

Amélioration de l'algorithme

- Schéma numérique :
 - schéma explicite
- Méthode numérique
 - Algorithme glouton
 - Programmation dynamique
- Calcul de la force externe
 - Gradient Vector Flow
- Evolution de la forme
 - Force interne tangentielle et normale
- Changement de topologie



Schéma explicite

- Estimation de la force interne à l'instant t

$$\begin{cases} X^{t+1} = (I + \Delta t K)X^t + \Delta t F_{ext}(X^t, Y^t) \\ Y^{t+1} = (I + \Delta t K)Y^t + \Delta t F_{ext}(X^t, Y^t) \end{cases}$$

- Critère de Stabilité : $\Delta t \leq \Delta t_{\text{critique}}$
 - $[K]$ est une matrice symétrique pentadiagonale
 - Critère de stabilité est lié aux valeurs propres de $I + \Delta t [K]$: $|v_m| < 1$



Schéma explicite

- Si le contour est fermé alors $[K]$ est une matrice circulante donc les valeurs propres sont de la forme :

$$\{v_m\}_{m \in [0, N-1]} \stackrel{0 < a < \frac{1}{4}}{=} \left\{ \sum_{k=0}^{N-1} c_k e^{\frac{2ipkm}{N}} \right\}_m$$

$$v_m = 1 + \frac{-2a \Delta t}{h^2} + \frac{-6b \Delta t}{h^4} + \Delta t \left(\frac{2a}{h^2} + \frac{8b}{h^4} \right) \cos \left(2p \frac{m}{N} \right) + \frac{-2b \Delta t}{h^4} \cos \left(4p \frac{m}{N} \right) < 1$$



Algorithme glouton

- Chaque sommet est modifié de manière indépendante (méthode asynchrone)
 - Pour chaque sommet, détermine la position qui minimise l'énergie globale
 - Séquence de mise à jour dépend de la valeur locale de $F_{\text{int}}(P_i) + F_{\text{ext}}(P_i)$
 - Plus efficace mais moins robuste
 - Equivalent à l'algorithme de Gauss-Siedel



Amélioration de l'algorithme

- Schéma numérique :
 - schéma explicite
- • Méthode d'optimisation
 - Algorithme glouton
 - Programmation dynamique
- Calcul de la force externe
 - Gradient Vector Flow
- Evolution de la forme
 - Force interne tangentielle et normale
- Changement de topologie



Programmation Dynamique (1)

- Hypothèses :
 - Sommets du contour coïncident avec la grille de l'image
 - Le voisin du sommet P_i est contraint à être dans une parmi m configurations
 - Energie est décomposable par arête :

$$E(P_0, \dots, P_{N-1}) = E(P_1, P_2) + \dots + E(P_{N-2}, P_{N-1})$$



Programmation Dynamique (2)

- On ne considère que l'énergie d'étirement

$$E(P_i, P_{i+1}) = \mathbf{a} \|P_{i+1} - P_i\|^2 + \mathbf{b} E_{\text{ext}}(P_{i+1})$$

- On initialise la position du contour
- On minimise successivement les variables

$$s_0(P_1) = \min_{P_0} E(P_0, P_1)$$

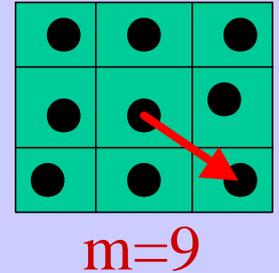
$$s_k(P_{k+1}) = \min_{P_k} (s_{k-1}(P_k) + E(P_k, P_{k+1}))$$


$$\min_{P_0 \dots P_{N-1}} E(P_0, \dots, P_{N-1}) = \min_{P_{N-1}} s_{N-1}(P_0)$$



Programmation Dynamique (3)

- On note d_i , $0 \leq i < m$ chacun des déplacements possibles



- Alors, il faut évaluer m valeurs pour $s_k(P_{k+1})$
- $$s_k(P_{k+1}) = \min_{i=0..m-1} (s_{k-1}(P_k + d_i) + E(P_k + d_i, P_{k+1}))$$
- Conduit à un algorithme de complexité $O(Nm^2)$
 - Utilisation d'une matrice des s_k et d'une matrice de position



Programmation Dynamique (4)

Calcul de $s_k(P_{k+1})$ avec $m=6$

Tableau de

$(P_{k+1} + d_i)$



Etape k+1

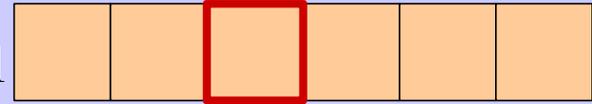


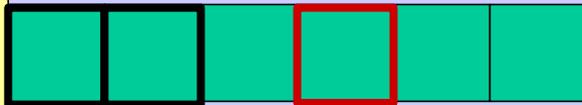
Tableau de

P_k^+

$$s_k(P_{k+1} + d_i) = \min_{j=0..m-1} (s_{k-1}(P_k + d_j) + E(P_k + d_j, P_{k+1} + d_i)) \quad P_k^+ = P_k + d_{j_{\min}}$$

Tableau de

$(P_k + d_j)$



Etape k

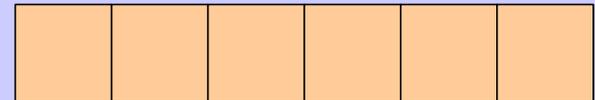


Tableau de

P_{k-1}



Programmation Dynamique (4)

- **Algorithme**

- Calcul de $S_0 = \min_{0 \leq i < m} E_{ext}(P_0 + d_i)$
- Initialisation de $S_0(i) = S_0$ et de $P_0^+(i) = P_0$
- $k=0$
- **Do**
 - $k \leftarrow k+1$
 - **For (i=0; i<m; ++i)**
 - $s_{\min} \leftarrow +\infty, j_{\min} \leftarrow 0$
 - **For (j=0; j<m; ++j)**
 - » Calcul de $s_k^j = S_{k-1}(j) + E(P_k + d_j, P_{k+1} + d_i)$
 - » Si $s_k^j < s_{\min}$ alors $s_{\min} = s_k^j$ et $j_{\min} = j$
 - Stockage de $S_k(i) = s_k(P_{k+1} + d_i) = s_{\min}$ $P_k^+(i) = P_k + d_{j_{\min}}$
 - **end**
- **Until (k=N)**



Amélioration de l'algorithme

- Schéma numérique :
 - schéma explicite
- Méthode d'optimisation
 - Algorithme glouton
 - Programmation dynamique
- • Calcul de la force externe
 - Gradient Vector Flow
- Evolution de la forme
 - Force interne tangentielle et normale
- Changement de topologie

