

Hum, analyse d'images par scans successifs

[05.11.08]

SCAN #1: sur toute l'image.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

On a donc 30,31,37,41,42,47,48,51,56,57,58,67,68,77,87 allumés

activeCells = [30,31,37,41,42,47,48,51,56,57,58,67,68,77,87];

On connaît la largeur et la hauteur du tableau (10x10)

Au cours de la première boucle, le + allumé est 57. Ce sera le point de départ au grouping
mainCell = 57;

GROUP #1

mainGroups = []

each group knows its cells

each cell knows its id and witch borders to seek

if mainCell!=-1

mainGroups[0].cell[0].id = mainCell (< 57)

a	b	c
h	57	d
g	f	e

mainGroups[0].cell[0].ckecks = abcdefgh

on boucle sur la liste des check

les opérations sont:

- a: id-1-largeur
[si id>largeur && id-(round(id/largeur))>0]
si id = 9, pas bon, bord du dessus
si id = 40, pas bon, bord de gauche
aussi non, on cherche en dehors du tableau
- b: id-largeur
[si id>largeur]
- c: id+1-largeur
[si id>largeur && (id+1)-(round((id+1)/largeur))>0]
si id = 39, pas bon, bord de droite
- d: id+1
[si (id+1)-(round((id+1)/largeur))>0]
- e: id+1+largeur
[si (id+largeur)<(largeur*hauteur) && (id+1)-(round((id+1)/largeur))>0]
si 92, pas bon, ligne du fond
- f: id+largeur
[si (id+largeur)<(largeur*hauteur)]
- g: id-1+largeur
[si (id+largeur)<(largeur*hauteur) && id-(round(id/largeur))>0]
- h: id-1
[si id-(round(id/largeur))>0]

chaque check retourne un id, qui doit être testé:

si le test est positif (if activeCells[returned id] exist) ->

mainGroups[0].cell[new].id = returned id

règle des checks (si le test est positif sur X, la nouvelle cellule devra checker Y)

- X:a > Y:abh
- X:b > Y:b
- X:c > Y:abc
- X:d > Y:d
- X:e > Y:efg
- X:f > Y:f
- X:g > Y:fg
- X:h > Y:h

a	b	b	a	b
h	a	b	c	c
h	h	57	d	d
h	g	f	e	e
g	f	f	g	f

A chaque mainGroups[0].cell[m].ckecks, on retire la lettre checkée. Tant que la checks.length>0, on boucle
Puis, on passe à la cell suivante.

A chaque ajout d'une activeCells dans un groupe, il faut la retirer de activeCells

A la fin de la première boucle (mainGroups[0]), on aura donc
mainGroups[0].cells = 57,47,48,58,68,67,77,56,87 dans cet ordre, vu la liste de check
activeCells = [30,31,37,41,42,51];

pour mettre le tout en musique:

```
while activeCells.length > 0
    mainCell = seekBiggestCell
    currentGroup = createGroup
    mainGroups[currentGroup].addCell(mainCell, abcdefgh)
    for (i<mainGroups[currentGroup].length)
        mainGroups[currentGroup][i].performChecks
            //results is addition of new cells to the group
            if check==true, removeActiveCell(check result id)
```

avec le tableau de départ, on obtient 2 groupes

lors de la création du groupe, on peut directement stocker les cellules extrêmes (haut, gauche, bas & droite)

On a alors le l'occupation maximum de chaque groupe.

Avec ce système, il est possible de grouper deux formes qui « dépassent » l'une sur l'autre

La finesse de la grille de départ est bien sûr une condition à la qualité du groupage!

UPDATE [08.11.08]

après implémentation, il y avait des problèmes dans les opérations:

les opérations de détections:

- a: id-1-largeur
[si id>largeur && id-(round(id/largeur)*largeur)>0]
si id = 9, pas bon, bord du dessus
si id = 40, pas bon, bord de gauche
aussi non, on cherche en dehors du tableau
- b: id-largeur
[si id>largeur]

- c: id+1-largeur
[si id > largeur && (id+1)-(round((id+1)/largeur)*largeur) > 0]
si id = 39, pas bon, bord de droite
- d: id+1
[si (id+1)-(round((id+1)/largeur)*largeur) > 0]
- e: id+1+largeur
[si (id+largeur) < (largeur*hauteur) && (id+1)-(round((id+1)/largeur)*largeur) > 0]
si 92, pas bon, ligne du fond
- f: id+largeur
[si (id+largeur) < (largeur*hauteur)]
- g: id-1+largeur
[si (id+largeur) < (largeur*hauteur) && id-(round(id)/largeur)*largeur > 0]
- h: id-1
[si id-(round(id)/largeur)*largeur > 0]

règle des checks

- X:a > Y:abh
- X:b > Y:abc
- X:c > Y:bcd
- X:d > Y:cde
- X:e > Y:def
- X:f > Y:efg
- X:g > Y:fgd
- X:h > Y:agh

a	b			
h	a	b	c	
	h	57	d	
	g	f	e	

	a	b	c	
	a	b	c	
	h	57	d	
	g	f	e	

			b	c
	a	b	c	d
	h	57	d	
	g	f	e	

	a	b	c	c
	h	57	d	d
	g	f	e	e

	a	b	c	
	h	57	d	
	g	f	e	d

			f	e
--	--	--	---	---

...

