

Algorithme de transformation en distance généralisée en un balayage, mises en œuvre séquentielle et vectorielle

Nicolas NORMAND¹, Olivier DÉFORGES², Patrick LE CALLET¹, Aurélien CHIRON¹

¹Laboratoire IRCCyN CNRS UMR 6795, équipe Image et Vidéocommunications, groupe Mojette
La Chantrerie, rue Christian Pauc, BP 50609, 44306 Nantes CEDEX 3

²Laboratoire CNRS UMR 6164 IETR/INSA de Rennes
20, av. des Buttes de Coëmes, CS 14315, 35043 Rennes CEDEX
Nicolas.Normand@polytech.univ-nantes.fr
Olivier.Deforges@insa-rennes.fr
Patrick.LeCallet@polytech.univ-nantes.fr

Résumé – Dans ce papier, un algorithme de transformation en distance généralisée est présenté et utilisé pour la mise en œuvre de l’opérateur d’érosion morphologique dans des images binaires. La généralité de la transformation permet d’utiliser en tant que famille de pseudo-disques toute suite d’éléments structurants obtenue par des dilations ce qui réunit les méthodes basées sur la décomposition d’éléments structurants par dilations et celles basées sur les transformations de distance en profitant de la variété de formes des premières et de la faible complexité (indépendante de la taille des éléments) des secondes. La décomposition en éléments convexes de Xu [Xu91], initialement élaborée pour une architecture parallèle, est prise en exemple et utilisée pour construire la famille de disques de la transformation en distance. Finalement, cette transformation est présentée sous les formes séquentielle et vectorielle.

Abstract – We introduce a generalized distance transform algorithm and use it to perform morphological erosion in binary images. This generic transform can use any dilation-based series of structuring elements as a set of generalized distance disks. Hence dilation-based decompositions and distance transform-based erosion methods are unified and we gain both the shape diversity of the first ones and the low computational complexity (independent from the structuring element size) of the second ones. Xu convex structuring element decomposition [Xu91], which was initially developed for parallel machines, is used as a base for generalized disks construction. Finally, a sequential and a vectoriel versions of the transformation are shown.

1 Introduction

La transformation en distance (qui consiste à affecter à chaque pixel objet de l’image sa distance au point du fond le plus proche) est souvent une étape préliminaire à d’autres traitements (squelettisation, érosion...). La complexité algorithmique de cette opération influe sur le coût global du traitement en particulier lorsqu’une telle opération est répétée (suite de traitements morphologiques). Dans cet article, un algorithme de transformation en distance généralisée (GDT) est présenté. Le mécanisme de construction de pseudo-disques qu’il utilise est suffisamment générale pour modéliser notamment les distances de chanfrein et les séries de formes obtenues par addition de Minkowski [Nor03]90.

Un exemple de mise en œuvre séquentielle sera donné à partir d’une décomposition d’éléments structurants [Xu91]. Après adaptation, une version vectorielle sera également présentée.

2 Méthodes existantes

Les mises en œuvre directes de l’érosion et de son opérateur dual, la dilatation, à partir de la transcription directe de leurs définitions (eq. 1) sont très inefficaces en termes de complexité de calcul notamment pour de grands éléments structurants car elles requièrent de multiples lectures des mêmes pixels dans

l’image.

$$\begin{aligned} A \ominus \check{B} &= \{p | (B)_p \subseteq A\} \\ A \oplus \check{B} &= (A^c \ominus \check{B})^c = \{p | (B)_p \cap A \neq \emptyset\} \\ A \oplus B &= \{a + b | a \in A, b \in B\} \text{ addition de Minkowski} \end{aligned} \quad (1)$$

Afin de minimiser les temps de calcul, de nombreuses méthodes ont été proposées. Certaines n’étant pas régulières, sont donc plus difficilement implantables efficacement et n’ont pas été retenues [Vin91]. Parmi les méthodes régulières, on distingue principalement deux classes de méthodes. Les premières procèdent par seuillage d’une carte de distance, les secondes utilisent le principe de décomposition d’éléments structurants par additions de Minkowski.

