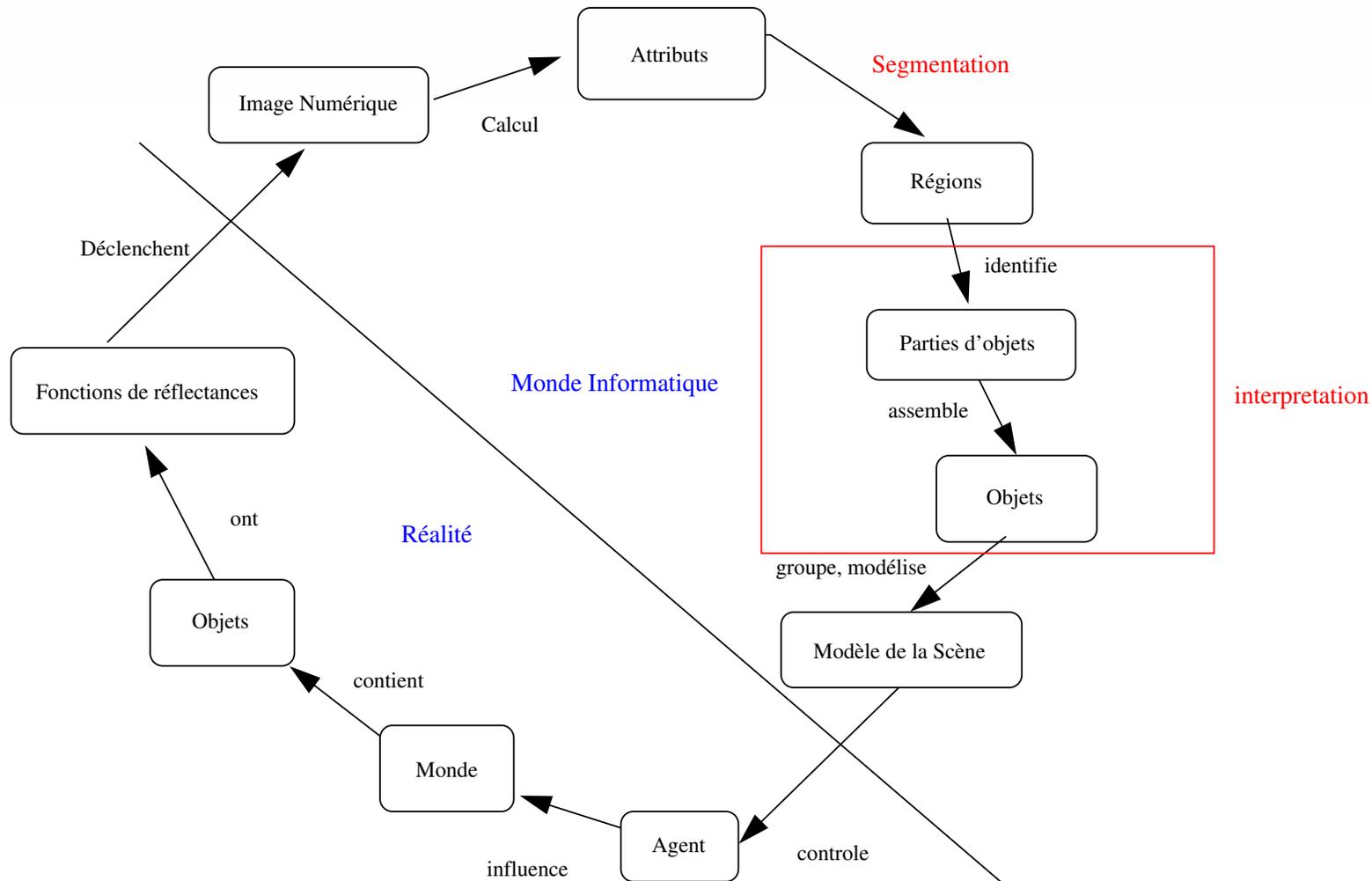
Luc Brun

Plan



1. De quoi parle t'on,
2. Les systèmes à base de règles,
 - Méthode de Hu,
 - Modèle de Levine,
 - Méthode de DarWish.
3. Les graphes sémantiques
 - Les algorithmes d'arc consistency
 - L'algorithme AC-4BC
4. Reconnaissance structurelle de scènes/objets
 - Introduction,
 - Les méthodes «algorithmiques»,
 - Les méthodes «numériques».

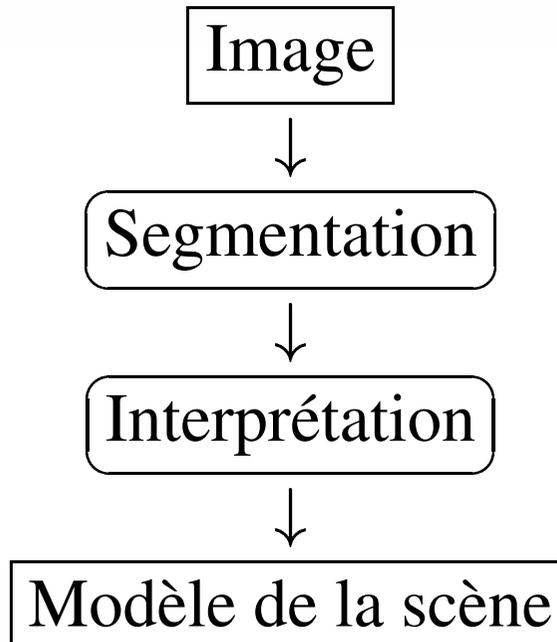
Le cycle du traitement



Segmentation/Interprétation



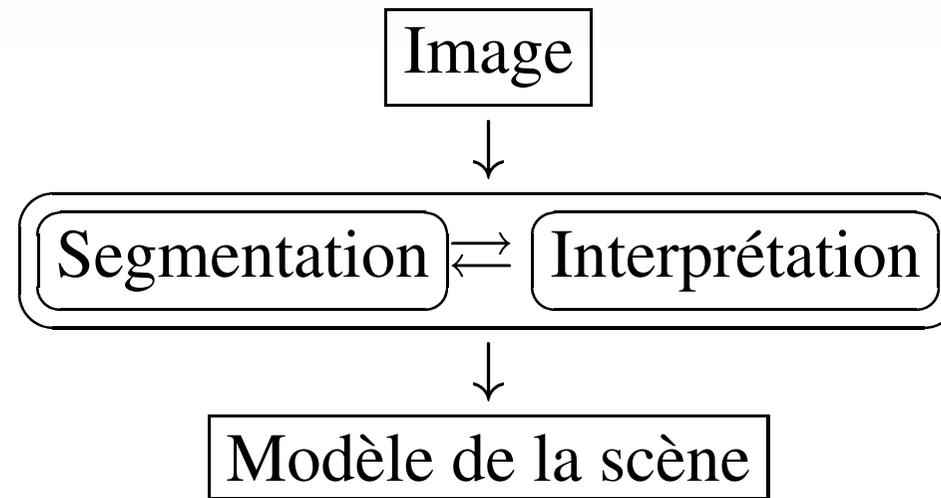
- Doit on avoir ceci :



