

14. Les textures

Nombre de participants : 10



Les textures

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Crédit photo: Enchantedgal-Stock, [deviantart.com](https://www.deviantart.com), merci à Derek Hoiem

Les textures

Introduction aux textures

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Crédit photo: Enchantedgal-Stock, deviantart.com, merci à Derek Hoiem

2

Qu'est-ce qu'une texture?



source: heifer12x12.com

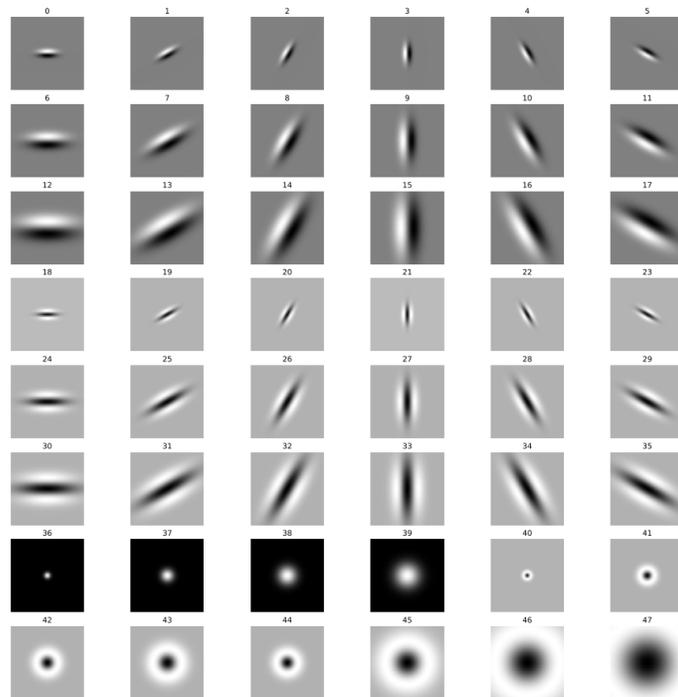
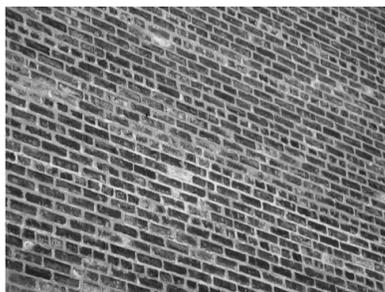
Types de textures