Informations connues pour chaque cellule:

Dans le tableau général des cellules (celui recouvrant toute l'image):

0: [0,1] occupation courante: nombre de pixels actifs / nombre de pixel total

1: [0,1] interpolation: alignement progressif en fonction de 0; à chaque analyse, on ajoute 50% de la différence entre 0 et 1.

2: [-1,1] direction de l'activité: avant analyse, on déclare que 2 = 1. Après analyse, on retire le 1 du 2 => on sait si 1 est plus grand ou plus petit que 2. Si 2 est plus grand, la cellule est en train de s'éteindre. Si plus petit, la cellule s'allume. Si = 0: la cellule est stable.

3: 0/1: est-ce que la cellule est déjà attribuée à un groupe. Évite qu'une cellule se retrouve dans 2 groupes

Dans les groupes, on stocke les références aux cellules du tableau général (clé « position »)

Description des groupes (hashmap)

cells: ArrayList de cellules

borders: table de 4 Integer: top, right, bottom, left. Ce sont les 4 points extrêmes du groupe. Top & bottom sont des rows, right & left des colonnes

baricenter: point d'équilibre de la forme != au centre des bords. En fonction de leur occupation, les cellules ont +/- de poids. Le baricentre en est la représentation.

baricenterNegatif: les cellules qui s'éteignent indiquent d'où la forme vient.

baricenterPositif: les cellules qui s'allument indiquent où la forme va.

Le baricentre est utilisé pour repéré les groupes d'une analyse à l'autre.

Après implémentation, ajout de 2 boundaries pour augmenter les perfs: maxNumOfGroup (pas besoin d'en sortir 300) & minNumOfCellsPerGroup (on vire les groupes de - de x cellules).

[09.11.08]

Big problem avec le groupage: dès qu'un deuxième groupe arrive, et qu'il est traité avant le premier, le simple groupage en fonction d'une distance au centres précédents ne fonctionne pas!

Pfffff...

[12.11.08]

En fait, il faut d'abord comparer tous les groupes courants aux groupes précédents, puis choisir!

Ca donne

Ancien groupes

ag.0.pos = x,y

ag.0.id = A

ag.0.proximity []

ag.1.pos = x,y

ag.1.id = B

ag.1.proximity []

etc.

Puis, on passe sur les nouveaux groupes et on compare aux AG

ng.0.pos = x,y

ng.0.id = ?

-> distance à ag.0 = 10

-> distance à ag.1 = 5

ng.1.pos = x,y

ng.1.id = ?

-> distance à ag.0 = 8

-> distance à ag.1 = 12

ng.2.pos = x,y

ng.2.id = ?

-> distance à ag.0 = 20

-> distance à ag.1 = 4

=>

ag.0.proximity = [0 = 10, 1 = 8, 2 = 20]
ag.1.proximity = [0 = 5, 1 = 12, 2 = 4]

Du coup, on sait qui est qui:
ag.0.id -> ng.1.id
ag.1.id -> ng.2.id

ng.0.id = ?
ng.1.id = A
ng.2.id = B

D'où on peut déduire que 0 est un nouveau groupe à qui il faut attribuer un nouvel id, disons C.
A l'opération suivante, simple: ag comptera 3 positions.

[13.11.08]

Ca fonctionne, mais le centre n'est pas suffisant vu qu'on travaille dans une grille et que donc les centres peuvent être très proche. Pour être certain, j'ai ajouté un calcul sur les 4 bords.
La distance au centre + abs différence des bords constitue la donnée de référence pour la comparaison.
Même si certains groupes « sautent » encore, les groupes principaux sont stables!

Je suis monté jusqu'à une grille de 40x40. A voir si on peut descendre jusqu'au pixel.

A faire maintenant:

L'occupation de chaque groupe: nombre de cellules/surface totale (donnée par les bords). Ca indiquera si la forme est « étendue » ou « ramassée ». Au moins l'occupation sera importante, au plus on peut prétendre que la personne a les bras et les jambes écartées.