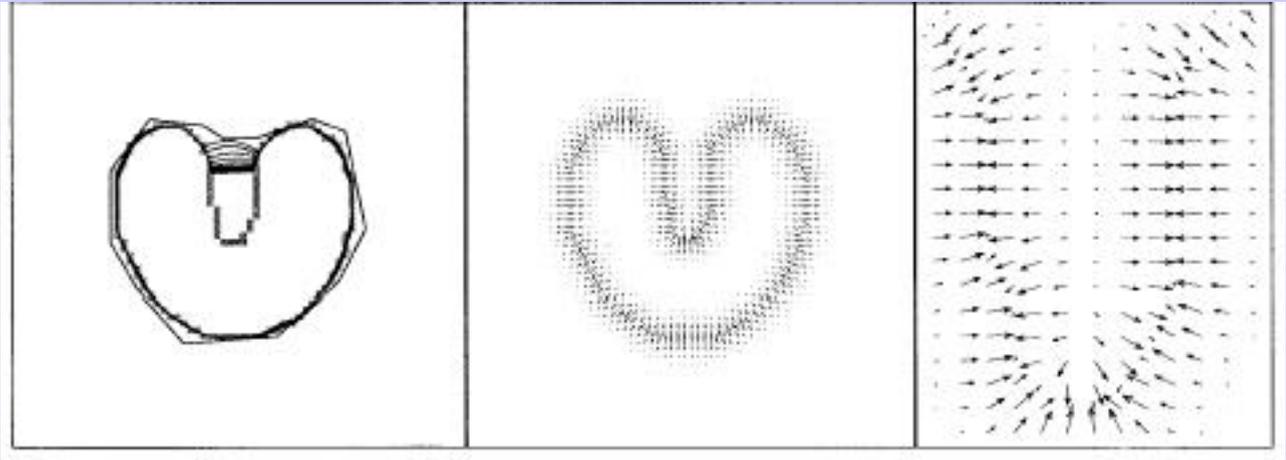


Gradient Vector Flow

- Problème dans le choix de l'énergie externe
 - Difficulté de pénétrer dans les concavités

Avec

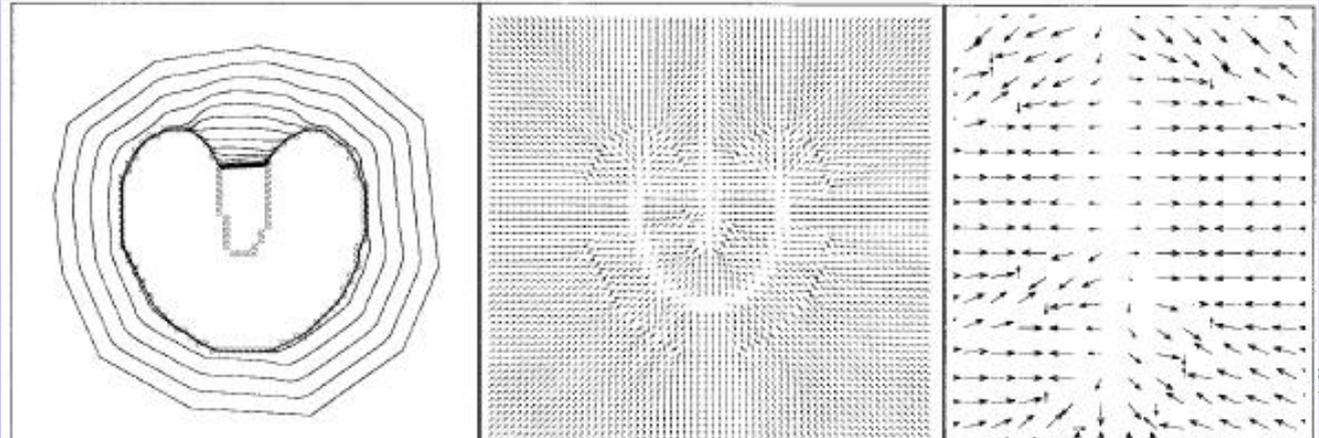
$$E_{ext} = -\|\nabla I(x, y)\|^2$$



Avec

$$E_{ext} = E_{Global}$$

Carte de
distance



Gradient Vector Flow

- Xu et Prince proposent de remplacer le champ de potentiel $-\nabla E_{\text{ext}}$ par un autre champ qui permet d'attirer les contours dans les concavités
- On choisit un champ de potentiel
 - par exemple $E_{\text{ext}} = -\|\nabla I\|^2$
 - On remplace $F_{\text{ext}} = -\nabla E_{\text{ext}}$ par un nouveau champ:

$$F_{\text{ext}} = GVF(\nabla E_{\text{ext}}) = \mathbf{v}(x, y) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix}$$



Gradient Vector Flow

- On note $-\nabla E_{\text{ext}} = \nabla f = \begin{bmatrix} f_x(x, y) \\ f_y(x, y) \end{bmatrix}$ le champ initial

- On détermine le nouveau champ $\mathbf{v}(x, y)$ en minimisant la fonctionnelle

$$\mathbf{e} = \iint \left(m \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right) + \|\nabla f\|^2 \|\mathbf{v} - \nabla f\|^2 \right) dx dy$$

- Quand $\|\nabla f\|$ est faible alors le champ est lisse

$$\Delta \mathbf{v} \approx 0$$

- Quand $\|\nabla f\|$ est fort alors $\mathbf{v}(x, y)$ coïncide avec $\|\nabla f\|$



Gradient Vector Flow

- Mise en œuvre:
 - On écrit le problème sous la forme d'une équation d'évolution après avoir effectué le calcul des variations

$$\frac{\partial u}{\partial t}(x, y, t) = \mathbf{m}\Delta u - (u - f_x)\|\nabla f\|^2$$

$$\frac{\partial v}{\partial t}(x, y, t) = \mathbf{m}\Delta v - (v - f_y)\|\nabla f\|^2$$

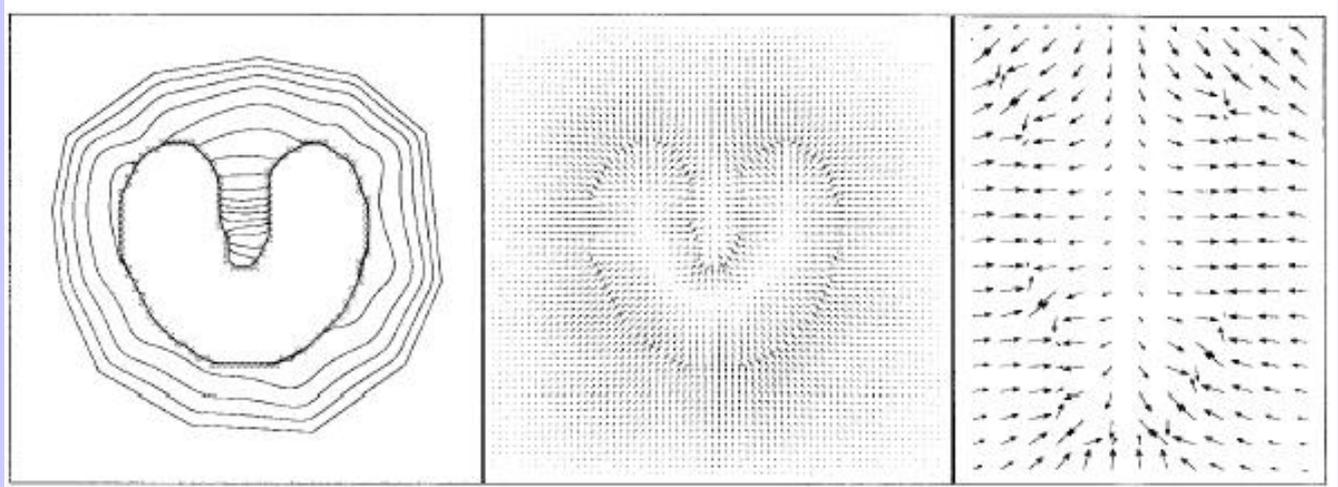
- On utilise les différences finies centrées pour la discrétisation
- Possibilité d'utiliser des approches multi-grilles pour accélérer le calcul