2.1 Transformation en distance

À partir d’une distance $d(x, y)$, on définit la transformation en distance d’un ensemble de points X (une image binaire) comme l’application qui associe à chaque point x , la distance au point hors de X le plus proche :

$$d_X(x) = \min_{y \notin X} (d(x, y)) \quad (2)$$

De manière équivalente, c’est le rayon du plus grand disque de centre x inscrit dans l’image :

$$d_X(x) = \operatorname{argmax}_r \{(D(r))_x \subseteq X\} \quad (3)$$

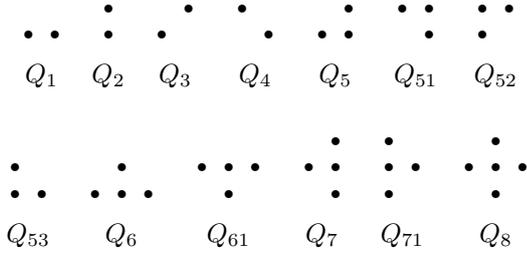


Figure 1: Formes élémentaires de Xu [Xu91]

où le disque ouvert de centre c et de rayon r est défini comme l'ensemble des points à une distance inférieure à r de c :

$$D(c, r) = (D(r))_c \text{ avec } D(r) = \{p | d(O, p) < r\}. \quad (4)$$

L'érosion par le disque de rayon r est alors obtenue par un simple seuillage de la carte de distance :

$$A \ominus \tilde{D}(r) = \{p | d_X(p) \geq r\}. \quad (5)$$

La complexité est donc directement liée à celle de la transformation en distance, or celle-ci ne dépend que du nombre de pixels et est donc indépendante de la taille de l'élément choisi. Les algorithmes classiques consistent à augmenter les distances obtenues pour les points voisins soit d'une valeur identique quel que soit le voisin [RP68], soit d'une valeur de distance locale attribuée en fonction du voisin (distances de chanfrein) [Bor86]. La propriété de symétrie des distances impose un minimum de deux balayages en sens contraires de l'image ce qui est incompatible avec une structure *pipeline* et impose la mémorisation d'un résultat intermédiaire de taille identique aux données. En relâchant cette contrainte (pseudo-distance) et en choisissant comme « centre » d'un disque le dernier point dans l'ordre de balayage, un seul balayage est alors nécessaire.

La limitation principale de ce genre de méthodes réside dans la faible variété des formes disponibles, très régulières même lorsque l'on fait alterner les voisinages [WB92].

2.2 Décomposition par addition de Minkowski

Pour plus de souplesse dans la géométrie des formes utilisables, on construit des séries d'éléments par suites d'additions de Minkowski. Éroder avec une forme obtenue par une série d'additions de Minkowski d'éléments structurants revient à effectuer une série d'érosions par les éléments.

$$((A \ominus \tilde{B}_1) \ominus \tilde{B}_2) \ominus \tilde{B}_3 = A \ominus (\tilde{B}_1 \oplus \tilde{B}_2 \oplus \tilde{B}_3). \quad (6)$$

Les méthodes de décomposition d'éléments structurants utilisant cette propriété sont généralement adaptées à des architectures spécifiques aptes à calculer une érosion simple en un cycle de calcul. Jianning Xu a, par exemple, décrit une méthode de décomposition adaptée à une architecture parallèle 8-connexe, capable d'éroder avec toute forme convexe inscrite dans un carré 3×3 en un cycle [Xu91]. Les 13 éléments qu'il a dénombrés ainsi qu'un exemple de décomposition d'élément structurant sont représentés sur les figures 1 et 2.

