

Traitements d'images et logiciel libre

Annexes au cours

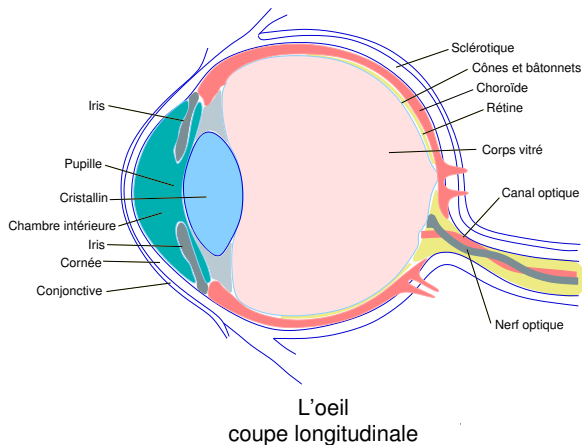
Deuxième Journée Méditerranéenne des Logiciels Libres

Olivier Lecarme

École Polytechnique Universitaire

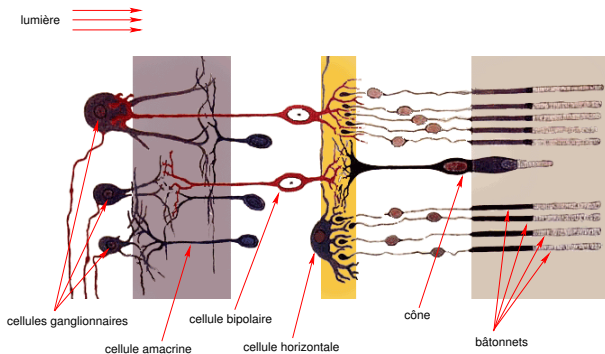
Samedi 6 mai 2006

Mécanismes de la vision



L'image se forme à travers le **système optique de l'œil**, sur sa partie arrière, la **rétine**.

Structure de la rétine

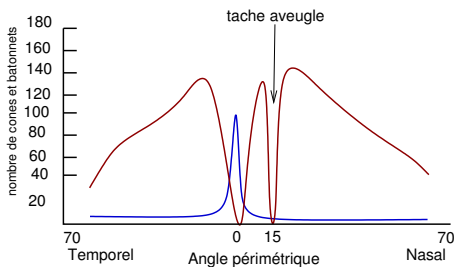
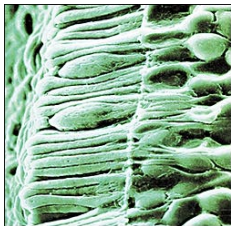


La lumière arrive **par la gauche de la figure**, côté nerveux des cellules. Les cellules réceptrices sont différenciées en **bâtonnets** et **cônes**. Les autres cellules transforment le signal reçu.

Cônes et bâtonnets

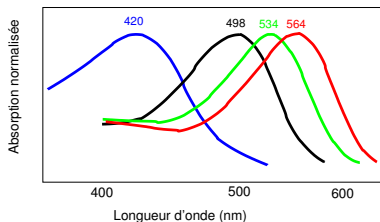
- les **bâtonnets** :
 - sont 120 à 150 millions
 - ils sont très sensibles à la luminosité
 - ils ne distinguent pas les longueurs d'ondes
- les **cônes** :
 - sont 6 à 7 millions
 - ont besoin d'une lumière suffisante
 - sont sensibles aux couleurs
 - donc on ne distingue pas les couleurs la nuit

Cônes et bâtonnets (suite)



- les cônes sont surtout au **centre de la rétine (fovéa)**
- c'est le cerveau qui **reconstitue les couleurs** dans les côtés de l'image
- c'est lui aussi qui **reconstitue l'image manquante** sur la **tache aveugle**

Vision des couleurs



- il y a **trois sortes** de cônes :
 - sensibles au **bleu** (autour de 420 nm, 1/50)
 - sensibles au **vert** (autour de 534 nm, 2/5)
 - sensibles au **rouge** (autour de 654 nm, 4/5)
- les **daltoniens** manquent de certains cônes

