Extraction et reconnaissance de primitives dans les façades de Paris à l'aide de similarités de graphes

HAUGEARD Jean-Emmanuel

ETIS - CNRS - ENSEA - Université Cergy-Pontoise

Directrice de thèse : Sylvie Philipp-Foliguet
Co-Directeurs de thèse : Frédéric Precioso et Philippe-Henri Gosselin

17 décembre 2010

iTowns

Projet financé par l'Agence nationale de la recherche

iTowns : image-based Town On-line Web Navigation and Searchengine.

- 5 partenaires (IGN, LIP6, ETIS, LCPC - LRPC , ARMINES)

iTowns

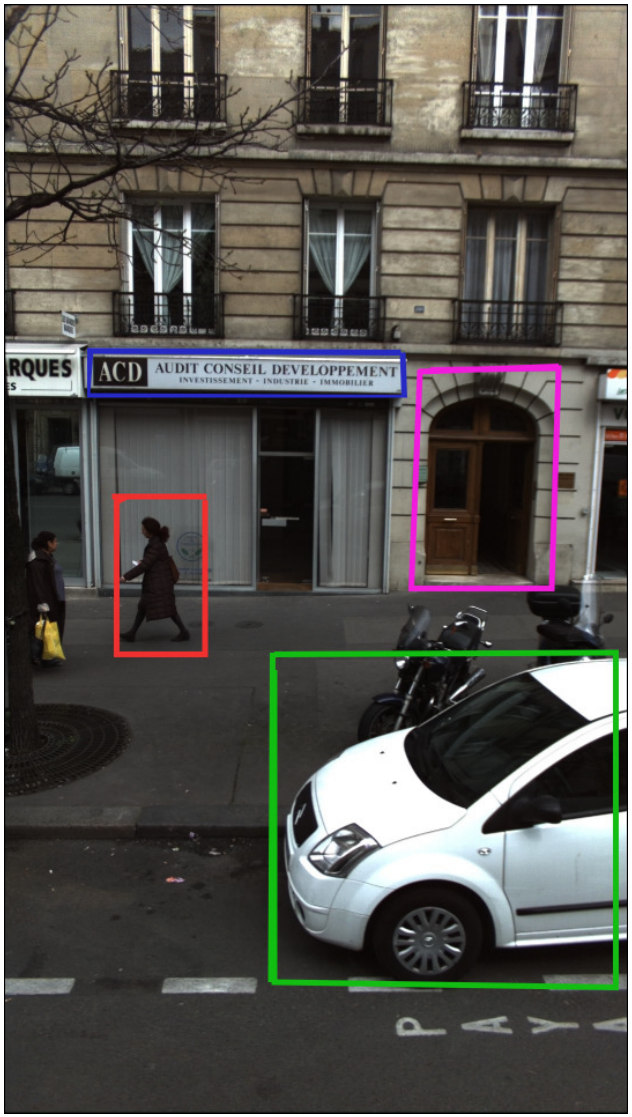
2 objectifs :

- naviguer de manière fluide dans un flux d'images panoramiques
- construire à partir des images un système d'information basé sur le contenu

Naviguer de manière fluide dans un flux d'images panoramiques

- un Teraoctet de données
- 12ème arrondissement de Paris
- 25000 vues panoramiques
- camion avec 12 caméras et 3 lasers

Construire un système d'information basé sur le contenu



Les contours [SHOTTON 05, OPELT 06]

Pouvez-vous identifier les objets présents dans ces deux images de contours ?

Plan

1. Extraction des fenêtres dans les images de façades
   - Etat de l'art
   - Notre démarche
   - Résultats
2. Graphe relationnel attribué de contours et similarités sur graphes
   - Représentation graphes de contours
   - Attributs sur sommets et arêtes
3. Noyau sur graphes de contours
   - Noyau sur chemins
   - Complexité de calcul
4. Localisation des fenêtres dans les façades à l'aide des graphes

Plan

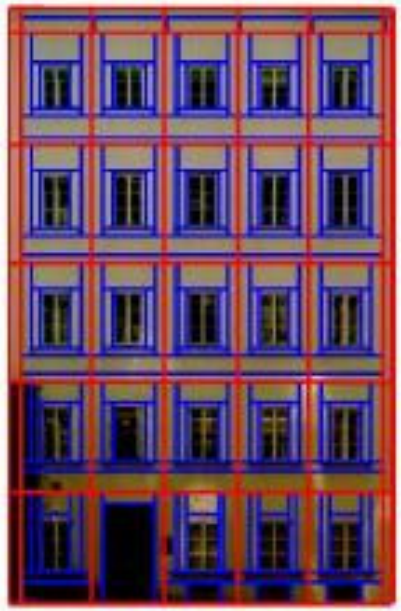
1. Extraction des fenêtres dans les images de façades
   - Etat de l'art
   - Notre démarche
   - Résultats
2. Graphe relationnel attribué de contours et similarités sur graphes
   - Représentation graphes de contours
   - Attributs sur sommets et arêtes
3. Noyau sur graphes de contours
   - Noyau sur chemins
   - Complexité de calcul
4. Localisation des fenêtres dans les façades à l'aide des graphes

Extraction des fenêtres : Etat de l'art [MULLER 07, MAYER 07, MOSLAH 10]

Modélisation grossière des bâtiments : besoin d'améliorer la visualisation et le contenu des façades

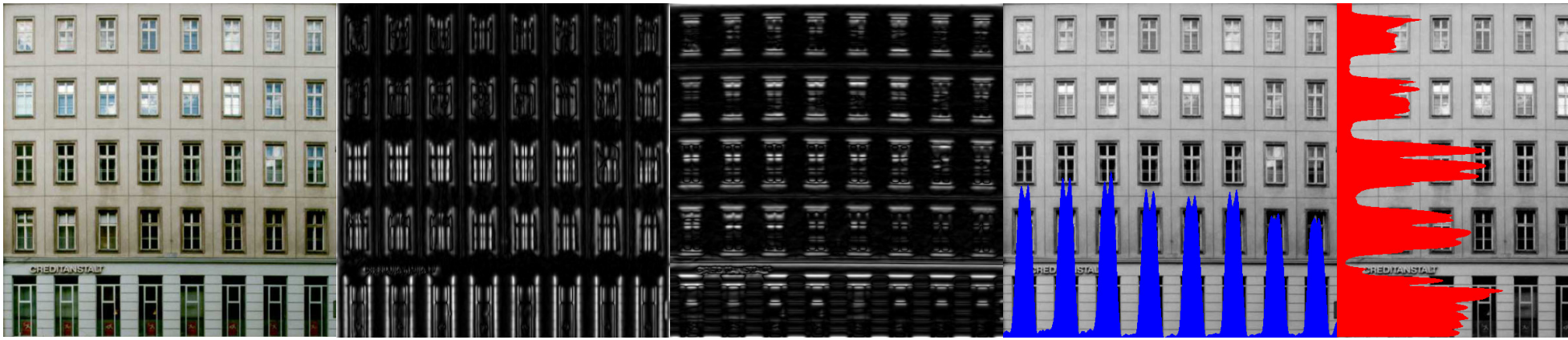
Analyse d'une façade :

- Symétrie dans les façades
- Géométrie des entités architecturales



Histogramme des projections du gradient [Lee et Nevatia 04]

- Calcul des composantes verticales et horizontales du gradient (Shen-Castan)
- Projection horizontale et verticale
- Seuillage

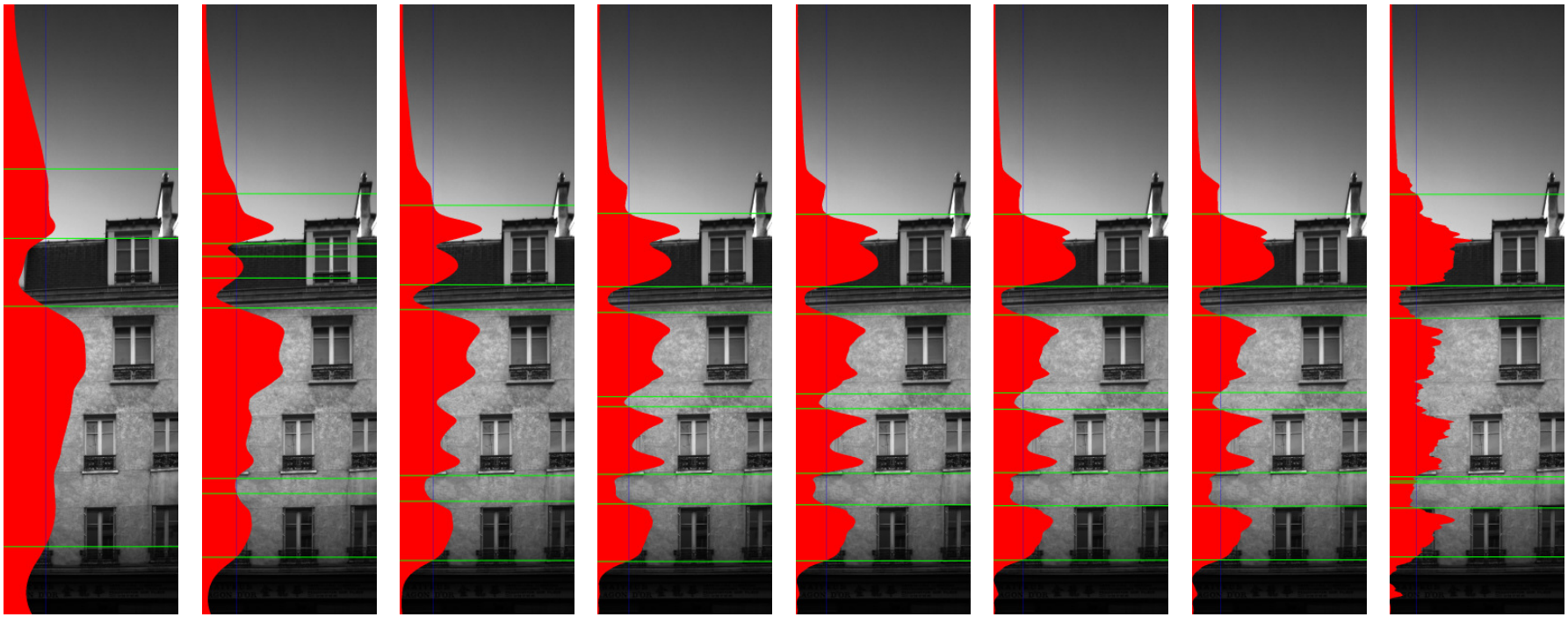


REMARQUE

Méthode efficace sur des façades sans texture, avec des fenêtres bien alignées et de même intensité lumineuse. Les résultats dépendent du paramètre de lissage et de dérivation β de l'algorithme de Shen-Castan [HUET 97].

Notre démarche

Automatiser le paramètre de lissage et travailler étage par étage.

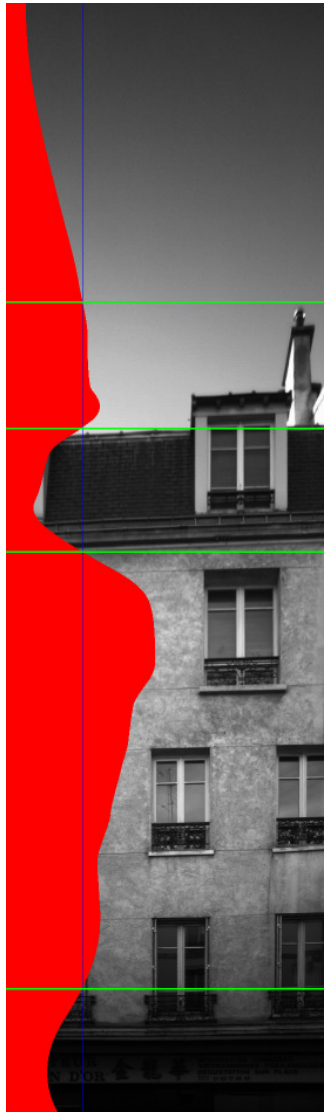


(a) $\beta = 0.01$ (b) 0.02 (c) 0.03 (d) 0.05 (e) 0.07 (f) 0.09 (g) 0.1 (h) 0.3

Algorithme

Soit *I* l'image originale :

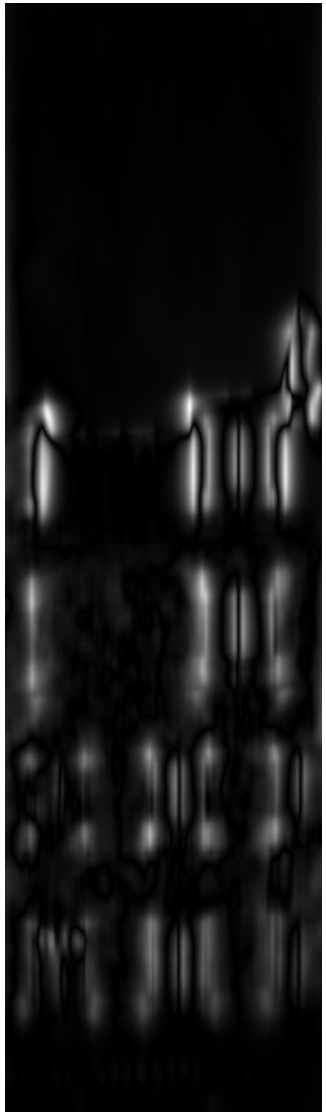
- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.



Algorithme

Soit *I* l'image originale :

- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.



Algorithme

Soit *I* l'image originale :

- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.

Algorithme

Soit *I* l'image originale :

- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.

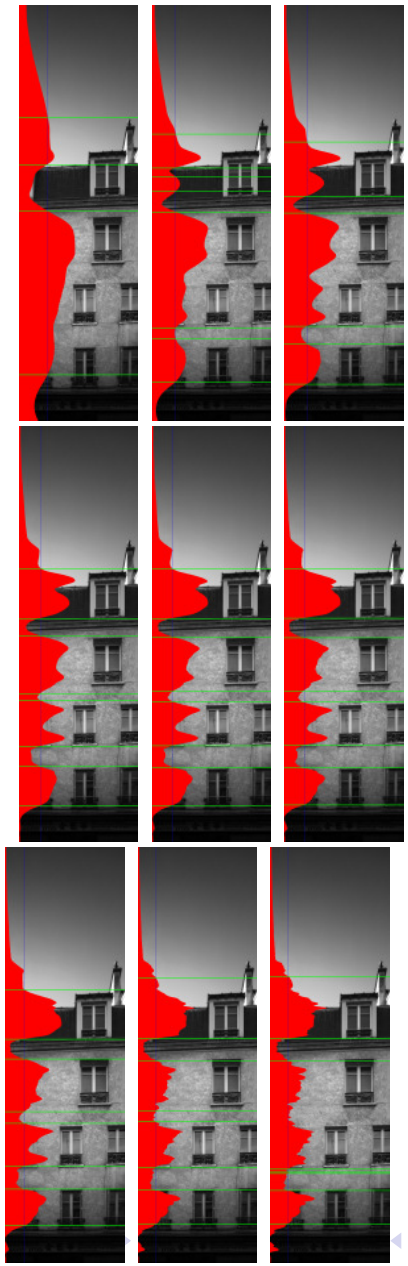
CRITÈRE

$$S_{\beta_i} = \begin{cases} \overbrace{\frac{p_{\beta_{i-1}}}{p_{\beta_i}}}^{Stabilite} \cdot \frac{\overbrace{\sum_{j=1}^{p_{\beta_i}} maxH_{pj}}^{AmplitudeMoyennedesPics}}{p_{\beta_i}} & \text{si } p_{\beta_{i-1}} \\ \frac{\sum_{j=1}^{p_{\beta_i}} maxH_{pj}}{p_{\beta_{i-1}}} & \text{sinon} \end{cases} \quad (1)$$

Algorithme

Soit *I* l'image originale :

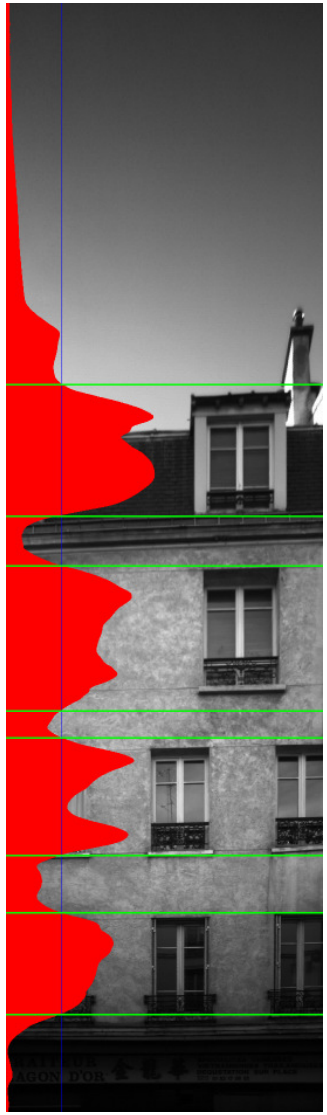
- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.