Cette méthode est revanche moins adaptée aux architectures séquentielles, le nombre d'itérations pour aboutir au résultat final étant égal au nombre d'éléments dans la décomposition. La complexité de l'érosion dépend dans ce cas de la taille de l'élément structurant.

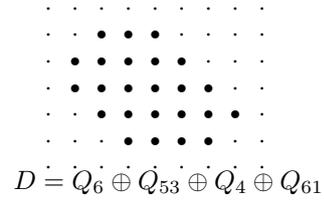


Figure 2: Décomposition d'un élément par la méthode de Xu

Dans la suite nous utiliserons une méthode unifiée, qui combine une transformation en distance pour la complexité et une décomposition par additions de Minkowski pour la définition des pseudo-disques.

3 Méthode proposée

3.1 Construction récursive des disques

Afin de construire récursivement une famille de disques de la manière la plus générale qui soit, chaque disque $D(i)$ est la réunion ensembliste de disques plus petits translétés. Les disques assemblés ne sont pas forcément identiques (comme c'est le cas pour une dilatation et les distances usuelles) et il n'ont pas nécessairement de relation fixe par type de voisinage (comme ce serait le cas pour les distances de chanfrein).

Les vecteurs de translation (voisinages) pris pour la construction sont dénotés p_k . La suite des disques $\{D(i)\}_{i \in [0, N]}$ est croissante ce qui implique que par construction chaque disque i est contenu dans le disque suivant $i + 1$. D'autre part, le disque $D(0)$ est l'ensemble vide et le disque $D(1)$ est le singleton composé de l'origine O .

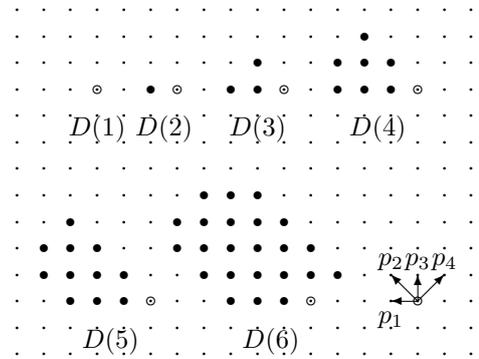


Figure 3: Construction des disques

Dans l'exemple de la figure 3, $D(3)$ est construit avec le translété de $D(2)$ par p_1 et de $D(1)$ par p_2 , il contient également $D(2)$ non translété pour garantir l'inclusion de $D(2)$ dans $D(3)$:

$$D(3) = D(2) \cup (D(2))_{p_1} \cup (D(1))_{p_2}.$$

D'une manière générale, la construction des disques s'écrit de la manière suivante :

$$\forall i = 2 \dots N \quad D(i) = \bigcup_{k \in [0, K]} (D(I_k(i)))_{p_k} \quad (7)$$

où $I_k(i) < i$, $I_0(i) = i - 1$ et $p_0 = O$.

La valeur de $I_k(i)$ représente l'indice du disque utilisé dans le voisinage p_k pour construire $D(i)$. Dans l'exemple précédent (figure 3), on a : $I_0(3) = 2, I_1(3) = 2$ et $I_2(3) = 1$.

Table 1: Tableau de construction des disques

i	2	3	4	5	6
$I_1(i)$	1	2	3	3	3
$I_2(i)$	0	1	3	4	5
$I_3(i)$	0	0	0	0	5
$I_4(i)$	0	0	0	0	5

Les valeurs de $I_k(i)$ sont consignées dans le tableau de construction (cf. tableau 1).

Cette construction est générale : elle permet de décrire toutes les familles croissantes de disques ou d'éléments structurants. Les disques de chanfrein sont obtenus en fixant la différence $i - I_k(i)$ par type de voisinage [Bor86].