Longueurs d'onde

- la **couleur pure** est déterminée par la **longueur d'onde** du rayonnement électro-magnétique :
 - 420 nm \Rightarrow violet
 - 470 nm \Rightarrow bleu
 - 530 nm \Rightarrow vert
 - 580 nm \Rightarrow jaune
 - 600 nm \Rightarrow orangé
 - 650 nm \Rightarrow rouge
- le **spectre visible** va de 400 nm à 780
- la plupart des **couleurs réelles** ne sont **pas monochromatiques**

Problèmes de représentation

- le **codage brut** de l'image donne un résultat **extrêmement encombrant** :
 - avec un **appareil de photo numérique** par exemple, une image représente **10 à 20 millions d'octets** (ou plus)
 - c'est **beaucoup** pour le **stockage**, **trop** pour les **échanges d'information**
 - il faut donc **compresser**
- la **définition** occasionne également de **grandes difficultés** :
 - **70 à 100 pixels par pouce** sur un écran (28 à 40 points/cm)
 - **300 à 2400 pixels par pouce** sur une imprimante (120 à 1000 points/cm)

Principes de compression

- deux grandes méthodes de compression :
- Compression sans perte d'information :
 - l'objectif est de pouvoir reconstruire entièrement l'image initiale
 - il faut donc qu'il y ait beaucoup de redondance pour permettre une représentation plus compacte
- Compression avec perte d'information :
 - on abandonne l'idée de reconstruction
 - on accepte d'interpoler des pixels manquants entre les pixels représentés

Compression par indexage

- on tient compte de la **redondance naturelle** des informations
 - soit **dans un texte** un caractère qui apparaît **100 fois de suite**
 - il occupe donc **100 octets**
 - on peut le représenter par un **caractère spécial**, suivi d'un **nombre d'apparitions** et du caractère à répéter
 - l'encombrement est alors de **3 octets**
- les **techniques courantes** utilisent un **codage de Huffman** ou dérivé :
 - les éléments sont codés en fonction de leur **fréquence d'apparition**
 - les plus fréquents sont **codés sur quelques bits**

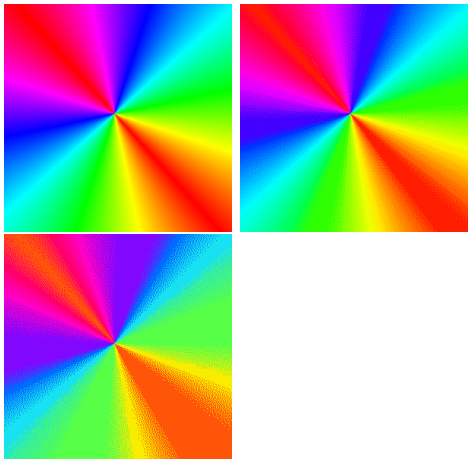
Principe de l'indexage

- dans un texte il y a **moins de 200 caractères différents**
- la **représentation ordinaire des pixels** permet d'en avoir **plus de 16 millions différents**
- dans une **image de 4 millions de pixels** il est donc normal qu'il n'y en ait **pas deux identiques**
- il n'y a donc **aucune redondance**
- mais l'œil est incapable de **distinguer des couleurs trop voisines**
- on choisit donc de **limiter le nombre de pixels distincts**

Fonctionnement de l'indexage

- on construit une **table finie de couleurs distinctes**
- chaque pixel de l'image est **ramené à la couleur la plus proche** dans la table
- chaque pixel est représenté par l'**indice de sa couleur dans la table**
- si la table fait **256 couleurs**, la taille de la représentation est **divisée par 3**
- comme on a **introduit beaucoup de redondance**, on peut utiliser une **compression sans perte d'information**
- si la **table est plus petite**, le gain est **encore plus grand**

