

# Module-projet 2006-2007

## **Planning :**

- Séances encadrées par un intervenant, (jeudi 9h-12h),
- 3ème semaine : Présentation orale des sujets,
- Dernière semaine : Rapport + Présentation orale.

## **Intervenants :**

M. Chaumont, W. Puech, J. Triboulet

## **Rappel :**

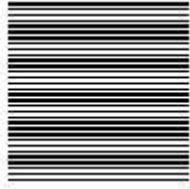
Vous devez travailler en **dehors** des séances encadrées.

Évaluation à partir :

- du manuscrit (15, 20 pages + annexes),
- des présentations orales,
- de la démonstration de vos logiciels,
- de la qualité du code (lisibilité, modularité, documentation, ...).

# 1- Étude et comparaison de quelques schémas numériques pour la magnification d'images

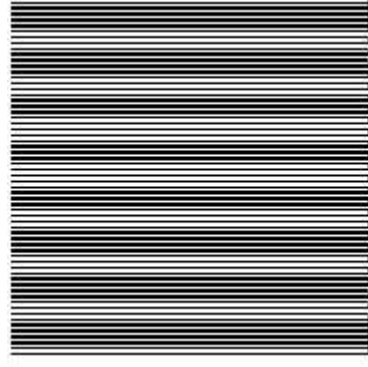
Magnification :



*Image  
cosinus*



*Magnification  
classique*



*Magnification  
adaptée*

[Montesinos S. Dattenny 1997]

**Phillipe Montesinos**  
**EMA- CLGI2P**  
**Philippe.Montesinos@ema.fr**

# 1- Étude et comparaison de quelques schémas numériques pour la magnification d'images

La magnification d'images consiste à agrandir une image tout en conservant des contours nets.

Nous allons travailler avec une méthode qui effectue une magnification lissée à l'aide d'un filtre gaussien sub-pixel. Nous rehaussons ensuite les contrastes avec une EDP de type équation de la chaleur inverse.

Le travail va consister en trois points :

- Implémenter plusieurs schémas numériques de résolution de l'équation de la chaleur inverse.
- Qualifier l'évolution de cette équation à l'aide d'une fonction "énergie".
- Tenter de définir un critère d'arrêt automatique des itérations.

**Phillipe Montesinos**

**EMA- CLGI2P**

**Philippe.Montesinos@ema.fr**

# 1- Étude et comparaison de quelques schémas numériques pour la magnification d'images

La magnification d'images consiste à agrandir une image tout en conservant des contours nets.

Nous allons travailler avec une méthode qui effectue une magnification lissée à l'aide d'un filtre gaussien sub-pixel. Nous rehaussons ensuite les contrastes avec une EDP de type équation de la chaleur inverse.

Le travail va consister en trois points :

- Implémenter plusieurs schémas numériques de résolution de l'équation de la chaleur inverse.
- Qualifier l'évolution de cette équation à l'aide d'une fonction "énergie".
- Tenter de définir un critère d'arrêt automatique des itérations.

« Sub-Pixel Accuracy Using recursive filtering », P. Montesinos, S. Dattenny, SCIA'97, Proceedings of the ``10th Scandinavian Conference on Image Analysis. pp. 523-530, Lappeenranta, Finland, June 9-11, 1997.

« Sur quelques schémas numériques de résolution d'équations aux dérivées partielles pour le traitement d'images », L. Lucido, R. Deriche, L Alvarez, V. Rigaud. RR3192, Juin 1997

## 2- Étude des algos de la librairie Boost et d'algos de dessin de graphes.

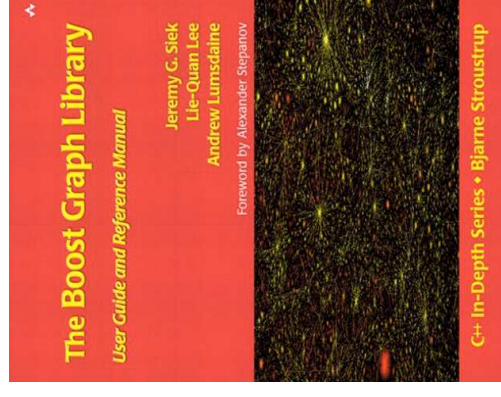
### A- Étude des algos de la librairie Boost :

Exemple d'algo de la « Boost Graph Library » à regarder :

- Johnson's All-Pairs Shortest Paths
- Dynamic Connected Components (using Disjoint Sets)
- Topological Sort
- Reverse Cuthill Mckee Ordering
- Smallest Last Vertex Ordering
- Sequential Vertex Coloring

### Travail à réaliser :

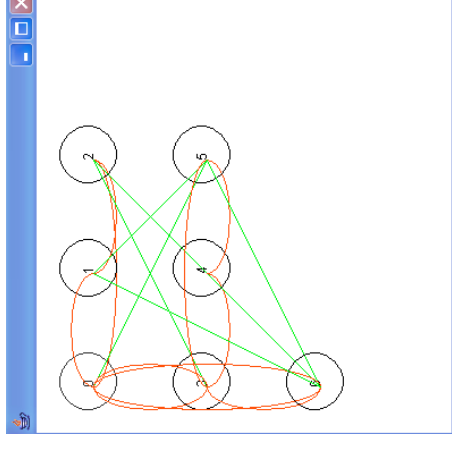
- Maîtriser la librairie,
- Découvrir et nous faire découvrir des algorithmes.



## 2- Étude des algos de la librairie Boost et d'algos de dessin de graphes.

### B- Dessin de graphes :

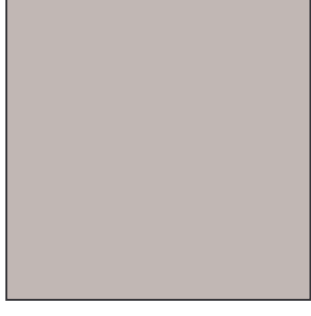
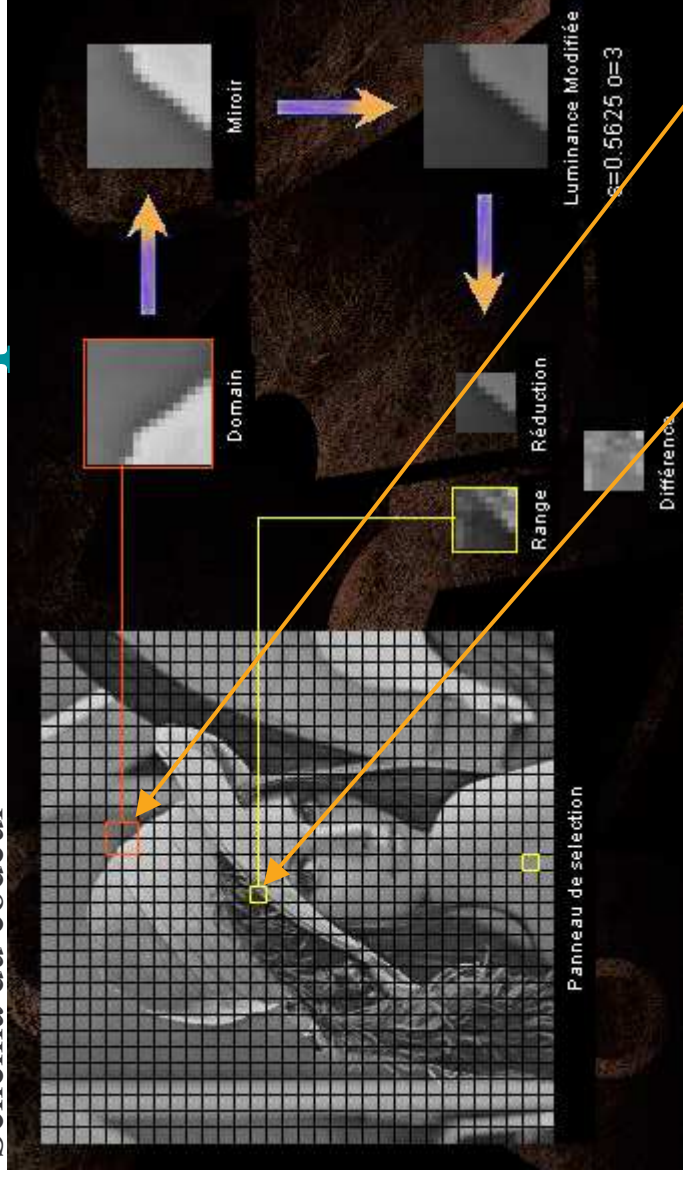
- Sans un minimum de théorie →