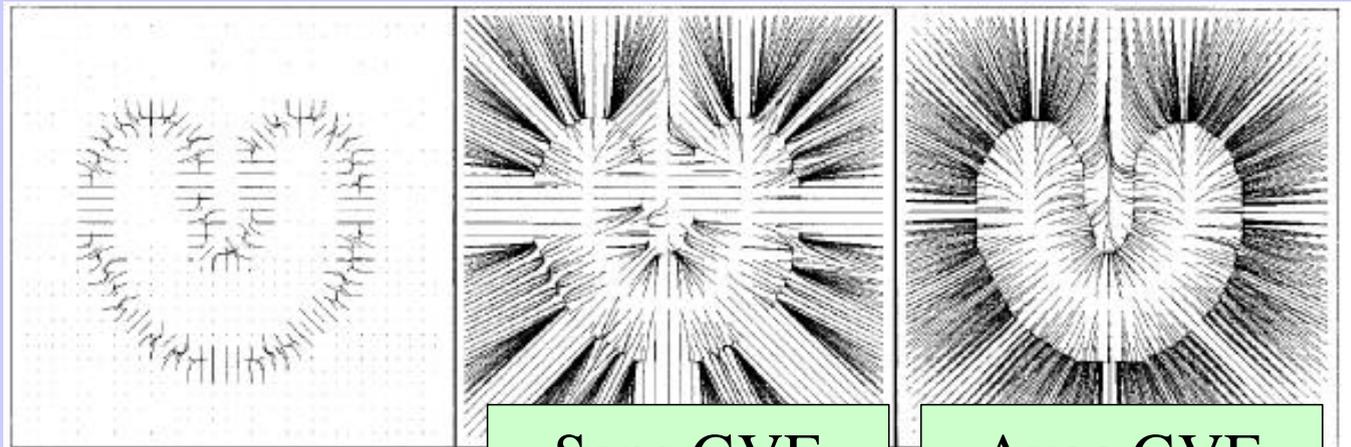
Gradient Vector Flow

- Résultat

Evolution du contour avec GVF



Lignes de champ



Sans GVF

Avec GVF

Gradient Vector Flow

- Comparaision

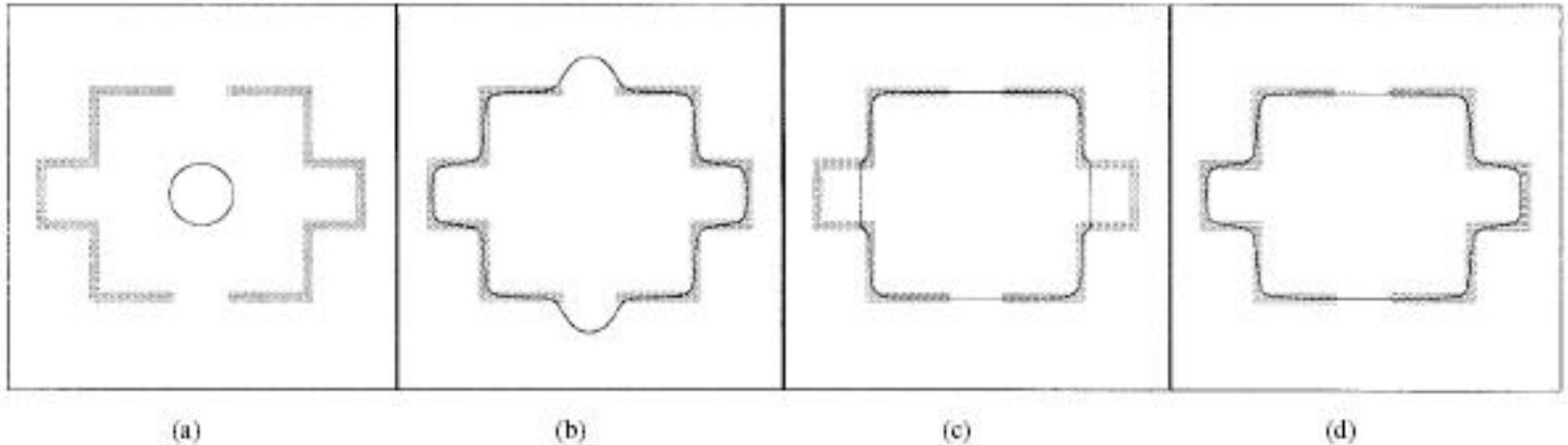


Fig. 5. (a) Initial curve and snake results from (b) a balloon with an outward pressure, (c) a distance potential force snake, and (d) a GVF snake.

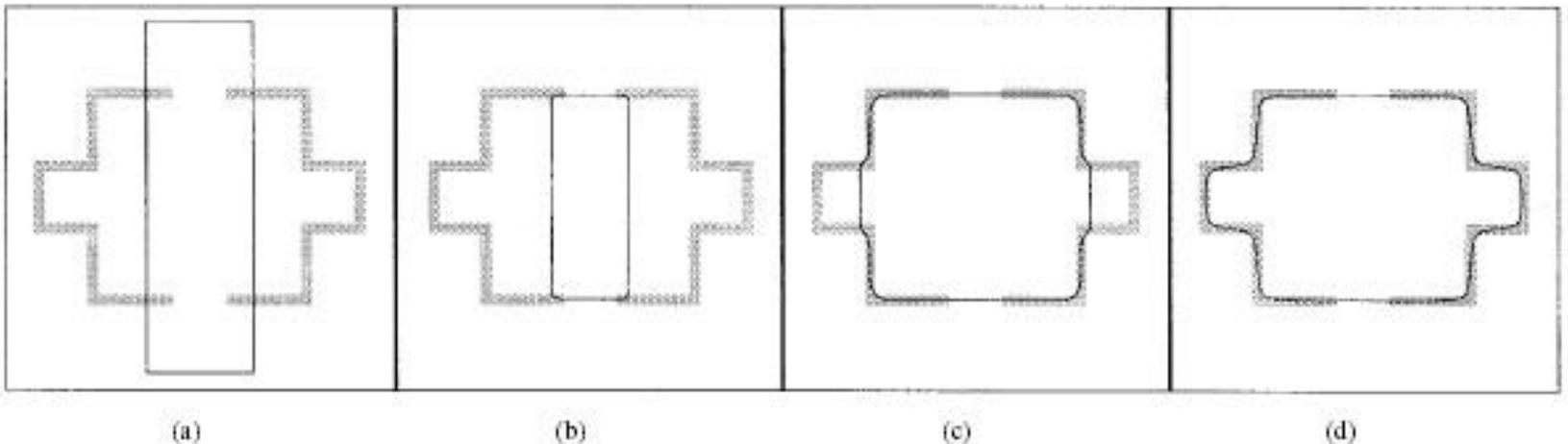


Fig. 6. (a) Initial curve and snake results from (b) a traditional snake, (c) a distance potential force snake, and (d) a GVF snake.

Amélioration de l'algorithme

- Schéma numérique :
 - schéma explicite
- Méthode d'optimisation
 - Algorithme glouton
 - Programmation dynamique
- Calcul de la force externe
 - Gradient Vector Flow
- • Evolution de la forme
 - Force interne tangentielle et normale
- Changement de topologie



Analyse de l'évolution de la forme

- On décompose la force interne suivant une composante normale et tangentielle

$$F_{\text{int}} = \mathbf{a}(u, t) \underbrace{\mathbf{t}(u)}_{\text{tangentielle}} + \mathbf{b}(u, t) \underbrace{\mathbf{n}(u)}_{\text{normale}}$$

- Influence des 2 termes ?