Inconvénients de l'indexage

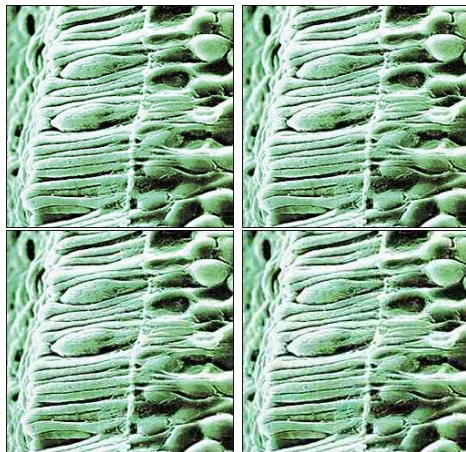


- pour obtenir une **redondance d'information** (nombreux pixels identiques) on accepte de **perdre de l'information** sur la couleur
- le problème est très apparent dans les **dégradés** (transformation de **transition**)
- il n'est pas gênant dans les **images schématiques**

Compression par interpolation

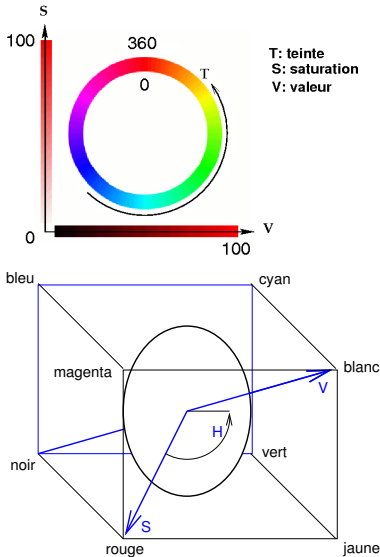
- dans un dégradé par exemple, il est inutile de représenter toutes les couleurs si on est capable de les reconstituer
- la compression par interpolation suppose donc qu'on peut reconstituer à l'affichage les pixels perdus
- c'est l'idée de l'interpolation
- on réduit donc de manière importante le nombre de pixels représentés
- en ajustant le degré de compression on peut choisir le compromis entre qualité et encombrement

Exemple de taux de compression



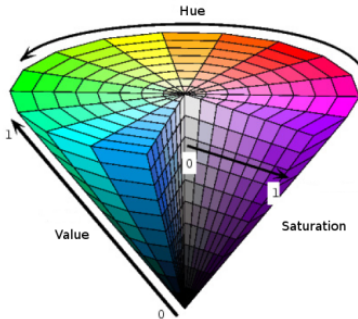
taux	taille (Ko)
85%	28
50%	17
30%	12
20%	9,4

Correspondance entre RGB et HSV



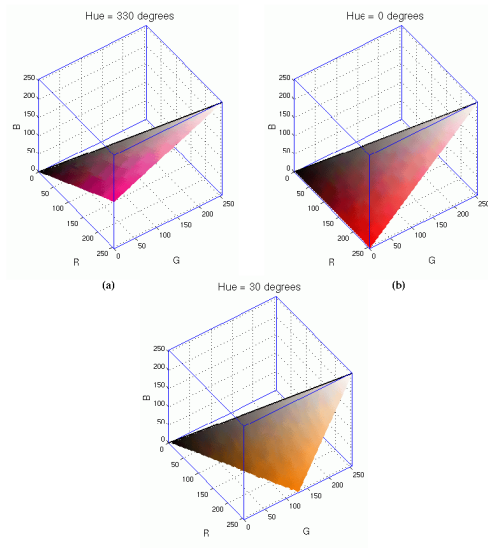
- le modèle HSV
- sa place dans le cube RGB

Le cône HSV



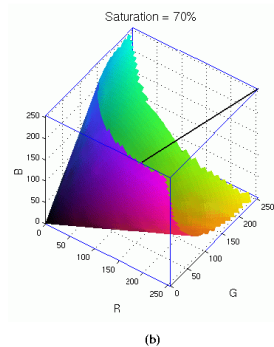
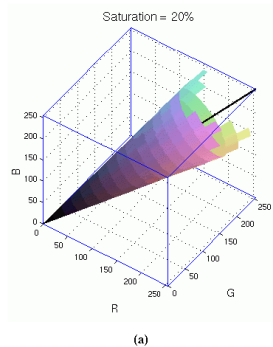
- le cône HSV est découpé dans le cube RGB
- il a pour axe la diagonale des gris