Segmentation/Interprétation



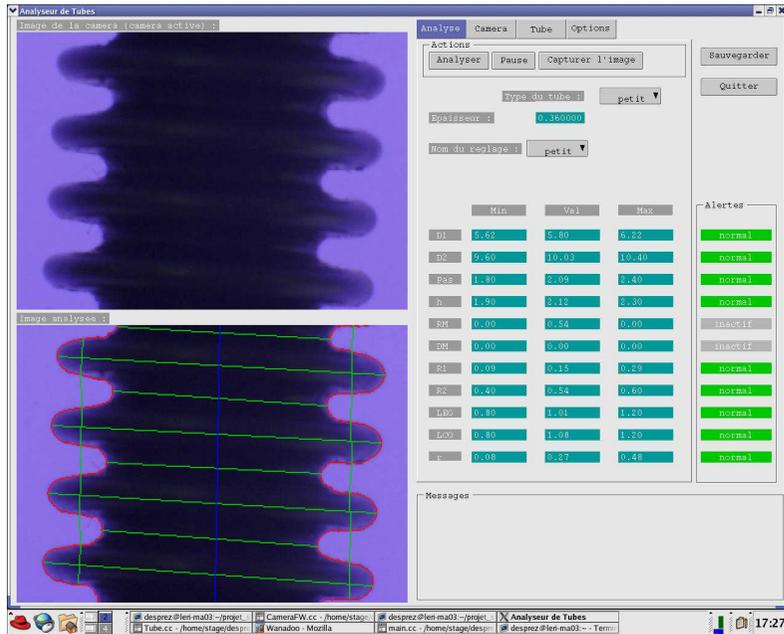
■ ou cela :



Complexité de la scène



Scène Simple



Segmentation «triviale»,
interprétation «facile»

Scène Complexe



Interprétation et
segmentation ardues.

Liens Segmentation/Interprétation



- Une bonne interprétation nécessite une bonne segmentation,
- une bonne segmentation requiert une bonne interprétation.

Comment intégrer des connaissances dans le processus de segmentation ?

Une première approche



Z.P Hu, T. Pun and C. Pellegrini, *An Expert System for guiding image segmentation*, Vol. 14, No. 1, 1990

- Combinaison de deux systèmes.
 - Un logiciel de traitement d'images (Labo Image),
 - Un système expert.
- Le logiciel dialogue avec le système expert pour déterminer une séquence «optimale» d'opérateurs.

Exemple de session (1/3)



U : compter les éléments

E : ■ Les éléments sont ils isolés $([-1..1])$? .5

■ les éléments sont ils de taille uniforme ? .2

■ l'image est elle bruitée ? .8

■ voulez vous tous les éléments ou seulement ceux d'une certaine taille ? tous

■ Existe t'il un fond non homogène : .4

Utilisez un filtre linéaire de grande taille pour avoir une version passe bas de l'image et soustrayez la à l'image originale

Exemple de session (2/3)



E : essayons une détection par morphologie mathématique : **Utilisez un filtre médian de taille $N \times N$ si les défauts à éliminer sont de taille $\frac{N \times N}{2}$.**

⋮

E : êtes vous satisfait (o/n) ? n

E : Détectons les éléments par leurs contours :

- Niveaux de gris maximal/minimal dans l'image : 128, 82
- Nombre de niveaux de gris : 255
- Voulez vous des contours fermés : n

Exemple de session (3/3)



⋮

Le système propose \neq détecteurs de contours jusqu'à ce que l'utilisateur soit satisfait.

⋮

Le système aide l'utilisateur à fixer le seuil pour déterminer les contours

⋮

Le système aide l'utilisateur à supprimer les contours indésirables,

⋮

Le système propose les éléments

Conclusion



- Le système expert et le logiciel de segmentation d'images sont disjoints :
 - Le système ne fait que recommander des actions (effectuées par l'utilisateur).
 - L'observation des images est faite par l'utilisateur,
 - Récupération de la «vision» du problème par l'utilisateur.
- Limitations :
 - Qualité de la segmentation faiblement appréhendée,
 - Description de l'image limitée,
 - Base de connaissances limitée.

Le modèle de Lévine



- **Ambition :**

Définir une méthode de segmentation **généraliste** intégrant un maximum de connaissances a priori sur ce que constitue une «bonne» segmentation et les méthodes à appliquer pour y parvenir.

- **Codage : Des ensembles**

- d'objets,
- de mesures sur ces objets,
- De règles de transformations (e.g. découpe/fusion),
- De règles définissant une stratégie de traitement.

Levine : Les objets



- Régions : définissent une partition de l'image,
- Contours (line) : ensemble de courbes résultat de l'application d'un opérateur gradient et d'un seuillage.
 - Relations :
 - adjacence de régions (R/R),
 - devant, derrière, parallèle à ligne (L/L),
 - région à gauche ou à droite d'une ligne (R/L),
 - ligne touchant ou intersectant une région (L/R).
- Zones (area) : regroupement de régions vérifiant un certain critère.
 - Types de zones :
 - lisses,
 - texturées,
 - créés par des lignes définissant un contour fermé.

Mesures des objets



- L'uniformité (1/2) :

- Erreur quadratique d'une région :

$$SE(R) = \sum_{x \in R} (x - \mu_R)^2, \mu_R = \frac{1}{|R|} \sum_{x \in R} x$$

- Erreur quadratique maximale :

$$SE_{max}(R) = \frac{1}{2} |R| (x_{max} - x_{min})^2$$

- Homogénéité d'un zone :

$$U_\alpha = 1 - \frac{1}{N} \left(\sum_{R_j \in \alpha} SE(R_j) \right) \quad N = \sum_{R_j \in \alpha} SE_{max}(R_j)$$

Mesures des objets



- L'uniformité (2/2) :

- Note : $U_\alpha \in [0, 1]$.

$$\begin{aligned}U_\alpha &= 1 - \frac{1}{N} \left(\sum_{R_j \in \alpha} SE(R_j) \right) \\&= 1 - \frac{2}{(x_{max} - x_{min})^2 \sum_{R_j \in \alpha} |R_j|} \left(\sum_{R_j \in \alpha} SE(R_j) \right) \\&= 1 - \frac{2}{(x_{max} - x_{min})^2 |\alpha|} \left(\sum_{R_j \in \alpha} SE(R_j) \right) \\&= 1 - \sum_{R_j \in \alpha} u_{j,\alpha}\end{aligned}$$

avec :

$$u_{j,\alpha} = \frac{2SE(R_j)}{(x_{max} - x_{min})^2 |\alpha|} = \frac{|R_j|}{|\alpha|} \frac{SE(R_j)}{SE_{max}(R_j)}$$

Mesure des objets



- Le contraste (1/2) :

- Entre deux régions R_i, R_j :

$$c_{ij} = \frac{|\mu_{R_j} - \mu_{R_i}|}{|\mu_{R_j} + \mu_{R_i}|} = c_{ji} \in [0, 1]$$

- D'une région :

$$c_i = \sum_{R_j \in V(R_i)} p_{ij} c_{ij} \text{ avec } p_{ij} \approx \frac{|R_i \cap R_j|}{|R_i|}$$

- Note :

$$\sum_{R_j \in V(R_i)} p_{ij} = 1 \Rightarrow c_i \in [0, 1]$$

Mesure des objets



- Le contraste (2/2) :
 - Mesure du contraste d'une zone :

$$C_\alpha = \frac{\sum_{R_i \in \alpha} v_i c_i}{\sum_{R_i \in \alpha} v_i}$$

avec

$$v_j = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(|R_j| - |R_{opt}|)^2}{\sigma^2}}$$

Sensibilité optimum au contraste pour $|R_j| = |R_{opt}|$ (Données psycho-visuelles).