- On étudie $\frac{\partial \mathbf{C}(u, t)}{\partial t} = \mathbf{F}_{\text{int}}(\mathbf{C})$



Evolution de la géométrie

- On étudie l'évolution de t, n, g, k, s au cours de l'évolution de la courbe
- On initialise le contour : $C(u, 0) = C_o(u)$
- On utilise la grandeur $g(u, t)$ = métrique de $C(u, t)$

$$g(u, t) = \left\| \frac{dC(u, t)}{du} \right\| = \sqrt{\left(\frac{dx(u, t)}{du} \right)^2 + \left(\frac{dy(u, t)}{du} \right)^2} = \frac{ds}{du}$$



Rappels de géométrie

- On rappelle que : $s(u_o) = \int_a^{u_o} \frac{ds}{du} du = \int_a^{u_o} g(u) du$

- Relations de Serret-Frenet

$$\frac{d\mathbf{t}}{ds} = k \mathbf{n} \quad \frac{d\mathbf{n}}{ds} = -k \mathbf{t}$$

- k est la courbure en un point :

$$k(u) = \frac{d\mathbf{q}}{ds} = \frac{\frac{d\mathbf{C}}{du} \cdot \frac{d^2\mathbf{C}}{du^2}}{\left\| \frac{d\mathbf{C}}{du} \right\|^{3/2}}$$



Evolution de la géométrie

- Remarque : les 2 paramètres u et t sont indépendants donc on peut commuter leurs dérivées.
- On commence par l'évolution de $g(u,t)$

$$(g(u,t))^2 = \left\| \frac{dC}{du} \right\|^2 = \left(\frac{dx(u,t)}{du} \right)^2 + \left(\frac{dy(u,t)}{du} \right)^2$$

$$\frac{\partial g^2}{\partial t} = 2 \frac{dC}{du} \cdot \frac{d}{du} \frac{\partial C}{\partial t} \quad \longrightarrow \quad \frac{\partial g^2}{\partial t} = 2g \left(\frac{\partial \mathbf{a}}{\partial u} - \mathbf{b}kg \right)$$

$$\frac{\partial g}{\partial t} = \frac{\partial \mathbf{a}}{\partial u} - \mathbf{b}kg$$



Dérivées temporelles de s

- Pour calculer l'évolution de t,n,s, on évalue :

$$\frac{\partial}{\partial t} \frac{\partial}{\partial s} = \frac{\partial}{\partial t} \left[\frac{1}{g} \frac{\partial}{\partial u} \right] = -\frac{1}{g^2} \frac{\partial g}{\partial t} \frac{\partial}{\partial u} + \frac{1}{g} \frac{\partial}{\partial t} \frac{\partial}{\partial u}$$

$$\frac{\partial}{\partial t} \frac{\partial}{\partial s} = -\frac{1}{g} \left[\frac{\partial \mathbf{a}}{\partial u} - \mathbf{bkg} \right] \frac{\partial}{\partial s} + \frac{\partial}{\partial s} \frac{\partial}{\partial t}$$



Evolution de \mathbf{t} , \mathbf{n} et k

- On utilise le fait que $\mathbf{t} = \frac{d\mathbf{C}(s)}{ds}$

$$\frac{\partial \mathbf{t}}{\partial t} = \frac{1}{g} \left[\frac{\partial \mathbf{b}}{\partial u} + \mathbf{a}kg \right] \mathbf{n} = \frac{\partial \mathbf{q}}{\partial t} \mathbf{n}$$

$$\frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{g} \left[\frac{\partial \mathbf{b}}{\partial u} + \mathbf{a}kg \right] \mathbf{t} = -\left[\frac{\partial \mathbf{b}}{\partial s} + \mathbf{a}k \right] \mathbf{t}$$

$$\frac{\partial k}{\partial t} = \frac{\partial^2 \mathbf{b}}{\partial s^2} + \mathbf{a} \frac{\partial k}{\partial s} + \mathbf{b}k^2$$



Changement de paramètres

- Existe-t'il une autre paramétrisation de la courbe u^* et une fonction $\beta^*(u^*,t)$ telle que la courbe évolue de la même manière que sous l'effet de $\alpha(u,t)$ et de $\beta(u,t)$

$$\frac{\partial C(u^*,t)}{\partial t} = \mathbf{b}^*(u^*,t) \mathbf{n}(u^*) \quad \frac{\partial C(u,t)}{\partial t} = \mathbf{a}(u,t) \mathbf{t}(u) + \mathbf{b}(u,t) \mathbf{n}(u)$$

- Il faut que
$$-\frac{1}{g} \left[\frac{\partial \mathbf{b}}{\partial u} - \mathbf{a}k g \right] = -\frac{1}{g^*} \left[\frac{\partial \mathbf{b}^*}{\partial u^*} \right]$$

$$-\left[\frac{\partial \mathbf{a}}{\partial s} - \mathbf{b}k \right] = -\mathbf{b}^* k$$



Influence de la composante tangentielle

- Résultat fondamental :
 - Etant donnée une évolution guidée par (α, β)
 - On peut trouver un fonction $\beta^*(u, t)$ et une paramétrisation $g^*(u, t)$ telle que l'évolution de t et n , guidée par $(0, \beta^*)$ est identique à celle guidée par (α, β)
- Interprétation :
 - uniquement le terme normal modifie la forme du contour. Le terme tangentiel contrôle l'évolution de la paramétrisation.



Simplification

- Il suffit de s'intéresser à la composante normale : $\alpha=0$

$$\frac{\partial \mathbf{t}}{\partial t} = \frac{1}{g} \left[\frac{\partial \mathbf{b}}{\partial u} \right] \mathbf{n} \quad \frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{g} \left[\frac{\partial \mathbf{b}}{\partial u} \right] \mathbf{t}$$

$$\frac{\partial k}{\partial t} = \frac{\partial^2 \mathbf{b}}{\partial s^2} + \mathbf{b}k^2 \quad \frac{\partial L(t)}{\partial t} = -\int_0^{L(t)} \mathbf{b}k \, ds$$



Evolution de la longueur

- Par définition, on a : $L(t) = \int_a^b g(u, t) du$
- D'où :

$$\frac{\partial L(t)}{\partial t} = \int_a^b \frac{\partial g(u, t)}{\partial t} du = \int_a^b \left[\frac{\partial \mathbf{a}}{\partial u} - \mathbf{b}k g \right] du$$

$$\frac{\partial L(t)}{\partial t} = \int_0^{L(t)} \left[\frac{1}{g^2} \frac{\partial \mathbf{a}}{\partial s} - \mathbf{b}k \right] ds$$



Evolution de l'aire sous-jacente

- Par définition, on a : $A(t) = \int_a^b (\mathbf{C} \cdot \mathbf{n}) du$

- D'où :

$$\frac{\partial A(t)}{\partial t} = \int_a^b \frac{\partial C(u,t)}{\partial t} \cdot n g du + \int_a^b C \cdot \frac{\partial n}{\partial t} g du$$

$$\frac{\partial A(t)}{\partial t} = \int_a^b \mathbf{b} g du$$



Evolution par la courbure

- L'évolution par la courbure a des propriétés remarquables :

$$\frac{\partial \mathbf{C}(u, t)}{\partial t} = k(u) \mathbf{n}(u)$$

- L'évolution par la courbure correspond à la minimisation de la longueur de la courbe :

$$dL = k(u) \mathbf{n}(u)$$



Evolution par la courbure (2)