Interprétation de H dans le cube RGB



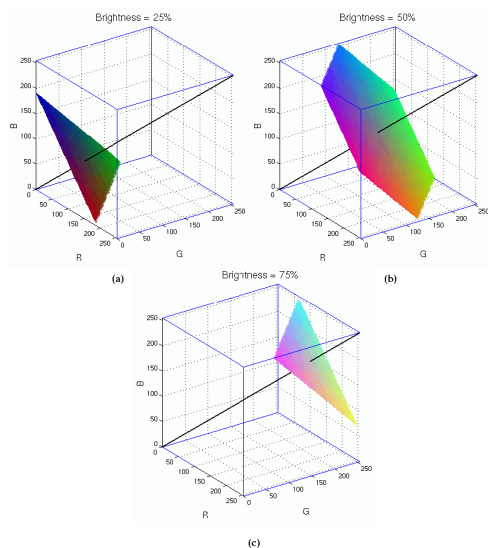
la teinte est la même sur
un demi-plan passant
par la diagonale des gris

Interprétation de S dans le cube RGB



la saturation est la
même sur un cône ayant
pour axe la diagonale
des gris et son sommet
sur le noir

Interprétation de V dans le cube RGB



la valeur est la même sur un plan perpendiculaire à la diagonale des gris

Représentation de la luminance



- en chaque pixel on prend la **moyenne pondérée des valeurs** des canaux RGB :
$$L = 0.3R + 0.59G + 0.11B$$
- c'est une représentation de la **luminance** de l'image
- on l'obtient par le choix
Image : Image \Rightarrow Mode
 \Rightarrow Niveaux de gris

Représentation de la clarté



- en chaque pixel on met à zéro le composant de saturation
- c'est une représentation de la *clarté* de l'image, obtenue par la formule
$$C = (\max(R, G, B) + \min(R, G, B))/2$$
- on l'obtient par le choix
Image : Calque \Rightarrow Couleur \Rightarrow Désaturer
- l'image reste en RGB, mais les trois canaux RGB ont la même valeur

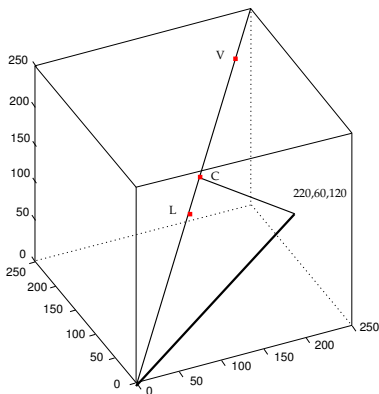
Représentation de la valeur



- en chaque pixel on ne conserve que le **composant de valeur du modèle HSV**
- c'est le **maximum des valeurs** des trois canaux RGB
- on l'obtient par le choix
Image : Filtres \Rightarrow Couleurs
 \Rightarrow Décomposer en choisissant le modèle TSV
- on obtient **trois calques**, le supérieur étant le **composant de valeur**

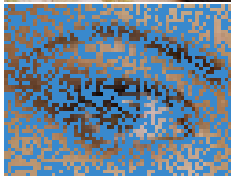
Explication des trois mécanismes

Projections sur la diagonale des gris



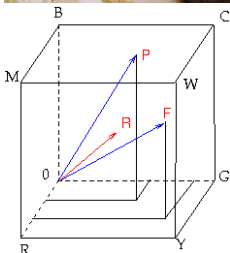
- les **trois points rouges** sont les représentants des trois transformations sur la **diagonale des gris**
- la **clarté C** est la projection du point sur cette diagonale

Mode Dissoudre



- le calque supérieur **se dissout** dans l'inférieur
- un **pourcentage égal à la transparence** des pixels inférieurs sont vus **à la place** des supérieurs
- ils sont **répartis de manière aléatoire**

Mode Multiplier



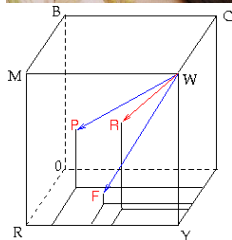
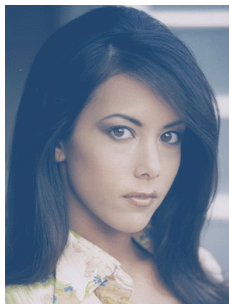
- on **multiplie terme à terme** les composants RGB des deux pixels, et on **normalise dans l'intervalle [0..255]** en divisant par 255
- on peut noter cela $R = \frac{F \times P}{255}$, où F est le **pixel de fond**, P le **pixel de premier plan** et R le **pixel résultant**
- le résultat est d'**assombrir** de la manière indiquée par le premier plan