4

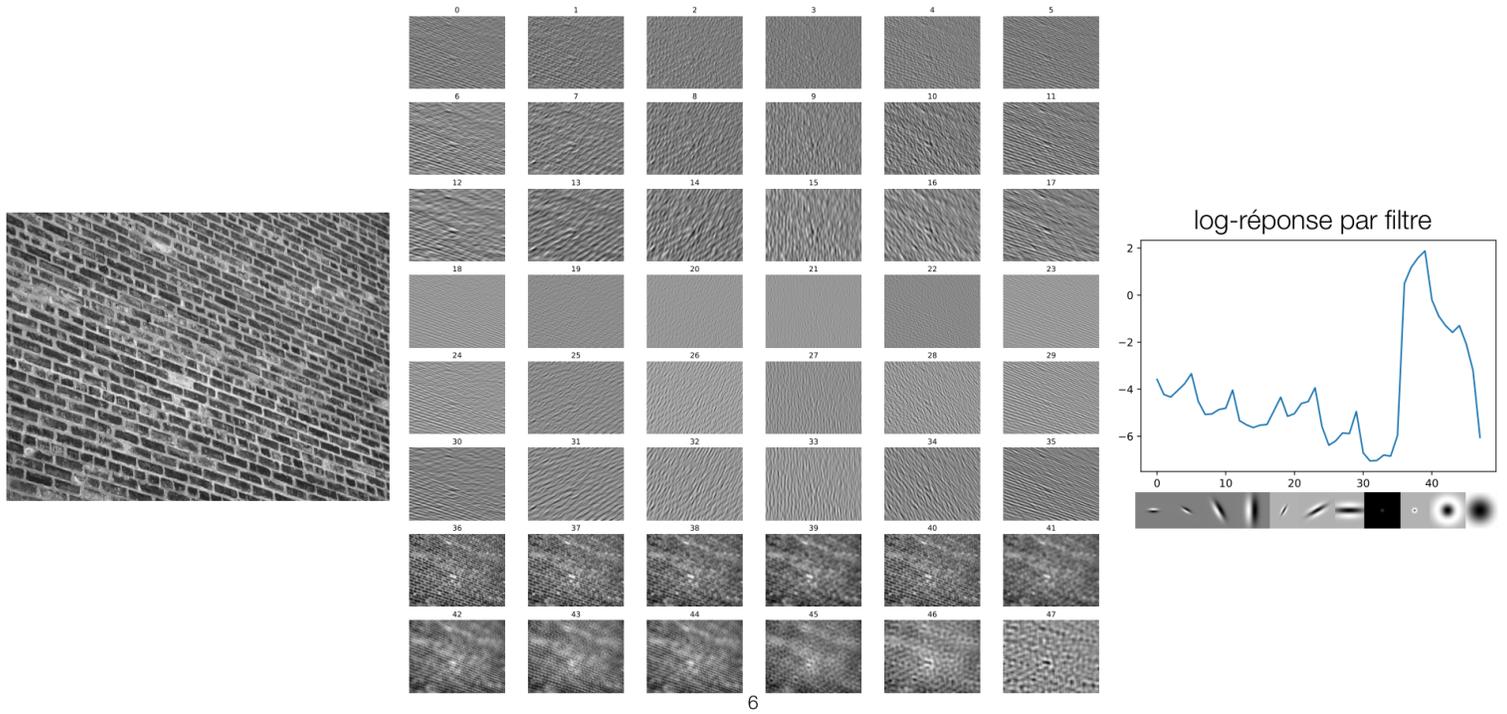
Source: Hoier

Banque de filtres

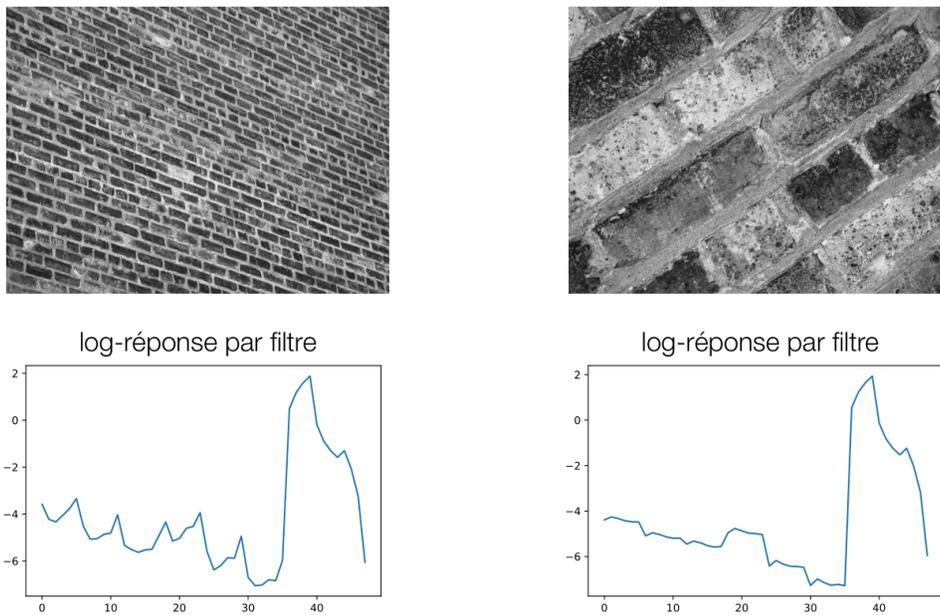


5

Banque de filtres



Banque de filtres



Les textures

Synthèse de textures par appariement de statistiques

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Crédit photo: Enchantedgal-Stock, deviantart.com, merci à Derek Hoiem

