

Décomposition en Polygones de forme Étoile – Application à la Détection de Pièces

Hervé Locteau

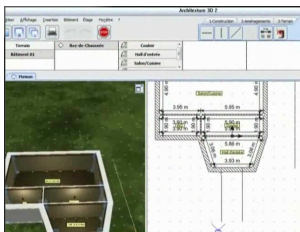
Laboratoire Lorrain de Recherche en Informatique et ses Applications, Université de Nancy 2
www.loria.fr

21 mars 2012



Contexte du projet ScanPlan

- Projet Euréka : coopération en R&D entre entreprises et instituts européens de recherche
- Partenaires : LiveCad (IdF, France), ICAR (Barcelone), Centre de Visio per Computador (Barcelone), Loria (Nancy)



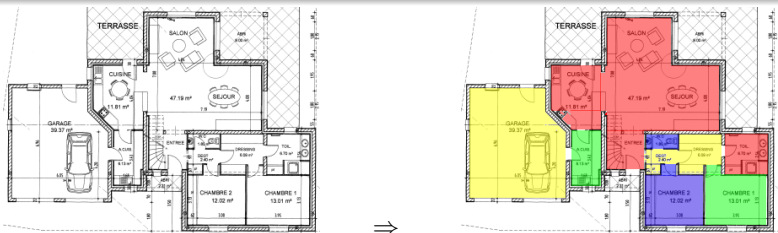
- Préparer une construction, un achat, une rénovation
- Clientèle principale (mais non limitée) : particulier



Modalité d'acquisition

Scenario

- Entrée : une image binaire
- Sortie : segmentation de la construction en pièces



Modules

- Détection de murs (de las Heras, 2011)
- Détection de pièces (Locteau, 2010)
- Détection de symboles (Boumaiza, 2012)

Une pièce

Une pièce est un polygone *approximativement convexe*. La convexité est quantifiée par le biais de la *profondeur* des concavités.

Stratégie

La scène est récursivement décomposée jusqu'à obtention de régions valides.

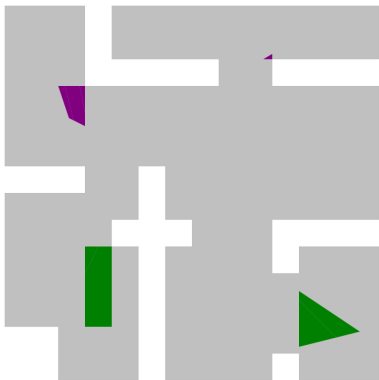
↑ Une approche descendante est rapide,

↑ Une décomposition récursive offre la possibilité de naviguer dans une pyramide de résolutions,

↓ Une étape de décomposition est faite localement (on ne peut garantir le fait d'isoler une unique pièce d'un ensemble de pièces); Sur-segmentation de certaines zones.

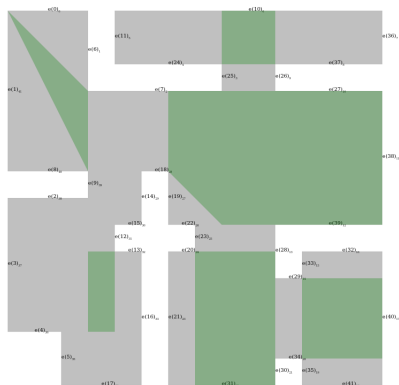
Couverture *versus* Partitionnement

$$\cup_i R_i = P$$



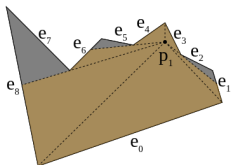
www.loria.fr/~locteau/scanplan/covering8.svg

$$\cup_i R_i = P \text{ and } R_i \cap R_j = \emptyset \forall i, j$$



www.loria.fr/~locteau/scanplan/partition8.svg

Polygone de visibilité $V(q)$ d'un point q



Soit un point q , $V(q)$ est l'ensemble des points $p \in P$ tels que $[pq] \cap P = \emptyset$.

Polygone de forme étoile

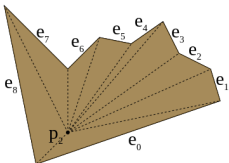
Noyau d'un polygone

Visibilité partielle d'un segment s

Visibilité entière d'un segment s

Polygone de visibilité $V(q)$ d'un point q

Polygone de forme étoile



Un polygone P est dit *de forme étoile*
 $\Leftrightarrow \exists p \in P$ tels que $V(p) = P$.