Mesures des objets



■ Contrastes de lignes (1/2) :

Une «bonne» ligne :

- Sépare des régions d'attributs différents ou
- possède un fort gradient

Soient :

$$F_l = \sum_{R_i} p_{il} \mu(R_i) \quad F_r = \sum_{R_j} p_{jl} \mu(R_j)$$

R_i, R_j respectivement à droite et à gauche de la ligne.

On défini :

$$d_j = \frac{|F_l - F_r|}{|F_l + F_r|} \quad \text{et} \quad g_j = \frac{G_j}{(x_{max} - x_{min})}$$

G_j gradient moyen le long de la ligne L_j .

Mesure des objets



- Contraste de lignes (2/2) :

- Synthèse de g_j et d_j :

$$h_j = \begin{cases} g_j & \text{si } g_j > 3d_j \text{ et } d_j < \epsilon \\ d_j & \text{sinon} \end{cases}$$

- Contraste de ligne de la zone :

$$H_\alpha = \frac{\sum_{L_j \in \alpha} |L_j| h_j}{\sum_{L_j \in \alpha} |L_j|}$$

Mesure des objets



■ Connectivité des lignes :

- Une extrémité de ligne est fermée si elle est en contact avec elle même ou une autre ligne.
- Soit t_i : nombre d'extrémités fermées de L_j ($t_j \in \{0, 1, 2\}$).

$$T_\alpha = \frac{\sum_{L_j \in \alpha} |L_j|^{\frac{t_j}{2}}}{\sum_{L_j \in \alpha} |L_j|} \in [0, 1]$$

- Idée : une segmentation est incomplète si de nombreuses lignes ne sont pas fermées.

Mesures des objets



■ Mesure de texture :

Une zone texturée est implicitement caractérisée par un grand nombre de régions (resp. de lignes) d'attributs différents (resp. de fort contraste).

■ Densité de régions :

$$R_{\alpha} = \frac{\frac{NR_{\alpha}}{|\alpha|}}{\frac{NR_I}{|I|}} = \frac{NR_{\alpha}|I|}{NR_I|\alpha|}$$

NR_{α} : Nombre de régions de α , NR_I : Nombre total de régions.

■ Densité de lignes :

$$L_{\alpha} = \frac{\frac{NL_{\alpha}}{|\alpha|}}{\frac{NL_I}{|I|}} = \frac{NL_{\alpha}|I|}{NL_I|\alpha|}$$

Vecteurs de mesures



- Si m attributs pour chaque pixel :

$$P_\alpha = [(U_\alpha^i)_{i \in \{1, \dots, m\}}, (C_\alpha^i)_{i \in \{1, \dots, m\}}, (H_\alpha^i)_{i \in \{1, \dots, m\}}, T_\alpha, R_\alpha, L_\alpha]$$

Vecteur de dimension $M = 3m + 3$.

- Caractérisation d'une zone :

$$\psi_\alpha \in \{lisse, \textit{texturé}, \textit{borné}\}$$

Types de zones



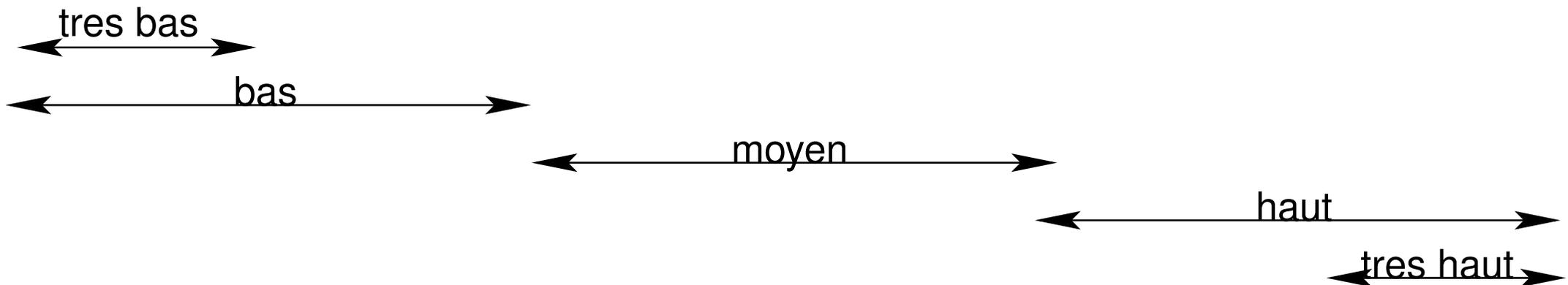
■ borné : une ligne fermée.

■ lisse :

R_α, L_α : bas,
 $\forall i \in \{1, \dots, m\} U_\alpha$ moyen

■ texturé :

$R_\alpha, L_\alpha, :$ haut,
 $\forall i \in \{1, \dots, m\} C_\alpha, H_\alpha$ haut



Ensemble des zones



- L'ensemble $\{\alpha_0, \dots, \alpha_n\}$ des zones est un ensemble **dynamique**.
- Les zones sont :
 - Modifiées : par ajout/suppression de régions/lignes,
 - Supprimées/ajoutées
 - changées d'étiquettes (lisse, texturé, borné).
- α_0 , zone par défaut : regroupe l'ensemble des régions qui ne peuvent être classées dans une zone lisse, texturé ou bornées. α_0 évolue également.

Stratégie



- On dispose :
 - D'un ensemble de données (régions, lignes, zones),
 - D'un ensemble de règles de transformation.

- Questions :
 - Où va t'on ? (Quelles données doit on considérer).
 - Quelle zone,
 - Quelles données dans la zone.
 - Que fait on une fois que l'on y est ? (quelles règles appliquer).

- Question supplémentaire :
 - Quelle donnée/règle appliquer après l'application d'une règle sur une donnée ?

Stratégie



- Question sous-jacente : Comment qualifier une «bonne» segmentation ?
- Critère flou :
 - Une bonne segmentation est une partition de l'image en régions homogènes possédant un fort contraste entre elles.*
- Maximisation d'un critère/d'une fonction de coût.

Sélection d'une zone (1/3)



- Une zone doit être (re)considérée si :
 1. Elle ne remplit pas un des objectifs (forte uniformité, fort contraste...).
 2. Elle n'a pas été examinée depuis longtemps par le système :

$$R_{\alpha} = \frac{\text{Nb zones visitées depuis } \alpha}{\text{Nb total de zones}}$$

Sélection d'une zone (2/3)



- Soient les zones $\Phi = \{\alpha_0, \dots, \alpha_N\}$ et

$$Q_\alpha = [(U_\alpha^i)_{i \in \{1, \dots, m\}}, (C_\alpha^i)_{i \in \{1, \dots, m\}}, (H_\alpha^i)_{i \in \{1, \dots, m\}}, T_\alpha, R_\alpha]$$

Q vecteur de taille $M = 3m + 2$.

- On défini :

$$\left. \begin{array}{l} \forall i \in \{1, \dots, N\} \\ \forall k \in \{1, \dots, M\} \end{array} \right\} Y_{ik} = 1 - Q_{\alpha_i}(k)$$

$Y_{ik} = 0$ si la zone α_i satisfait pleinement la contrainte k (uniformité, contraste...).