8

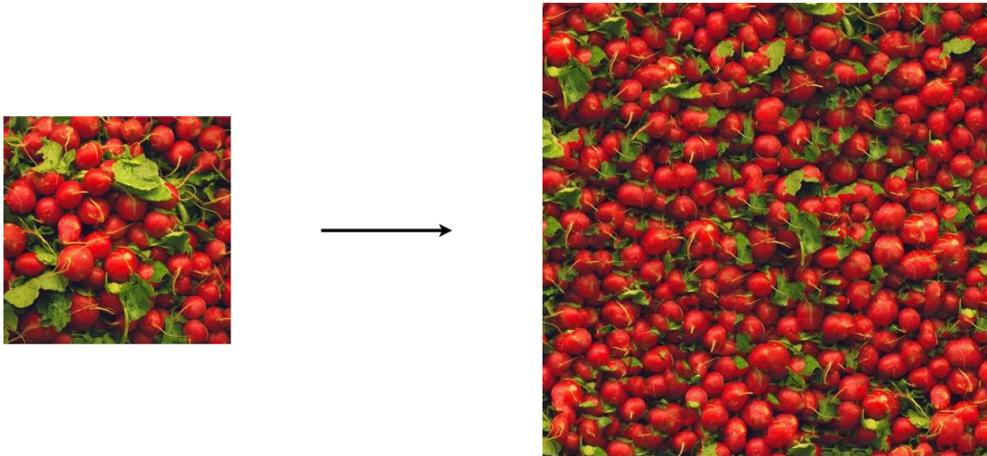
Synthèse de textures

- En entrée : exemple de texture
- But : répliquer l'exemple de la texture sur une plus grande surface



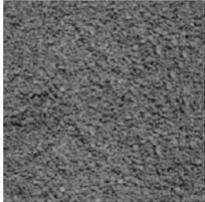
Synthèse de textures

- En entrée : exemple de texture
- But : répliquer l'exemple de la texture sur une plus grande surface

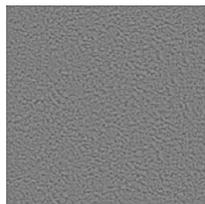


10

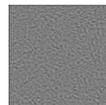
Exemple de texture



Niveau 0



Niveau 1



Niveau 2

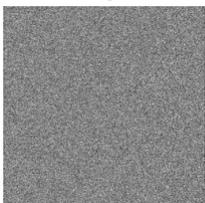


Niveau 3



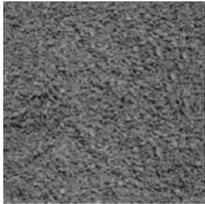
Statistiques
Histogrammes de différentes
bandes de fréquences

Texture générée

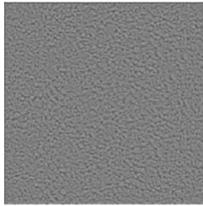


11

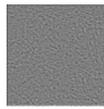
Exemple de texture



Niveau 0



Niveau 1



Niveau 2

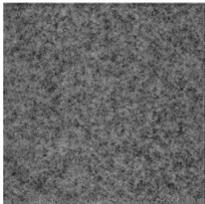


Niveau 3



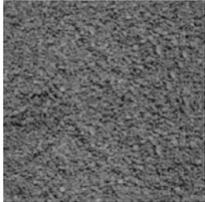
Statistiques
Histogrammes de différentes
bandes de fréquences

Texture générée

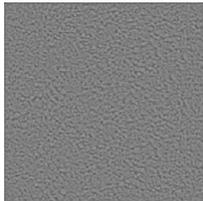


Après l'itération 1

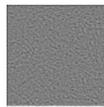
Exemple de texture



Niveau 0



Niveau 1



Niveau 2

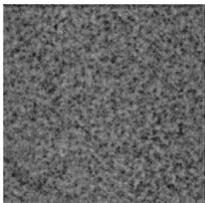


Niveau 3



Statistiques
Histogrammes de différentes
bandes de fréquences

Texture générée



Après l'itération 5

Synthèse de texture par appariement d'histogrammes

[Heeger et Bergen 1995]

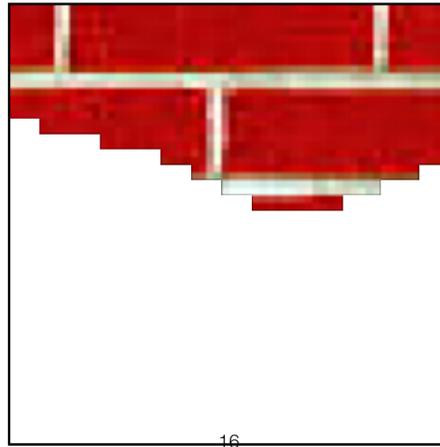
- Initialiser : image générée = image de bruit aléatoire
- Apparier l'histogramme de l'image générée avec celui de la texture d'exemple
- Calculer la pyramide laplacienne de l'exemple
- Pour N itérations :
 - Calculer la pyramide laplacienne de l'image générée
 - Pour tous les niveaux de la pyramide :
 - Apparier l'histogramme de l'image générée avec celui de l'exemple
 - Reconstruire l'image générée à partir de sa pyramide
 - Apparier l'histogramme de l'image générée avec celui de l'exemple à nouveau

14



Idée : échantillonner l'image

- On remplit l'image un pixel à la fois
- Comment faire pour déterminer la valeur de ce pixel?