- Avec $\beta=k$, on a :

$$\frac{\partial \mathbf{t}}{\partial t} = \frac{\partial k}{\partial s} \mathbf{n} \quad \frac{\partial \mathbf{n}}{\partial t} = -\frac{\partial k}{\partial s} \mathbf{t} \quad \frac{\partial k}{\partial t} = \frac{\partial^2 k}{\partial s^2} + k^3 \quad \frac{\partial L(t)}{\partial t} = -\int_0^{L(t)} k^2 ds$$

- En particulier, si le contour initial est un cercle de courbure k_0 , alors ce cercle évolue en un cercle de plus en plus petit de courbure :

$$k(t) = \frac{k_0}{\sqrt{1 - 2k_0^2 t}}$$



Evolution par la courbure (3)

- Autres Résultats remarquables (Gage/Hamilton) :
 - Toute courbe fermée converge vers un point (son barycentre) en prenant asymptotiquement la forme d'un cercle.
 - Les courbes successives issues de l'évolution par la courbure d'une courbe fermée convexe ne s'intersectent jamais



Autres évolutions remarquables

- Propagation $\beta=1$ (force de ballon) :
 - Minimisation d'une fonctionnelle

$$E(C) = \int_a^b x dy - y dx = \iint_{\Omega} dA$$

- Conservation du vecteur tangent au cours de l'évolution
- Création de chocs lors de l'évolution



Autre évolutions remarquables

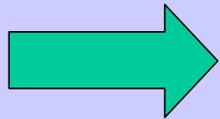
- $\mathbf{b} = k^3(u) + 2 \frac{d^2 k}{ds^2}$
- Minimisation d'une fonctionnelle $E(C) = \int_a^b k^2 ds$
- $\mathbf{b} = k - \frac{\mathbf{p}(\mathbf{C} \cdot \mathbf{n})}{A_0}$
- Préserve l'aire interne au cours de l'évolution (Gage)
- $\mathbf{b} = \frac{1}{L^2} \left(1 - \frac{2Ak}{L} \right)$
- Minimisation d'une fonctionnelle $E(C) = \frac{4A}{L^2}$
- Invariant par similarité



Cas de la force d'étirement

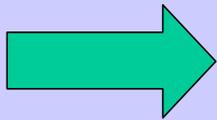
$$F_{\text{int}} = \frac{d^2 C(u)}{du^2} = \frac{d[gt]}{du} = \frac{dg}{du} \mathbf{t} + g^2 k \mathbf{n}$$

- Terme Tangentiel : $\frac{\partial g}{\partial t} = \frac{\partial^2 g}{\partial u^2} - k^2 g^3$



g a tendance à diffuser le long du contour
les points tendent à être équidistants

- Terme Normal : évolution par la courbure

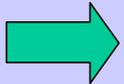


Convergence vers un point
Equivalent à minimiser la longueur de la courbe



Amélioration de l'algorithme

- Schéma numérique :
 - schéma explicite
- Méthode d'optimisation
 - Algorithme glouton
 - Programmation dynamique
- Calcul de la force externe
 - Gradient Vector Flow
- Evolution de la forme
 - Force interne tangentielle et normale
- Changement de topologie



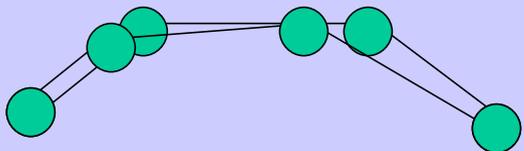
Active Contours

- Dynamic equation of active contours

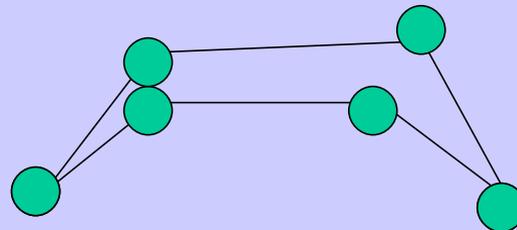
$$\frac{\partial^2 C}{\partial t^2} = -\mathbf{g} \frac{\partial C}{\partial t} + f_{\text{internal}}(C) + f_{\text{external}}(C)$$

$$f_{\text{tangent}}(C) + f_{\text{normal}}(C)$$

Vertex Spacing
(Parametrization)



Shape
Regularization

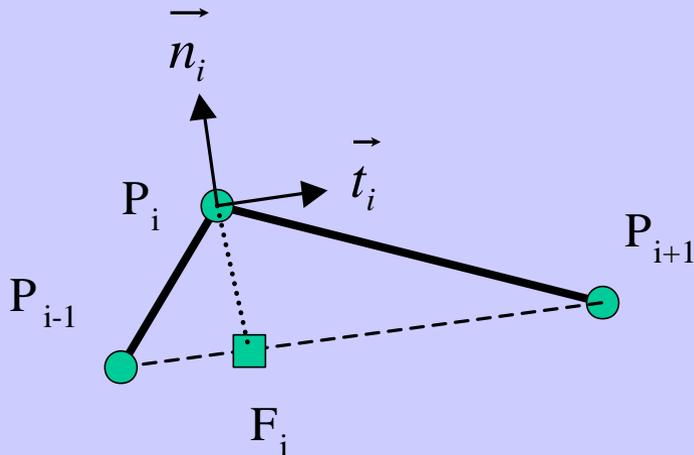


Discretization

- Replace classical finite differences approach with a discretisation based on geometry

$$\cancel{(x_i, y_i)} \quad \longrightarrow \quad (\mathbf{e}_i^1, \mathbf{f}_i)$$

- Contour discrete geometry :



Tangent

Normal

$$\vec{t}_i = \frac{P_{i-1}P_{i+1}}{\|P_{i-1}P_{i+1}\|}$$

$$\vec{n}_i = \vec{t}_i^\perp$$

Metrics Parameters

$$F_i = \mathbf{e}_i^1 P_{i-1} + \mathbf{e}_i^2 P_{i+1}$$

$$\mathbf{e}_i^1 + \mathbf{e}_i^2 = 1$$



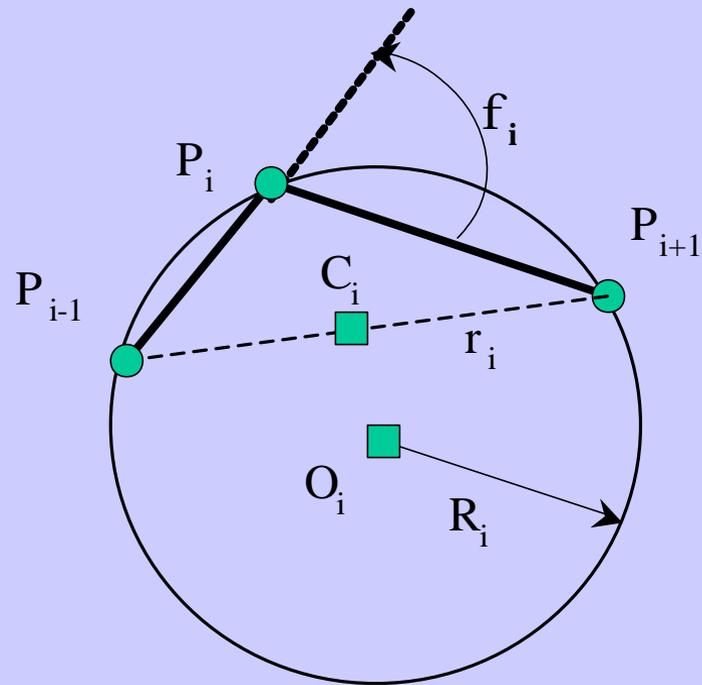
Discretization (2)