Algorithme

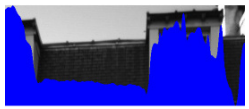
Soit *I* l'image originale :

- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.

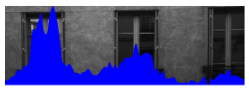


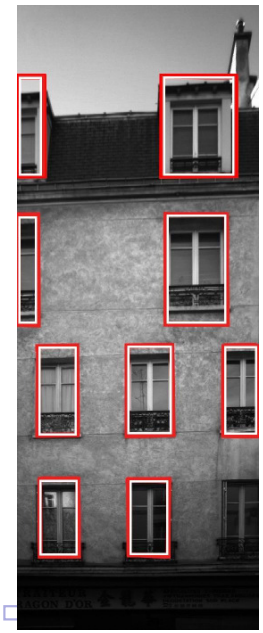
Algorithme

Soit *I* l'image originale :



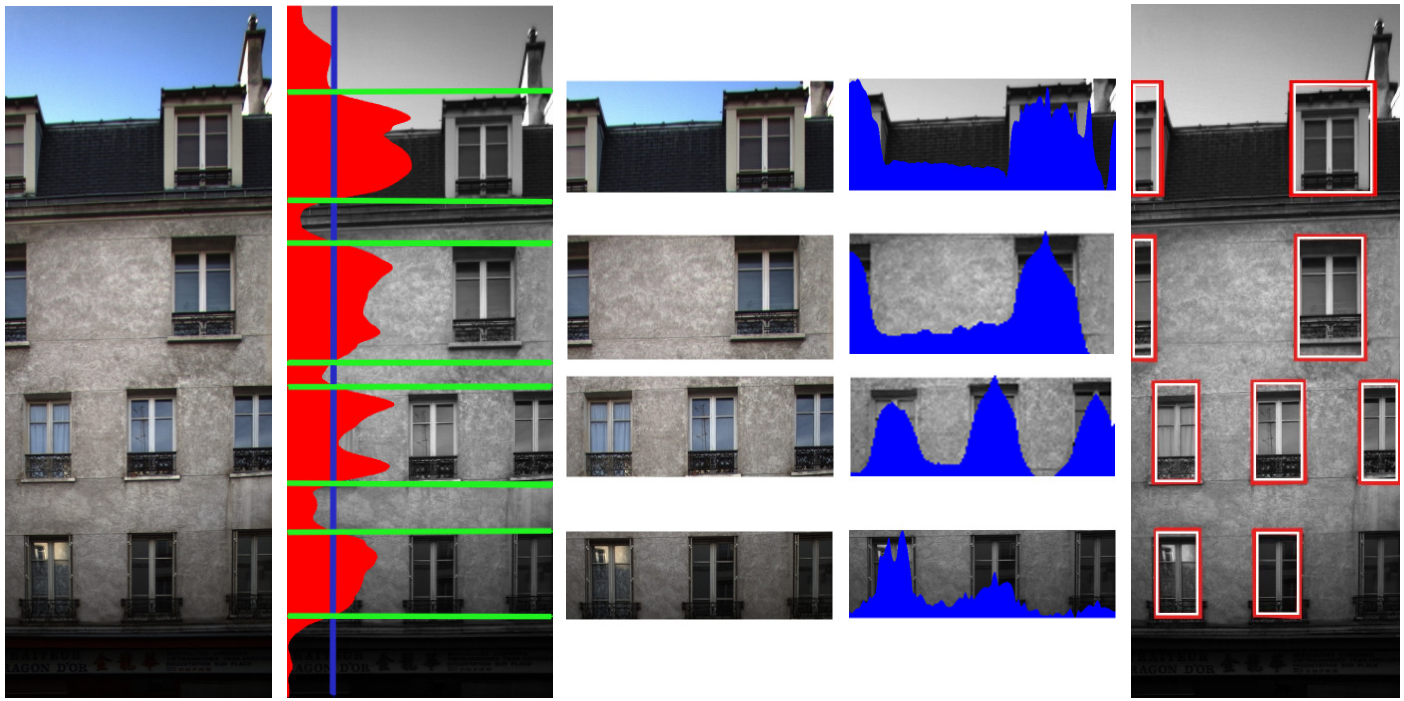
- Initialisation : lissage fort donc β faible
- Tant que $\beta < seuil$
  - Calcul de la composante horizontale du gradient
  - Projection de la norme
  - Seuillage et découpe en étage
  - Calcul d'un score
  - J'augmente β
- Je choisis β qui maximise mon score
- Pour chaque étage, je calcule la composante verticale avec le même β du gradient, je projette sur x et cherche les bassins.





Bilan : Histogramme des projections du gradient

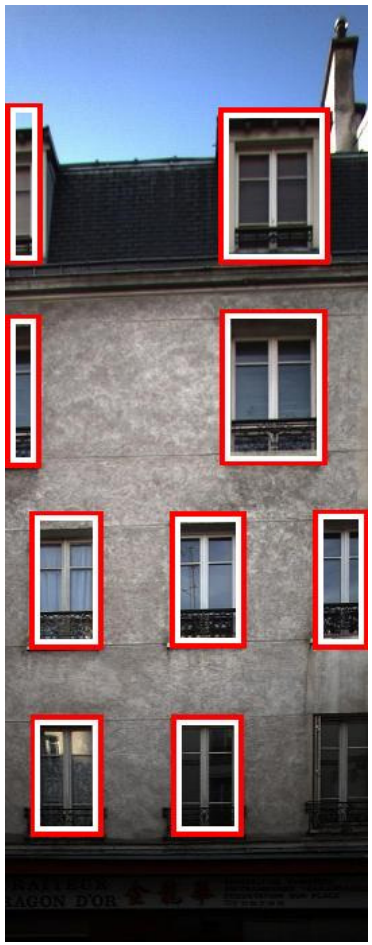
Résumé :



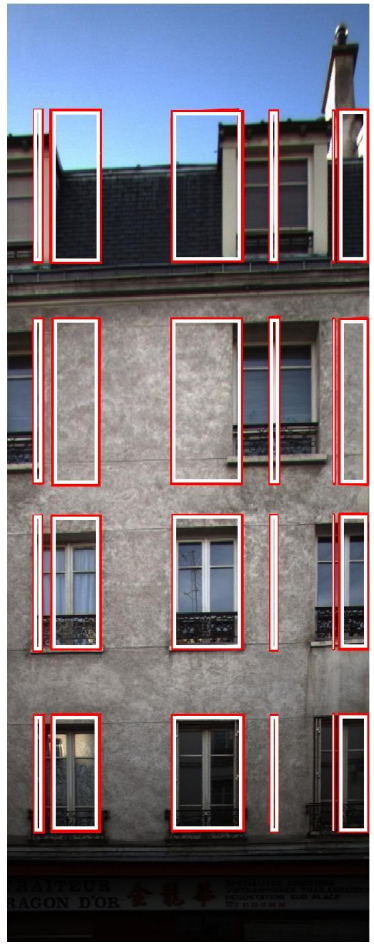
Histogramme des projections du gradient

Apports :

- Relaxation sur la contrainte verticale (alignement des fenêtres)
- Automatiser l'extraction en déterminant la bonne échelle



(i) Notre algorithme

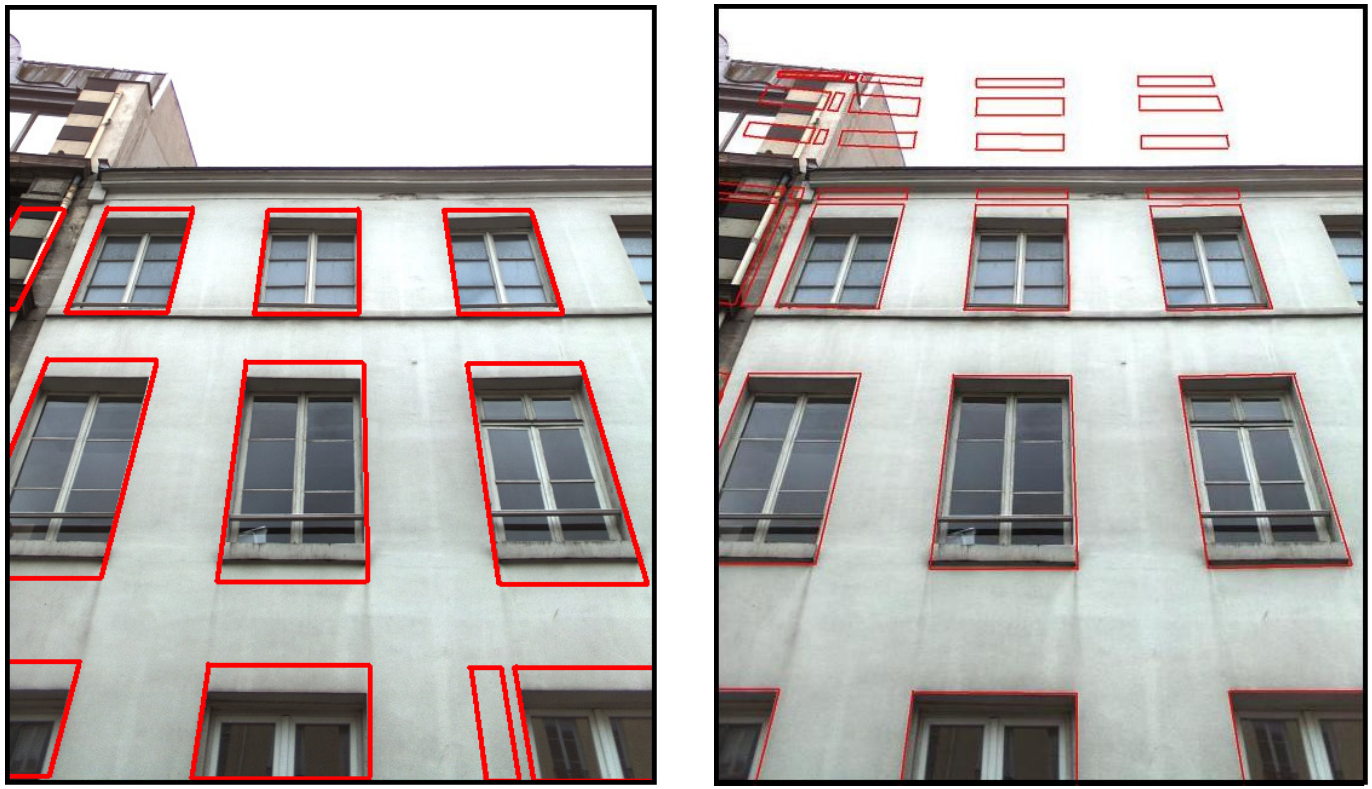


(j) Lee

Histogramme des projections du gradient

Apports :

- Relaxation sur la contrainte verticale (alignement des fenêtres)
- Automatiser l'extraction en déterminant la bonne échelle



(a) Notre algorithme (b) Lee

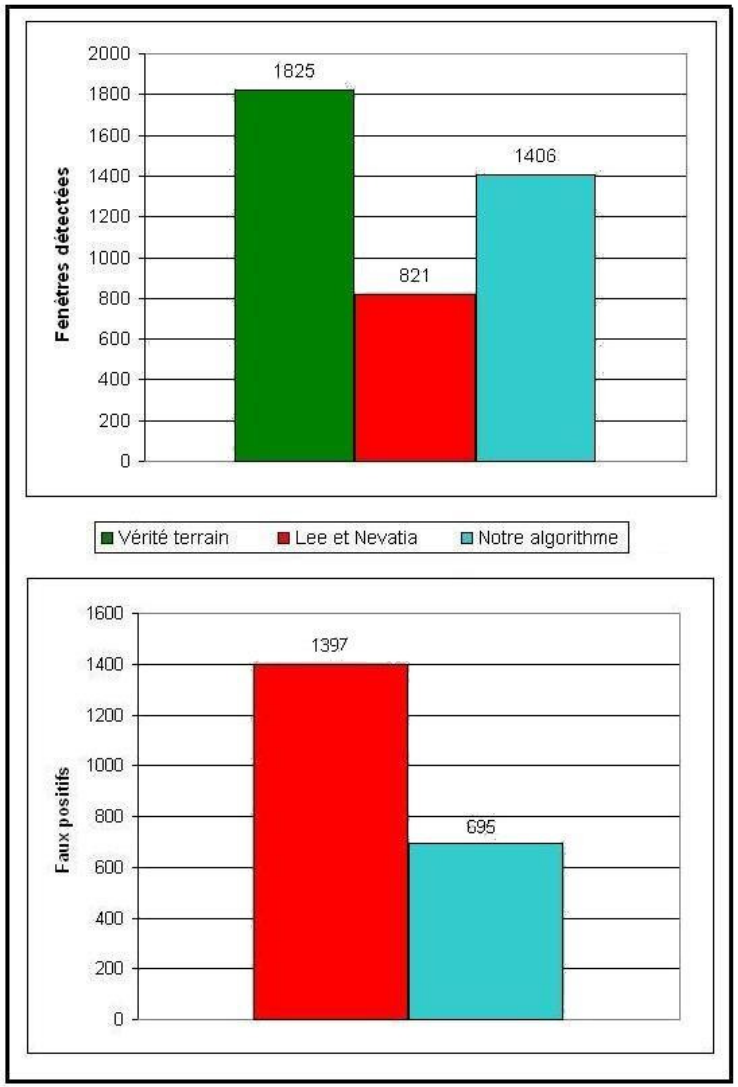
FIGURE: **Extraction des hypothèses de fenêtres.**

Comparaison de notre algorithme par rapport à celui de Lee et Nevatia

Test sur une base de 169 façades (rue Saint Antoine)

Nos résultats

- une meilleure détection
- moins de faux positifs



Plan

1. Extraction des fenêtres dans les images de façades
   - Etat de l'art
   - Notre démarche
   - Résultats
2. Graphe relationnel attribué de contours et similarités sur graphes
   - Représentation graphes de contours
   - Attributs sur sommets et arêtes
3. Noyau sur graphes de contours
   - Noyau sur chemins
   - Complexité de calcul
4. Localisation des fenêtres dans les façades à l'aide des graphes