Sélection d'une zone (3/3)



- Une zone à une faible priorité si elle satisfait au moins 1 des critères.
- Priorité d'une zone :

$$\forall i \in \{1, \dots, N\} \quad d_i = \min_{j \in \{1, \dots, M\}} y_{ij}$$

- Zone sélectionné : Celle de plus forte priorité :

$$\alpha_i \mid d_i = \max_{j \in \{1, \dots, N\}} d_j$$

- Région/ligne sélectionné : fonction du type de la zone sélectionné. Par exemple :
 - si ψ_{α_i} = lisse alors sélectionner R de plus grand cardinal.

Stratégie dans une zone



- Choix de la prochaine région(S_1),
- Choix de la prochaine ligne(S_2),
- Répéter (ou pas) la même sélection (S_3),
- Agir sur les lignes ou les régions (S_4),
- Choix de la règle agissant sur des régions(S_5),
- Choix de la règle agissant sur des lignes(S_6),

Choix possibles produisent un vecteur de stratégie de dimension 6 :

$$S = [S_1, S_2, S_3, S_4, S_5, S_6]$$

Vecteur de stratégie : S_1, S_2



- S_1 : Choix de la prochaine région :
 - plus grande frontière commune (s_{11})
 - adjacente de plus grand cardinal (s_{12})
 - plus petite frontière commune (s_{13})
 - région adjacente de plus petit cardinal (s_{14})
 - région en haut à droite de la zone (s_{15})
 - région de plus grand label (s_{16})
- S_2 : choix de la prochaine ligne :
 - plus proche devant (s_{21})/derrière (s_{22}),
 - plus proche et parallèle (s_{23}),
 - plus longue (s_{24})/courte (s_{25}) à proximité,
 - plus fort (gradient) à proximité (s_{26}),
 - plus faible (gradient) à proximité (s_{27}),
 - en haut à droite (s_{28}),
 - plus grand label (s_{29}).

Vecteur de stratégie : S_3, S_4, S_5, S_6



- S_3 : mise à jour :
 - alterner(s_{31}),
 - garder la même région(s_{32}).
- S_4 : type d'objet :
 - Région(s_{41}),
 - ligne (s_{42}).
- S_5 : action sur région
 - découpe(s_{51}),
 - fusion(s_{52}).
- S_6 : action sur ligne
 - ajouter(s_{61}),
 - supprimer(s_{62}).

Chaque variable S_i peut prendre un certain nombre d'états s_{ik} . L'instanciation de chaque S_i (et donc de S) définit une stratégie.

Détermination du vecteur Stratégie



- Problème : Étant donné α (donc ψ_α) et P_α comment déterminer S ?
- Influence de P_α sur s_{ij} :

	U	C	H	T	NR	NL
plus grande frontière commune	+	+			-	
adjacente de plus grand cardinal	-	-			+	
plus petite frontière commune	+	-			+	
région adjacente de plus petit cardinal	-	+			-	

- Influence de ψ_α, P_α sur S_i :

S_i	ψ_α	U	C	H	T	R	L
choix Region	lisse	VH	H	VL	VL	M	VL
	texturé	H	VH	VL	VL	M	VL
	borné	H	H	VL	VL	H	VL

Détermination de la stratégie



- Pour chaque s_{ik} construire le vecteur :

$$Y_k = [\mu_{i1}, \dots, \mu_{iM}]$$

μ_{ij} contribution de $P_\alpha(i)$ au choix de s_{ik} :

$$\mu_{ij} = \begin{cases} P_\alpha(i) & \text{Si influence positive} \\ 1 - P_\alpha(i) & \text{Si influence négative} \end{cases}$$

- Pour chaque s_{ik} :

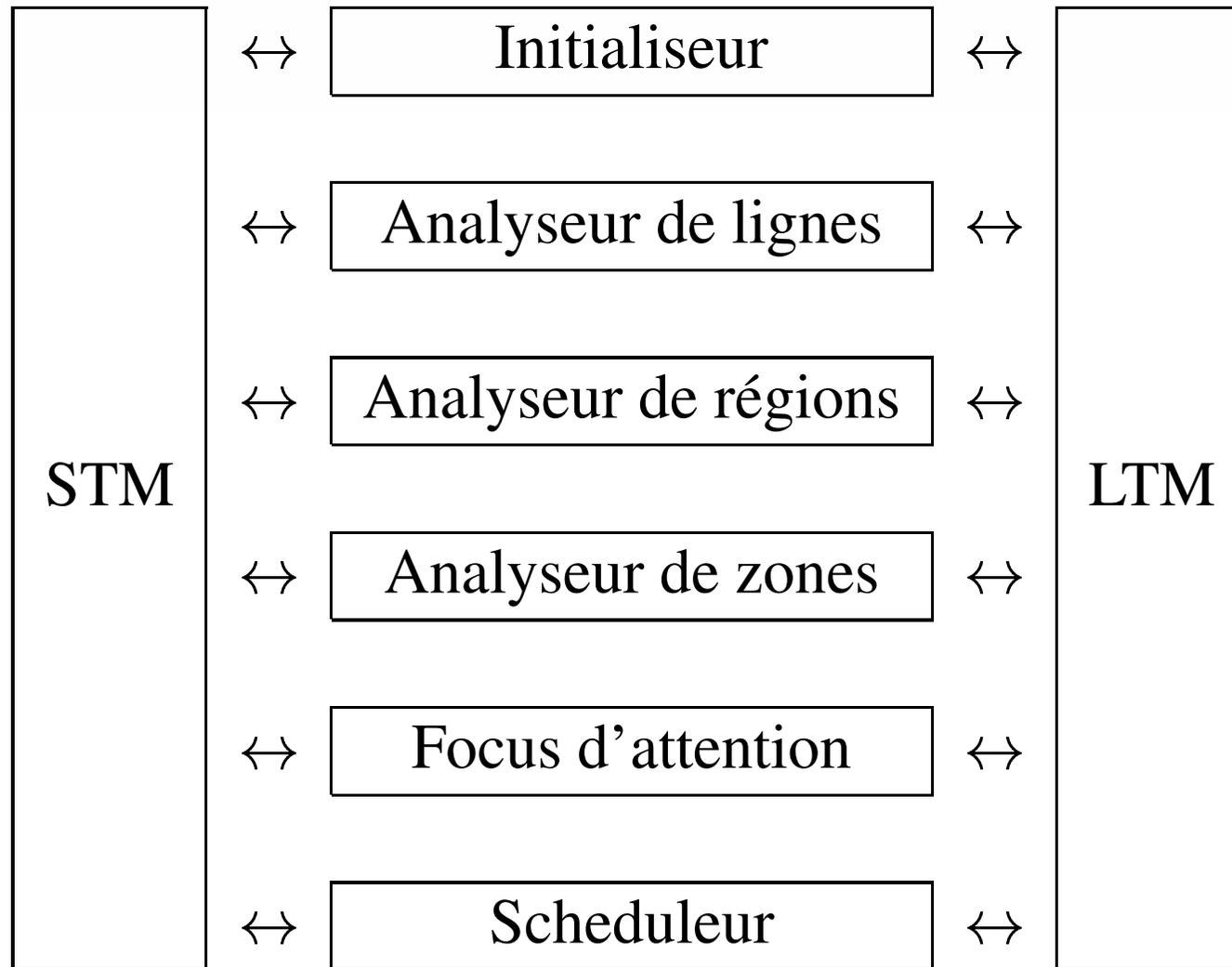
$$d_{ik} = \min_{j \in \{1, \dots, M\}} \mu_{ij}^{\gamma_{ij}}$$

γ_{ij} : importance relative de ψ_α dans l'utilisation de $P_\alpha(j)$ pour l'état i .