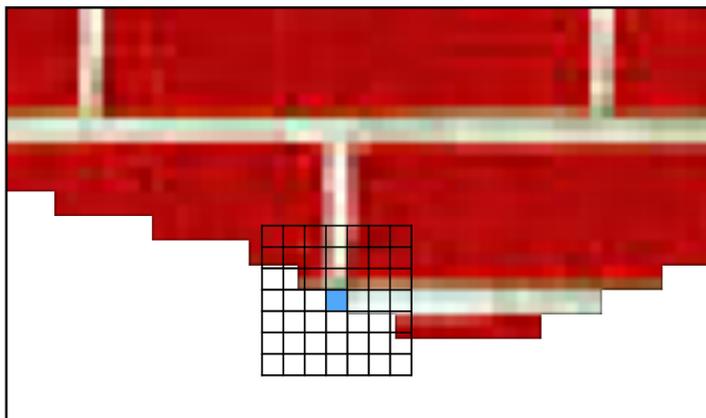
Idée : échantillonner l'image

- Faisons l'hypothèse que la valeur d'un pixel ne dépend que de celles de ses voisins



Solution?

- Plutôt que de calculer $P(\mathbf{p}|N(\mathbf{p}))$, utiliser l'image source et trouver des endroits similaires à $N(\mathbf{p})$



18

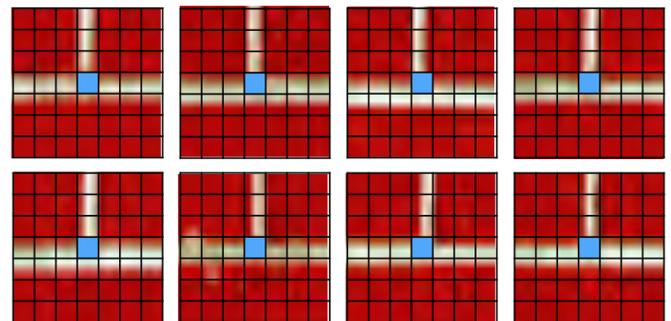
Source



Solution?

Approximation de
 $P(\mathbf{p}|N(\mathbf{p}))$
par échantillonnage non-paramétrique

- En pratique:
 - trouver les k voisinages les plus similaires
 - sélectionner aléatoirement (pourquoi pas moyenne?)



Source

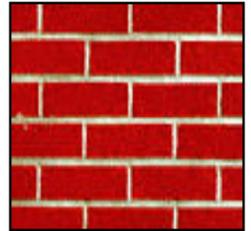


19

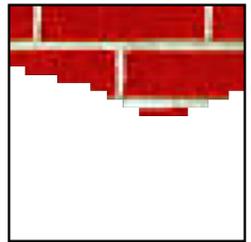
Algorithme : échantillonnage non-paramétrique

- Tant que l'image de destination n'est pas remplie :
 - Trouver, dans l'image de destination, le pixel inconnu qui a le plus de voisins;
 - Trouver, dans l'image source, les N pixels dont le voisinage est le plus similaire à celui du pixel inconnu
 - Somme des différences au carré, pondérée par gaussienne
 - Sélectionner aléatoirement parmi les pixels semblables, et copier sa valeur dans l'image.

Source



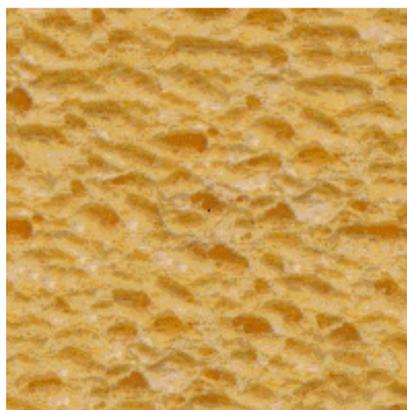
Destination



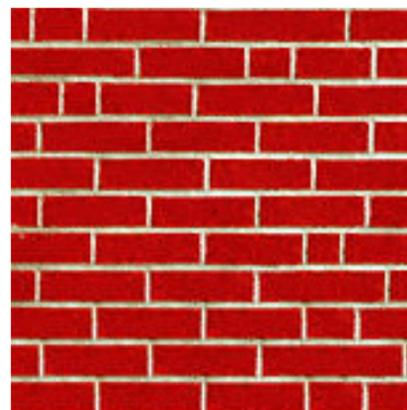
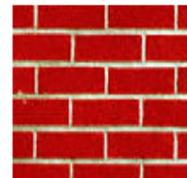
20