Notre représentation en graphes de contours : extraction des contours principaux

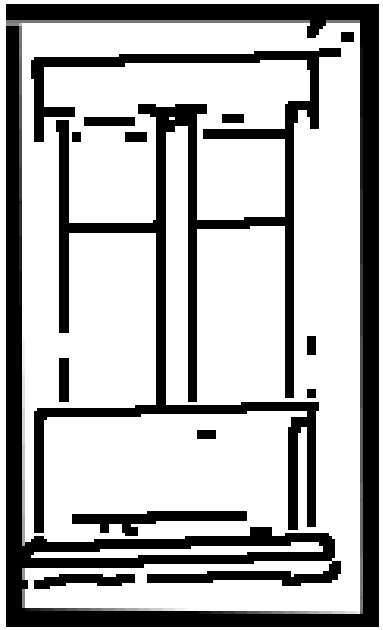
Soit *I* l'image originale :

- Recherche des contours par la méthode de Shen-Castan
  - Lissage
  - Calcul du gradient en chaque point de l'image
  - Extraction des maxima locaux
- Binarisation
- Polygonalisation et suppression des coins
- Recherche des régions d'influence (Voronoï)

Notre représentation en graphes de contours : extraction des contours principaux

Soit *I* l'image originale :

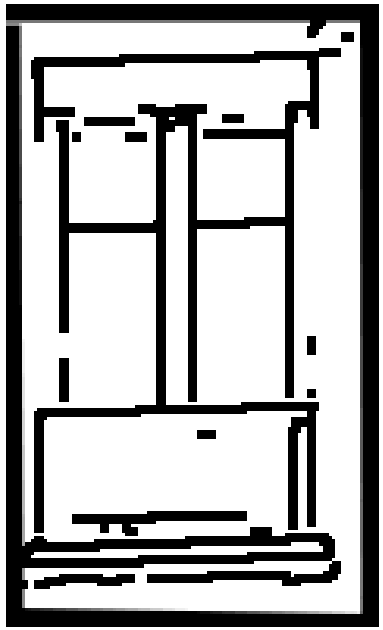
- Recherche des contours par la méthode de Shen-Castan
  - Lissage
  - Calcul du gradient en chaque point de l'image
  - Extraction des maxima locaux
- Binarisation
- Polygonalisation et suppression des coins
- Recherche des régions d'influence (Voronoï)



Notre représentation en graphes de contours : extraction des contours principaux

Soit *I* l'image originale :

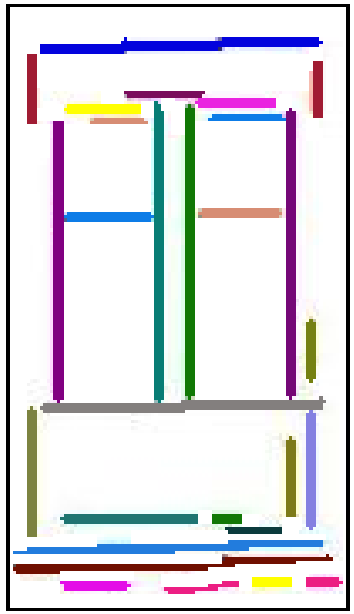
- Recherche des contours par la méthode de Shen-Castan
  - Lissage
  - Calcul du gradient en chaque point de l'image
  - Extraction des maxima locaux
- Binarisation
- Polygonalisation et suppression des coins
- Recherche des régions d'influence (Voronoï)



Notre représentation en graphes de contours : extraction des contours principaux

Soit *I* l'image originale :

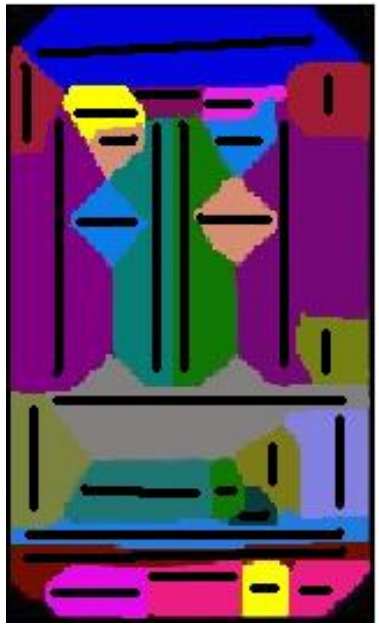
- Recherche des contours par la méthode de Shen-Castan
  - Lissage
  - Calcul du gradient en chaque point de l'image
  - Extraction des maxima locaux
- Binarisation
- Polygonalisation et suppression des coins
- Recherche des régions d'influence (Voronoï)



Notre représentation en graphes de contours : extraction des contours principaux

Soit *I* l'image originale :

- Recherche des contours par la méthode de Shen-Castan
  - Lissage
  - Calcul du gradient en chaque point de l'image
  - Extraction des maxima locaux
- Binarisation
- Polygonalisation et suppression des coins
- Recherche des régions d'influence (Voronoï)



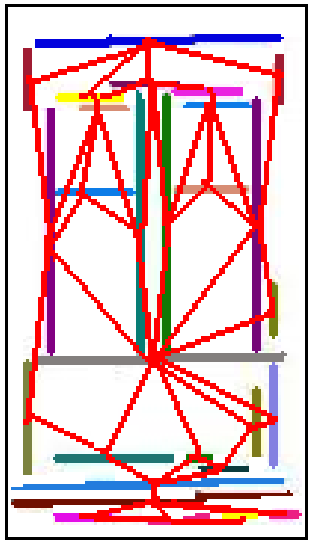
Graphes de contours : attributs sur sommets et arêtes

IDEE : graphe relationnel de contours

- Sommets représentent les segments de contours
- Arêtes modélisent la relation entre régions adjacentes

Soit G le graphe de contours :

- Chaque segment est représenté par son orientation : $\Theta \in [0, 180[$
- Chaque segment est "relié" par un arc à chacun des contours les plus proches



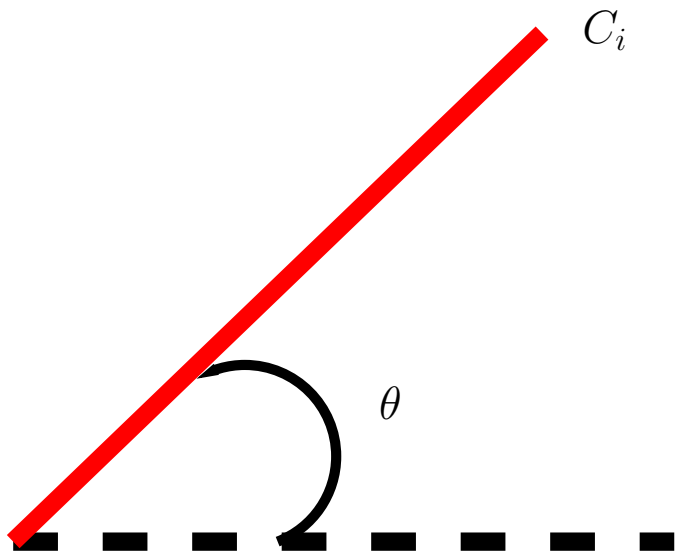
Graphes de contours : attributs sur sommets et arêtes

IDEE : graphe relationnel de contours

- Sommets représentent les segments de contours
- Arêtes modélisent la relation entre régions adjacentes

Soit G le graphe de contours :

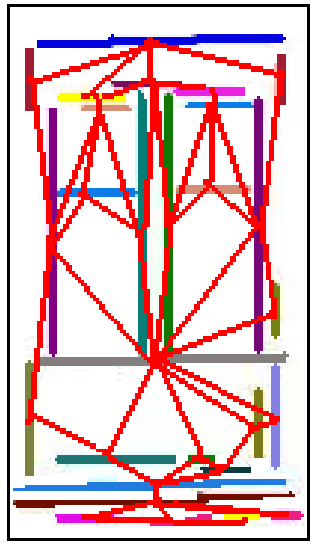
- Chaque segment est représenté par son orientation : $\Theta \in [0, 180[$
- Chaque segment est "relié" par un arc à chacun des contours les plus proches



Graphes de contours : attributs sur sommets et arêtes

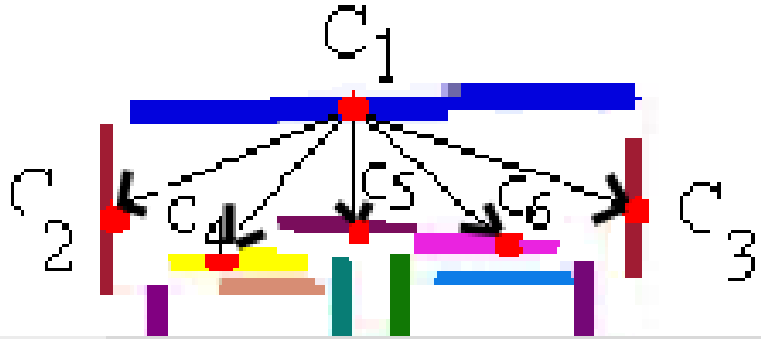
IDEE : graphe relationnel de contours

- Sommets représentent les segments de contours
- Arêtes modélisent la relation entre régions adjacentes

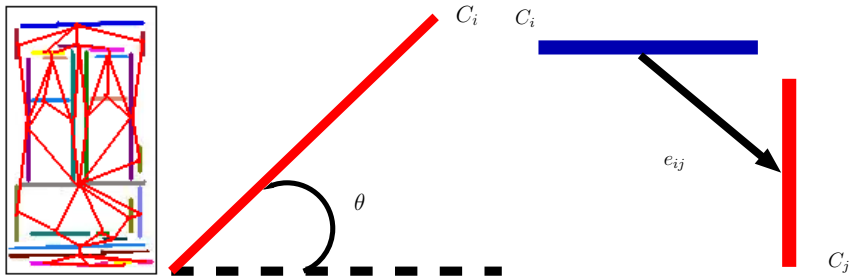


Soit G le graphe de contours :

- Chaque segment est représenté par son orientation : $\Theta \in [0, 180[$
- Chaque segment est "relié" par un arc à chacun des contours les plus proches



Graphes de contours : attributs sur sommets et arêtes



Graphe relationnel de contours

- Sommet représente un segment

$$\vec{v} = \begin{pmatrix} cos(2\Theta) \\ sin(2\Theta) \end{pmatrix}$$

avec Θ l'angle entre le contour et l'axe horizontal

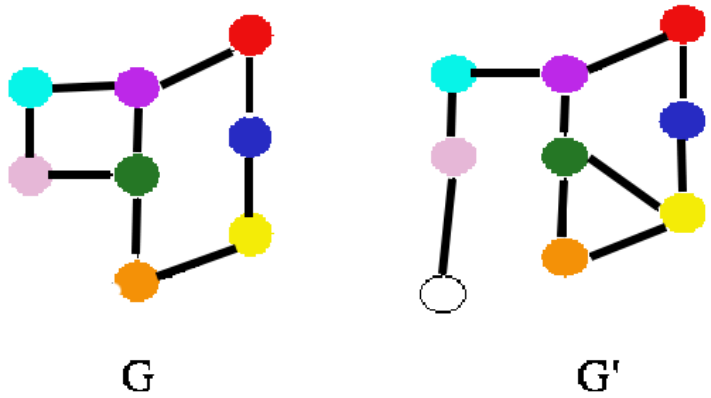
- Arc modélise la relation entre les centres de gravité des segments voisins (C_i et C_j)

$$\vec{e_{ij}} = \begin{pmatrix} Xg_{C_j} - Xg_{C_i} \\ Yg_{C_j} - Yg_{C_i} \end{pmatrix}$$

Plan

1. Extraction des fenêtres dans les images de façades
   - Etat de l'art
   - Notre démarche
   - Résultats
2. Graphe relationnel attribué de contours et similarités sur graphes
   - Représentation graphes de contours
   - Attributs sur sommets et arêtes
3. Noyau sur graphes de contours
   - Noyau sur chemins
   - Complexité de calcul
4. Localisation des fenêtres dans les façades à l'aide des graphes

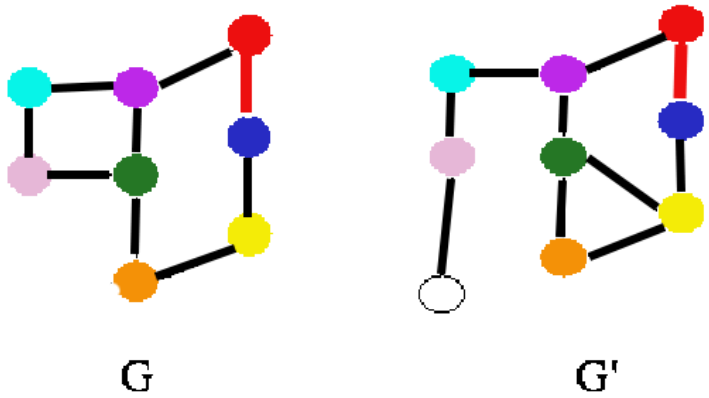
Graphe relationnel attribué de contours : Noyau sur graphes



Comment comparer deux graphes entre eux ?

- graphe non isomorphe
- graphe de grande taille (20 à 100 sommets)
- graphe valué sur les sommets et arcs
- recherche de similarités entre chemin du graphes

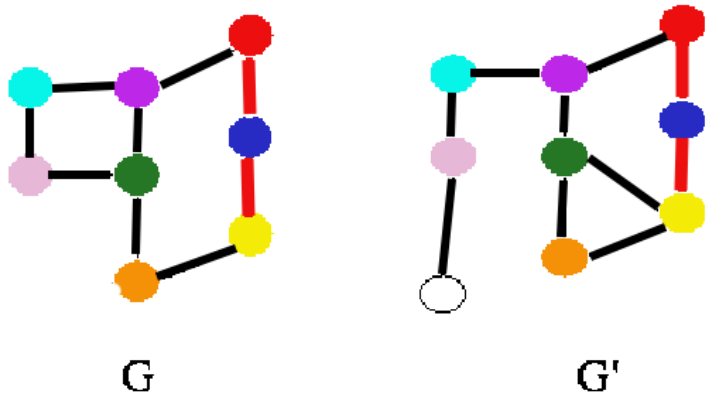
Graphe relationnel attribué de contours : Noyau sur graphes



Comment comparer deux graphes entre eux ?

- graphe non isomorphe
- graphe de grande taille (20 à 100 sommets)
- graphe valué sur les sommets et arcs
- recherche de similarités entre chemin du graphes

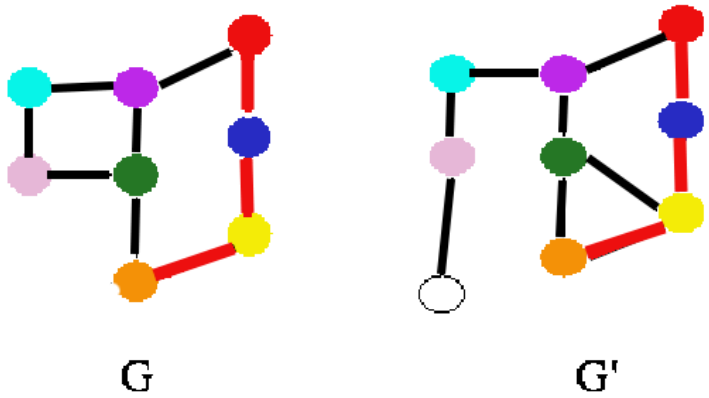
Graphe relationnel attribué de contours : Noyau sur graphes



Comment comparer deux graphes entre eux ?

- graphe non isomorphe
- graphe de grande taille (20 à 100 sommets)
- graphe valué sur les sommets et arcs
- recherche de similarités entre chemin du graphes

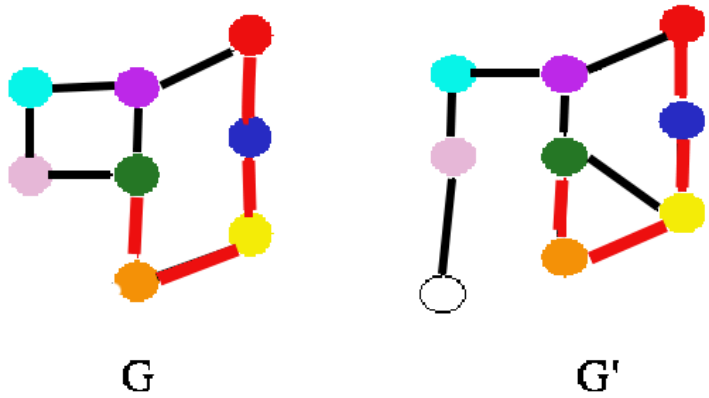
Graphe relationnel attribué de contours : Noyau sur graphes



Comment comparer deux graphes entre eux ?