- Avec un peu de théorie (modèle de Tutte, 1963, Eades, 1984...) Idée de Eades et Tutte : Les sommets du graphe représentent des masses (boules) chargées, qui se repoussent deux à deux. Les arêtes entre deux sommets induisent l'existence d'un ressort entre les deux masses correspondantes, qui retiennent les masses à proximité selon la cinétique propre aux ressorts (lois de Hooke).

# 3- Implémentation d'un codeur- décodeur par fractale

Schéma du codeur



T : transfo. affine  
s : Facteur d'échelle  
o : Facteur de translation

- 1- Découpage en grille de l'image,
- 2- Recherche pour chaque bloc  $R_i$  un bloc  $D_i$  tel que
$$\min (R_i - (s.T(D_i) + o))^2$$
*{n°bloc, T,s,o} est appelée l'application contractante*
- 3- Stockage des paramètres {n°bloc, T, s, o} dans un fichier.

# 3- Implémentation d'un codeur- décodeur par fractale

## Décodeur :

Lecture des paramètres {n°bloc, T, s, o} du fichier;

ITERER JUSQU' A CONVERGENCE

début

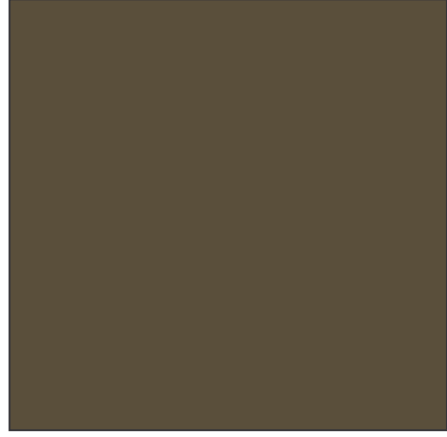
Pour chaque bloc  $R_i$  faire  $R_i = s.T(D_i) + o$  ;

fin

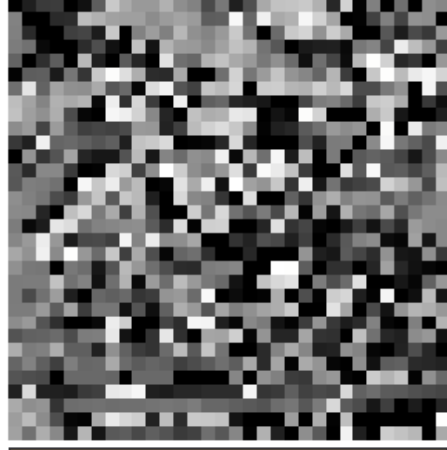
T : transfo. affine

s : Facteur d'échelle

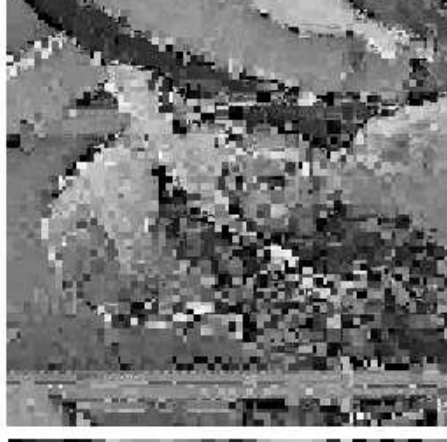
o : Facteur de translation



*Image initiale*



*Après 1 itération*



*Après 2 itérations*

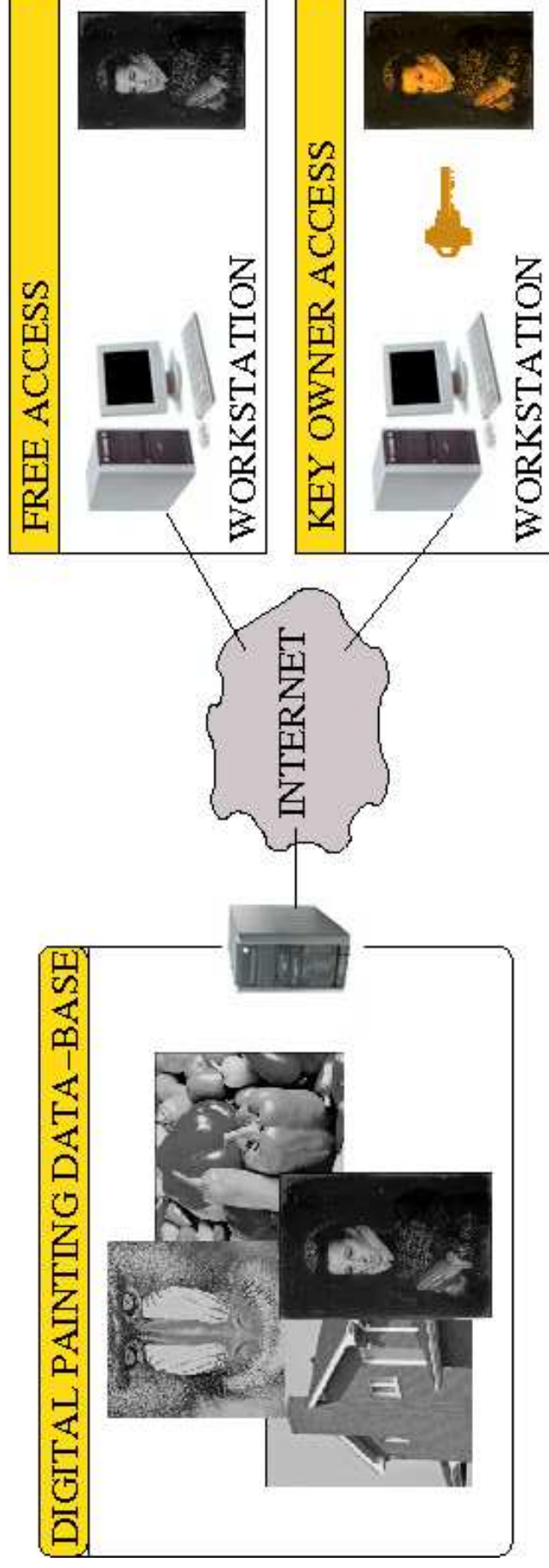


*Après 5 itérations*



# 4- Attaque d'une technique de tatouage couleur

Contexte : BD image numérique avec sécurisation de la couleur :



# 4- Attaque d'une technique de tatouage couleur

Contexte : BD image numérique avec sécurisation de la couleur :