Résultats

pain



briques



21

En hommage à Shannon

coming in the unsensational
for Dick Gephardt was fair
rful riff on the looming
only asked, "What's your
tions?" A heartfelt sigh
story about the emergen
es against Clinton. "Boy
g people about continuin
ardt began, patiently obs
; that the legal system h
e with this latest tanzer



thaim. them . "Wnephartfe lartifelintomimen
tel ck Clirtioout omaim thartfelins.f out s anetc
the ry onst wartfe lck Gephtoomimeationl sigat
Clhoufnt Clinut Cil riff on. hat's yodn, parut tly
ons yoonstehst wasked, paim t sahe loo riff on l
nskoneploourtfeas leil A nst Cliit, "Wleontongal s
k Cirtioouirtfepe ong pme abegal fartfenstemem
diensteneltorydt telemephinsverdt was agemer
ff ons artientont Cling peme as urtfe atih, "Boui s
nal s fartfelt sig pedrirdt ske abounutie aboutioo
tfeonewas yow abowonhardt thatins fain, ped, '
ains. them, pabout wasy arfuit coutly d, l n A h
ole emthringbooreme agas fa bontinsyst Clinut
ory about continst Clipequinst Cloke agatiff out C
stome ainemen tly ardt beorabou n, thenly as t C
cons faimeme Diontont wat coutlyohgans as fan
ien, phrtfaul, "Wbout cout congagal comingga
mifmst Clily abon al coountha.emungaitf toun
The loocrysta loontieph. tntly on, theoplegatick t
ul tatieeontly atie Diontiomi wal s f tbegae ener
nthahgat's enenhibnas fan. "intchthorw abons v

22

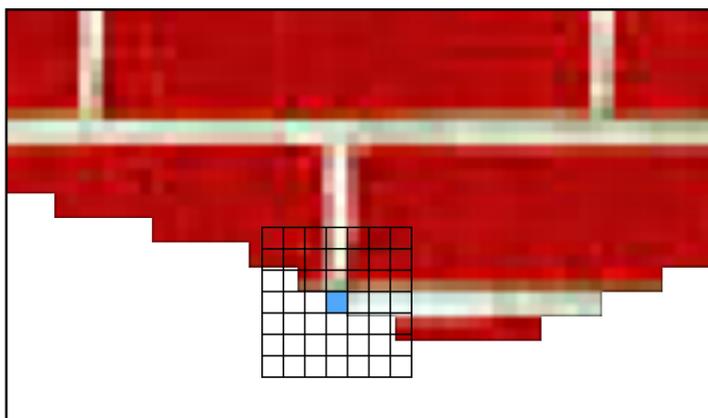
Les textures Synthèse de texture par courtepoin

Crédit photo: Enchantedgal-Stock, [deviantart.com](https://www.deviantart.com), merci à Derek Hoiem