Noyau d'un polygone

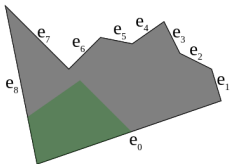
Visibilité partielle d'un segment s

Visibilité entière d'un segment s

Polygone de visibilité $V(q)$ d'un point q

Polygone de forme étoile

Noyau d'un polygone



Soit un polygone de forme étoile P , le noyau $K(P)$ désigne l'ensemble des points $p \in P$ tels que $V(p) = P$.

Visibilité partielle d'un segment s

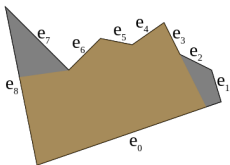
Visibilité entière d'un segment s

Polygone de visibilité $V(q)$ d'un point q

Polygone de forme étoile

Noyau d'un polygone

Visibilité partielle d'un segment s



L'ensemble $VP(s)$ défini par $\cup_{p \in s} V(p)$ constitue la visibilité partielle du segment s .

Visibilité entière d'un segment s

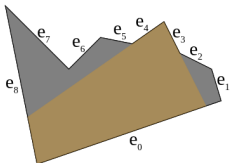
Polygone de visibilité $V(q)$ d'un point q

Polygone de forme étoile

Noyau d'un polygone

Visibilité partielle d'un segment s

Visibilité entière d'un segment s

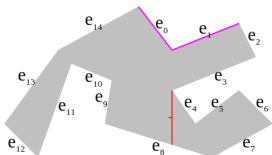


L'ensemble $VE(s)$ défini par $\bigcap_{p \in s} V(p)$ constitue la visibilité entière du segment s .

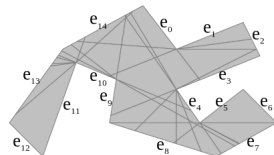
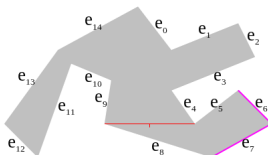
(Bottino 2011) Bottino A. et Laurenini A., *A nearly optimal algorithm for covering the interior of an Art Gallery*, *Pattern recognition*, vol. 44, p. 1048–1056, 2011

Complexe de visibilité d'une scène polygonale

Il s'agit d'une structure de données condensant les relations de visibilité entre les différents objets de l'espace. Dans (Bottino 2011), son extraction est obtenue à partir des *segments actifs*, segments définissant des frontières de visibilité.



Segments actifs & événements visuels

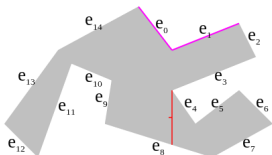


Partition $\Pi^0 = \{z_i^0\}_i$

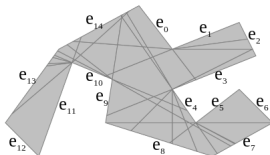
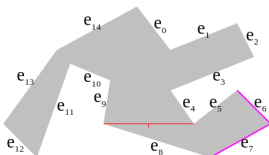
(Bottino 2011) Bottino A. et Laurenini A., *A nearly optimal algorithm for covering the interior of an Art Gallery*, *Pattern recognition*, vol. 44, p. 1048–1056, 2011

Complexe de visibilité d'une scène polygonale

Il s'agit d'une structure de données condensant les relations de visibilité entre les différents objets de l'espace. Dans (Bottino 2011), son extraction est obtenue à partir des *segments actifs*, segments définissant des frontières de visibilité.



Segments actifs & événements visuels



Partition $\Pi^0 = \{z_i^0\}_i$

Avec cette structure, le test de visibilité entre objets trouve sa réponse au niveau symbolique.

Une tâche de classification non supervisée des segments

Le **nombre de régions** dans la décomposition du polygone P , ou encore le nombre de *cluster*, est inconnu et doit être **minimisé**.

Un cluster doit être défini en fonction des inter-relations de visibilité entre ces individus afin d'**obtenir des polygones de forme étoile**.

L'ensemble des clusters doit définir une **partition** de P .