- graphe non isomorphe
- graphe de grande taille (20 à 100 sommets)
- graphe valué sur les sommets et arcs
- recherche de similarités entre chemin du graphes

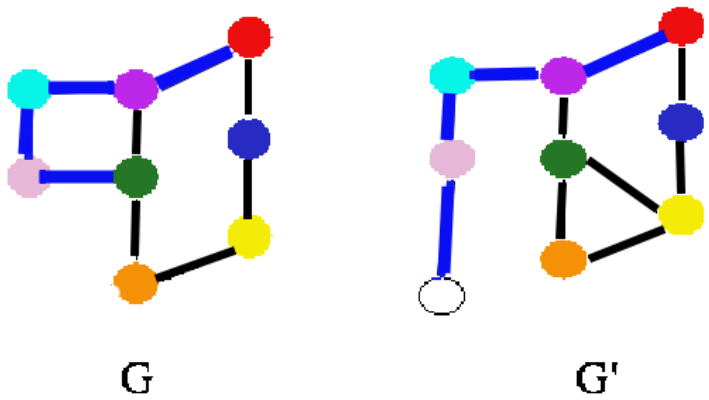
Graphe relationnel attribué de contours : Noyau sur graphes



Comment comparer deux graphes entre eux ?

- graphe non isomorphe
- graphe de grande taille (20 à 100 sommets)
- graphe valué sur les sommets et arcs
- recherche de similarités entre chemin du graphes

Graphe relationnel attribué de contours : Noyau sur graphes



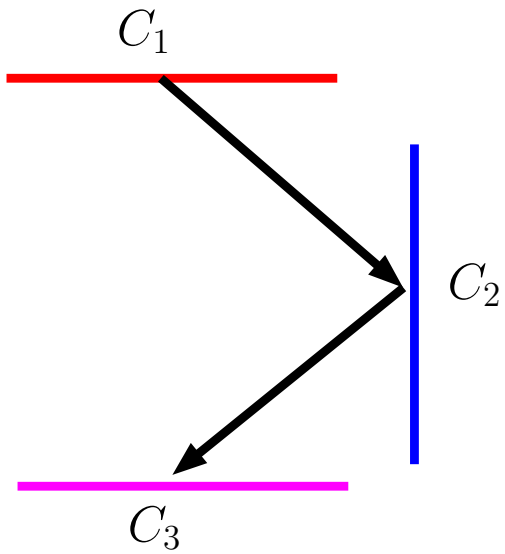
Comment comparer deux graphes entre eux ?

- graphe non isomorphe
- graphe de grande taille (20 à 100 sommets)
- graphe valué sur les sommets et arcs
- recherche de similarités entre chemin du graphes

Graphe relationnel attribué de contours : Notations

Notations

- $G = (V, E)$ Graphe G composé d'un ensemble V de sommets et d'un ensemble E d'arêtes.
- $h = (v_0, e_{01}, v_1, ..., v_n)$ chemin h succession de sommets et d'arêtes.
- $|h|$ taille du chemin = nombre d'arêtes.
- $H(G)$ ensemble de chemins possibles.
- h_{v_i} chemin débutant par le sommet v_i



Graphe relationnel attribué de contours : Noyau sur graphes [KASHIMA 04],[SUARD 06],[LEBRUN 08]

- Graphes sur molécules [KASHIMA 04]

$$K_{Kashima}(G, G') = \sum_{h \in H(G)} \sum_{\substack{h' \in H(G') \\ |h'|=|h|}} K_C(h, h') p(h|G) p(h'|G')$$

avec K_C un noyau sur chemins.

- Graphes sur points particuliers du squelette de l'objet [SUARD 06]

$$K_{Suard}(G, G') = \frac{1}{2} \left(\sum_{h \in H(G)} \max_{\substack{h' \in H(G') \\ |h'|=|h|}} K_C(h, h') + \sum_{h' \in H(G')} \max_{\substack{h \in H(G) \\ |h|=|h'|}} K_C(h, h') \right)$$

- Graphes sur régions floues [LEBRUN 08]

Graphe relationnel attribué de contours : Noyau sur graphes [Lebrun], [Gosselin], [Suard], [Kashima]

Soient G, G' deux graphes de contours, h_{v_i} chemin commençant par le sommet v_i.

Notre similarité est basée sur le noyau suivant :

$$K_{struct}(G,G') = \frac{1}{|V|}\sum_{i=1}^{|V|}\max_{\substack{h'\in H(G')\\ h_{v_i}\in H_{v_i}(G)\\ |h'|=|h_{v_i}|}} K_C(h_{v_i},h') + \frac{1}{|V'|}\sum_{i=1}^{|V'|}\max_{\substack{h\in H(G)\\ h'_{v_i}\in H_{v'_i}(G')\\ |h|=|h'_{v'_i}|}} K_C(h,h'_{v'_i}). \quad (1)$$

avec K_C un noyau sur chemins.

Remarques

Ce noyau est indéfini mais bien adapté à l'algorithme de "séparation et évaluation".

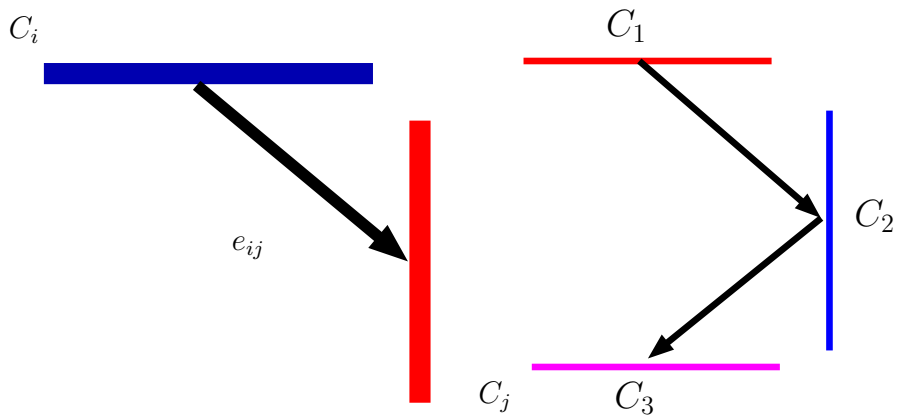
Noyau sur chemins

Noyau sur chemins proposé entre $h_{v_i} \in G$ et $h' \in G'$:

$$K_C(h_{v_i}, h') = K_v(v_i, v'_0) + \sum_{j=1}^{|h|} S_{e_i e'_i} O_{j,j-1} K_e(e_j, e'_j)\, K_v(v_j, v'_j). \quad (2)$$

avec :

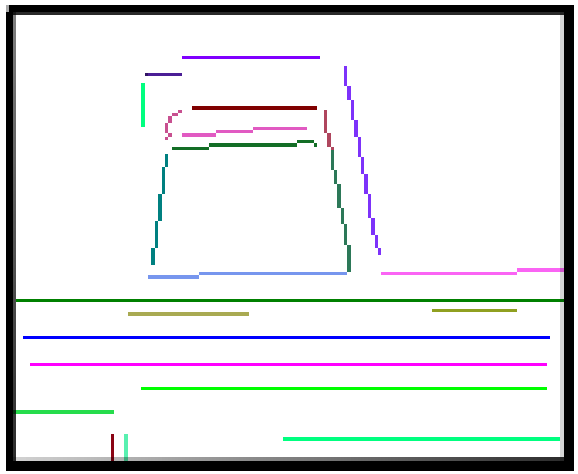
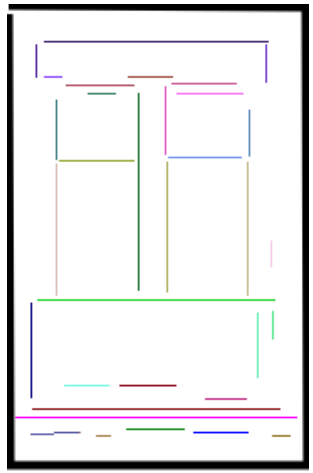
- noyau sur sommets : $K_v(v_j, v'_j) = \frac{\langle v_j, v'_j \rangle}{||v_j||.||v'_j||} + 1.$
- noyau sur arêtes : $K_e(e_j, e'_j) = \frac{\langle e_j, e'_j \rangle}{||e_j||.||e'_j||} + 1.$



Pondération orientation

Problème

Eviter de choisir les chemins ne possèdant que des contours dans la même orientation : effet "contours parallèles"

$$O_{i,j} = sin(\phi_{ij}) \times sin(\phi'_{ij}) = \sqrt{\frac{1}{2}(1 - \langle v_i, v_j \rangle)} \times \sqrt{\frac{1}{2}(1 - \langle v'_i, v'_j \rangle)}.$$

avec ϕ_{ij} (respectivement ϕ'_{ij}) l'angle entre les sommets i et j dans G (G').

Pondération d'échelle

Problème

Réaliser le meilleur groupement perceptuel.
Ne pas choisir un contour trop éloigné de la structure de l'objet.

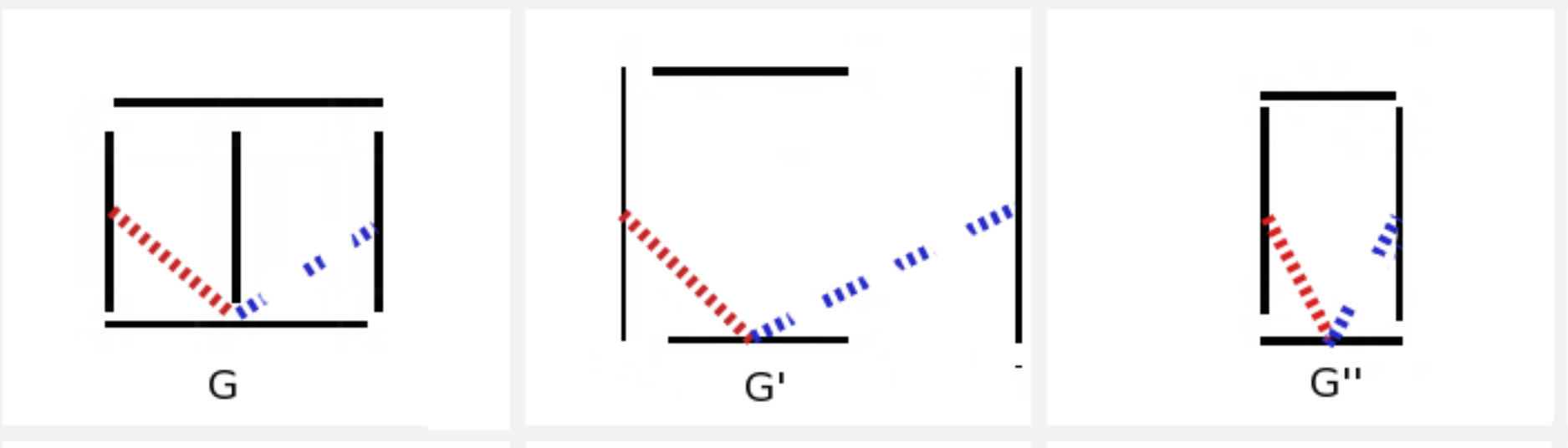


FIGURE: Exemple : structure et problème d'échelle. Le segment de contour sur la droite du graphe G' est-t-il un contour de l'objet cherché ou non ?

$$S_{e_i e'_i} = \min_{\substack{e_i \in h \\ e'_i \in h'}} \left(\frac{\frac{||e_{i-1}||}{||e'_{i-1}||}}{\frac{||e_i||}{||e'_i||}} \right)$$

Evaluations : classifications des hypothèses de fenêtres

220 imagettes extraites par l'algorithme précédent :

- 70 fenêtres
- 150 faux-positifs

(a) Fenêtres (b) Négatifs

FIGURE: Exemples d'hypothèses de notre base : 70 fenêtres et 150 négatifs

La matrice de Gram est définie positive sur l'ensemble des données de la base.

Recherche interactive : RETIN

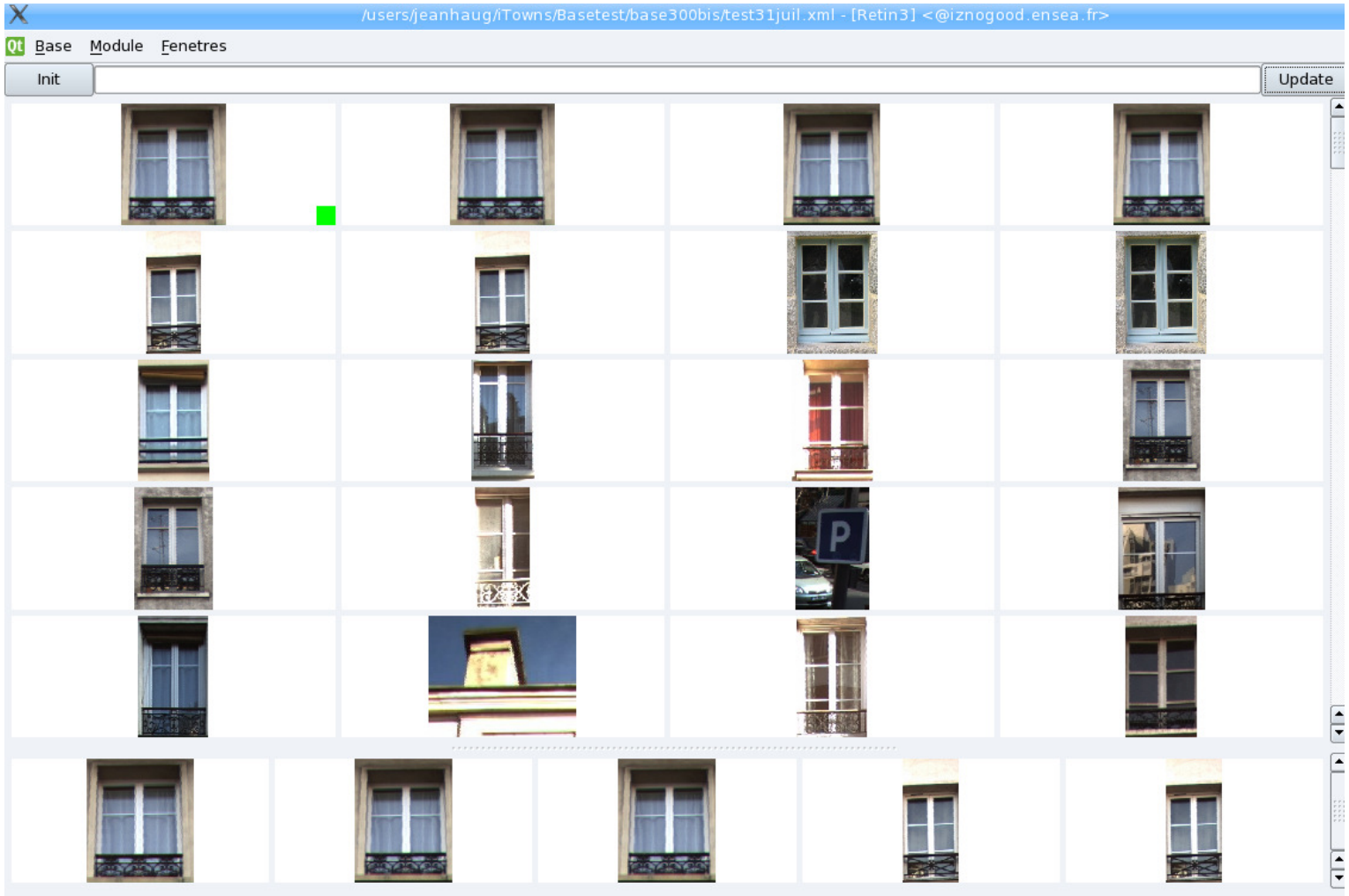


FIGURE: Exemple d'une session de recherche. L'image en haut à gauche avec un petit carré vert est l'image annotée positivement, et les images qui suivent sont ses plus proches voisines. Le classement est fait de gauche à droite puis de haut en bas.