Mode Diviser



- on **divise terme à terme** les composants RGB des deux pixels, et on **normalise dans l'intervalle $[0..255]$** en multipliant par 256
- pour éviter une division par zéro, on **ajoute 1 au pixel diviseur**
- enfin, on **tronque ce qui dépasse du cube RGB**
- on peut noter cela $R = \min(W, F \div \frac{P+1}{256})$, où W est le blanc pur
- le résultat est d'**éclaircir en tirant vers le blanc**

Mode Écran



- on **inverse** le cube RGB, on **multiplie** terme à terme et on **inverse** à nouveau, avec normalisation
- on peut noter cela

$$R = W - \frac{(W-P) \times (W-F)}{255}$$
- le résultat est d'**éclaircir** de la même manière que la multiplication assombrit

Mode Superposer



- ce mode est une **combinaison** des modes **multiplier** et **écran**
- on peut noter cela
$$R = \frac{F \times R_e + (1-F) \times R_m}{255}$$
, où R_e est le pixel calculé par le mode écran, et R_m le pixel calculé en mode multiplier
- le résultat est **proche du mode multiplier** si le pixel inférieur est **sombre**, et du **mode écran** s'il est **clair**

Mode Éclaircir



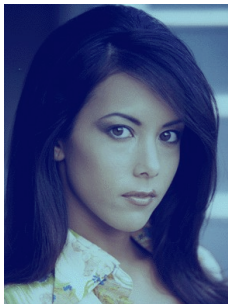
- on **inverse** le cube RGB, on **divise** et on **inverse** à nouveau
- on peut noter cela $R = W - \min(0, \frac{W-F}{W-P+1} \times 256)$
- le résultat est d'**éclaircir le pixel supérieur**, comme dans une **photographie sur-exposée**

Mode Assombrir



- on **inverse** le cube RGB, on **multiplie** et on **inverse** à nouveau
- on peut noter cela $R = W - \max(W, \frac{(W-P) \times (W-F)}{255})$
- le résultat est d'**assombrir le pixel supérieur**, comme dans une **photographie sous-exposée**

Mode Lumière dure



- on combine les **modes Écran et Multiplier**
- on peut noter cela ... je ne sais pas comment !
- le résultat est d'**appliquer fortement la lumière portée** par le pixel supérieur

Mode Lumière douce



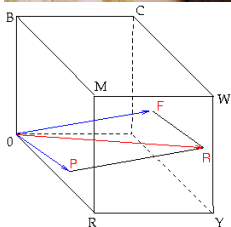
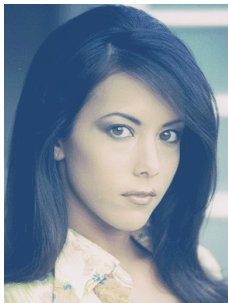
- ce mode n'est pas décrit dans la documentation...
- son résultat est d'**appliquer faiblement la lumière portée** par le pixel supérieur
- en particulier, **les couleurs sont affaiblies**

Modes touchant au grain de l'image



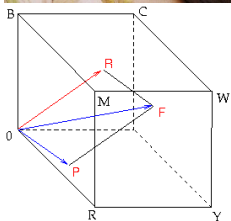
- deux modes touchent au **grain d'une photographie** sur papier
- le mode **Extraction de grain** produit une image qui ne contient que le grain
- le mode **Fusion de grain** utilise une image produite par le mode précédent pour **l'ajouter à une image existante**
- les deux exemples ci-contre ne sont pas du tout parlants...