3.2 Carte de distance généralisée

Il s'agit pour chaque pixel x de déterminer le rayon du plus grand disque de centre x inscrit dans l'image (eq. 3). Par définition (eq. 7), ce disque est la réunion de disques plus petits, ceux-ci doivent donc être inclus dans l'image :

$$d_X(x) = \operatorname{argmax}_r \{ \forall k, (D(I_k(r)))_{x+p_k} \subseteq X \}$$

Le problème est donc de déterminer le rayon du plus grand disque que l'on puisse construire compte tenu de ceux trouvés dans le voisinage. Si les voisins de x sont tous antérieurs dans l'ordre de balayage, on connaît lors du calcul de $d_X(x)$, les plus grands disques inscrits dans le voisinage de x . Il suffit donc de chercher dans le tableau de construction, le plus grand indice de disque i dont les composants sont inclus dans les disques voisins :

$$\begin{aligned} d_X(x) &= \operatorname{argmax}_i \{ \forall k, I_k(i) \leq d_X(x + p_k) \} \\ &= \min_k \{ \operatorname{argmax}_i \{ I_k(i) \leq d_X(x + p_k) \} \} \end{aligned} \quad (8)$$

Tableau de transformation en distance Pour éviter le recours à la recherche systématique du maximum pendant le calcul de la transformation en distance, on l'évalue une fois pour toutes pour chaque voisin p_k et chaque rayon de disque i . On pose :

$$M_k(j) = \operatorname{argmax}_i \{ I_k(i) \leq j \} \quad (9)$$

La transformation en distance s'écrit alors :

$$d_X(x) = \begin{cases} \min_k (M_k(d_X(x + p_k))) & \text{si } x \in X \\ 0 & \text{sinon} \end{cases} \quad (10)$$

La transformation en distance consiste donc à consulter les tableaux M_k indicés par les valeurs déjà calculées de la carte de distance puis à extraire la valeur minimale.

3.3 Mise en œuvre séquentielle

Le schéma de construction décrit précédemment peut utiliser toute famille de disques ou d'éléments structurants basée sur des incréments de distance locaux ou des additions de Minkowski. Dans l'exemple de la figure 3, nous avons utilisé les formes élémentaires de Xu (figure 1) pour obtenir $D(6)$ par une suite d'additions de Minkowski ($D(6) = Q_6 \oplus Q_{53} \oplus Q_4 \oplus Q_{61}$) ;

Table 2: Tableau de transformation en distance

i	0	1	2	3	4	5	6
$M_1(i)$	1	2	3	6	6	6	6
$M_2(i)$	2	3	3	4	5	6	6
$M_3(i)$	5	5	5	5	5	6	6
$M_4(i)$	5	5	5	5	5	6	6

la démarche est similaire avec d'autres schémas de décomposition ([JPT89], [Gon88]).

Parmi les éléments de Xu, certains ne contiennent que l'origine du plan O et un sous-ensemble des voisins p_k . Obtenir l'addition de Minkowski du disque $D(i)$ avec l'un de ces éléments, revient à porter dans la colonne $i+1$ du tableau de construction la valeur i pour chacun des p_k concernés. Dans notre exemple (figure 3), $D(4) = D(3) \oplus Q_{53}$ se traduit par $I_1(4) = 3$ et $I_2(4) = 3$ car $Q_{53} = \{O, p_1, p_2\}$. Pour les autres voisinages, on se contente de reporter la valeur précédente (nulle dans l'exemple) de façon à préserver la monotonie des tableaux I_k . Les autres éléments de Xu (Q_6, Q_7, Q_{71} et Q_8) nécessitent deux étapes mais le principe reste identique. $D(2) = D(1) \oplus \{O, p_1\}$ et $D(3) = D(2) \oplus \{O, p_1\} \cup D(1) \oplus \{O, p_2\}$ ce qui donne bien : $D(3) = D(1) \oplus \{O, p_1 + p_1, p_1, p_2\} = D(1) \oplus Q_6$.