# 4- Attaque d'une technique de tatouage couleur

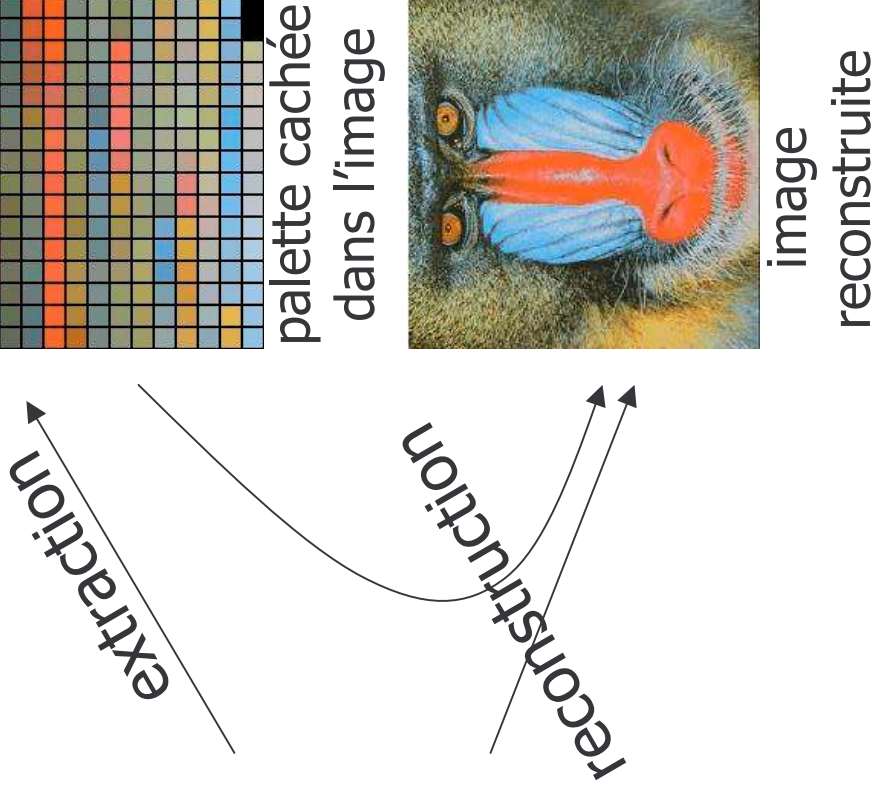
Contexte : BD image numérique avec sécurisation de la couleur :





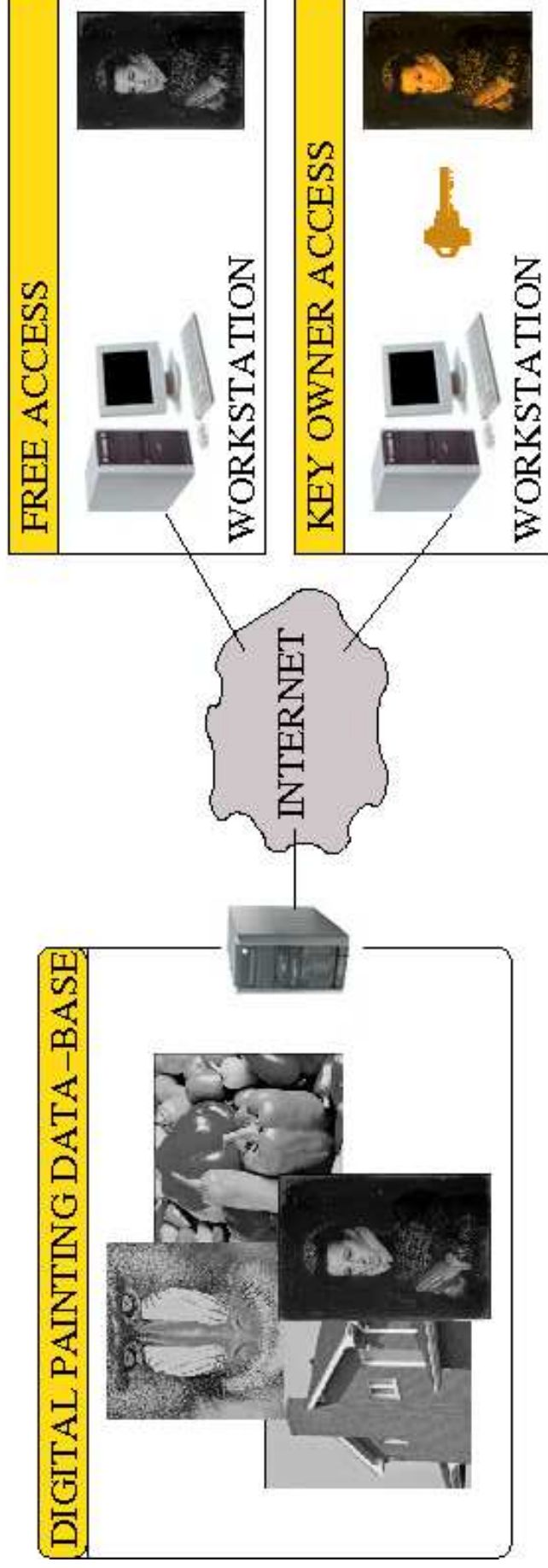
# Attaque d'une technique de tatouage couleur

Reconstruction de l'image couleur :



# 4- Attaque d'une technique de tatouage couleur

Contexte : BD image numérique avec sécurisation de la couleur :



**De nombreuses failles de sécurité :**

→ OBJECTIF Attaquer le système de tatouage :

Exemple : A partir d'une image tatouée, retrouver l'image couleur, c'est-à-dire : **re-colorier** l'image !!

# 4- Attaque d'une technique de tatouage couleur

→ OBJECTIF Attaquer le système de tatouage :  
**re-colorier** l'image tatouée !!!



Image tatouée

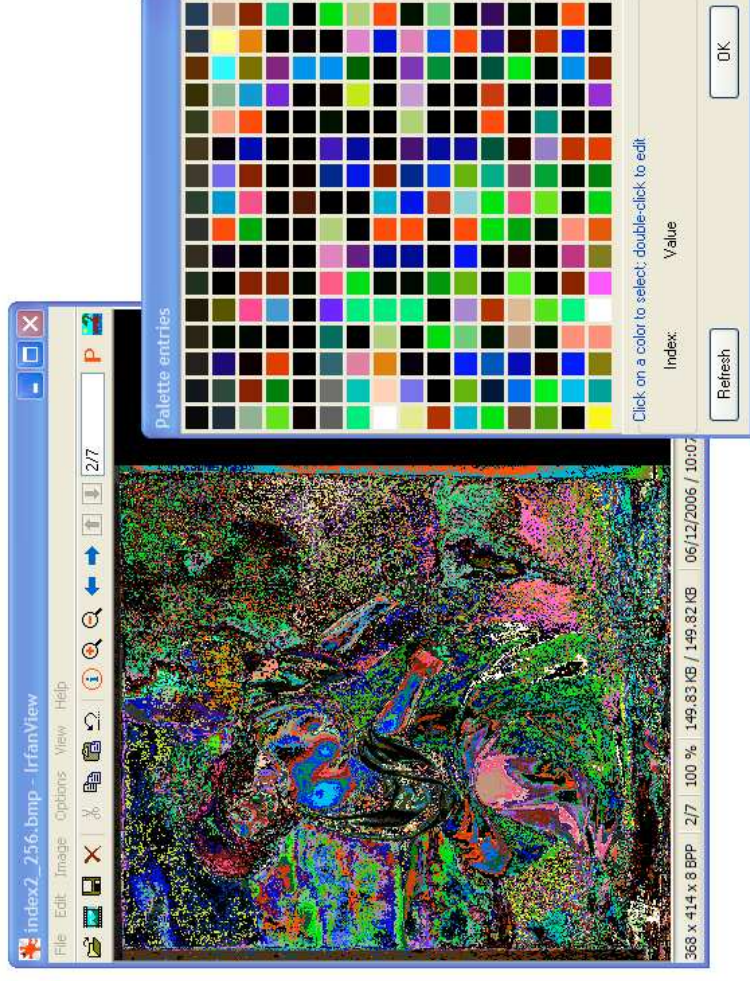


Image re-colorée aléatoirement

# 5- Étude des algorithmes de quantification + dithering + halftoning pour les images couleur

Illustration de l'effet problème

« plage de couleur » due à la quantification



Image originale  
Skyline arch (400\*594)



Quantifiée  
sur 256 couleurs



Quantifiée  
sur 16 couleurs

# 5- Étude des algorithmes de quantification + dithering + halftoning pour les images couleur



Image originale



Après application de l'algorithme  
de Floyd-Steinberg dithering



## 5- Étude des algorithmes de quantification + dithering + halftoning pour les images couleur

### Travail demandé :

Implémentation d’algo de dithering et/ou  
halftoning sur des images couleurs  
quantifiées réduites à 256 couleurs.