Mode Addition



- on **ajoute terme à terme** les composants RGB des pixels, en **tronquant aux limites** du cube RGB
- on peut noter cela
 $R = \min(W, P + F)$
- la couleur résultante est toujours **plus claire** que les deux couleurs additionnées
- c'est bien la **somme des couleurs** dans le modèle RGB

Mode Soustraction



- on **soustrait** les composants du pixel supérieur à ceux du pixel inférieur, en **tronquant aux limites** du cube RGB
- on peut noter cela $R = \max(P - F, 0)$
- la couleur résultante est **assombrie** par les pixels supérieurs

Mode Différence



- on prend la **valeur absolue** de la **différence** des pixels
- on peut noter cela
$$R = |P - F|$$
- le résultat est **symétrique par rapport aux deux pixels**, contrairement à la soustraction
- les pixels **complètement noirs** avec la soustraction reviennent vers la **couleur la plus sombre**

Mode Noircir seulement



- on prend la **valeur minimale** du composant supérieur et du composant inférieur
- on peut noter cela
 $R = \min(P, F)$
- le résultat prend chaque fois le **pixel le plus sombre** des deux

Mode Éclaircir seulement



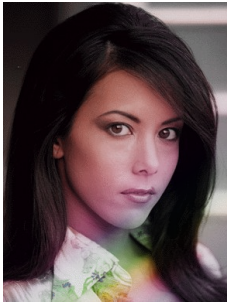
- on prend la **valeur maximale** du composant supérieur et du composant inférieur
- on peut noter cela
 $R = \max(P, F)$
- le résultat prend chaque fois le **pixel le plus clair** des deux

Modes du modèle HSV



- pour ces quatre modes, le principe est le même :
 - l'un des composants du modèle HSV est pris dans le pixel supérieur
 - les deux autres sont pris dans le pixel inférieur
 - pour les exemples suivants, le calque supérieur est celui qui apparaît à gauche

Modes Teinte et Saturation



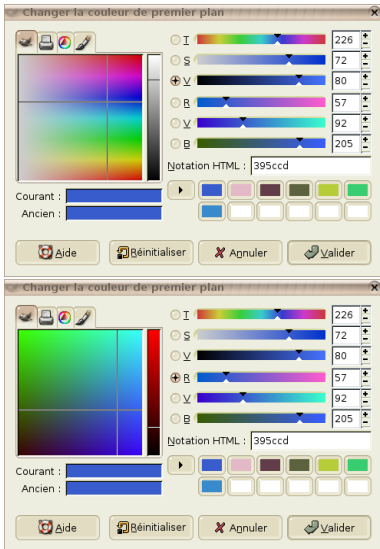
- le mode **Teinte** permet de **colorier** l'image avec les pixels supérieurs
- le mode **Saturation** est plus difficile à justifier
- dans le dernier exemple, on a utilisé la saturation du portrait pour modifier le dégradé

Modes Valeur et Couleur



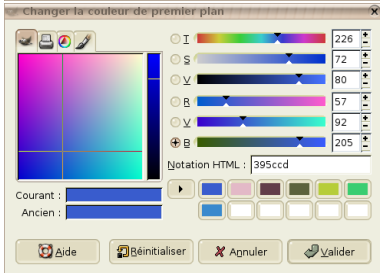
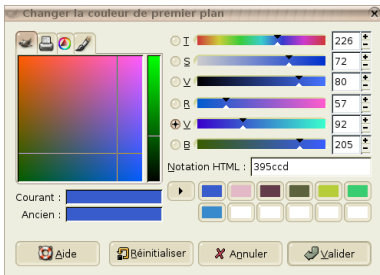
- le mode **Valeur** est aussi difficile à justifier que le mode **Saturation**
- le mode **Couleur** combine la **teinte** et la **saturation** des pixels supérieurs avec la **clarté** des pixels inférieurs, c'est-à-dire la moyenne entre les valeurs maximale et minimale des trois composants RGB

Outil de couleur, modèle de Gimp (suite)



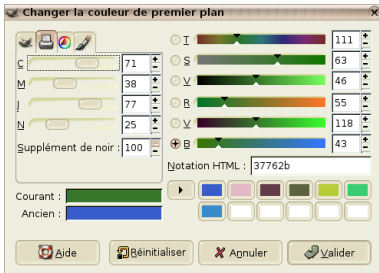
- on a vu les trois cas correspondant au **modèle HSV**
- toujours dans ce premier onglet, on peut choisir les composants du **modèle RGB**
- le rectangle montre les **nuances accessibles** à partir de la valeur de la **couleur directrice**