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

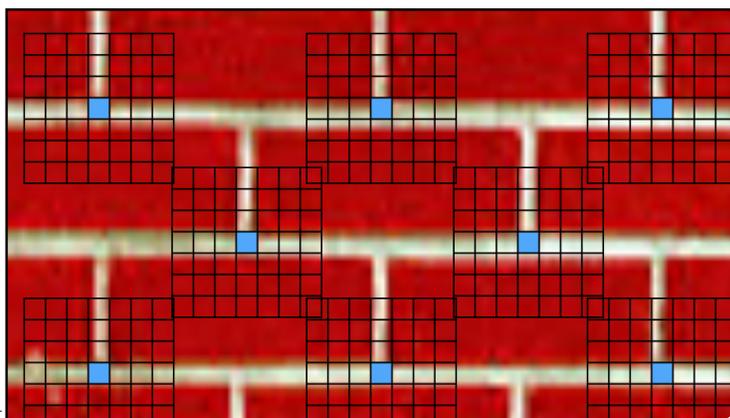
23

Observation

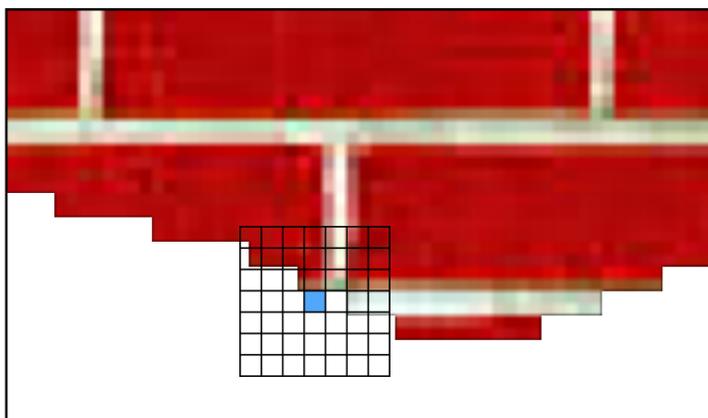


24

Source

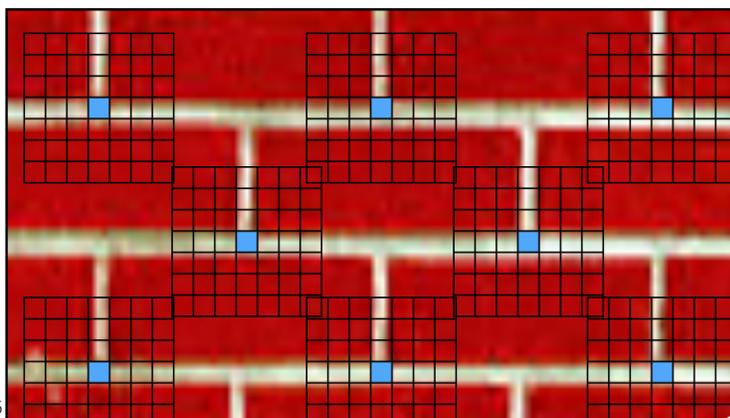


Observation

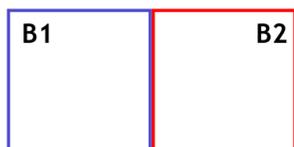
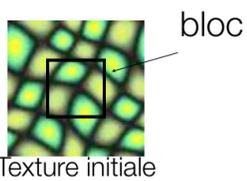
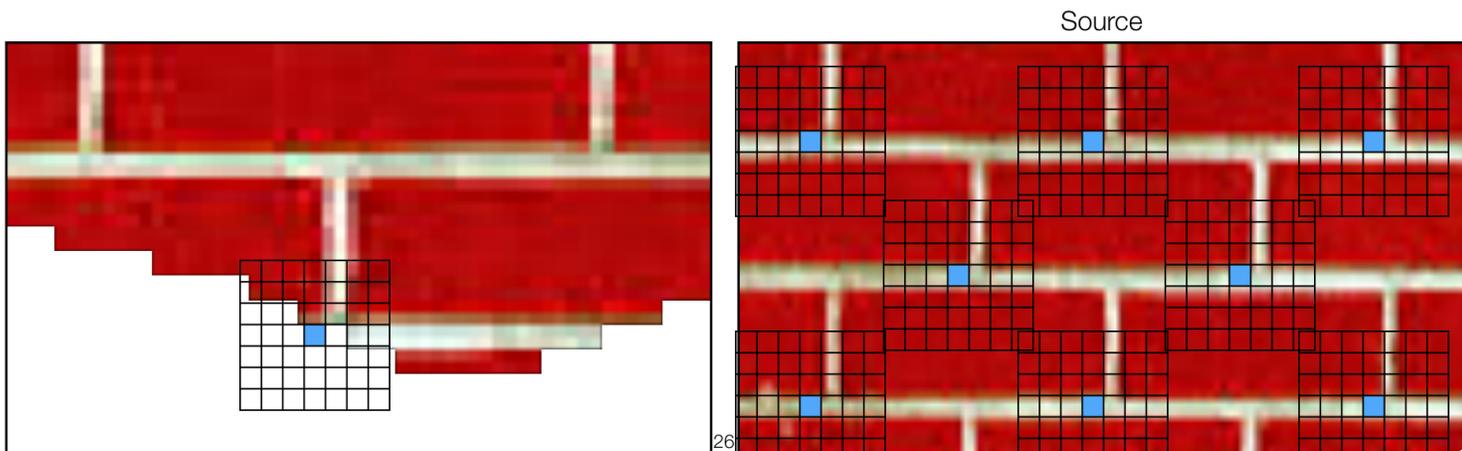