- Choix de l'état final de S_i : s_{ik} tel que :

$$d_{i,k} = \max_{j \in \{1, n_i\}} d_{ij}$$

Architecture du système



Exemples de règles (1/5)



■ Analyseur de régions

■ Règle (801) :

■ Si

1. il y a une faible différence sur la composante 1
2. il y a une faible différence sur la composante 2
3. il y a une faible différence sur la composante 3

■ Alors fusionner les deux régions.

■ Règle (905) :

■ Si

1. La variance 1 de région est haute
2. La variance 2 de région est haute
3. La variance 3 de région est haute
4. L'histogramme est bimodal

■ Alors découper la région en utilisant l'histogramme.

Exemples de règles (2/5)



■ Analyseur de lignes

■ Règle (1501) :

■ Si

1. l'extrémité terminale est ouverte
2. la distance avec la ligne de devant est basse

■ Alors joindre les deux lignes.



■ Règle (1507) :

■ Si

1. l'extrémité terminale est ouverte
2. le gradient moyen n'est pas bas
3. la longueur de la ligne n'est pas basse

■ Alors étendre la ligne en avant

Exemples de règles (3/5)



■ Analyseur de zone

■ Règle (401) :

■ Si

1. la taille de la région est haute
2. la variance 1 de la région est basse
3. la variance 2 de la région est basse
4. la variance 3 de la région est basse

■ Alors créer une zone lisse

■ Règle (402) :

■ Si

1. la zone est lisse
2. l'adjacence de la région n'est pas très basse
3. la taille de la région n'est pas très basse
4. les variances (1,2,3) de la région sont basses

■ Alors ajouter la région à la zone lisse.

Exemples de règles (4/5)



■ focus d'attention

■ Règle (401) :

■ Si

1. le processus précédent n'était pas focus

■ Alors

1. Obtenir la prochaine région
2. Obtenir la prochaine ligne

■ Règle (101) :

■ Si

1. le processus précédent n'était pas focus
2. le processus précédent n'était pas actif
3. le processus précédent ($\times 2$) n'était pas focus

■ Alors obtenir la prochaine zone

Exemples de règles (5/5)



■ Les méta règles.

■ Méta Règle (9) :

■ Si

1. le processus précédent était focus
2. le processus précédent était actif

■ Alors appliquer les règles d'analyse de régions

■ Règle (6) :

■ Si

1. le processus précédent était focus
2. le processus précédent n'était pas actif
3. le processus précédent ($\times 2$) était focus
4. le processus précédent ($\times 2$) n'était pas actif

■ Alors STOP

Conclusion



- Points positifs :
 - travail novateur (démarche, concepts),
 - Exploration des fondements sous-jacent au traitement d'images,
 - Mise en évidence des limites de l'approche
- Points «négatifs» :
 - Complexité du système de règles \Rightarrow difficulté à maintenir.
 - Quel sera l'effet de la 2001^e règle ?
 - Logiciel généraliste :
 - Les applications sont par natures particulières.
 - Si application particulière (Attributs particuliers, règles particulières, méta règles particulières...) : Il faut presque tout recommencer.

Projet immense mais trop ambitieux.

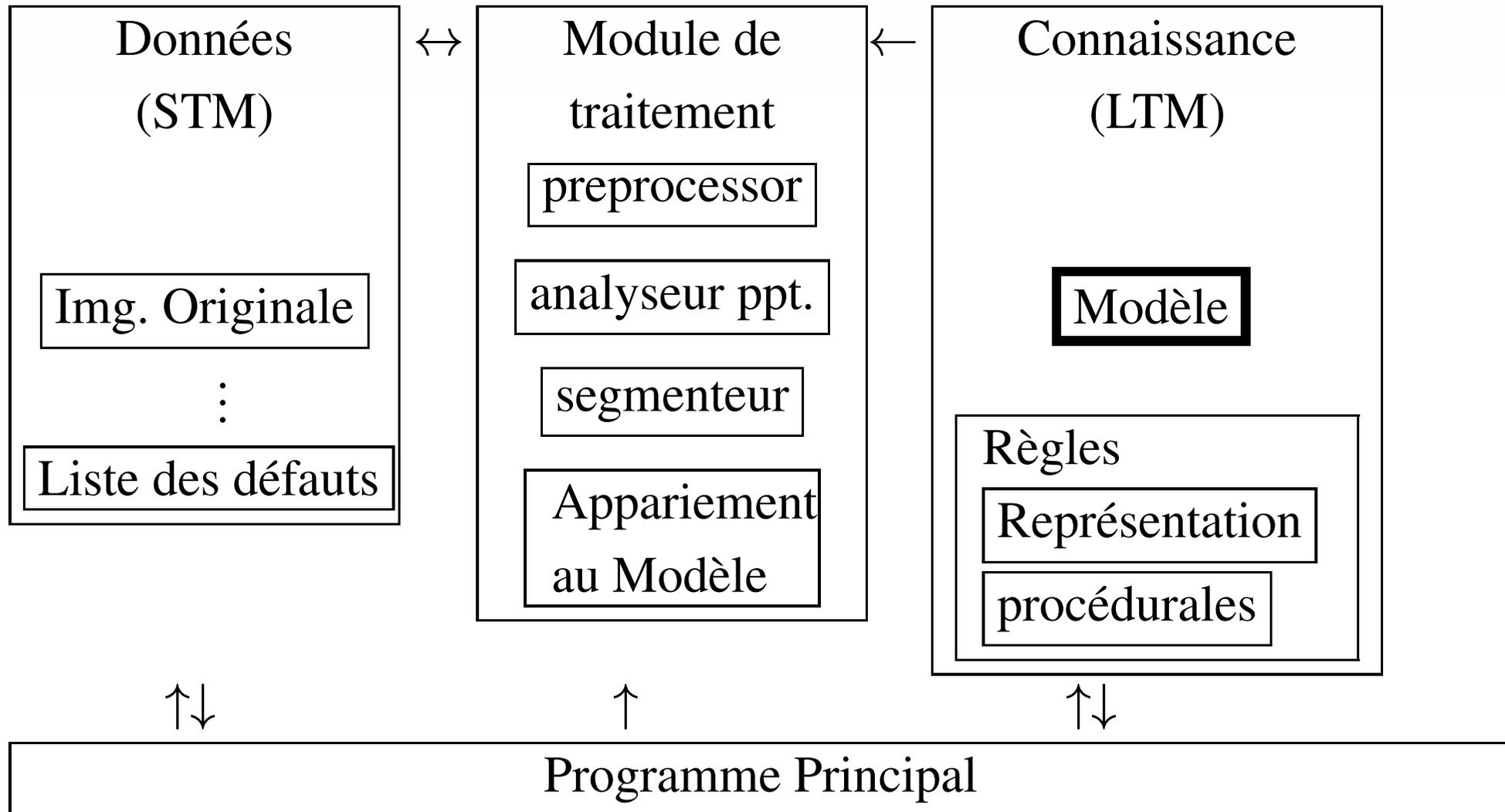
Redéfinition de la problématique



Ahmed M. DarWish and Anil K. Jain, *A Rule Based Approach for Visual Pattern Inspection*, Vol. 10, No 1, 1988

- Segmenter une classe précise d'images dans laquelle on peut :
 - définir les objets,
 - les propriétés vérifiées par ceux-ci,
 - les relations entre les objets.
- Problème de la mise en relation des objets avec le modèle.