Recherche interactive : RETIN

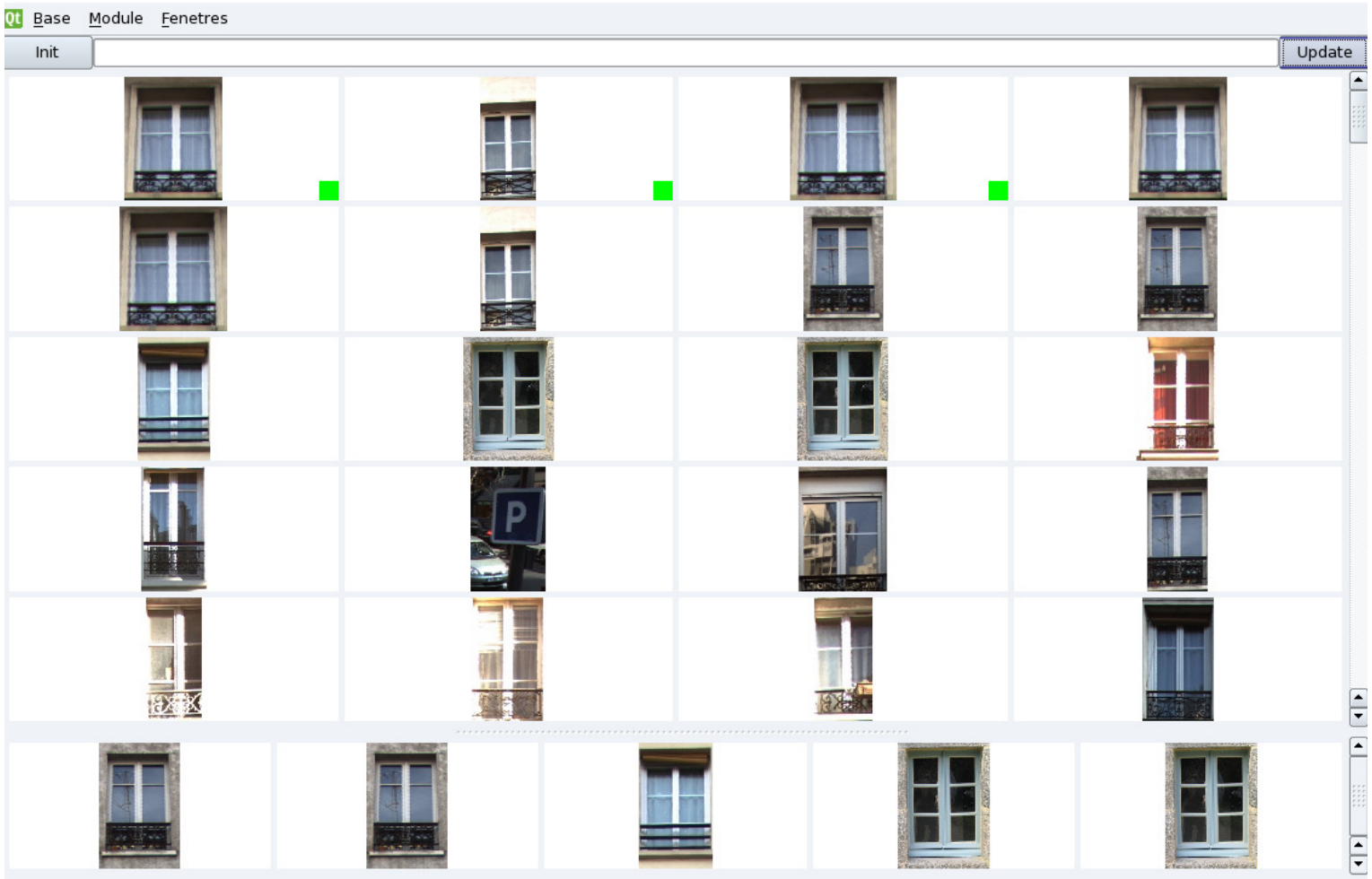


FIGURE: Classement après une itération avec 3 labels positifs pour affiner la recherche.

Recherche interactive : RETIN

FIGURE: Annotations suite au classement précédent. Nous annotons les éléments les plus pertinents sélectionnés par le système.

Recherche interactive : RETIN

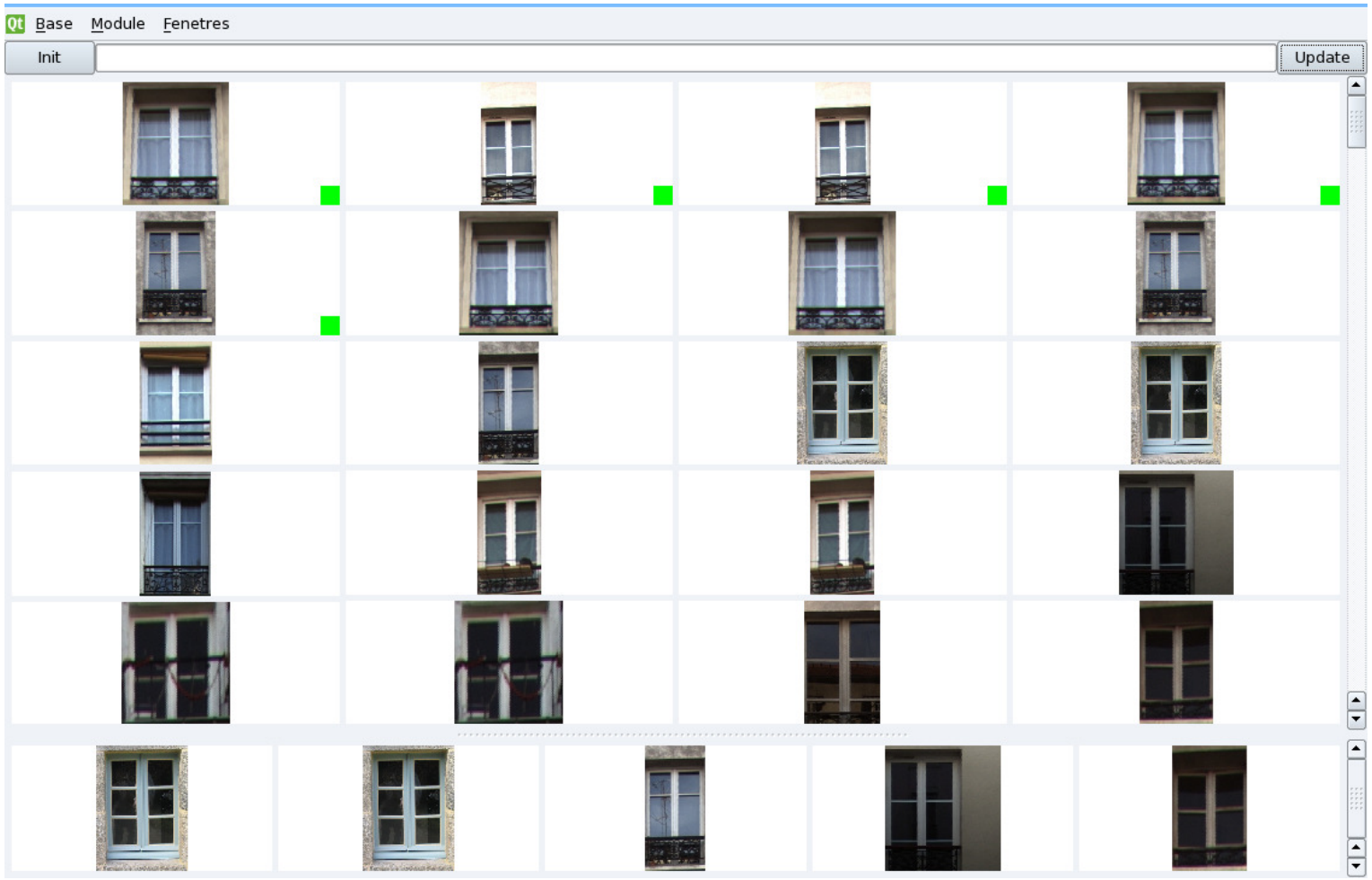
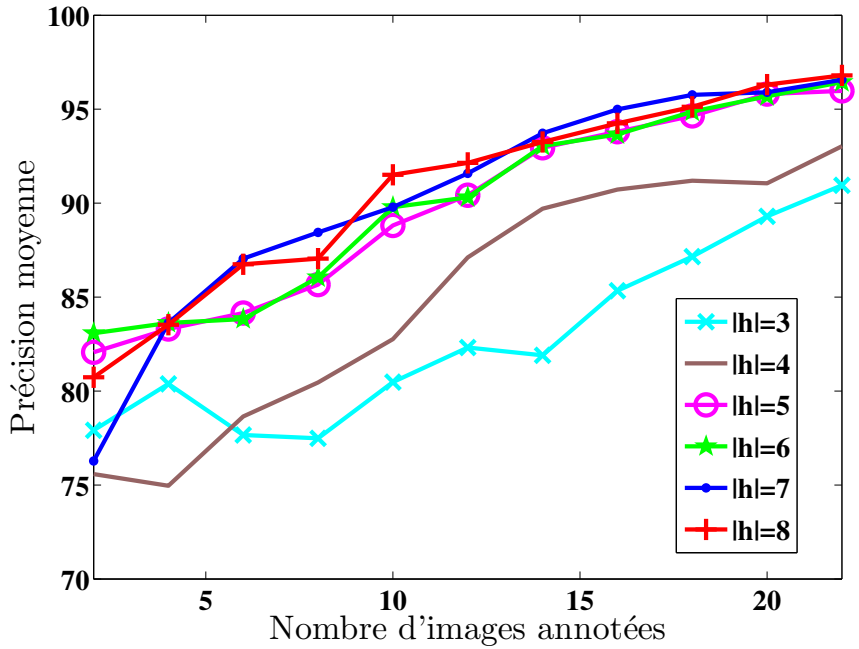


FIGURE: Résultats après 4 itérations : 5 images annotées positivement et 4 négativement.

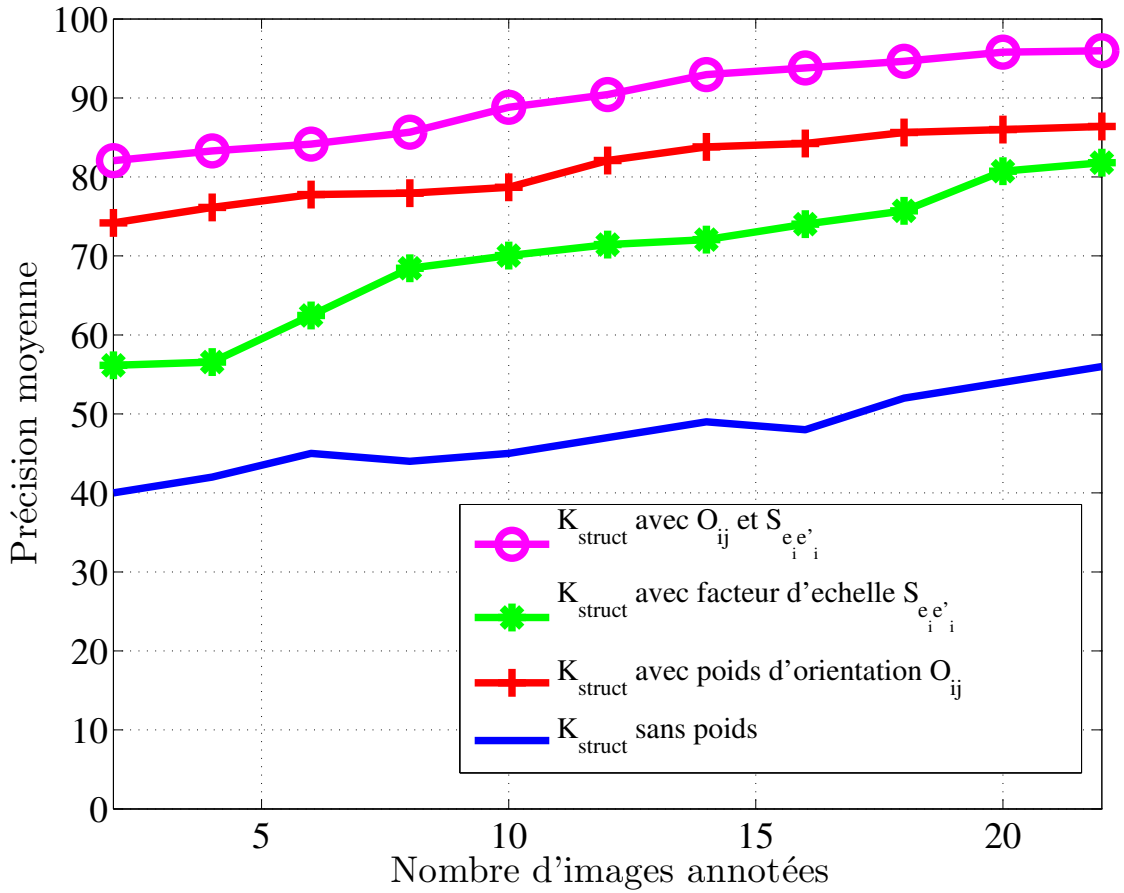
Evaluation de la taille des chemins



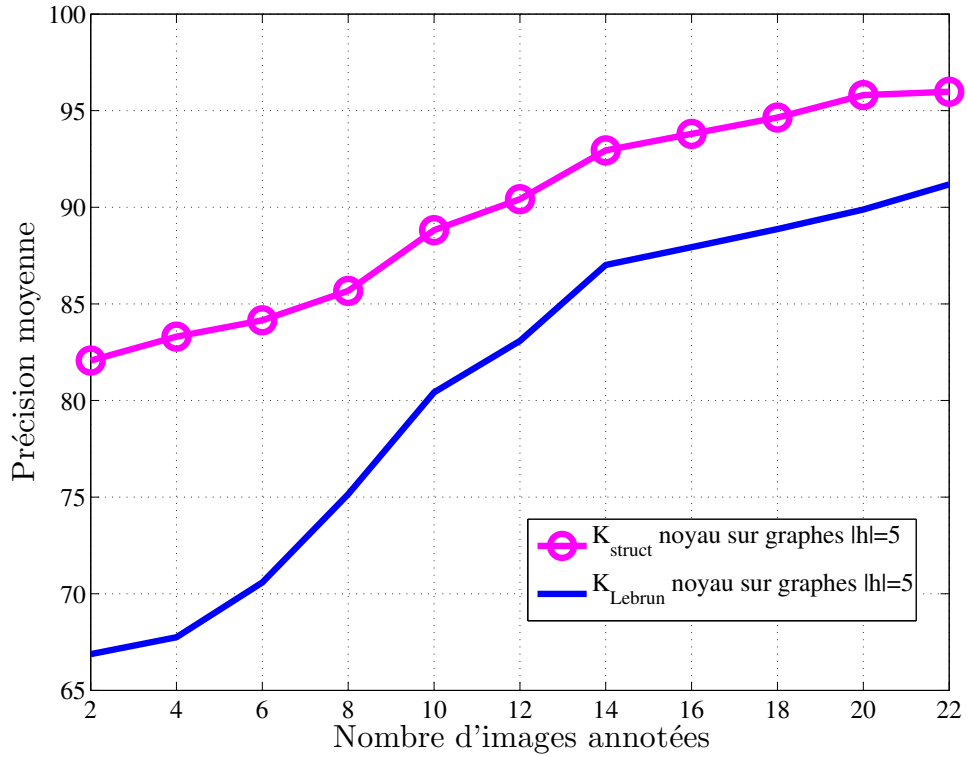
Longueur	3	4	5	6	7	8
Temps moyen (millisecondes)	55	61	67	77	84	94

TABLE: Temps de calcul moyen (en millisecondes) pour calculer la similarité entre un graphe G de 30 contours et un autre graphe, pour différentes longueurs de chemins.