Definition of angle

$$\mathbf{f}_i$$

Definition of discrete curvature

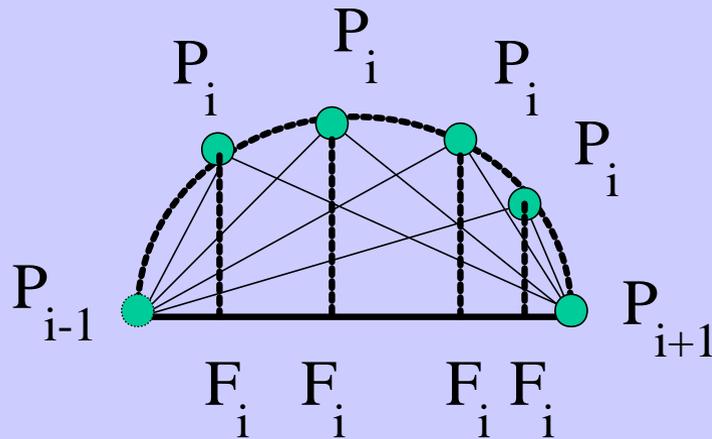
$$k_i = \frac{1}{R_i} = \frac{\sin \mathbf{f}_i}{r_i}$$



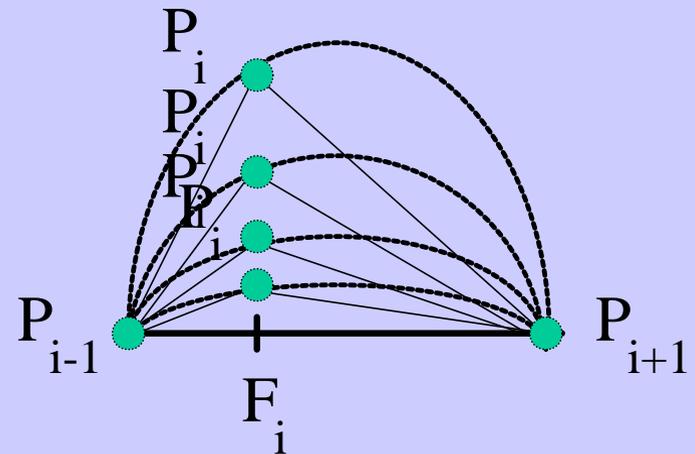
Discretization (3)

- Fundamental relation

$$P_i = \mathbf{e}_i^1 P_{i-1} + \mathbf{e}_i^2 P_{i+1} + L(r_i, \mathbf{e}_i^1, \mathbf{f}_i) \vec{n}_i$$



Modifying \mathbf{e}_i^1



Modifying \mathbf{f}_i



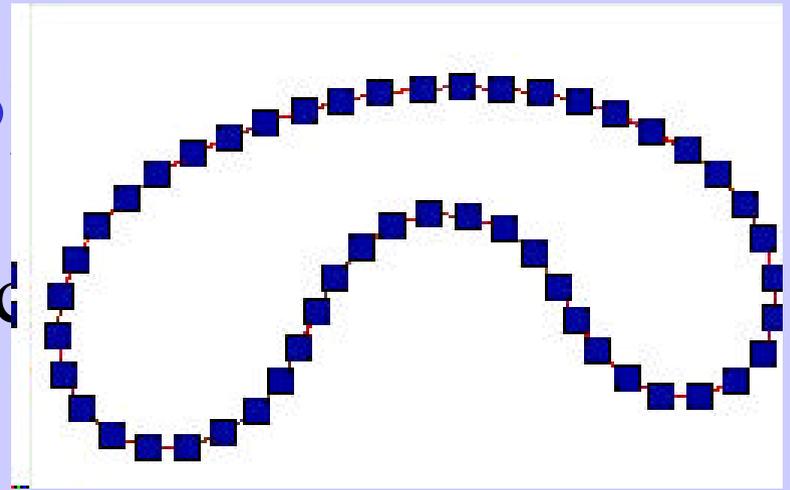
Objectives

- Improve 3 key aspects of active contours :
 - **Parametrization**
 - Vertex Spacing
 - Refinement-Decimation
 - **Shape regularization**
 - **Topological changes**



Vertex Spacing

- Component $f_{\text{tangent}}(C)$ of



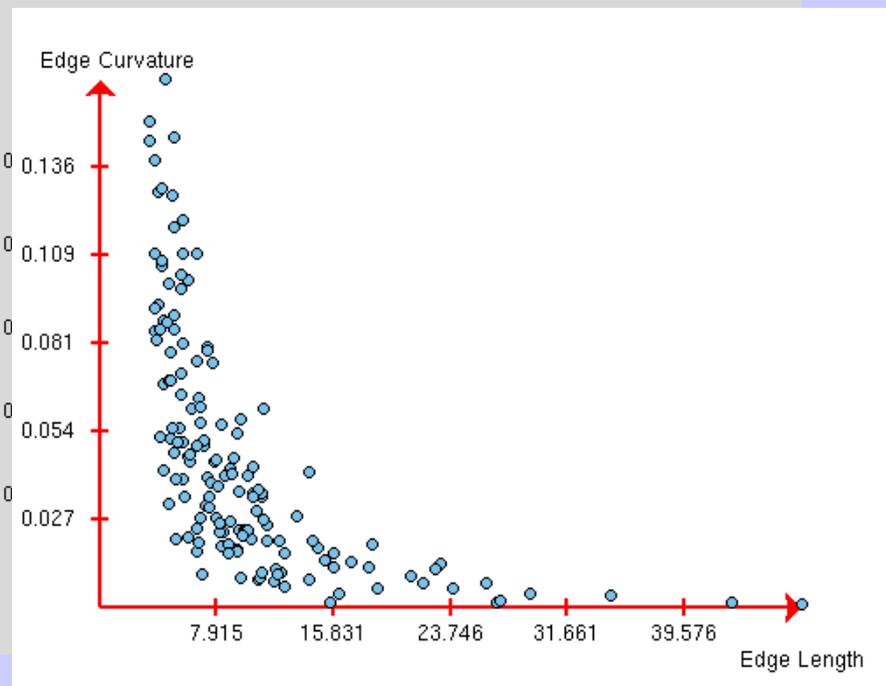
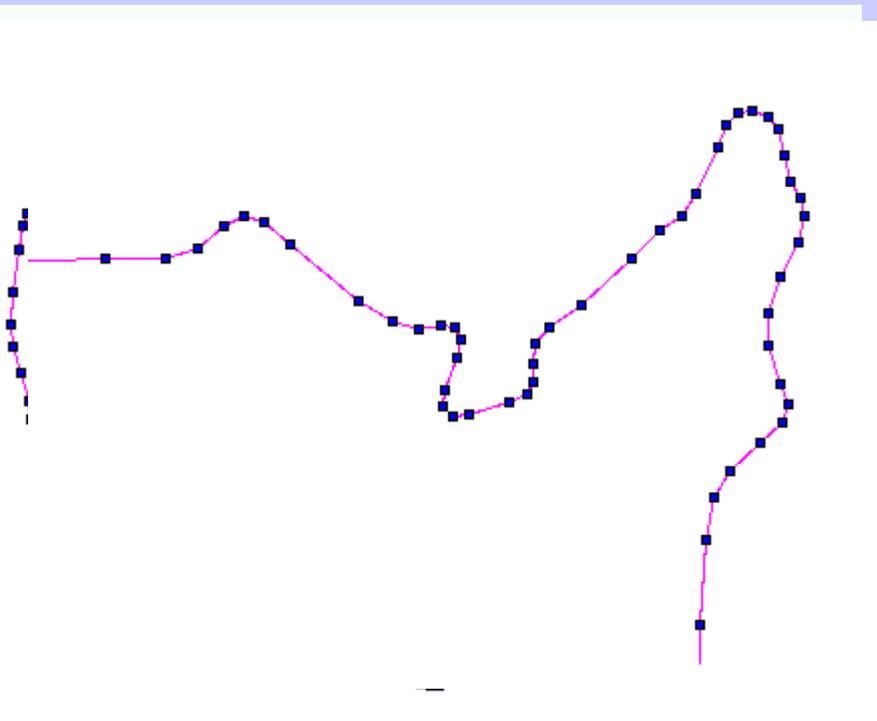
$$\overrightarrow{f_{\text{tangent}}}(P_i) = (\mathbf{e}_i^* - \mathbf{e}_i)(P_{i+1} - P_{i-1})$$