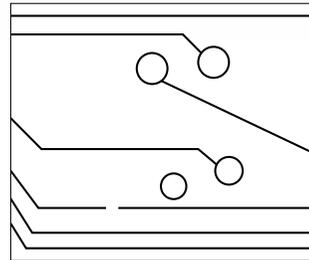
Architecture



Application au contrôle de cartes imprimées

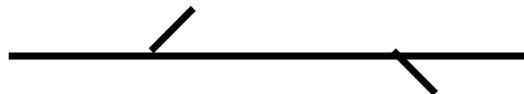


- lignes et connecteurs



- Segmentation (Module traitement, Morpho Math),

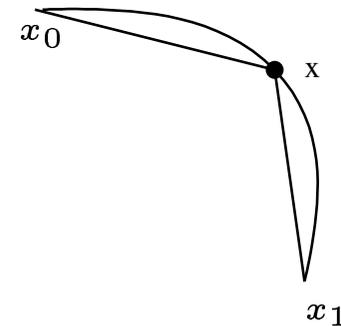
- Binariser l'image,
- Squelettiser les circuits,
- Supprimer les branches parasites issues de la squelettisation.



Structuration de l'image



- Décomposition de l'image en segments (traitement) :
 - plus long chemin tel que tout pixel est 2 voisins.
- Fusion/découpe/suppression de segments (règles)
 - Découpe de segments :
 - Si
 - segment non fermé ET
 - longueur inférieure au seuil ET
 - il existe x tel que $\widehat{(x_0, x, x_1)} > \epsilon$
 - Alors
 - découper le segment au point x



Fusion de segments



■ Exemple

■ Si

- les 2 segments ont une extrémité commune ET
- aucun autre segment est connecté à cette extrémité ET
- l'angle entre les deux segments est inférieur à un seuil

■ Alors

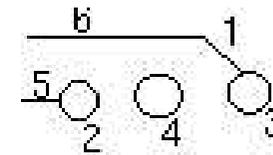
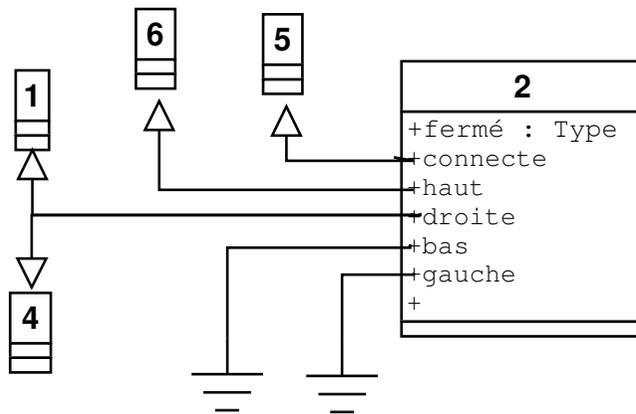
- Fusionner les deux segments



Création du graphe sémantique



- Décrit l'ensemble des segments avec
 - leurs attributs (direction, longueur, extrémités),
 - leurs types (ligne, courbe, cercle, courbe fermée)
 - les segments connectés,
 - les segments en haut (droite, bas, gauche)



Appariement avec un graphe modèle (1/3)



- Comparaison avec un graphe sémantique décrivant un circuit modèle.
- Mise en relation de chaque segment avec le segment correspondant du graphe modèle.

$$\text{confidence}(u, v) = \sum_{i=1}^m w_i T(f_{iu}, f_{iv}) + \sum_{j=1}^r \frac{w_{m+j}}{|N_{ju}|} \left[\sum_{k \in N_{ju}} \max_{l \in N_{jv}} \text{conf}(k, l) \right]$$

- u, v : noeuds,
- f_{iu} attribut i du noeud u , m nombre d'attributs,
- w_i poids de l'attribut i

$$T(f_{iu}, f_{iv}) = \begin{cases} 1, & \text{si } |f_{iu} - f_{iv}| < \epsilon \\ \frac{|f_{iu} - f_{iv}|}{\epsilon} & \text{sinon.} \end{cases}$$

Appariement avec un graphe modèle (2/3)



$$\text{confidence}(u, v) = \sum_{i=1}^m w_i T(f_{iu}, f_{iv}) + \sum_{j=1}^r \frac{w_{m+j}}{|N_{ju}|} \left[\sum_{k \in N_{ju}} \max_{l \in N_{jv}} \text{conf}(k, l) \right]$$

- r nombre de relations de voisinages,
- N_{ju} ensemble des voisins de u suivant la relation j (connecté, droite, gauche...)
- $\text{conf}(k, l)$: confidence (non réursive)

$$\text{conf}(k, l) = \sum_{i=1}^m T(f_{ik} f_{il}) \frac{w_i}{\sum_{j=1}^m w_j}$$

Appariement avec un graphe modèle (3/3)



- Pour chaque noeud u du graphe segmenté calculer son plus proche noeud v_u correspondant dans le graphe modèle :

$$confidence(u, v_u) = \max_{v \in V_m} confidence(u, v)$$

V_m : ensemble des noeuds du graphe modèle.

- Calculer la similarité des 2 graphes :

$$Sim \approx \frac{1}{|V_s|} \sum_{u \in V_s} confidence(u, v_u)$$

v_s ensemble des noeuds du graphe segmenté.

- Si la similarité est au dessus d'un seuil accepter l'appariement.

Détection des défauts



- Si l'appariement à réussis
 - Tester si des fusions sont possibles (segment artificiellement découpé).
 - Étiqueter chaque noeud segmenté en fonction du noeud modèle correspondant,
 - Appliquer les règles de vérification spécifiques à chaque segment étiqueté.
 - Largeur (min, max) de ligne,
 - Espacement des lignes...

Conclusion



- Avantages :
 - Problème beaucoup mieux cerné
 - Utilisation de graphe sémantiques et d'appariements de graphes pour faire la correspondance entre les données et le modèle.
 - Généralisable à une classe d'images
- Inconvénients :
 - Extension parfois difficile pour des applications hors du contrôle industriel.
 - Segmentation triviale, utilisation du graphe uniquement pour le contrôle.
- Est il possible d'utiliser un modèle pour aider la segmentation de scènes complexes ?

Ce que l'on voudrait faire



- Définir un graphe modèle contenant :
 - l'ensemble des objets à traiter avec leurs propriétés (contraintes internes) et
 - leurs relations (contraintes externes).
- Étant donné une sur segmentation d'une image, définir des regroupements de régions permettant de satisfaire l'ensemble des contraintes.



ciel

dessus↓, ↑dessous

montagne

dessus↓, ↑dessous

lac

dessus↓, ↑dessous

herbe

- Rem : reflet du ciel sur le lac

Initiation au pb d'Arc Consistency



Notre problème se modélise sous forme d'un graphe de contrainte $G = (V, E)$.