Evaluation des pondérations ($|h| = 5$)



Comparaison avec le noyau sur chemins de Lebrun



$$K_{Lebrun}(G,G') = \frac{1}{|V|}\sum_{i=1}^{|V|} \max_{\substack{h_{v_i}\in H_{v_i}(G)\\ h'_{s(v_i)}\in H_{s(v_i)}(G')\\ |h_{v_i}|=|h'_{s(v_i)}|}} K_C(h_{v_i}, h'_{s(v_i)}) + \frac{1}{|V'|}\sum_{i=1}^{|V'|} \max_{\substack{h_{s(v'_i)}\in H_{s(v'_i)}(G)\\ h'_{v'_i}\in H_{v'_i}(G')\\ |h_{s(v'_i)}|=|h'_{v'_i}|}} K_C(h_{s(v'_i)}, h'_{v'_i}).$$

Complexité de calcul

- Arbre de recherche et algorithme d'"évaluation et séparation"
  - Avantage de ne pas explorer l'ensemble de l'arbre de recherche
- Dictionnaire de chemins et noyau incrémental
  - Réduire le nombre de chemins à comparer
  - Calcul incrémental
  - Concevoir des noyaux de Mercer

Théorie [GOSSELIN 11]

Soit un dictionnaire de chemins : $D_L = \{\mathbf{h}_l\}_{l\in[1,L]}$ de chemins $\mathbf{h}_l$

Noyau sur dictionnaire entre deux graphes G_i et G_j :

$$K_L(G_i, G_j) = S\left(\sum_{l=1}^{L} k_{\mathbf{h}_l}(G_i, G_j)\right)$$

Le calcul incrémental devient :

$$K_{L+1}(G_i, G_j) = S\left(\sum_{l=1}^{L} k_{\mathbf{h}_l}(G_i, G_j) + k_{\mathbf{h}_{L+1}}(G_i, G_j)\right)$$

Le noyau mineur :

$$k_{\mathbf{h}_l}(G_i, G_j) = -\delta(e_{\mathbf{h}_l}(G_i), e_{\mathbf{h}_l}(G_j))$$

avec $e_{\mathbf{h}_l}$ une fonction d'évaluation

Notre cas : dictionnaire de contours

La fonction d'évaluation choisie

$$e_{\mathbf{h}_l}(G_i) = \max_{\mathbf{h} \in G_i} K_C(\mathbf{h}_l, \mathbf{h})$$

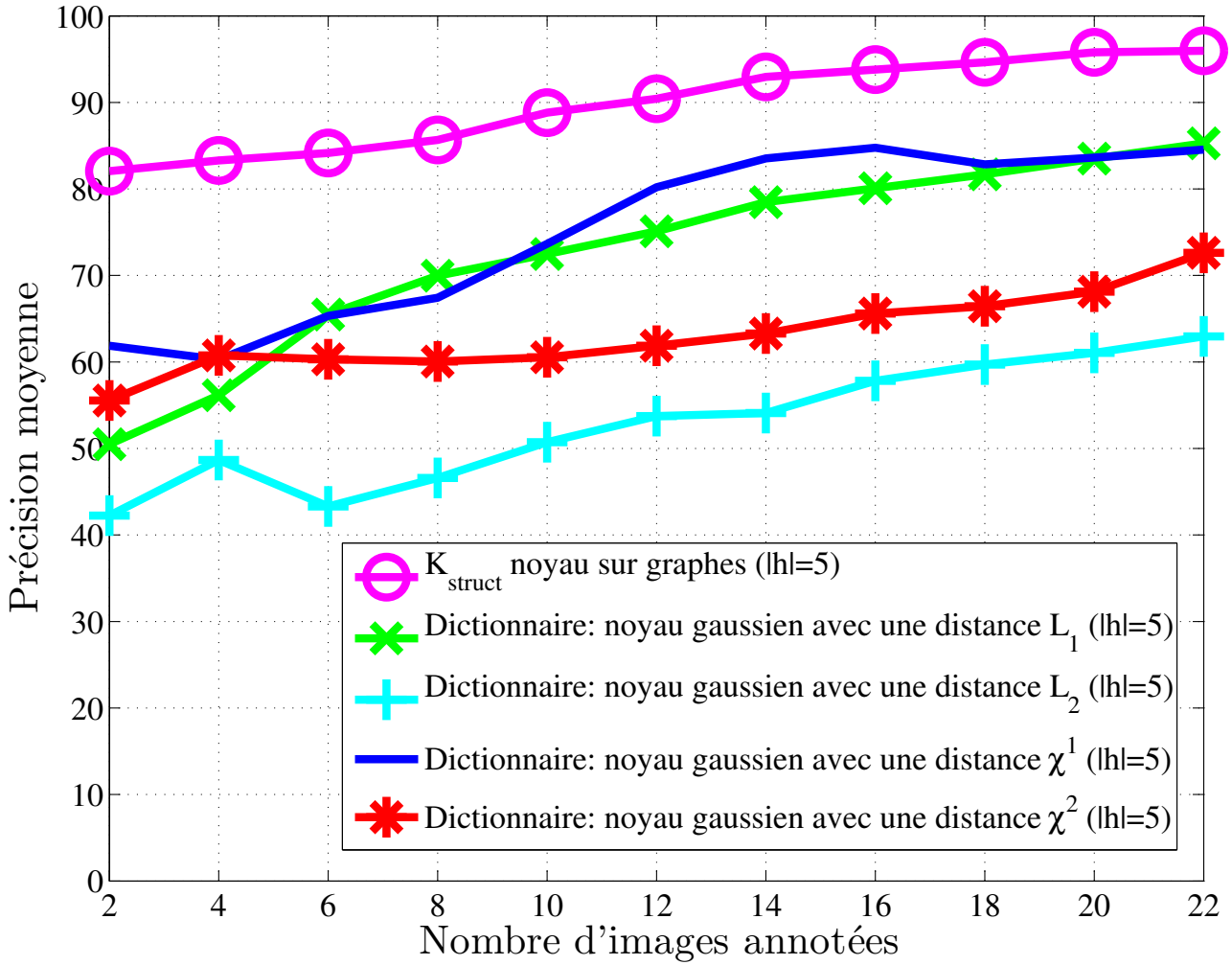
- Noyau triangulaire

$$\begin{aligned} K_L^{Tri}(G_i, G_j) &= \sum_{l=1}^{L} k_{\mathbf{h}_l}(G_i, G_j) \\ &= -\sum_{l} \delta(e_{\mathbf{h}_l}(G_i), e_{\mathbf{h}_l}(G_j)) \\ &= -d(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

- Noyau gaussien

$$\begin{aligned} K_L^{Gau}(G_i, G_j) &= \exp\left(\frac{1}{2\sigma^2} \sum_{l=1}^{L} k_{\mathbf{h}_l}(G_i, G_j)\right) \\ &= \exp\left(-\frac{1}{2\sigma^2} \sum_{l=1}^{L} \delta(e_{\mathbf{h}_l}(G_i), e_{\mathbf{h}_l}(G_j))\right) \\ &= \exp\left(-\frac{1}{2\sigma^2} d(\mathbf{x}_i, \mathbf{x}_j)\right) \end{aligned}$$

Evaluation : dictionnaire de contours



Evaluation : dictionnaire de contours

	Nombre de noeuds dans le graphe G			
	26	30	40	124
K_{struct}	28	68	132	362
Dictionnaire avec noyau triangulaire	3	4	6	34
Dictionnaire avec noyau gaussien	4	5	8	37

TABLE: Temps de calcul moyen (en millisecondes) pour le calcul de similarité de graphes entre un graphe G et un autre graphe pour des chemins de taille $|h| = 5$.

Plan

1. Extraction des fenêtres dans les images de façades
   - Etat de l'art
   - Notre démarche
   - Résultats
2. Graphe relationnel attribué de contours et similarités sur graphes
   - Représentation graphes de contours
   - Attributs sur sommets et arêtes
3. Noyau sur graphes de contours
   - Noyau sur chemins
   - Complexité de calcul
4. Localisation des fenêtres dans les façades à l'aide des graphes

Localisation à l'aide des contours

Comparaison d'un graphe G de fenêtre et d'un graphe G' d'une façade, K_{struct} est -il toujours pertinent ?

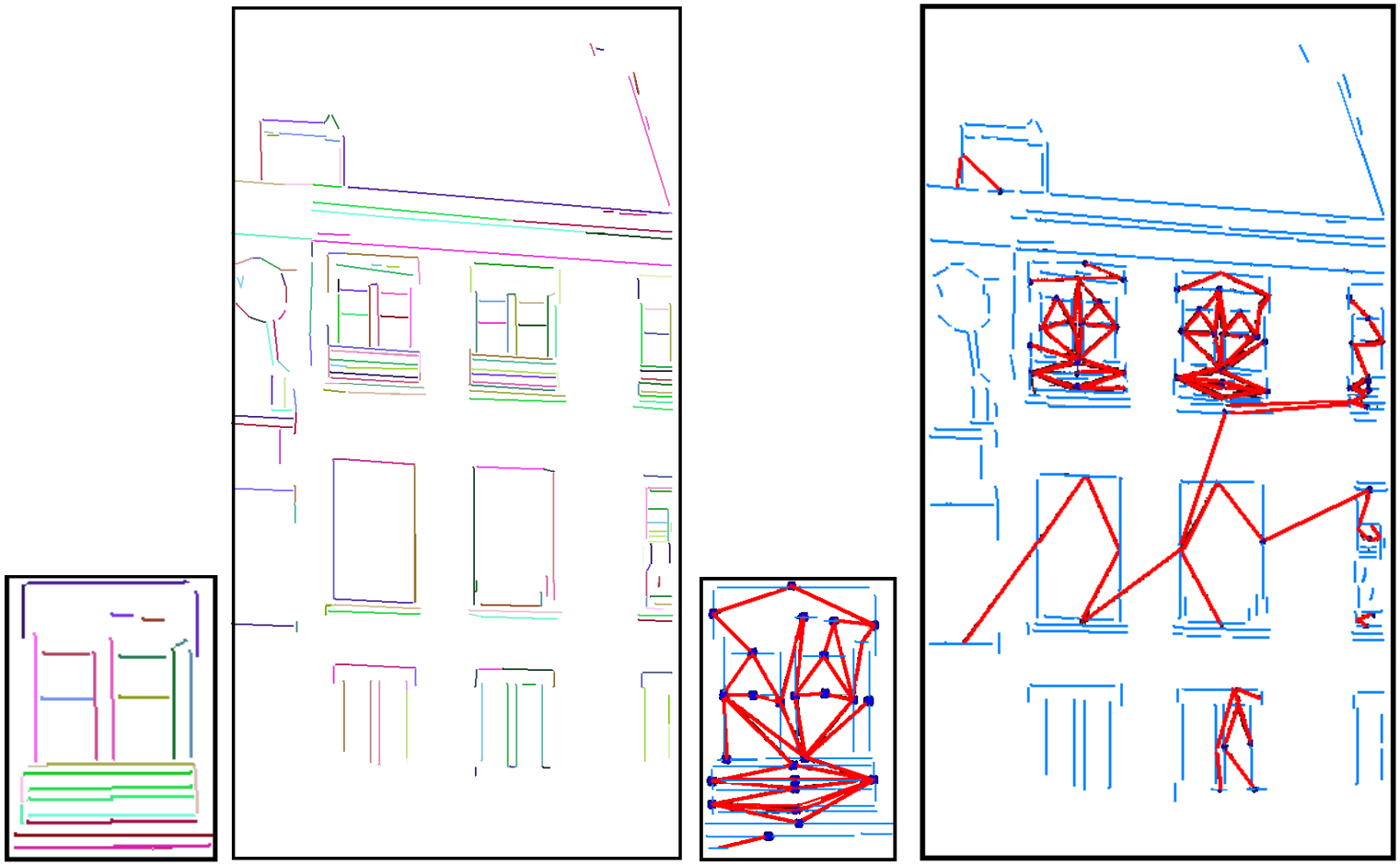
$$K_{struct}(G, G') = \frac{1}{|V|} \sum_{i=1}^{|V|} \max_{\substack{h' \in H(G') \\ h_{v_i} \in H_{v_i}(G) \\ |h'| = |h_{v_i}|}} K_C(h_{v_i}, h') + \frac{1}{|V'|} \sum_{i=1}^{|V'|} \max_{\substack{h \in H(G) \\ h'_{v_i} \in H_{v'_i}(G') \\ |h| = |h'_{v'_i}|}} K_C(h, h'_{v'_i}).$$

Nous redéfinissons une autre similarité non symétrique :

$$S_{struct} = \frac{1}{|V|} \sum_{i=1}^{|V|} \max_{\substack{h' \in H(G') \\ h_{v_i} \in H_{v_i}(G) \\ |h'| = |h_{v_i}|}} S_C(h_{v_i}, h').$$

avec $S_C(h_{v_i}, h') = K_v(v_i, v'_0) + \sum^{|h|} S_{e_i e'_i} O_{j,j-1} K_e(e_j, e'_j) K_v(v_j, v'_j)$.