25

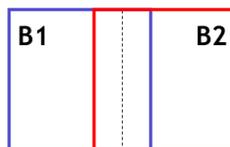
Source



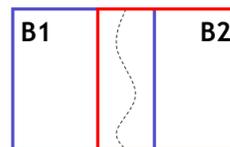
Observation



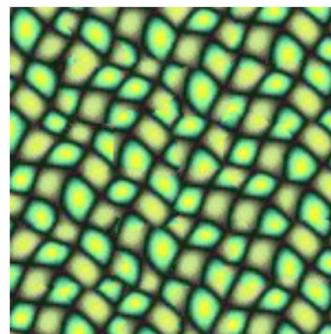
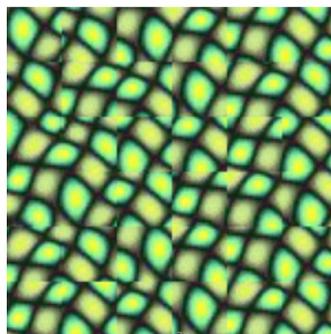
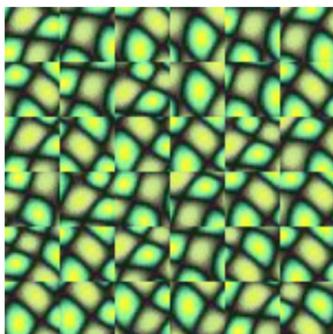
Placement des blocs aléatoire



Blocs voisins se chevauchent

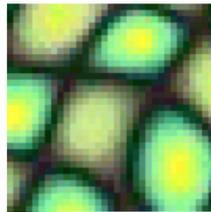
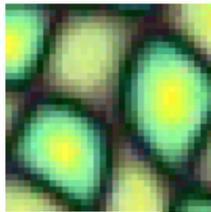


Coupure minimisant les discontinuités

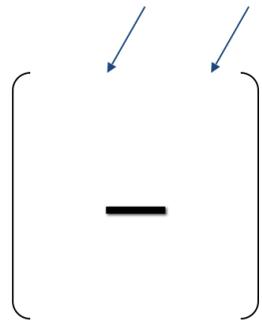
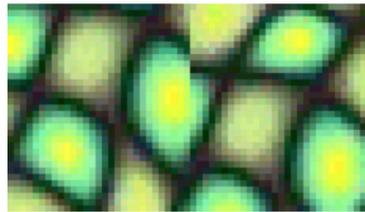


Coupure minimisant les discontinuités

blocs se chevauchant



discontinuité verticale

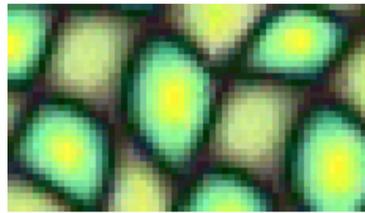


2

=



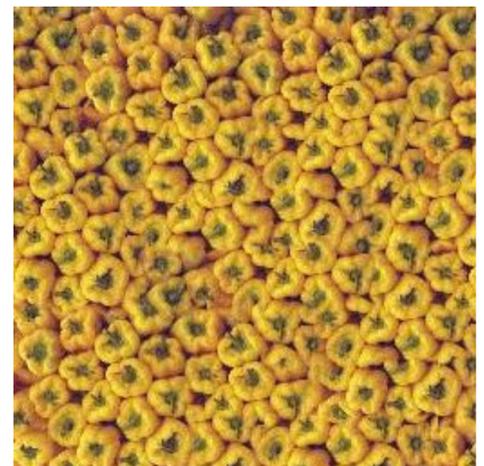
erreur de
chevauchement



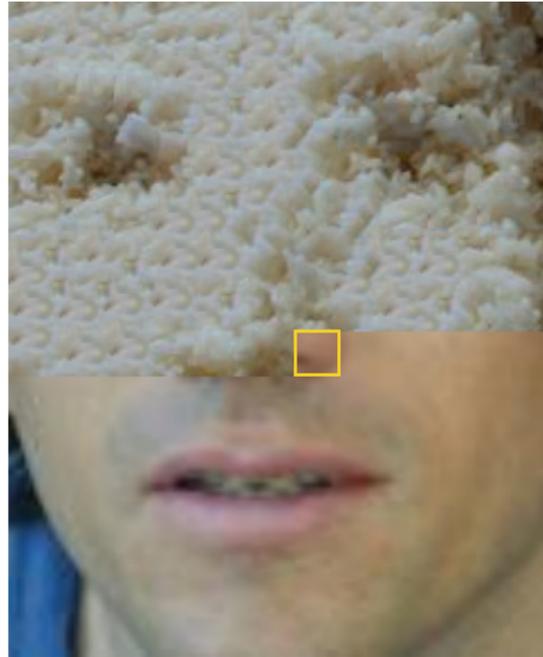
erreur minimale

28

Crédit : Efror



Transfert de texture



30



Les textures

Synthèse de texture par appariement de blocs

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Crédit photo: Enchantedgal-Stock, [deviantart.com](https://www.deviantart.com), merci à Derek Hoiem