### Extension possible ...

Amélioration du tatouage couleur avec  
quantification et dithering simultané.

# 6- Étude de la mise en place de pluggin (tatouage vidéo) sous virtual dub

Démo logiciel *VirtualDub*

v1.6.17(stable) <http://www.virtualdub.org/>  
<http://virtualdub.sourceforge.net/>



**Mise en place des Filtres** [Tutorial \(http://www.virtualdub.org/filtersdk\)](http://www.virtualdub.org/filtersdk)

Download filter SDK 1.05 (filtersdk-1.05.zip, 66K zip file)

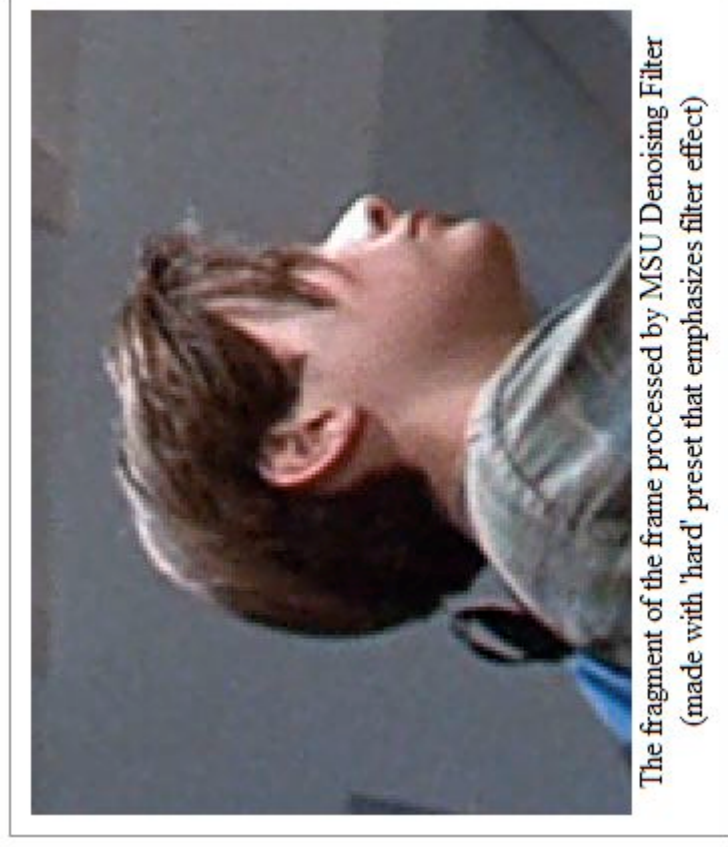
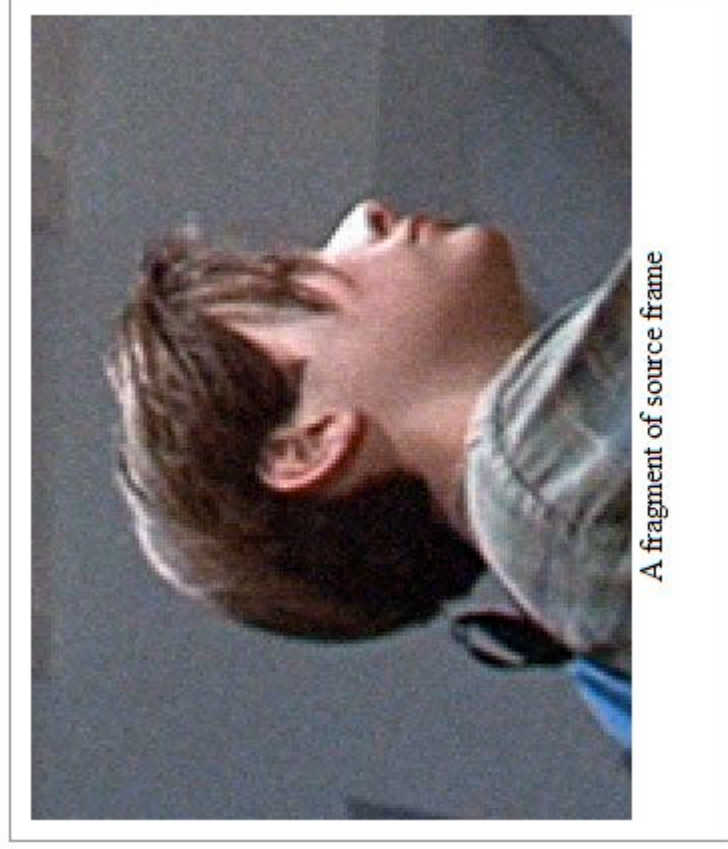
Requirements for the filter SDK:

- A 32-bit Windows C++ compiler. Microsoft Visual C++ 6.0 Service Pack 5 recommended; it is possible to write filters in Borland C++ Builder but I cannot help you with this. Microsoft Visual Studio .NET has not been tested with this SDK and is not recommended.
- VirtualDub 1.2 or newer (viable V1.4.11 build tree recommended to build debug builds).

# 6- Étude de la mise en place de pluggin (tatouage vidéo) sous virtual dub

MSU Graphics & Media Lab (Video Group)

[http://compression.ru/video/denoising/index\\_en.html](http://compression.ru/video/denoising/index_en.html)



MSU Denoising

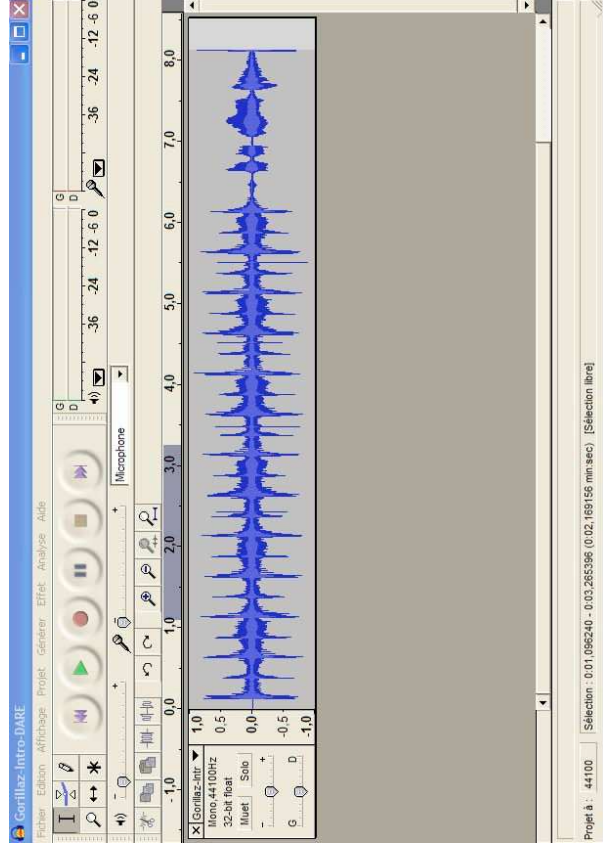
# 6- Étude de la mise en place de pluggin (tatouage vidéo) sous virtual dub

Exemples de filtrages envisageables :

- filtres vus en cours d'image,
- insertion-suppression de logos,
- débruitage de vidéo,
- détection et cryptage de visages,
- tatouage couleur (cf. autre sujet),
- tatouage par étalement de spectre,
- décryptage Canal+ ...