- Chaque sommet de V représente une étiquette (ciel, montagne...),
- Chaque arête orienté (i, j) représente une contrainte entre le noeud i et le noeud j .
- Calcul du domaine de chaque noeud :

Calcul du domaine de chaque noeud



procédure domaine ()

début

Pour chaque v dans V faire

Pour chaque x dans domaine D de v faire

si x ne satisfait pas une contrainte de v alors

enlever x de D

finsi

finpour

finpour

fin

Prise en compte des contraintes externes



procédure REVISE (V_i, V_j)

début

DELETE \leftarrow faux

Pour chaque X dans D_i faire

si il n'existe pas d'Y dans D_j tel que (X,Y) soit consistant **alors**

delete X de D_i

DELETE \leftarrow vrai

finsi

finpour

retourner DELETE

fin

- Pb : X pouvait être le support d'une contrainte...

AC-1



procédure AC-1 (G)

Déclaration Q : Liste

début

$Q \leftarrow \{(V_i, V_j) \text{ dans arcs}(G)\}$;

répéter

CHANGE \leftarrow faux

Pour chaque (V_i, V_j) **dans Q faire**

CHANGE \leftarrow REVISE(V_i, V_j) ou CHANGE

finpour

jusqu'à ce que non CHANGE

fin

- Pb : on revisite tous les arcs à chaque modification.

AC-3



procédure AC-3 (G)

Déclaration Q : Liste

début

$Q \leftarrow \{(V_i, V_j) \in \text{arcs}(G)\}$

tant que Q pas vide **faire**

selectionner et supprimer (V_k, V_m) de Q

si REVISE(V_k, V_m) **alors**

$Q \leftarrow Q \cup \{(V_i, V_k) \text{ tel que } (V_i, V_k) \in \text{arcs}(G)\}$

finsi

fintantque

fin

- Pb : on revisite tout le domaine des D_i

AC-4 : définitions



Définition 1 un arc (i, j) est consistant par rapport à D_i et D_j si et seulement si :

$$\forall u \in D_i \exists v \in D_j \text{ tel que } C_{i,j}(u, v) \text{ est vrai}$$

D_i domaine du noeud i , $C_{i,j}$ contrainte entre i et j .

Définition 2 Soit $P = D_1 \times D_2 \cdots \times D_n$. Un graphe G est arc consistant par rapport à P si et seulement si :

$$\forall (i, j) \in \text{arc}(G), (i, j) \text{ est consistant par rapport à } D_i \text{ et } D_j$$

- Notre Problème : Trouver $P' \subset P$ de taille maximale consistant par rapport à P .

AC-4 : notations



- $S_{j,b} : \langle i, a \rangle \in S_{j,b} \Leftrightarrow a \in D_i$ est consistant par rapport à $b \in D_j$. Ensemble des éléments qui utilisent $b \in D_j$ pour vérifier leur consistance.
- $counter[(i, j), a]$ nombre de $b \in D_j$ tels que (a, b) est consistant par rapport à (V_i, V_j) .

$\exists(i, j)$ tels que $counter[(i, j), a] = 0 \Leftrightarrow a$ est inconsistant

AC-4 : Initialisation (1/2)



procédure initialisation (G)

Déclaration Q,S : Listes

début

$Q \leftarrow \{\}$

$S \leftarrow \{\}$

Pour chaque (V_i, V_j) **dans** arcs(G) **faire**

Pour chaque a **dans** D_i **faire**

total \leftarrow 0

Pour chaque b **dans** D_j **faire**

si (a,b) est consistant par rapport à (V_i, V_j) **alors**

total \leftarrow total+1 ;

$S_{j,b} \leftarrow S_{j,b} \cup \{ \langle i, a \rangle \}$

finsi

finpour

AC-4 : Initialisation (2/2)



■ Initialisation (suite)

Pour chaque (V_i, V_j) **dans** arcs(G) **faire**
Pour chaque a **dans** D_i **faire**

⋮

 counter[(i,j),a] ← total

si counter[(i,j),a]=0 **alors**

 delete a de D_i

$Q \leftarrow Q \cup \{ \langle i, a \rangle \}$;

finsi

finpour

finpour

AC-4



procédure AC-4 (G)

Déclaration Q : Liste

début

$Q \leftarrow \text{initialisation}(G)$

tant que Q non vide **faire**

selectionner et supprimer $\langle j, b \rangle$ de Q

Pour chaque $\langle i, a \rangle$ **dans** $S_{j,b}$ **faire**

$\text{counter}[(i,j),a] \leftarrow \text{counter}[(i,j),a] - 1$

si $\text{counter}[(i,j),a]=0$ et a est encore dans D_i **alors**

supprimer a de D_i ;

$Q \leftarrow Q \cup \{ \langle i, a \rangle \}$;

finsi

finpour

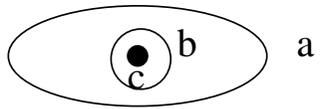
fintantque

fin

AC-4 : Application et limites



■ Exemple trivial



1 : ellipse $D_1 = \{a\}$

contient $\uparrow \downarrow$ *est contenu*

2 : cercle $D_2 = \{b, c\}$

contient $\uparrow \downarrow$ *est contenu*

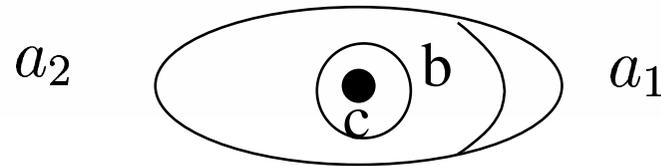
3 : cercle $D_3 = \{b, c\}$

■ AC-1 nous permet d'isoler la région a ,

■ AC-4 va nous permettre de différencier b et c .

■ Problème : que se passe t'il si a est découpé en deux régions a_1 et a_2 .

AC-4 : limitations



- ni a_1 ni a_2 ne satisfont les contraintes internes (ellipse),
- a_1 ne satisfait pas les contraintes externes.
- a_1 et a_2 ne sont pas reconnus.

- l'algorithme AC-4 ne peut fonctionner qu'avec une segmentation parfaite (impossible à garantir en pratique).
- Il faut pouvoir inclure dans l'algorithme AC-4 une notion d'équivalence de régions $\{a_1, a_2\}$ ne formant qu'un seul groupe.