Localisation à l'aide des contours



(a) (b) (c) (d)

FIGURE: **Les meilleurs appariements.**

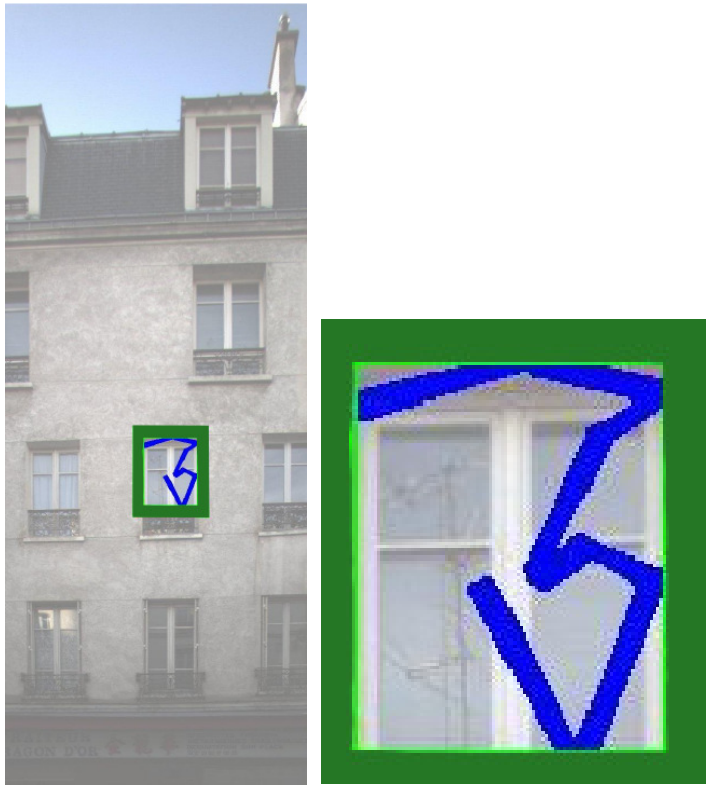
Localisation à l'aide des contours

Nous avons besoin de plus de chemins, la similarité devient :

$$S_{struct}(G, G^{'}) = \frac{1}{n_a} \sum_{i=1}^{|V|} \sum_{|h|=3}^{|h|=l} \max_{h_{v_i}, h'} S_C(h_{v_i}, h^{'})$$

Similarités des façades par rapport à une requête

Définition d'une région d'intérêt

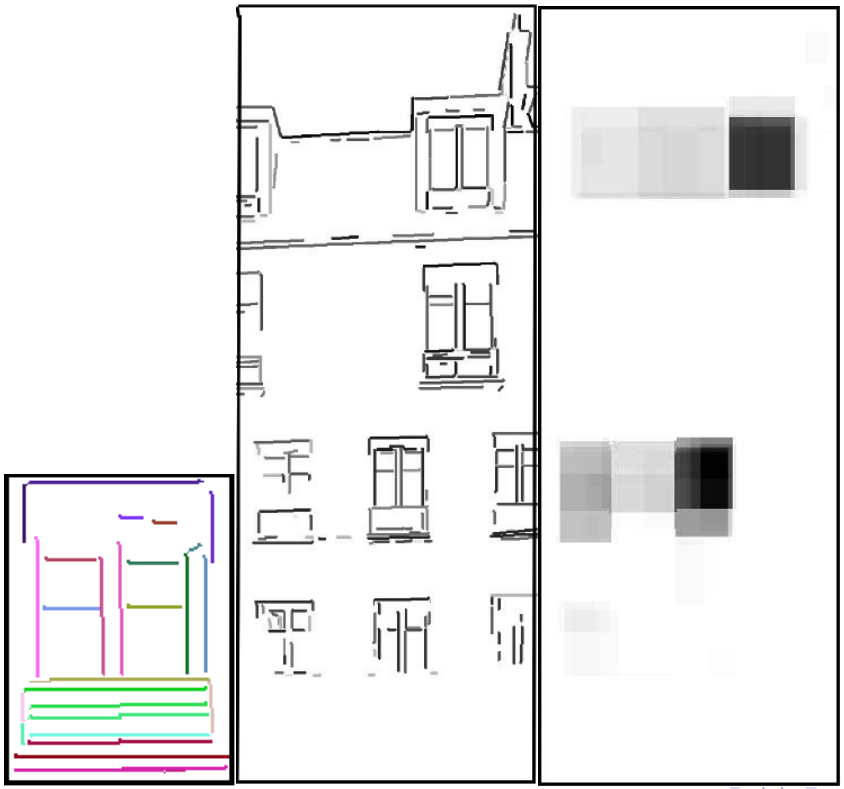


(a) (b) (c)

FIGURE: **Localisation des fenêtres. (a)** Un chemin et sa région d'intéret dans la façade. **(b)** Zoom sur le chemin.

Vote des régions d'intérêt

$$J_{G,G'}(x,y) = \sum_{i=1}^{|V|} J_{h_{v_i},h'}(x,y)$$
$$= \sum_{i=1}^{|V|} S_C(h_{v_i}, h^{'})$$



Localisation à l'aide d'une approche contours puis régions

Utiliser les régions pour mieux discriminer les mauvais appariements.

La similarité devient :

$$S_{struct}(G, G') = \frac{1}{n_a} \sum_{i=1}^{|V|} \sum_{|h|=3}^{|h|=l} S_R(B_{h_{v_i}}, B_{h'}) \max_{h_{v_i}, h'} S_C(h_{v_i}, h')$$

L'accumulation de votes devient :

$$J_{h_{v_i}, h'}(x, y) = S_R(B_{h_{v_i}}, B_{h'}) S_C(h_{v_i}, h_{v_i})$$

Similarités par rapport aux façades

L'extraction des fenêtres

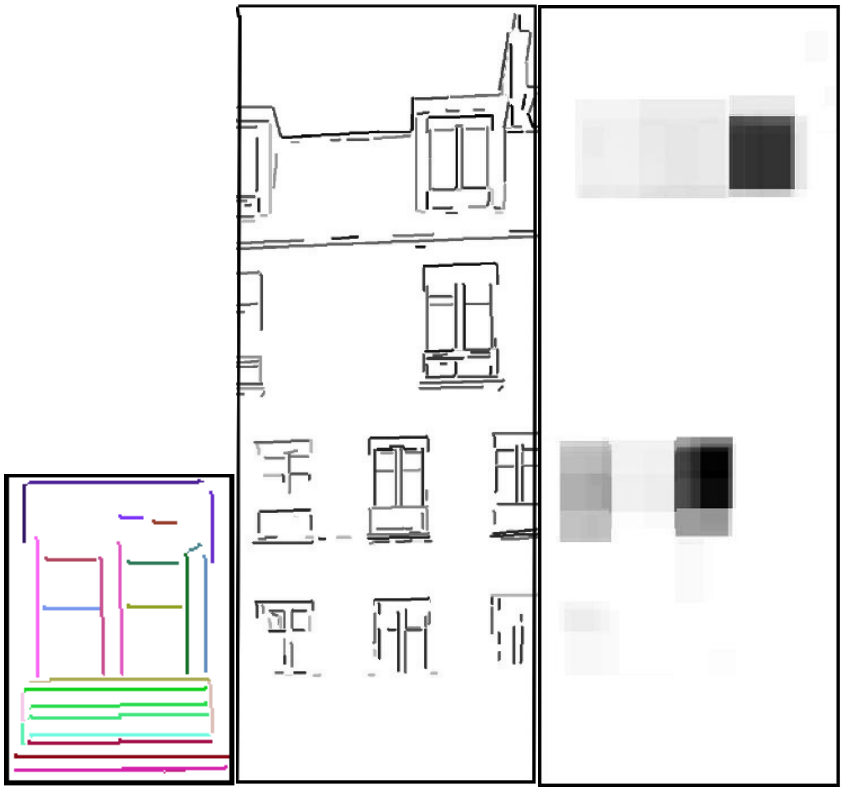


FIGURE: **Accumulation des votes de régions d'intérêt sélectionnées sur la similarité chemins, puis combinées avec la similarité régions.**

Les mauvais appariements sont moins importants qu'avec l'accumulation uniquement sur la similarité chemins.

Localisation à l'aide d'une approche contours et régions

Utiliser les régions pour mieux selectionner les chemins.

La similarité devient :

$$S_{struct}(G, G') = \frac{1}{n_a} \sum_{i=1}^{|V|} \sum_{|h|=3}^{|h|=l} \max_{h_{v_i}, h'} S_{C_{CheReg}}(h_{v_i}, h')$$

avec

$$\begin{aligned} S_{C_{CheReg}}(h_{v_i}, h') &= K_v(v_i, v_0') \\ &+ \sum_{j=1}^{|h|} S_R(B_{h_{v_i}} B_{h'_{v_j'}}) S_{e_i e_i'} O_{j,j-1} K_e(e_j, e_j') K_v(v_j, v_j') \end{aligned}$$

L'accumulation de votes devient :

$$\begin{aligned} J_{G,G'}(x, y) &= \sum_{i=1}^{|V|} J_{h_{v_i}, h'}(x, y) \\ &= \sum_{i=1}^{|V|} S_{C_{CheReg}}(h_{v_i}, h') \end{aligned}$$

Similarités par rapport aux façades

FIGURE: Une requête et les images retournées par ordre de similarité conjointe sur les contours et les régions.

L'extraction des fenêtres

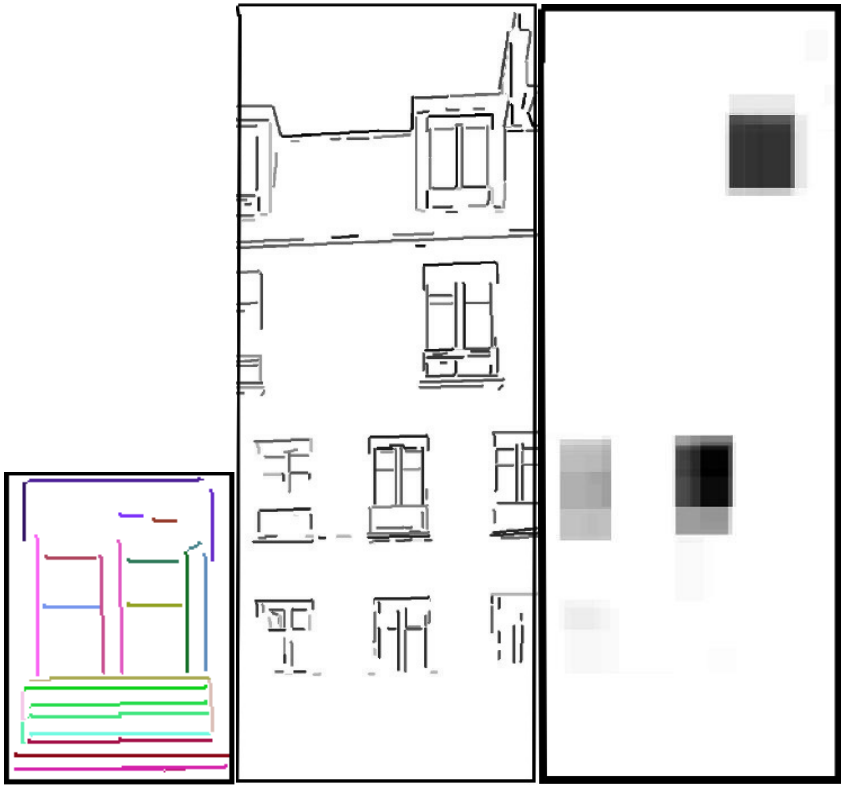
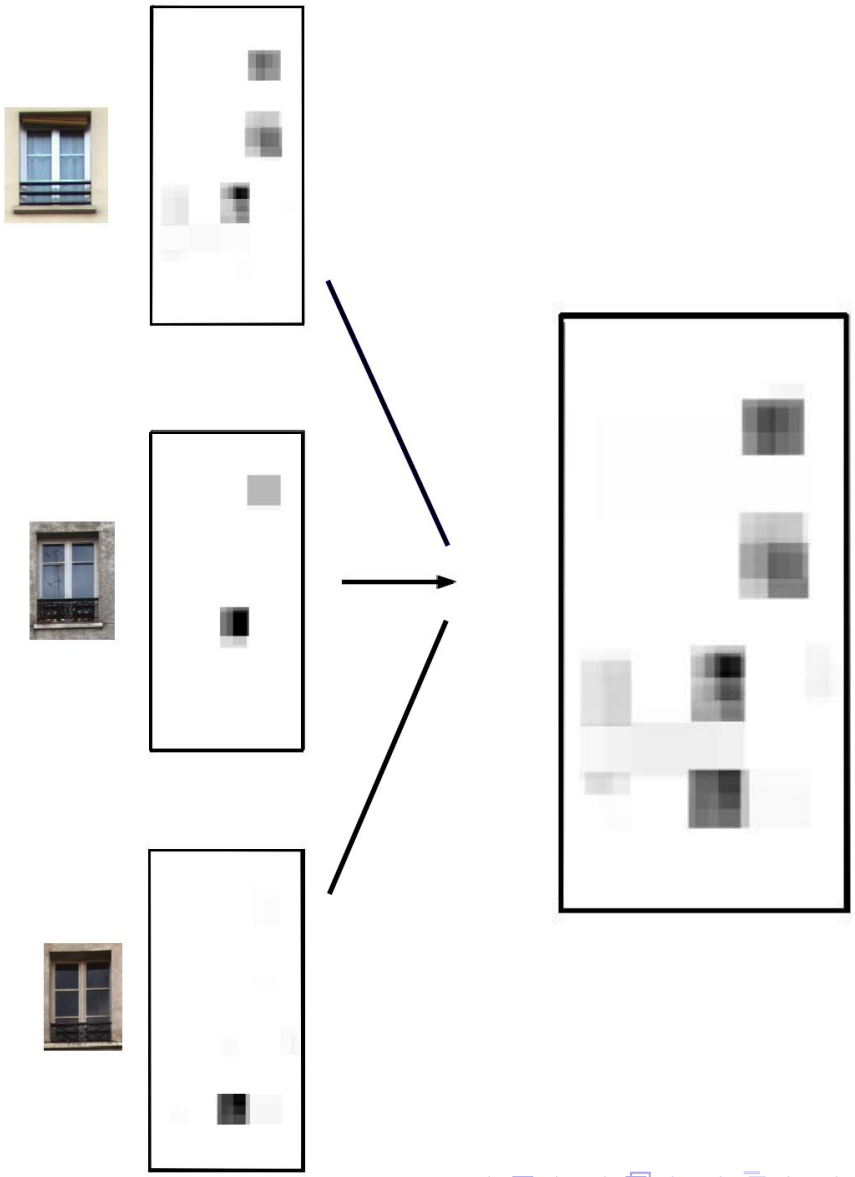


FIGURE: Accumulation des votes de régions d'intérêt sélectionnées sur la similarité combinant chemins et régions.

Le choix des régions d'intérêt devient plus pertinent.

Apprentissage par k-ppv, Accumulation sur une ensemble d'apprentissage de 3 fenêtres

$$J_{G'}(x,y) = \frac{1}{k}\sum_{i=1}^{k} y_i J_{G_i,G'}(x,y)$$



Conclusion

Mise en place d'une chaine de traitement :

- Détection : Extraire des fenêtres à l'aide d'un algorithme automatique
- Description : Décrire les objets géométriques par des graphes de contours
- Mise en place d'un noyau sur chemins et de similarité sur chemins
  - attributs
  - pondérations
  - combiner région et contours

iTowns

- Mise en production sur la rue Saint Antoine de l'algorithme d'extraction
- Apprentissage sur un ensemble de fenêtres

temps de calcul

- Utilisation de l'algorithme de "séparation et d'évaluation"
- Dictionnaire de chemins de segments

Publications

- 3 conférences internationales : SCIA 09, ICIP 09, ICPR 10
- 1 conférence nationale : RFIA 10
- 1 article de journal soumis PRL

Perspectives

- Généraliser à d'autres objets, modifier les pondérations et attributs
- Accélérer les temps de calcul
- Eliminer les chemins redondants dans le dictionnaire

Extraction et reconnaissance de primitives dans les façades de Paris à l'aide de similarités de graphes

HAUGEARD Jean-Emmanuel

ETIS - CNRS - ENSEA - Université Cergy-Pontoise

Directrice de thèse : Sylvie Philipp-Foliguet
Co-Directeurs de thèse : Frédéric Precioso et Philippe-Henri Gosselin