# 7- Étude de la mise en place de pluggin (tatouage son) sous audacity

Audacity <http://audacity.sourceforge.net/>



LADSPA Plug-Ins (<http://www.ladspa.org/>)

Nyquist Plug-Ins (langage interprété basé Lisp)

VST Plug-Ins

## 8- Construction de modèle 3D visage en VRML + openCV

**OpenCV : Librairie de traitement d'images**

Exemple : [code source](#)  
image + calcul histogramme + affichage

L'exécution du programme : [demo.exe](#)

# 8- Construction de modèle 3D visage en VRML + openCV

**VRML** : Description d'un fichier VRML:

```
coord Coordinate {  
  point [  
    0      1.061  -0.371,  
    0.174  0.8    -0.024,  
    0      0.539  0.085,  
    ...]  
  coordIndex [  
    0,      11,      -1,  
    0,      1,      -1,  
    0,      34,      -1,  
    ...]]
```

Modèle 3D de visage : [model3D.wrl](#)

## 8- Construction de modèle 3D visage en VRML + openCV

Ce que l'on veut :

Écrire une application openCV qui :

1- charge une image contenant un visage.



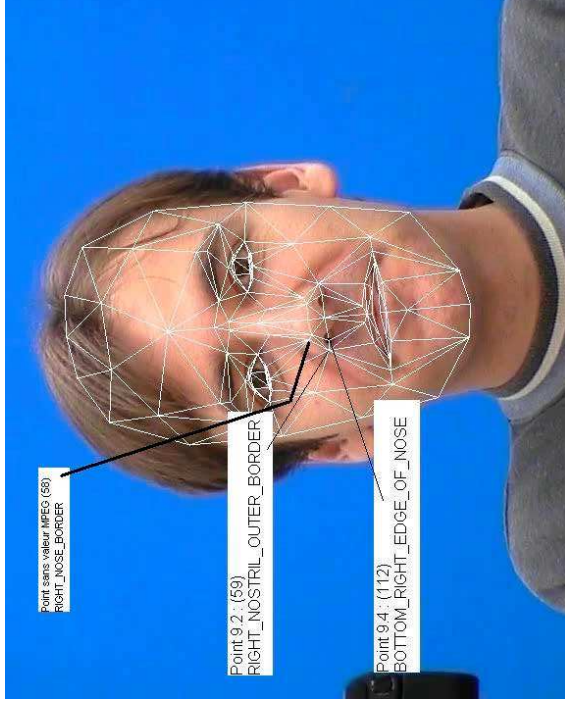


# 8- Construction de modèle 3D visage en VRML + openCV

Ce que l'on veut :

Écrire une application openCV qui :

- 1- charge une image contenant un visage.
- 2- plaque le maillage du modèle 3D sur l'image.



Ici, le maillage est plaqué au bon endroit et adapté à la forme du visage !!

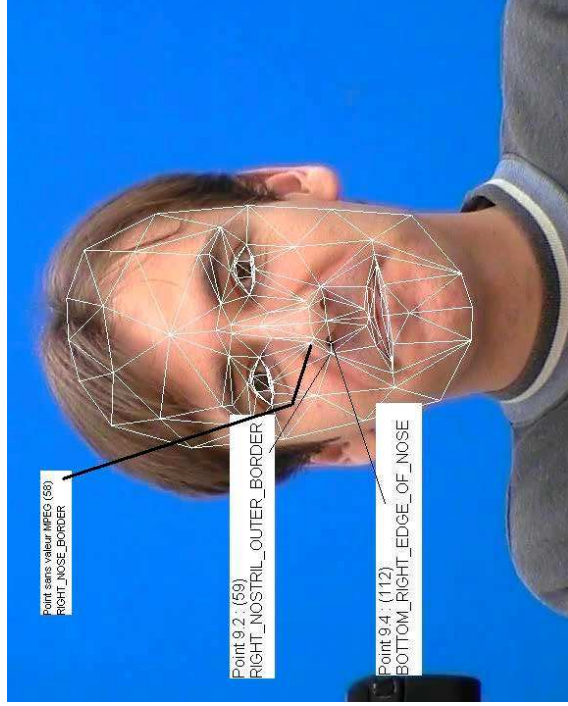
# 8- Construction de modèle 3D visage en VRML + openCV

Ce que l'on veut :

Écrire une application openCV qui :

- 1- charge une image contenant un visage.
- 2- plaque le maillage 3D sur l'image.

*Fichier fait à la main...*



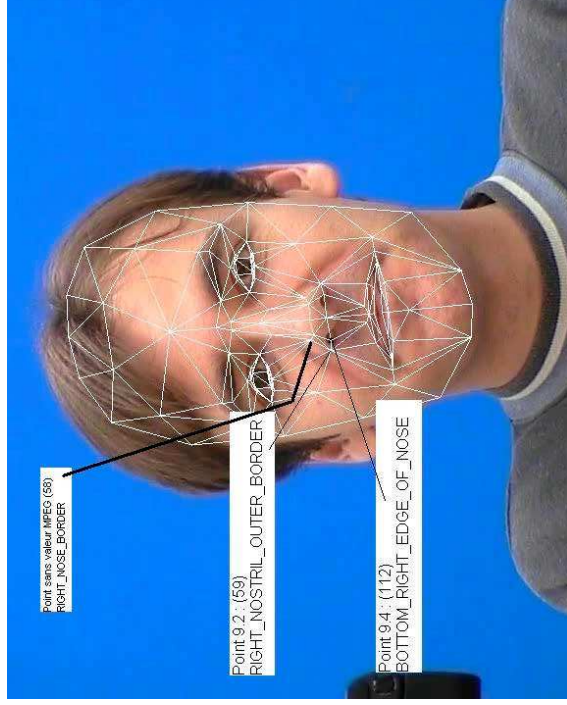
NOSE_TIP	384.744 305.778
OUTER_CORNER_OF_RIGHT_EYEBROW	275 252
INNER_CORNER_OF_RIGHT_EYEBROW	351 218
UPPERMOST_POINT_OF_RIGHT_EYEBROW	306 216
LOWER_POINT_OF_RIGHT_EYEBROW	314 232
OUTER_CORNER_OF_LEFT_EYEBROW	485 218
INNER_CORNER_OF_LEFT_EYEBROW	404 213
UPPERMOST_POINT_OF_LEFT_EYEBROW	454 195
LOWER_POINT_OF_LEFT_EYEBROW	448 212
OUTER_CORNER_OF_RIGHT_EYE	299.112 270.123
INNER_CORNER_OF_RIGHT_EYE	351.344 258.943

# 8- Construction de modèle 3D visage en VRML + openCV

Ce que l'on veut :

Écrire une application openCV qui :

- 1- charge une image contenant un visage.
- 2- plaque le maillage du modèle 3D sur l'image.