La mise en œuvre séquentielle est ensuite directe à partir de eq. 10. La carte de distance est calculée en un balayage et permet d'obtenir l'érosion par un des disques $D(i)$ par simple seuillage. La complexité dépend uniquement du nombre de voisinages élémentaires p_k et non de la taille de l'élément structurant.

Un prototype a été réalisé sur une carte équipée d'un FPGA Xilinx Spartan IIE. Il prend en entrée un signal vidéo PAL (échantillonné à 720 pixels par ligne pour 312,5 lignes par trame et 50 trames par seconde). Ce signal est ensuite binarisé, érodé selon la méthode présentée puis restitué au format PAL. Le traitement consomme de la mémoire RAM embarquée utilisée en registre à décalage dont la contenance est égale au nombre de pixels par ligne plus un et qui permet de conserver une ligne de la carte de distance pour fournir les valeurs de $d_X(x + p_2)$, $d_X(x + p_3)$ et $d_X(x + p_4)$. La profondeur de la carte de distance est de 8 bits par pixels, ce qui est largement suffisant pour la plupart des applications et permet d'obtenir de grands éléments structurants. Les ressources monopolisées restent minimales et permettent d'envisager l'enchaînement de plusieurs érosions ou dilatations selon la méthode proposée dans un FPGA modeste en taille et en coût. De même, il est possible de cadencer le traitement à des fréquences supérieures à celle nécessaire pour le format PAL (environ 14 MHz). Sur la cible choisie, nous garantissons sans optimisation le bon fonctionnement pour des fréquences triples.

3.4 Mise en œuvre vectorielle

La mise en œuvre vectorielle est moins immédiate, car dans l'expression eq. 10, chaque pixel dépend du pixel immédiatement précédent, ce qui interdit la parallélisation. Afin de la rendre possible, nous avons remplacé l'utilisation du voisinage p_1 par la prise en compte des longueurs de plages de pixels à gauche (dont le calcul est très rapide). La dilatation d'un élément B par les formes de Xu qui ne contiennent pas p_1 est

évidemment assurée. En revanche, pour les formes incluant p_1 , soit Q_1 , Q_5 , Q_{53} et Q_6 , le résultat est incomplet tel quel et dépend de la géométrie de B . En effet, le dilaté de B par la forme élémentaire Q privée de p_1 ($Q^* = Q \setminus \{p_1\}$) est un sous-ensemble du dilaté de B par Q_1 . L'ensemble des points manquants est donné par : $(B \oplus Q) \setminus (B \oplus Q_1^*) = \{p | p \in B_{p_1} \setminus (B \oplus Q^*)\}$. Pour Q_1^* , chaque point de $B \oplus Q_1$ qui n'a pas de voisin à gauche (*i.e.* le premier pixel de chaque ligne) est manquant. Pour Q_5^* et Q_6^* il s'agit du premier pixel de chaque ligne sans voisin en bas à gauche et pour Q_{53}^* chaque pixel sans voisin en dessous ou à gauche engendrera un pixel manquant.

Le premier pixel de la dernière ligne, systématiquement absent, sera traité séparément. On vérifie que si B est constitué d'une seule ligne, aucun autre pixel ne manque pour Q_1^* , que si le bord gauche de B est orienté à 45° cela vaut aussi pour Q_5^* et Q_6^* de même pour Q_{53}^* si le bord gauche de B est vertical ou à 45° . La commutativité de l'addition de Minkowski nous permet sans perte de généralité d'utiliser Q_1^* en premier lieu, Q_5^* et Q_6^* ensuite, puis Q_{53}^* et finalement les autres formes élémentaires. Pour le pixel manquant de la dernière ligne, nous utilisons la longueur des plages de pixels objet antérieurs à partir du pixel courant :

$$R(x) = \begin{cases} R(x + p_1) + 1 & \text{si } x \in B \\ 0 & \text{sinon} \end{cases} \quad (11)$$