31



32

Image de [Fried et al. 2015]



33

Image de [Fried et al. 2015]

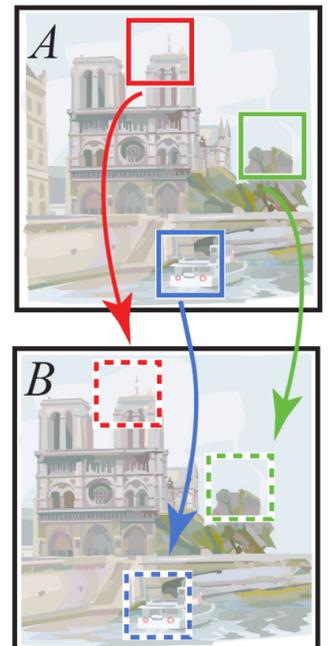


34

Image de [Fried et al. 2015]

PatchMatch [Barnes et al. 2009]

- Algorithme semi-aléatoire pour trouver les correspondances entre les blocs d'une image *de façon très rapide*
- Définition du problème:
 - Nous avons deux images, A et B.
 - Pour chaque bloc dans l'image A, calculer la translation (t_x, t_y) qui entre ce bloc et son plus proche voisin dans l'image B
 - Nous avons donc une translation (t_x, t_y) pour chaque pixel



35

Crédit: Kayvon Fatahaliar

PatchMatch [Barnes et al. 2009]

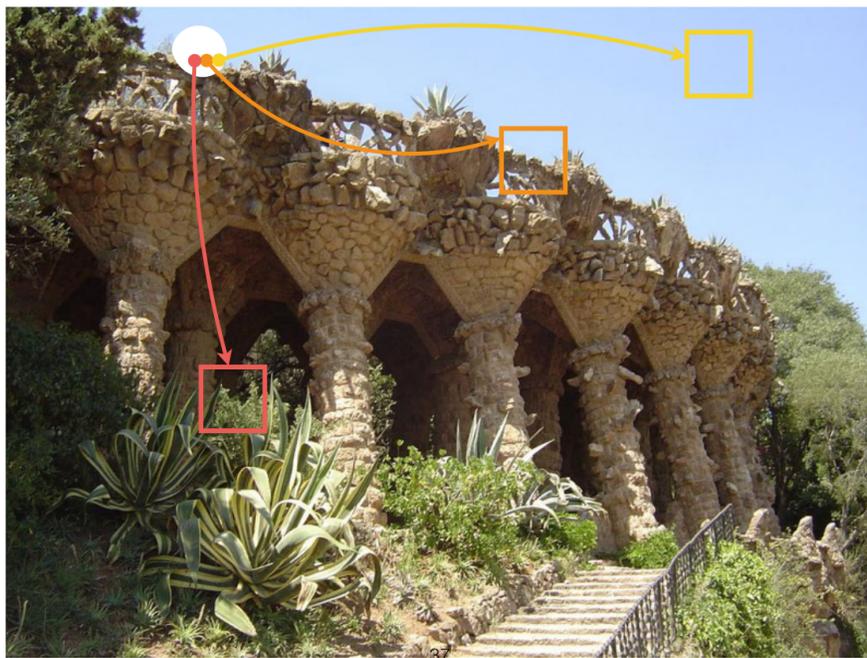
Idée #1 : une translation **aléatoire** devrait bien fonctionner pour **certains pixels**



36

PatchMatch [Barnes et al. 2009]

Idée #2 : les **voisins** sont **cohérents**



37

PatchMatch [Barnes et al. 2009]

Idée #2 : les **voisins** sont **cohérents**



Le plus proche voisin d'un bloc centré à (x, y) devrait être un bon indice pour trouver le plus proche voisin du bloc à $(x+1, y)$

- Boucler sur chaque pixel :
- Regarder le voisin à droite : si le bloc à sa droite est un meilleur candidat pour le bloc courant, alors remplacer le voisin du bloc courant. Sinon, garder le résultat précédent.
- Répéter l'opération avec le voisin en haut.
- À la prochaine itération, utiliser les voisins en bas et à droite