- Two different choice of target metric parameter:

- uniform vertex spacing $\mathbf{e}_i^* = \frac{1}{2}$
- curvature-based vertex spacing $\mathbf{e}_i^* = \frac{1}{2} - 0.4 \Delta K_i$

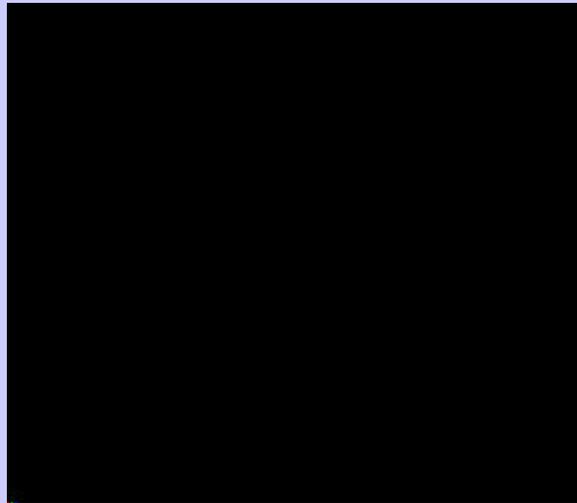


Vertex Spacing (2)



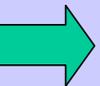
Refinement-Decimation

- Dynamic update of contour resolution
 - Add vertices if an edge is too elongated
 - Remove vertices if an edge is too short



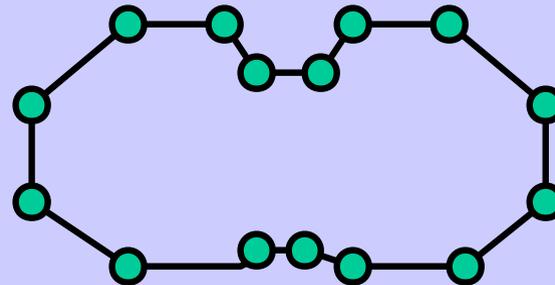
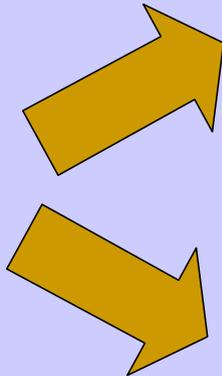
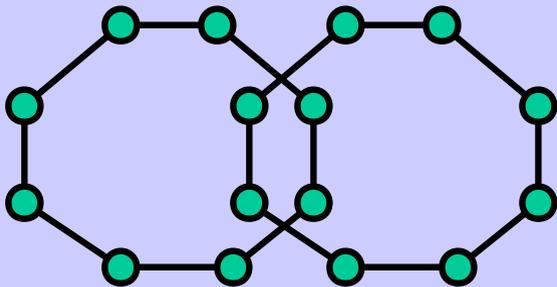
Amélioration de l'algorithme

- Schéma numérique :
 - schéma explicite
- Méthode d'optimisation
 - Algorithme glouton
 - Programmation dynamique
- Calcul de la force externe
 - Gradient Vector Flow
- Evolution de la forme
 - Force interne tangentielle et normale
- Changement de topologie