On veut que l'utilisateur de l'application puisse facilement positionner le maillage

## 8- Construction de modèle 3D visage en VRML + openCV

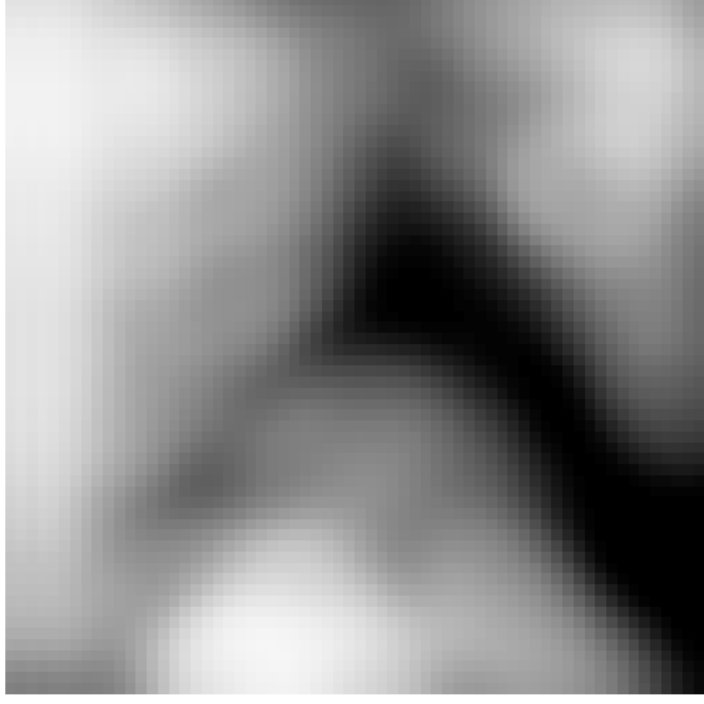
**Ce que l'on veut :**

Écrire une application openCV qui :

- 1- Charge une image contenant un visage.
- 2- Plaqué le maillage du modèle 3D sur l'image.
- 3- Sauvegarde le modèle 3D VRML texturé.

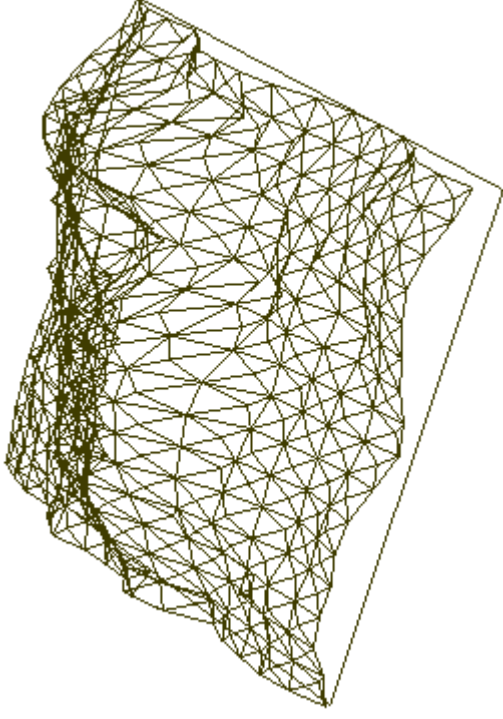
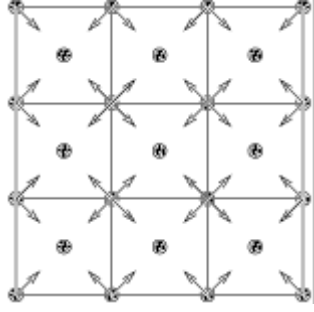
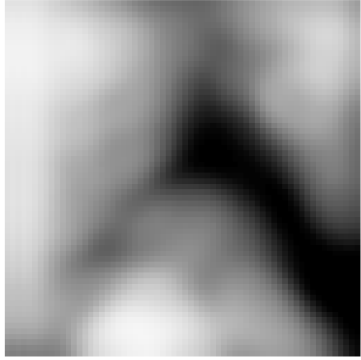
# 9- Visualisation 3D de Modèles Numériques de Terrains (MNT) en temps réel

Données initiales :  
carte de texture + altitudes



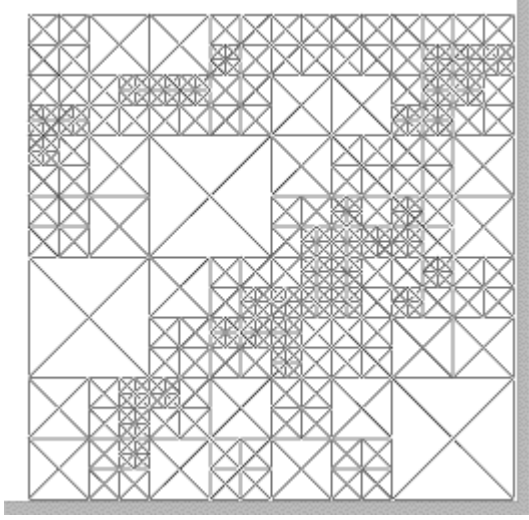
# 9- Visualisation 3D de Modèles Numériques de Terrains (MNT) en temps réel

Construction de la surface 3D



# 9- Visualisation 3D de Modèles Numériques de Terrains (MNT) en temps réel

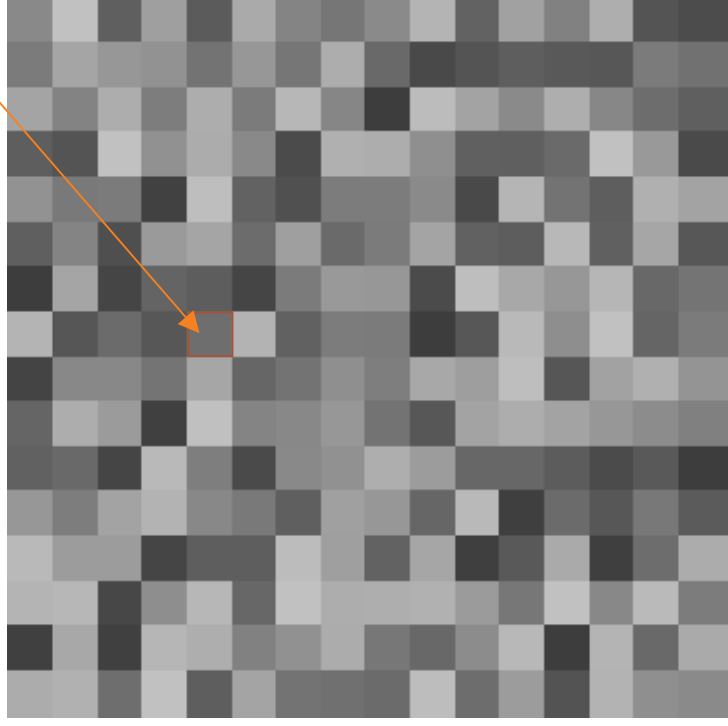
Optimisation à partir d'un algorithme basé sur un quadtree