Avec le baricentre, on sait en plus si de quel « côté » penche la forme. Un calcul entre le centre de la surface et le baricentre donne la direction de la forme. (angle et rayon).

A tout ça, on ajoute en plus un baricentre des cellules s'allumant et un baricentre des cellules s'éteignant.
J'aurai alors une représentation « énergétique » du corps.

Baricentre neutre (comment est réparti le poids dans la forme)
Baricentre positif (vers où la forme va)
Baricentre négatif (d'où la forme vient)

Problème: si la personne écarte les 2 bras en même temps, les baricentres signés seront relativement stable. Une autre donnée est à prendre en compte. Comme occupation et baricentre neutre sont liés, les baricentres signés doivent être balancés d'une occupation signée.

3 couples donc:

Baricentre et occupation neutre
Baricentre et occupation positif
Baricentre et occupation négatif

On a alors une représentation énergétique « précise » du corps observé.

Les baricentres s'alimentent sur:

neutre: somme cellules[1]
positif: somme if cellules[2]>0
négatif: somme if cellules[2]<0

Comment on calcule un baricentre?

Le centre, c'est simple:

$x = \text{bord gauche} - (\text{bord droit} - \text{bord gauche})/2$
 $y = \text{bord haut} - (\text{bord bas} - \text{bord haut})/2$

Le baricentre:

si je dis x += position x de la cellule

avec un groupe qui commence à 5 et qui fait 10 de long (se finit à 14)

$5 + (10/2) = 10$, on a la position absolue du centre.

Chaque valeur va être multipliée par son opacité, entre 0 et 1 donc. Ca donnera toujours un résultat plus bas que la moitié.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

[14.11.08]

Etude de cas simples:

1. Si les cellules 5 à 10 sont à 1 et que toutes les autres sont à 0, le baricentre doit être à 8, ceci représentant le milieu des cellules actives

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

2. Si les cellules 5 et 6 sont à 1 et que toutes les autres sont à 0, le baricentre doit être à 6, ceci représentant le milieu des cellules actives

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

3. Si les cellules 12 à 14 sont à 1 et que toutes les autres sont à 0, le baricentre doit être à 13.5, ceci représentant le milieu des cellules actives

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

Avec des cellules à 0 et 1, le calcul est relativement simple...

1: distance au centre de chaque cellule: $-5-4-3-2-1-0$ (-15), divisé par nombre de cells active (6) $\rightarrow -2.5$. Additionner de la position du centre (10), on arrive à 7.5, pas à 8. Avec un floor, ça donne 2... Testons le cas suivant.

2: $-5-4$ (-9)/cells actives (2) $\rightarrow -4.5$, floor $\rightarrow 4$, centre $10-4 \rightarrow 6$. Ca tombe juste.

3: $2+3+4$ (11)/3 $\rightarrow 3,666$, ... avec un floor basique, on arrive à 3, et pas à 3.5. Par contre, en disant if $((x-\text{floor}(x))\leq 0.5) \rightarrow x=\text{floor}(x)$, else if $(x-\text{floor}(x)>0.5) \rightarrow x=\text{floor}(x)+0.5$

Je viens de détecter une incohérence dans le calcul.

5 et 14 sont à distance 5 du milieu, ou plutôt leur centre à 4.5 chacun! Autant que 10 n'est pas à 0 mais à 0.5 du centre de la forme. On parle d'une surface, pas d'un point!

La distance au centre est donc, pour chaque cellule, augmentée/diminuée de 0.5 if (distance!=0) { distance+0.5 }. En négatif ça retire, en positif augmente.

Donc:

1: $-4.5-3.5-2.5-1.5-0.5+0.5 \rightarrow -12 / 6 \rightarrow -2$, additionné au centre 8!

2: $-4.5-3.5 \rightarrow -8 / 2 \rightarrow -4$, additionné au centre 6!

3: $2.5+3.5+4.5 \rightarrow 10,5 / 3 \rightarrow 3.5$, additionné au centre 13.5!

C'était donc ça!!!!

Ca marche aussi si le nombre de cellules est impair. 11 cellules actives, centre 10.5. Pour la cellule 10, $10+0.5 - \text{centre} = 0$.

Pour une ligne et des cellules pleine...

Il faut pondérer.

1: 12=1, 13=0.5, 14=1

La baricentre est toujours 13.5

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

$(2.5*1)+(3.5*0.5)+(4.5*1) = 8,75$

activeCells = $1 + 0.5 + 1 = 2.5$

$8.75 / 2.5 = 3.5$, cool.

Une cellule allumée à 50% a une influence de 0.5, que ce soit dans le nombre de cellules actives aussi bien que dans la distance.