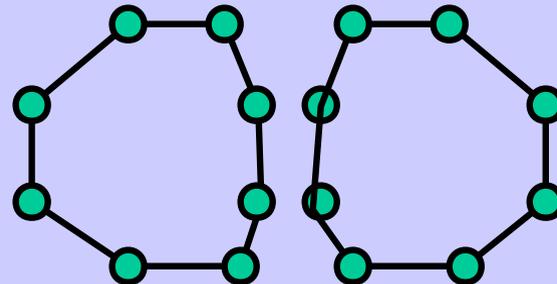


Topology Changes

- Objective : Process intersecting contours



- Fuse Contours



- Push Contours



Topology Changes

- Constraints
 - Efficiency



Use a regular grid for edge intersection tests

- Independence from vertex spacing

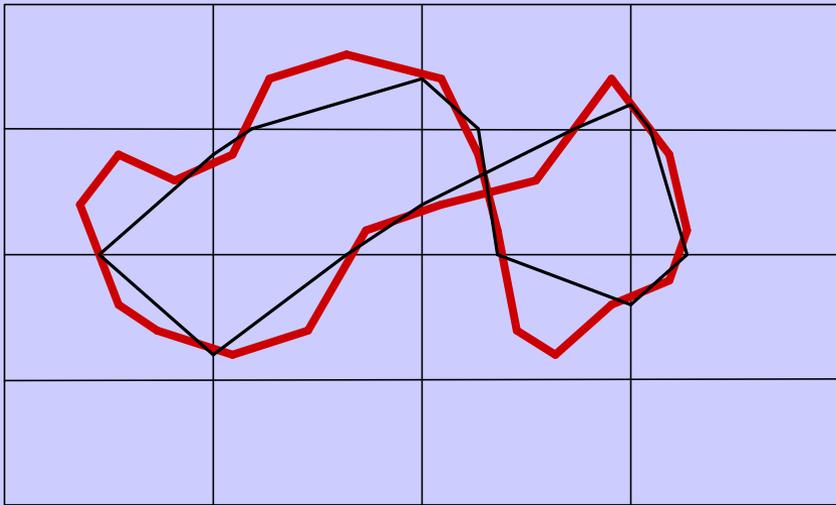


Use topological operators for fusing contours

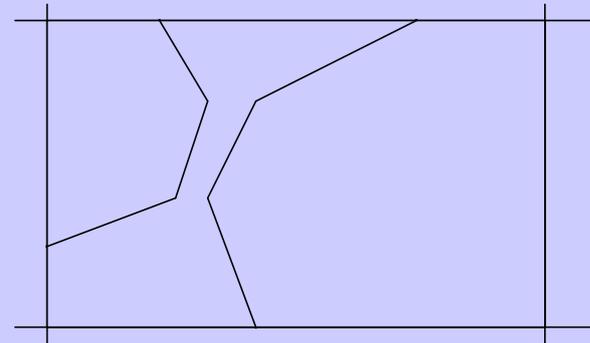


Idea 1

- Use regular grid : for detecting intersecting edges



Grid approximation

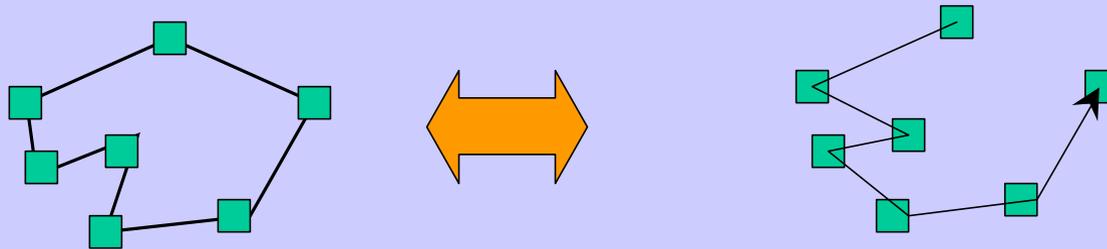


Cell by cell detection

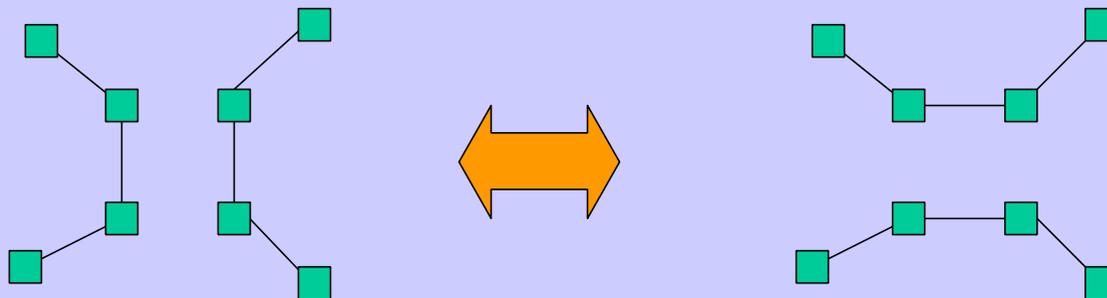


Idea 2

- Use topological operators



Opening-Closing Contours



Merging Contour Edges



Implementation details

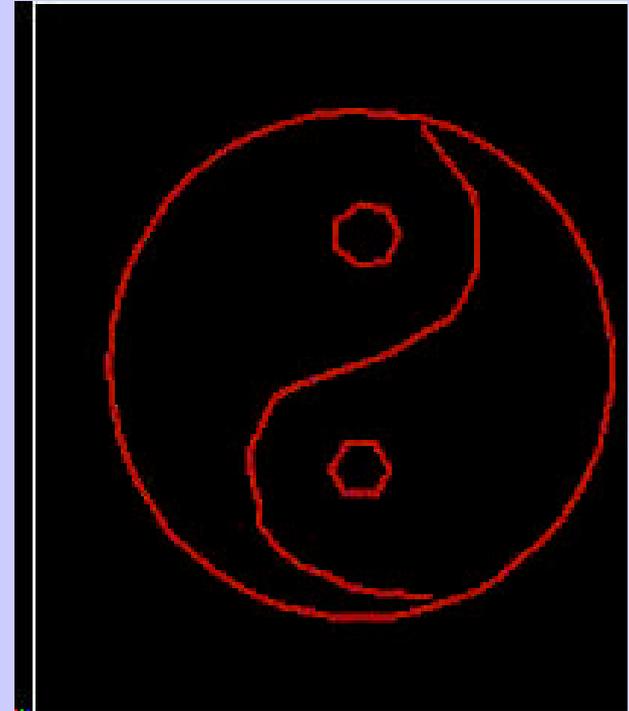
- Origin of the grid is chosen randomly
- Grid size may be chosen fairly large
- Applied every 5-10 iterations
- Process edge intersection by pairs



Examples



Two components
splitting



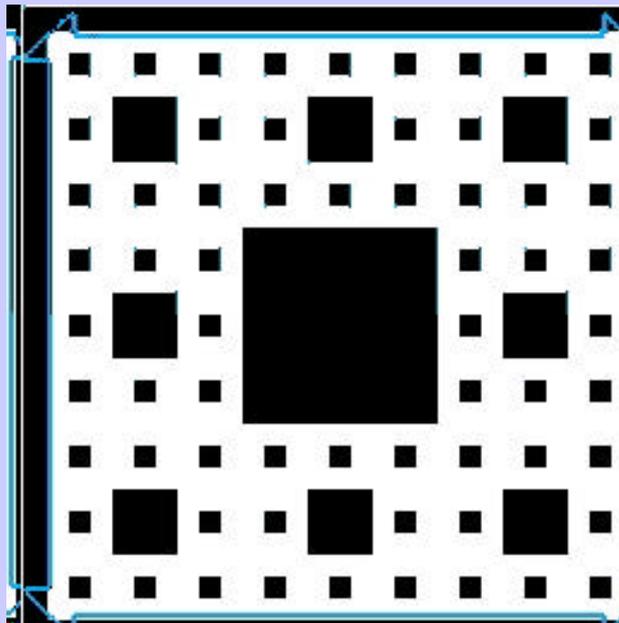
Opened and closed
contours



Comparison with level-sets

Level sets

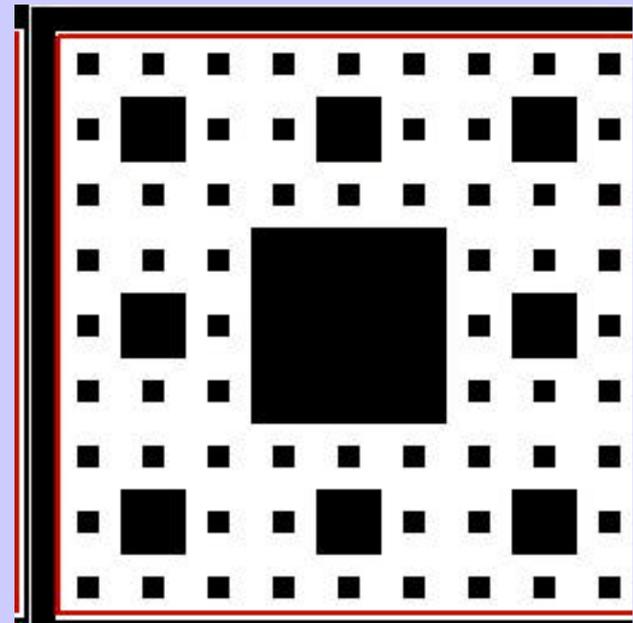
$$u(\mathbf{p}) = b(\mathbf{p})(k(\mathbf{p}) + c)$$



Real time: 3,3 s

Discrete contour

$$f_{\text{int}}(\mathbf{p}) \quad f_{\text{ext}}(\mathbf{p}) = b(\mathbf{p})c\mathbf{n}$$



Real time: 0,42 s