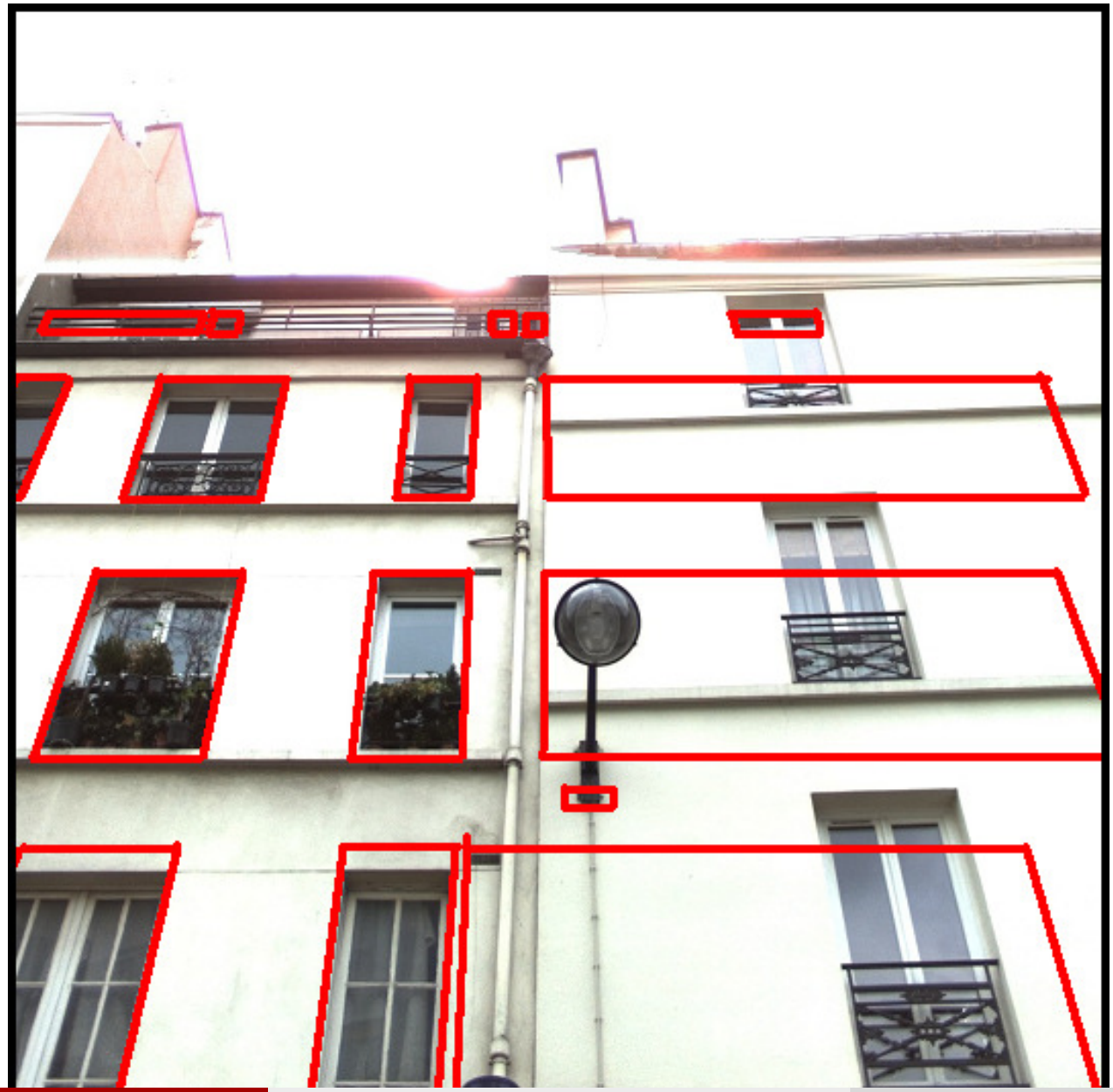
17 décembre 2010

Limites

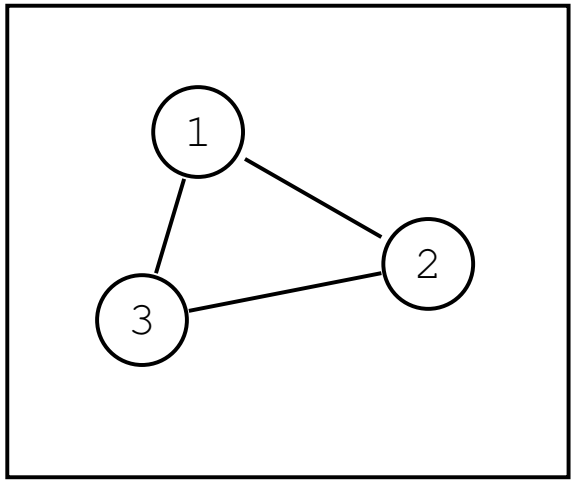
(a) Notre algorithme (b) Lee

FIGURE: **Extraction des hypothèses de fenêtres.** Nous constatons que la recherche des étages par notre algorithme permet de ne pas bruiter la recherche des fenêtres sur les autres étages.

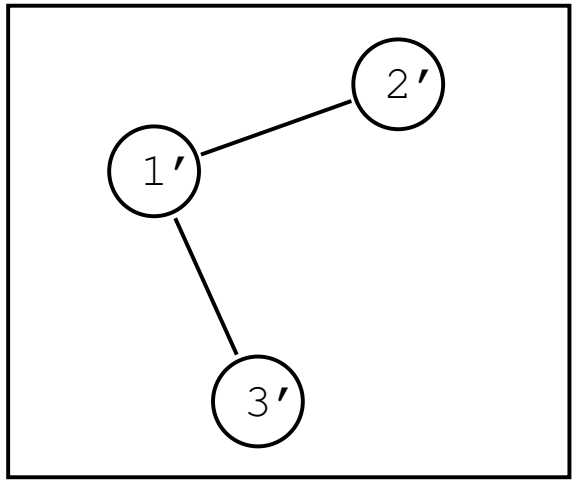
Limites



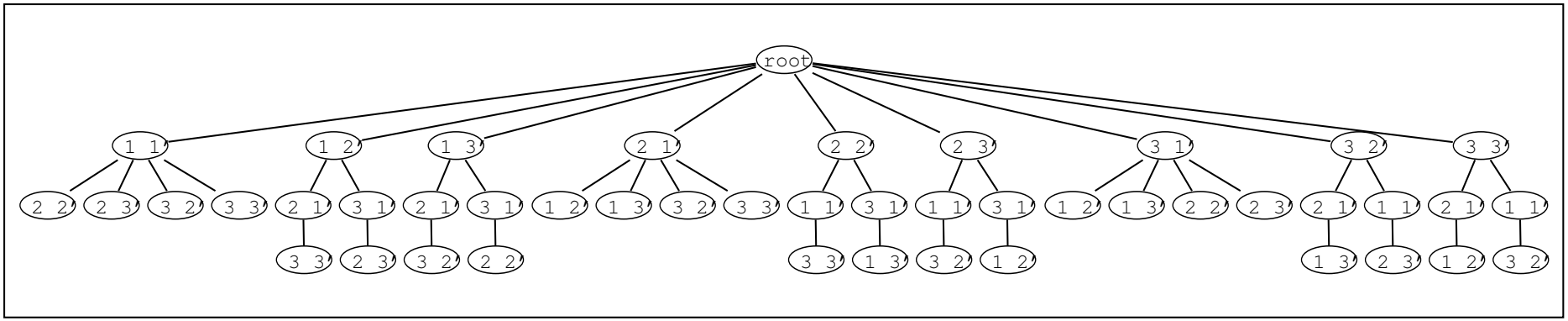
Implémentation : arbre de recherche



(a) Un exemple de graphe G



(b) Un exemple de graphe G'



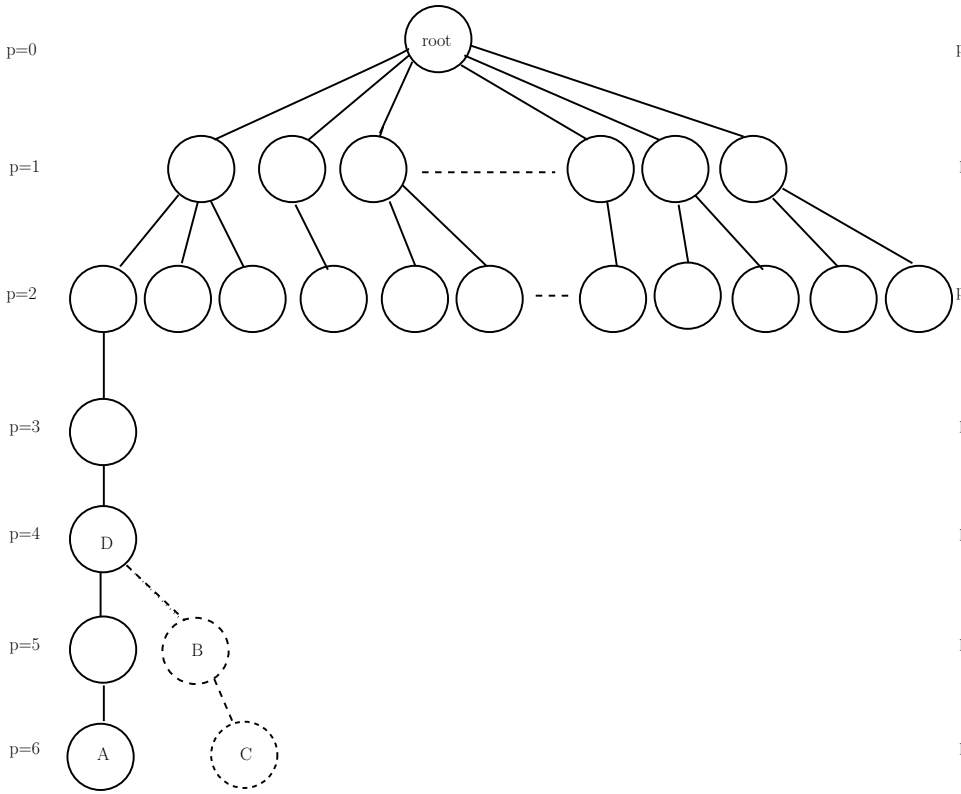
(c) L'arbre de recherche correspondant, sans cycle et boucle.

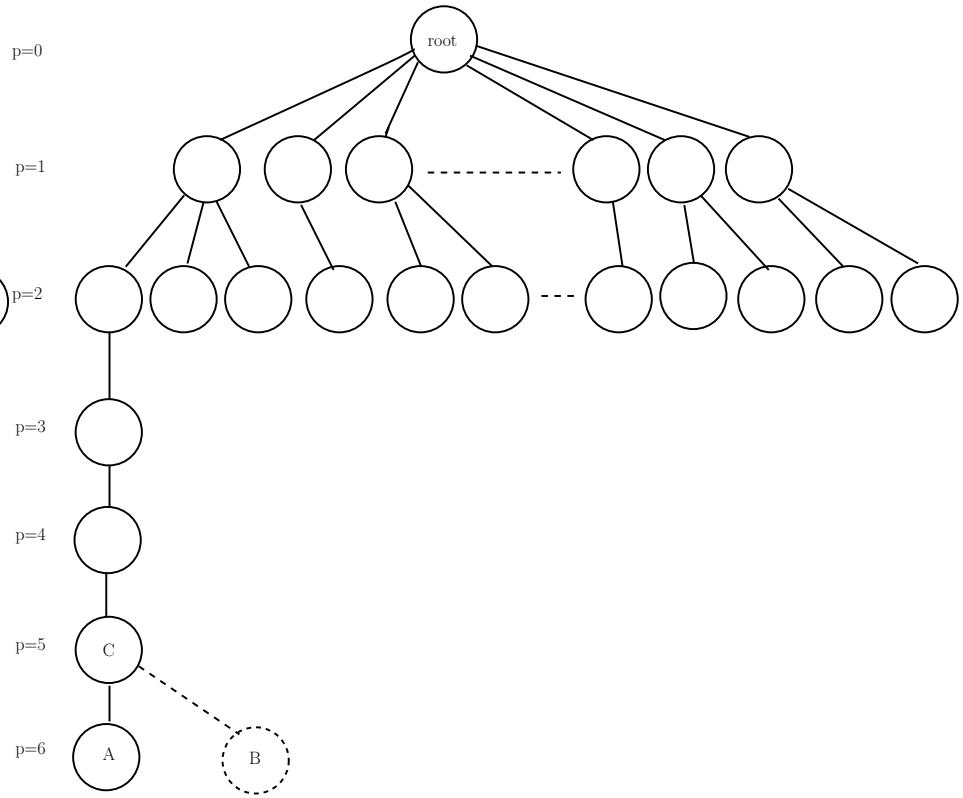
FIGURE: **Un exemple d'arbre de recherche.** Chaque parcours depuis la racine vers une feuille est un appariement possible entre deux chemins de G et G'. Par exemple, le chemin correspondant à la comparaison des chemins (213) and ($3'1'2'$) est ($root$) $\rightarrow$ ($23'$) $\rightarrow$ ($11'$) $\rightarrow$ ($32'$).

Implémentation : arbre de recherche

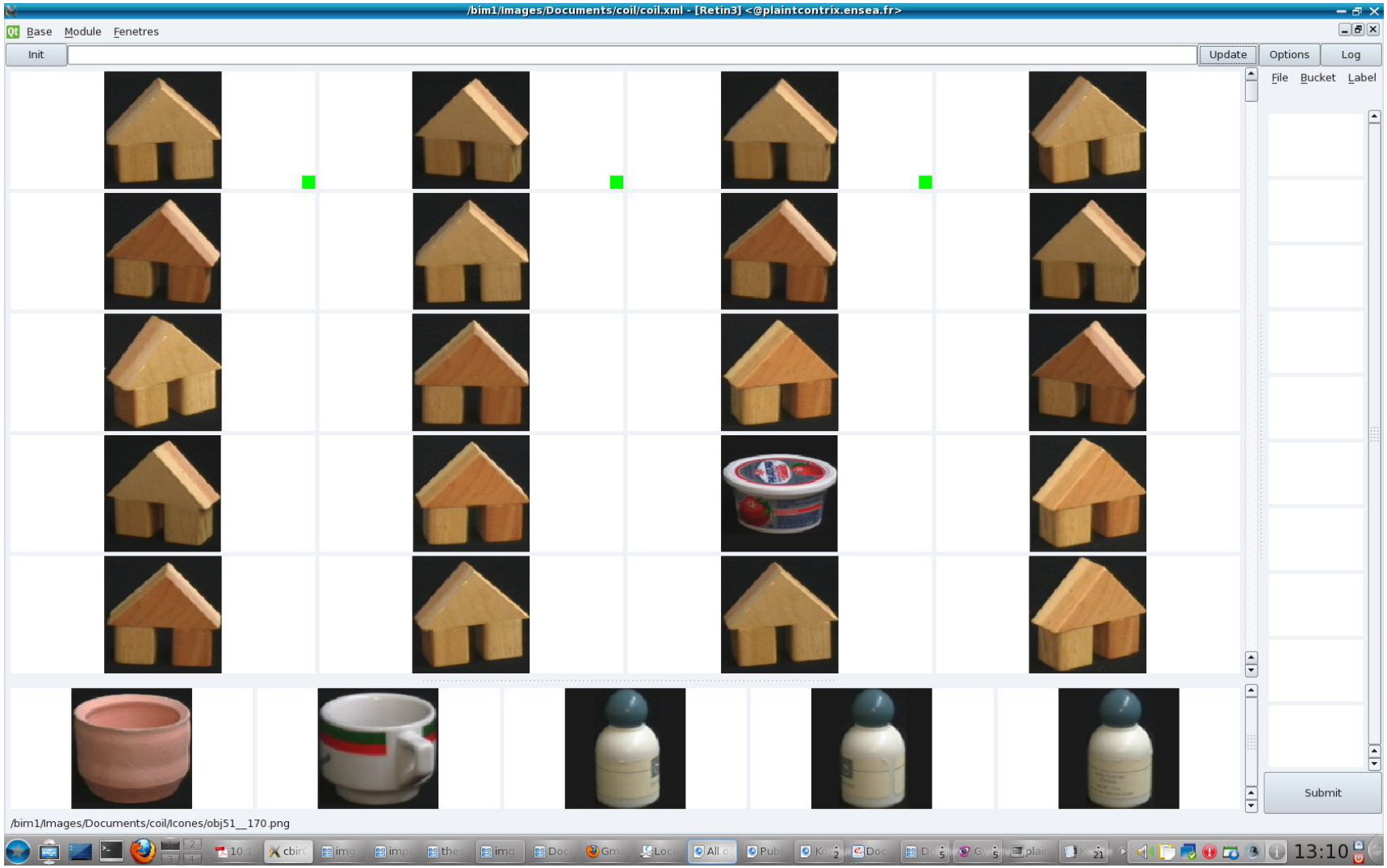
Algorithme d'évaluation et séparation [Clausen]

Seules les solutions potentiellement bonnes sont explorées.





Objets géométriques



Autres objets