2: 12=0.5, 13=1, 14=1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

$(2.5*0.5)+3.5+4.5 = 9.25$

activeCells = $0.5 + 1 + 1 = 2.5$

$9.25 / 2.5 = 3.7$

Correct aussi, la 12 ne tire pas beaucoup, donc le baricentre va vers le centre de 13 et 14.

Maintenant, si j'ai 3 lignes, le calcul n'est plus le même

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

On a 9 activeCells sur 3 lignes. La largeur du groupe n'est plus le seul facteur, il faut prendre en compte le nombre de lignes. A vue de nez, le baricentre x doit être autour de 7.

Ligne 1: $-4.5-3.5 = -8$

Ligne 2: $-4.5-3.5-2.5-1.5-0.5 = -12.5$

Ligne 3: $-4.5-3.5 = -8$

Total: $-28.5 / 9 = 3.16666$. Retirer de 10, on arrive bien à -6.83333 . Pas besoin de se casser plus la tête, on se fout du nombre de lignes, seul le nombre de cellules pondérées compte.

Calcul:

totalDistanceX = 0

totalDistanceY = 0

for each (cellule) {

if (cellule %allumage>0)

totalDistanceX += ((cellule x - centre x)+0.5) * abs (cellule %allumage)

totalDistanceY += ((cellule y - centre y)+0.5) * abs (cellule %allumage)

activeCellsRatio += abs (cellule %allumage)

}

baricentre x = centre x + (totalDistanceX/activeCellsRatio)

baricentre y = centre y + (totalDistanceY/activeCellsRatio)

Opération à effectuer pour les 3 couples, adapter la condition de la boucle pour chaque couple.

[17.11.08]

Problème: on ne connaît le centre qu'à la fin de la boucle d'ajout des cellules. Or, il serait bien plus « rentable » en temps processeur d'ajouter les valeurs au fur et à mesure et d'enlever le centre à la fin. En reprenant une seule ligne de cellules (voir les 3 ci-dessus), On aurait:

$5+6 = 11$ (il faut arriver à -8!)

Le centre est à 10, et on a 2 cellules -> 20, et à chaque opération, on a ajouté +0.5

Donc, $(-10*2)+(0.5*2) = -19$

$11-19 = -8$ (bon, bon, ça m'a l'air bien): nombre de $-(\text{nombre de cellules}*(\text{centre} - 0.5))$.

On continue et on teste sur les 3 lignes

$5+6+5+6+7+8+9+5+6 = 57$

$-(9*(10-0.5)) = -85,5$

-> -28.5. cool, c'est bon. On reprend avec des pourcentage.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

Avant:

$(2.5*1)+(3.5*0.5)+(4.5*1) = 8,75$

activeCells = $1 + 0.5 + 1 = 2.5$

$8.75 / 2.5 = 3.5$.

Maintenant:

$12+(13*0.5)+14 = 32,5$

$9.5*2.5 = 23.75$

$32.5-23.75 = 8.75$

Bon, c'est réglé.

On commence d'abord par additionner toutes les positions pondérées, puis on retire le centre-0.5 * le total pondéré.

Pour le baricentre neutre, pas de problème. Mais les baricentres positifs et négatifs sont très instables quand il y a peu de mouvement... A revoir.

[18.11.08]

Problème de baricentres signés.

[24.11.08]

exportation svg

problème de définition de taille de sortie...

le plus simple étant d'indiquer en mm la taille de fichier désirée, et là calculer l'épaisseur des traits...

normalement donc, un grand fichier aura des traits plus épais qu'un petit...

De plus, dans inkscape, tout est exprimé en pixels, la conversion en mm est faite dans l'interface...

Donc, conversion:

100mm -> 354.33069 px

=> si je veux une image de n mm, je dois multiplier n par 3.5433069 (constante px/mm)

Si maintenant je rentre 100px, ça me donne $100 \cdot (1/3.5433069)$. La constante dans ce sens est de 0.28222223709.

SvgOuput (la classe), va fonctionner comme suit:

Déclaration de la taille de l'export en mm (par ex: 1500x910, ça, ça serait cool!)

Et chaque stroke par contre est lui donné en pixels. Donc, pas de problème, on peut le donner directement à l'export.

Seulement, il faut le balancé avec l'opacité du trait du moteur..

Basiquement, on fait: $\text{stroke animé} \cdot \text{opacité animé} = \text{stroke print}$

Autre chose: proportionnalité écran/print

Si je veux garder l'écran et le print proportionnels, il faut diviser la taille du print par la taille écran

=>

$w \text{ print} / w \text{ screen} = wp$

$h \text{ print} / h \text{ print} = hp$

on prends la + petite, on on fixe la proportionnalité.

Chaque $\text{stroke print} = \text{stroke animé} \cdot \text{opacité animé} \cdot \text{prop}$

On y va