Ensemble des contraintes :

- ① $\min \text{card}(\{R_i\}_i)$,
- ② $K(R_i) \neq \emptyset \forall i$,
- ③ $\cup_i R_i = P$,
- ④ $R_i \cap R_j = \emptyset \forall i, j$.

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :

$\{\{1\}\}$

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :

$$\{\{1\}\}$$

$$\{\{1,2\}\}$$

$$\{\{1\}\{2\}\}$$

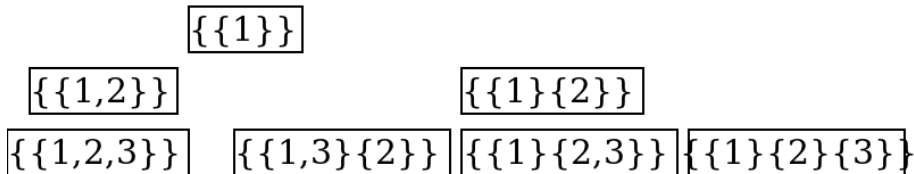
si $\mu(s_1) = \mu(s_2)$ est valide

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :



if $\mu(s_1) = \mu(s_2) = \mu(3)$ est valide

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :

$\{\{1\}\}$

Puisque nous voulons minimiser la taille de la partition...

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :

$$\boxed{\{\{1\}\}}$$

$$\boxed{\{\{1,2\}\}}$$

Puisque nous voulons minimiser la taille de la partition...

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :

$$\{\{1\}\}$$

$$\{\{1,2\}\}$$

$$\{\{1,2,3\}\}$$

Puisque nous voulons minimiser la taille de la partition...

Contrainte (1) : $\min \text{card}(\{R_i\}_i)$

Soit $\{s_j\}_j$ l'ensemble ordonné des segments définissant P .

La DPfE de P est définie en classant successivement chaque s_j par une fonction $\mu(\cdot)$ soit dans un cluster existant, soit dans un nouveau cluster.

Pour une énumération exhaustive des partitions, on peut construire pas à pas un arbre d'exploration d'un espace d'états à évaluer, nous aurions :

$\{\{1\}\}$

$\{\{1,2\}\}$

$\{\{1,2,3\}\}$

Puisque nous voulons minimiser la taille de la partition...

La recherche A^* se prête à notre objectif, par le biais d'une file de priorité triée en premier lieu par le nombre de clusters, en second lieu par le nombre de segments impliqués.

Contrainte (2) : $K(R_i) \neq \emptyset \forall i \dots$ et optimisation (A)

$\omega_k = \mu(s_i) = \mu(s_j)$ est valide $\Leftrightarrow VE(s_j) \cap (\bigcap_{i < j, \mu(s_i) = \omega_k} VE(s_i)) \neq \emptyset$. L'ajout d'un segment s_j ne peut conduire à un noyau associé qui soit vide.

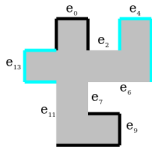
Contrainte (2) : $K(R_i) \neq \emptyset \forall i \dots$ et optimisation (A)

$\omega_k = \mu(s_i) = \mu(s_j)$ est valide $\Leftrightarrow VE(s_j) \cap (\bigcap_{i < j, \mu(s_i) = \omega_k} VE(s_i)) \neq \emptyset$. L'ajout d'un segment s_j ne peut conduire à un noyau associé qui soit vide.

Super-segment

$ss_i = (e_a e_{a+1} \dots e_{i-1} e_i e_{i+1} \dots e_{b-1} e_b)$ est appelé *super-segment* lorsque, en fonction des $VE(e_i)$, la visibilité entière du segment e_i (élément pivot) implique celle de chacun des segments de la séquence $e_a \dots e_b$: $VE(e_j) \subset VE(e_i) \forall a < j < b$.

Exemple :



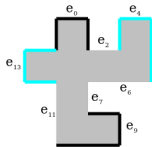
Contrainte (2) : $K(R_i) \neq \emptyset \forall i \dots$ et optimisation (A)

$\omega_k = \mu(s_i) = \mu(s_j)$ est valide $\Leftrightarrow VE(s_j) \cap (\bigcap_{i < j, \mu(s_i) = \omega_k} VE(s_i)) \neq \emptyset$. L'ajout d'un segment s_j ne peut conduire à un noyau associé qui soit vide.