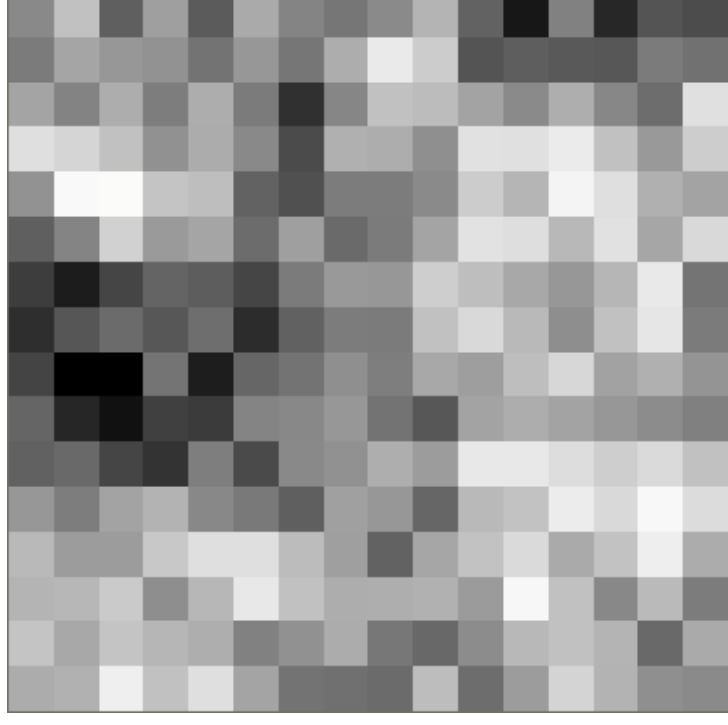
# 10- Régularisation d'une carte et modification d'une image

**Objectif 1 :** Partant d'une carte (damier = matrice de valeurs), construire la carte (damier) la plus « lisse » visuellement.

site  $i$  dont la valeur  $K_i \in [-64, 64]$



*Carte originale*



*Carte après régularisation*



# 10- Régularisation d'une carte et modification d'une image

Pour chaque site  $i$  on doit choisir entre  
la valeur  $K_i$  ou bien la valeur complémentaire  $K'_i$

site  $i$  dont la valeur  $K_i \in [-64, 64]$

42	42	47	-18	64	-34	34	0	16	64	58	-33	8	7	-14	-31	-7	29	-47	-52	53	-11	3	3
-16	-16	42	44	29	30	16	22	14	-5	23	-22	-5	46	5	43	-11	-11						
-21	-21	-10	45	-30	21	45	-13	-3	-7	21	-6	-5	44	-64	-23	9	9						
59	59	-24	47	36	-26	26	-40	38	-64	-52	34	9	14	58	-54	50	50						
-19	-19	11	25	-63	55	-25	33	28	-40	60	-31	-54	-32	33	-44	-30	-30						
26	26	54	-10	-39	-63	-25	44	60	55	38	-35	51	-33	9	-34	31	31						
-45	-45	-64	64	40	-21	-36	33	-42	13	21	54	-13	-22	45	-39	0	0						
49	49	50	7	-63	-40	-52	21	32	64	52	-32	-34	64	6	-41	45	45						
14	14	-23	56	-19	-9	-39	11	46	-27	-24	36	46	23	-19	-6	-43	-43						
9	9	41	-5	42	-37	-64	-1	19	-6	-12	-40	33	-53	-33	-15	-51	-51						
9	9	41	-5	42	-37	-64	-1	19	-6	-12	-40	33	-53	-33	-15	-51	-51						

$$K'_i = K_i + 128 \text{ si } K_i < 0$$

$$K'_i = K_i - 128 \text{ si } K_i > 0$$

$$K'_i = K_i \text{ ou } K_i - 128 \text{ ou } K_i + 128 \text{ si } K_i = 0$$

**CHOIX**

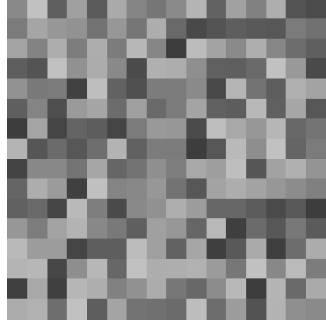


*Carte régularisée*

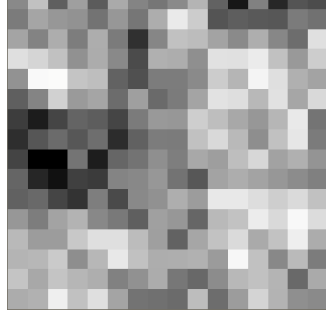
*Carte originale*

# 10- Régularisation d'une carte et modification d'une image

**Objectif 1 :** Partant d'une carte (damier) = matrice de valeurs construire la carte (damier) la plus « lisse » visuellement.



*Carte originale*



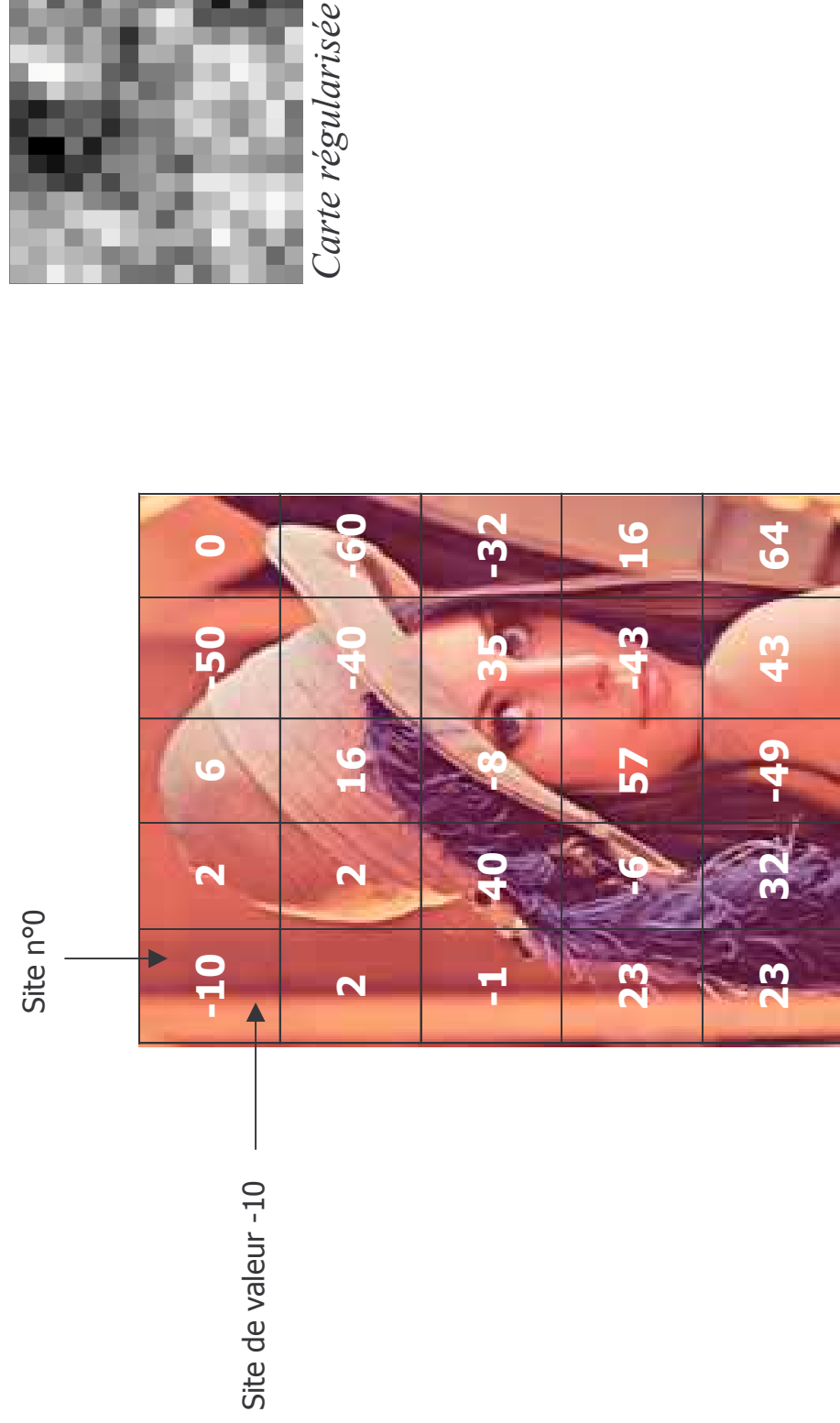
*Carte régularisée*

Technique : modélisation par un automate cellulaire  
(Idée similaire au jeu de la vie).