- AC-4 avec «Bilevel constraints».
- Dans chaque D_i on considère une relation $Cmp_i(u, v)$ qui indique si les régions u et v peuvent appartenir à un même objet.
- La relation d'arc consistance est alors modifiée comme suit :

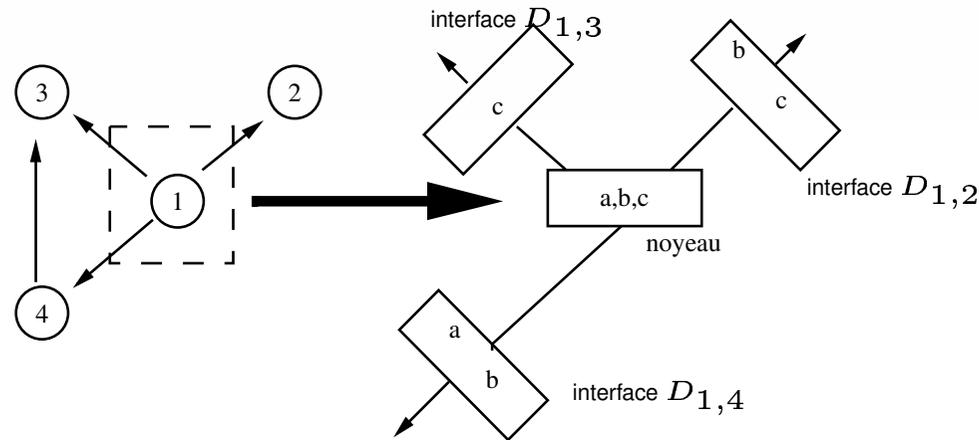
Définition

un arc $(i, j) \in arc(G)$ est consistant par rapport à D_i et D_j si et seulement si :

$$\forall t \in D_i \exists (u, v) \in D_i \times D_j \text{ tels que } Cmp_i(t, u) \text{ et } C_{i,j}(u, v)$$

Exemple a_1 est compatible avec a_2 ($Cmp_i(a_1, a_2)$) et a_2 contient un cercle ($C_{1,2}(a_2, b)$).

AC-4BC : Noyeau et interface



- I_i ensemble des interfaces du noeud i .
- $Path_i(b, D_{i,j})$: compatibilité avec l'interface $D_{i,j}$

$$Path(b, D_{i,j}) \Leftrightarrow \exists c \in D_{i,j} \text{ tel que } Cmp_i(b, c)$$

AC-4BC : l'algorithme



- Supprimer tout élément $a \in D_{i,j}$ tel que :

$$\nexists c \in D_{j,i} \mid C_{i,j}(b, c).$$

- Supprimer tout élément a de D_i tel qu'il existe $D_{i,j} \in I_i$ avec *non* $Path_i(a, D_{i,j})$ (a a perdu son support pour l'interface $D_{i,j}$).
 - Supprimer a de toutes les interfaces où il intervient.
- Itérer jusqu'à stabilité (voir transparent).

CleanKernel



procédure CleanKernel ($E D_i, I_i, S Q$)

début

Pour chaque b dans D_i faire

Pour chaque $D_{i,j}$ dans I_i faire

si non $Path_i(b, D_{i,j})$ alors

Ajouter (i, b) à Q

$D_i \leftarrow D_i - \{b\}$

Pour chaque $D_{i,j}$ dans I_i faire

$D_{i,j} \leftarrow D_{i,j} - \{b\}$

finpour

finsi

finpour

finpour

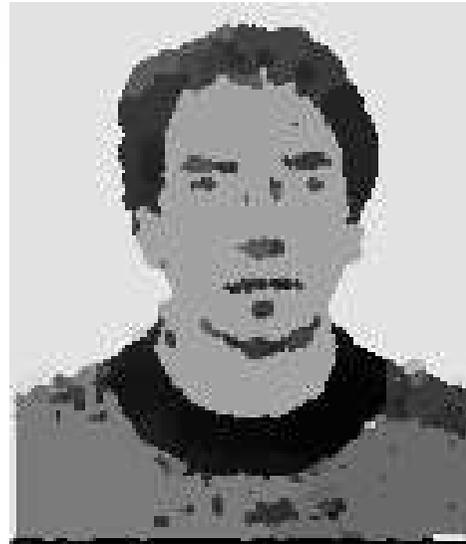
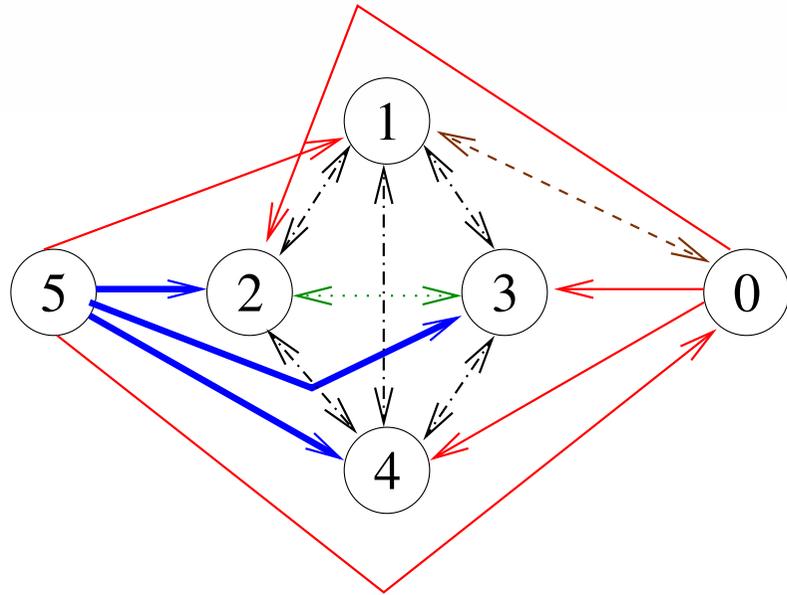
fin

Application à la segmentation



- Partir d'une image sur segmentée : P_0
- Fixer un seuil, s
- Faire
 1. $P_{n+1} \leftarrow \text{Simplification}(P_n, s)$
 2. $s \leftarrow s + \epsilon$
- Tant que P_{n+1} est consistant
- Renvoyer P_n .

Exemple



0	:	peau
1	:	cheveu
2	:	oeil gauche
3	:	oeil droit
4	:	bouche
5	:	fond

	autour
	autour et proche de
	gauche/droite
	dessus/dessous et proche
	dessus/dessous

Bibliographie (1/2)



- Graphes sémantiques et arc consistency

- **Segmentation 3D de cerveaux à partir d'Atlas**

- A. Deruyver, Y. Hodé, *Constraint satisfaction problem with bilevel constraint : application to interpretation of over-segmented images*, Artificial Intelligence, Vol. 93, (1997) 321-335,

- **Détection de tumeurs (AC-4BC modifié)**

- A. Deruyver, Y. Hodé, *Image Interpretation with a semantic Graph : Labeling over-segmented Images and Detection of Unespected Objects*, 3^eIAPR TC15 Workshop on graph based Representations in Pattern Recognition, pp 137-148,

- **Description/Reconnaissance de scène**

- H. Niemann, G.F. Sagerer, S. Schröder and F. Kummert, *a semantic network system for pattern understanding*, PAMI Vol. 12 (1990), 883-905,

Bibliographie (2/2)



A. Deruyver, Y. Hodé, E. Leammer and J.M Jolion, *Adaptative pyramid and Semantic graph : Knowledge driven Segmentation*, 5^eIAPR TC15 Workshop on graph based Representations in Pattern Recognition (GbR'2005), 213-222,

- Utilisation de graphes sémantiques

- **Mise à jour de cartes géographiques**

- M. Weis, S. Müller, C.E Liedtke and M. Pahl, *A framework for GIS and Imagery data fusion in support of cartographic updating*, Information fusion xxx(2004)In Press,

- **Analyse de Vidéos par le contenu**

- Volke Roth, *Content-based retrieval from digital video*, Image and Vision Computing, Vol. 17(1999) pp 531-540

Fin de la partie I



On continue sur la partie II...