Super-segment

$ss_i = (e_a e_{a+1} \dots e_{i-1} e_i e_{i+1} \dots e_{b-1} e_b)$ est appelé *super-segment* lorsque, en fonction des $VE(e_i)$, la visibilité entière du segment e_i (élément pivot) implique celle de chacun des segments de la séquence $e_a \dots e_b$: $VE(e_j) \subset VE(e_i) \forall a < j < b$.

Exemple :



\Rightarrow **Optimisation (A)** : La taille du problème initial peut être ramenée de n à $n - nss$ où nss désigne le nombre de segments inclus dans la définition d'un super-segment, à l'exception des éléments pivots.

Contrainte (4) : $R_i \cap R_j = \emptyset \forall i, j$

(1/2)

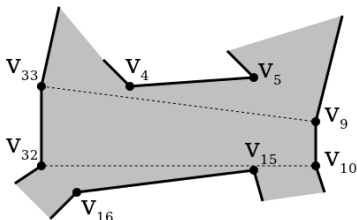
Prérequis pour la planarité

Soient deux régions $R_i = \{e_{i,0}, e_{i,1} \dots e_{i,a}\}$ et $R_j = \{e_{j,0}, e_{j,1} \dots e_{j,b}\}$,

$$\forall 0 \leq k \leq a, \text{dst}(e_{i,k}) \neq \text{src}(e_{i,k+1}), \forall 0 \leq l \leq b, \text{dst}(e_{j,l}) \neq \text{src}(e_{j,l+1})$$

 \Rightarrow

$$\begin{cases} \text{dst}(e_{i,k}) \leq \text{dst}(e_{j,l}) < \text{src}(e_{j,l+1}) \leq \text{src}(e_{i,k+1}) \\ \vee \\ \text{dst}(e_{j,l}) \leq \text{dst}(e_{i,k}) < \text{src}(e_{i,k+1}) \leq \text{src}(e_{j,l+1}) \end{cases}$$



Illustration

Nous ne pouvons avoir à la fois $\mu(e_4) = \mu(e_{15})$ et $\mu(e_9) = \mu(e_{32})$.

Contrainte (4) : $R_i \cap R_j = \emptyset \forall i, j$

(2/2)

Disjonction

Pour fermer le contour d'une région, un chemin liant deux discontinuités doit exister. Ce chemin peut impliquer des discontinuités d'autres régions, des segments non classifiés jusqu'à présent. De plus, un chemin n'est valide entre deux sommets v_s et v_d si :

- $s < c_0 < c_1 \dots < c_p < d$ lorsque $s < d$,
- $s < c_0 < c_1 \dots < c_k \leq n$ et $0 \leq c_{k+1} < c_{k+2} \dots < c_p < d$ sinon.

Contrainte (4) : $R_i \cap R_j = \emptyset \forall i, j$

(2/2)

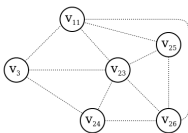
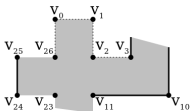
Disjonction

Pour fermer le contour d'une région, un chemin liant deux discontinuités doit exister. Ce chemin peut impliquer des discontinuités d'autres régions, des segments non classifiés jusqu'à présent. De plus, un chemin n'est valide entre deux sommets v_s et v_d si :

- $s < c_0 < c_1 \dots < c_p < d$ lorsque $s < d$,
- $s < c_0 < c_1 \dots < c_k \leq n$ et $0 \leq c_{k+1} < c_{k+2} \dots < c_p < d$ sinon.

Illustration : ajout de (v_2, v_3) à R_1 ?

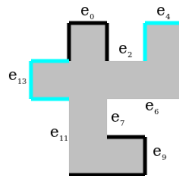
Soient $R_2 = \{e_{2,0} = (v_3, v_4), e_{2,1} = (v_9, v_{10}), e_{2,2} = (v_{10}, v_{11}), e_{2,3} = (v_{24}, v_{25})\}$ et $R_1 = \{e_{1,0} = (v_{26}, v_0), e_{1,1} = (v_0, v_1), e_{1,2} = (v_1, v_2), e_{1,3} = (v_2, v_3)\}$. Le prérequis pour la planarité est validé mais aucun chemin à partir de v_3 ne conduit à v_{26} en présence de R_2 .