Règle de transition fonction de l'écart-type local

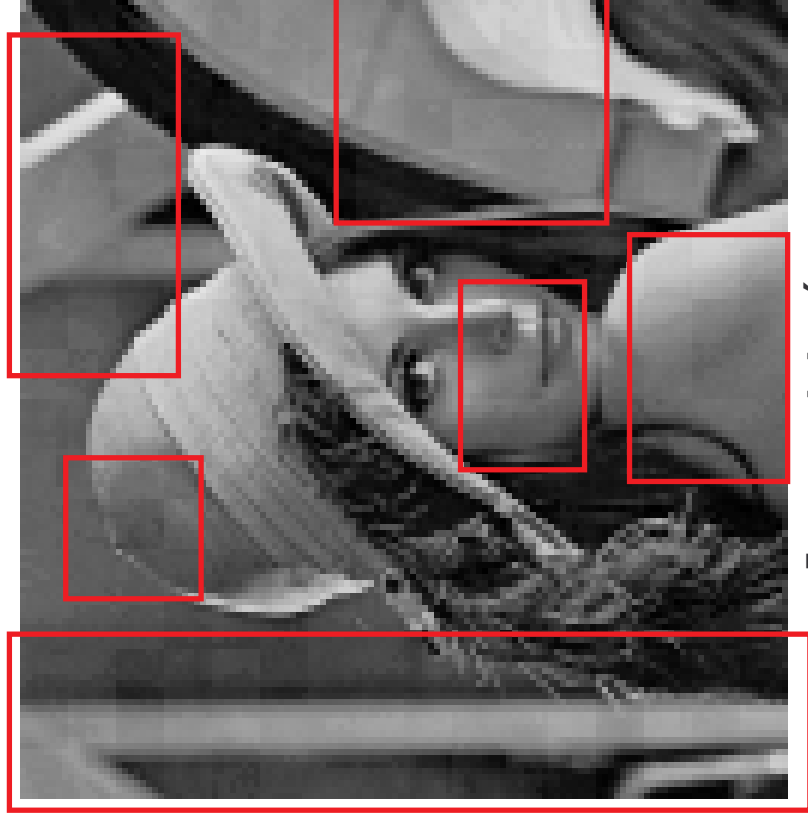
# 10- Régularisation d'une carte et modification d'une image

**Objectif 2 :** « Tatouer » une image à partir de la carte régularisée et ceci sans « effets de blocs ».

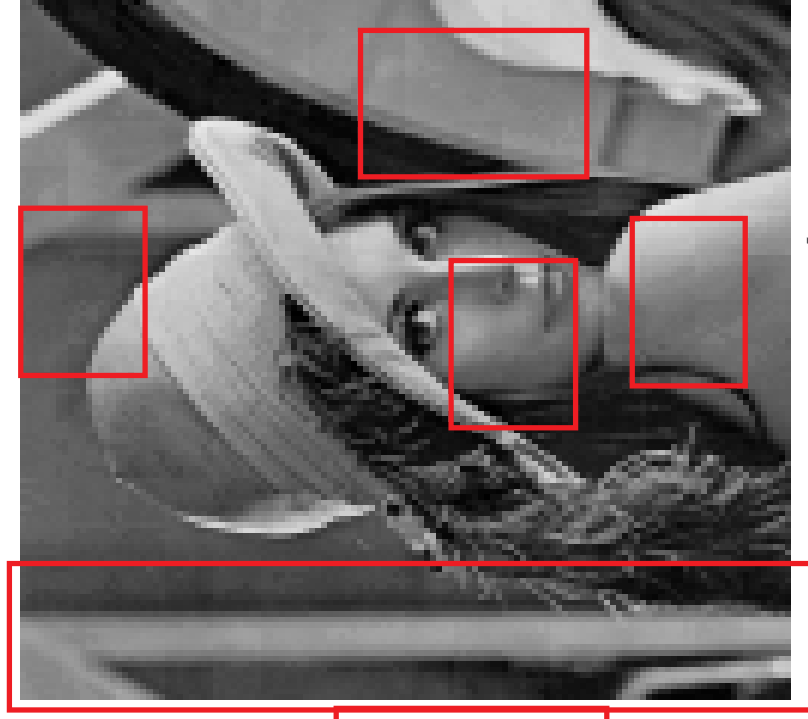


# 10- Régularisation d'une carte et modification d'une image

**Objectif 2 :** « Tatouer » une image à partir de la carte régularisée et ceci sans « effets de blocs ».



**Image « tatouée »  
à partir de la carte originale**



**Image « tatouée »  
à partir de la carte régularisée**

# 10- Régularisation d'une carte et modification d'une image

**Objectif 2 :** « Tatouer » une image à partir de la carte régularisée et ceci sans « effets de blocs ».

Technique : Choix des pixels à modifier dans un bloc :  
modification en priorité des pixels de plus fort  
ecart-type (les plus éloignées de la moyenne).

**Objectif 3 :** Mettre en place le schéma complet de tatouage. (A partir d'une image et d'un message, insérer de manière invisible le message dans l'image).

# 11- Mise en place d'un cours/td/tp

## MAPLE sur le traitement d'image

(public L2 MI MIP)

Mise en place du cours :

- Définition, représentation binaire, chargement d'un fichier en Maple,
- Illustration de quelques filtrage simples et explications,

Mise en place d'un TD :

- Mise en place du TD : exercices autour des concepts math-info.

Mise en place du TP :

- Support de cours existant pour java; cours à adapter...

Autres traitements possibles sous Maple :

- tatouage LSB, changement d'espace couleur, utilisation de librairie existante ?

# 12- Mise en place d'un cours/td/tp

## MAPLE sur la cryptographie

### (public L2 MI MIP)

Mise en place du cours :

- Léger historique de la crypto,
- Illustration des premières techniques (avec implémentation Maple et démonstration),
- Explication du système de cryptographie RSA (Rivest, Shamir et Adleman); explication des théorèmes (voir des démonstrations).

Mise en place d'un TD :

- Mise en place du TD : exercices autour des concepts math-info.
- Mise en place d'un TP.

Mise en place du TP :

- Remarque : de nombreux exemples et explications existent sur le web.  
<http://damien.faucillon.free.fr/tipe/>

Exemple de Cours Maple sous power-point disponible.

# 13- Pilotage d'une caméra Pan Tilt par USB

## Contexte de vision active

- Localiser un objet
- Le suivre dans une image
- Déplacer la caméra en fonction du mouvement de l'objet

## Outils

- Une caméra (Sweex WC040 Motion Tracking Webcam )
- Librairie OpenCV (outils de traitement d'image adaptés : pas besoin de les réécrire)

## Problématique forte

- Pas de drivers accessibles pour l'USB de la caméra
- Mise en œuvre de cette liaison pour la commande





# 14- Localisation 3D en vision stéréoscopique

## Contexte vision stéréoscopique

- Détermination des paramètres d'une mire 3D
- Calibrage de la paire de caméras
- Estimation des erreurs du modèle

## Outils

- Paires d'images Stéréo pour débiter
- 2 WEBCAM puis 2 Caméras couleur (Teli TOSHIBA)
- Librairie OpenCV :

*Traitement des images*

*Calibrage*

## Problématique forte

- Calibrage de chaque caméra
- Estimation de la liaison entre les 2 capteurs
- Robustesse du protocole