L'ordre des formes utilisées dans la figure 3 ($D(6) = Q_6 \oplus Q_{53} \oplus Q_4 \oplus Q_{61}$) respecte les contraintes imposées et, pour prendre en compte $R(x)$, il suffit de remplacer M_1 par M_R à la fois dans le tableau de transformation en distance (tableau 2) et dans l'équation de calcul de la carte de distance ($M_1(d_X(x + p_1))$ devient $M_R(R(x))$ dans eq. 10). M_R est calculé (eq. 9) à partir d'une ligne I_R du tableau de construction qui remplace I_1 et que l'on obtient de la manière suivante :

$$\begin{cases} I_R(1) = 1 \\ \forall i > 1, I_R(i) = I_R(i-1) + 1 \text{ si } I_1(i) = i-1 \\ = I_R(i-1) \text{ sinon} \end{cases}$$

L'algorithme a été mis en œuvre sur une architecture généraliste à base de PowerPC disposant du jeu d'instructions vectorielles AltiVec en langage C pour le chargement d'un vecteur, sa sauvegarde, la lecture dans une table de correspondance (*lookup-table*) et le calcul d'un minimum. Pour toutes ces opérations, les données sont organisées en vecteurs de 128 bits ce qui permet de traiter 16 pixels simultanément dans notre application.

Un exemple de carte de distance est donné dans la figure 4 pour la famille d'élément de la figure 3, les programmes séquentiel (avec tableau 2) et vectoriel (avec tableau 4) fournissent un résultat identique.

4 Conclusion

Dans cet article, un schéma de construction de pseudo-disques ou d'éléments structurants a été introduit. Une transformation en distance généralisée exploite la définition récursive des disques et permet un calcul en un balayage d'image lorsque les voisinages utilisés sont dans la partie antérieure. Deux versions de cet algorithme ont été présentées ; l'un, séquentiel, est la transposition directe de l'expression de transformation en distance ; le second, vectoriel, autorise un calcul parallèle ligne

Table 3: Utilisation des plages de pixels dans la construction des disques

i	1	2	3	4	5	6
$I_R(i)$	1	2	3	4	4	4

Table 4: Utilisation des plages de pixels dans la transformation en distance

i	0	1	2	3	4	5	6
$M_R(i)$	0	1	2	3	6	6	6

	1	2				1	
1	2	3	3		1	2	3
1	2	3	4	4	2	3	3
1	2	3	4	5	5	3	4
1	2	3	4	5	5	5	4
		1	2	3	6	5	5

Figure 4: Transformation en distance

par ligne moyennant certaines précautions dans la construction des disques sans restriction sur leur géométrie.

Références

- [Bor86] G. Borgefors. Distance transformations in digital images. *CVGIP*, 34:344–371, 1986.
- [Gon88] W. Gong. On decomposition of structure element for mathematical morphology. In *ICPR*, pages 836–838, 1988.
- [JPT89] L. Ji, J. Piper, and J. Tang. Erosion and dilation of binary images by arbitrary structuring elements using interval coding. *Pattern Recognition Letters*, (9):201–209, 1989.
- [Nor03] Nicolas Normand. Convex structuring element decomposition for single scan binary mathematical morphology. In *Discrete Geometry for Computer Imagery*, Naples, à paraître, 2003.
- [RP68] A. Rosenfeld and J.L. Pfaltz. Distances functions on digital pictures. *Pattern Recognition Letters*, 1(1):33–61, 1968.
- [Vin91] L. Vincent. Morphological transformations of binary images with arbitrary structuring elements. *Signal Processing*, (22):3–23, 1991.
- [WB92] X. Wang and G. Bertrand. Some sequential algorithms for a generalized distance transformation based on minkowski operations. *IEEE Trans. on PAMI*, 14(11):1114–1121, 1992.
- [Xu91] J. Xu. Decomposition of convex polygonal morphological structuring elements into neighborhood subsets. *IEEE trans. on PAMI*, 13(2):153–162, 1991.