Optimisation (B)

(1/2)

π	I.R.	I.R.S.	I.V.	N.I.
0	14	15	32	56
1	9	9	9	31
4	12	24	24	38
6	9	11	49	50
7	7	8	8	32
11	14	15	19	60



I.R. : Itération à partir de laquelle est atteint le nombre optimum de Régions

I.R.S. : Idem en traitant tous les Segments

I.V. : Itération à laquelle on identifie la première partition Valide

N.I. : Nombre d'Itérations (dans le cas d'une recherche exhaustive)

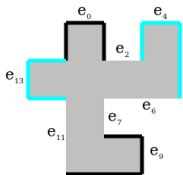
Combinatoires pour des permutations cycliques des segments

Impact critique de l'ordonnancement des segments

Optimisation (B)

(2/2)

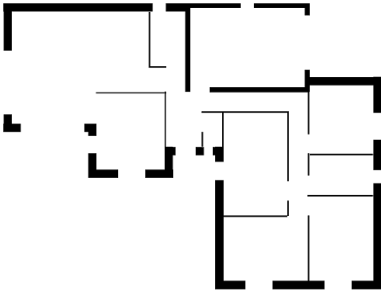
e_i	$N(e_i)$	$w(e_i)$	$w(e_i)$							
e_0	{2 6 7 11}	4	4	4	-	-	-	-	-	-
e_2	{0 4 6 7 11 13}	6	6	22	38	54	-	-	-	-
e_4	{2 6 13}	3	3	-	-	-	-	-	-	-
e_6	{0 2 4 7 11 13}	6	6	22	38	54	70	-	-	-
e_7	{0 2 6 9 11 13}	6	22	22	38	54	70	86	-	-
e_9	{7 11}	2	-	-	-	-	-	-	-	-
e_{11}	{0 2 6 7 9 13}	6	22	22	38	54	70	86	102	-
e_{13}	{2 4 6 7 11}	5	5	21	21	-	-	-	-	-



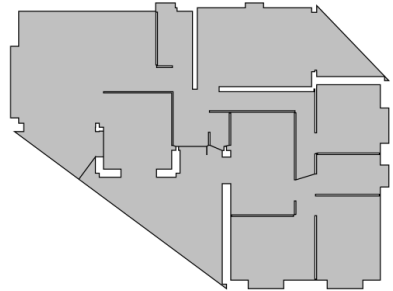
⇒ Nous obtenons une partition valide à la 7e itération.



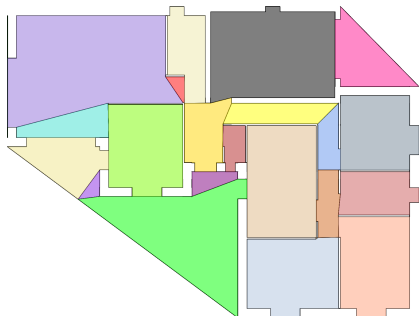
Plan architectural



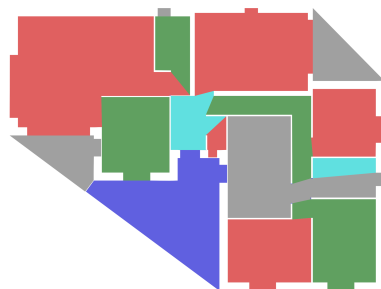
Détection de murs



Scène à analyser



(Locteau, 2010)



Notre approche
 (première partition valide, itération 147)
 $n = |P| = 163$
 $nss = 50$ (26 super-segments)

Conclusion et perspectives

Proposition d'un algorithme de DPfE minimisant le nombre de régions.

Pour un contexte applicatif donné, des contraintes supplémentaires doivent être introduites.

Alors que l'exhaustivité des solutions est envisageable pour une sélection *a posteriori* du résultat pour des problèmes de taille réduite, **l'espace des solutions croît rapidement.**

Perspectives

- Ajout au sein de l'algorithme A^* d'une caractéristique en vue de traiter en priorité les **partitions dont les noyaux sont plus étendus** (estimation de cette caractéristique strictement décroissante en ajoutant des segments),
- Évaluation quantitative,
- Apport de robustesse en présence de **données bruitées** (au niveau des noyaux, au niveau de la cardinalité des partitions.)