Outil de couleur, modèle de Gimp (suite)



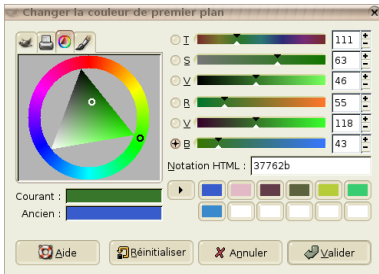
- les six **curseurs de droite** montrent en permanence les valeurs des composants
- ils sont **manœuvrables** individuellement
- on peut ajouter la couleur construite à l'**historique des couleurs**

Outil de couleur, modèle d'impression



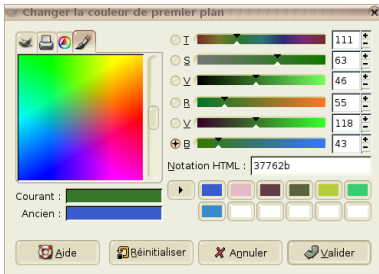
- le deuxième onglet utilise le **modèle CMYK**, spécialisé pour l'impression
- on retrouve les **composants de l'onglet précédent**
- les quatre curseurs des composants CMYK sont **manœuvrables** individuellement
- on peut aussi **réduire la quantité de noir**, qui par défaut est maximale

Outil de couleur, modèle du triangle



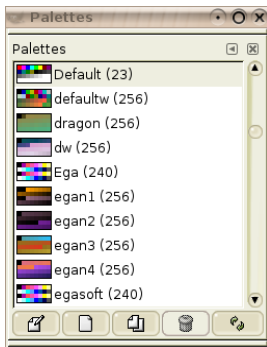
- le troisième onglet est spécialisé pour le **modèle HSV**
- on retrouve les **composants déjà vus**
- le cercle représente la **valeur H**
- le triangle représente la combinaison des **valeurs S et V**

Outil de couleur, modèle d'aquarelle



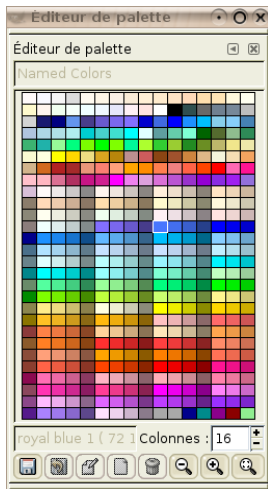
- le quatrième onglet rappelle le **travail de l'aquarelliste**
- on retrouve les **composants déjà vus**
- le carré est la **palette des couleurs complète**
- chaque clic dans la palette **ajoute de la couleur**
- l'emplacement **détermine la teinte**, chaque clic **augmente la saturation et diminue la valeur**
- le curseur vertical permet de régler la **quantité de couleur** prise à chaque clic

Manipulation de palettes



- une *palette* est un ensemble fini de couleurs distinctes
- on peut s'en servir comme outil de choix de couleurs qui respectent certaines contraintes ou qui sont apparentées
- Outils : Fichier \Rightarrow Dialogues \Rightarrow Palettes ouvre le dialogue des palettes
- la liste est longue et extensible
- le nombre de couleurs distinctes est indiqué

Manipulation de palettes (suite)



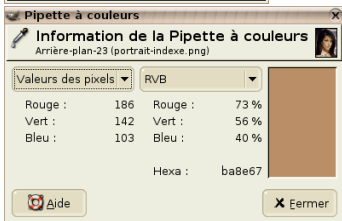
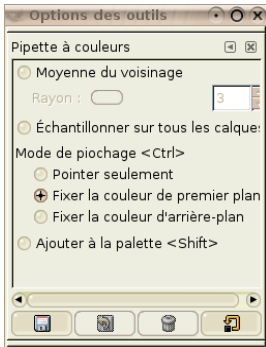
- double-cliquer sur une palette ouvre le dialogue de l'**éditeur de palettes**
- on peut l'utiliser pour :
 - choisir une couleur précise pour le premier ou l'arrière-plan
 - modifier une couleur
 - créer une nouvelle palette
 - on voit ici les **couleurs nommées**

Palette indexée



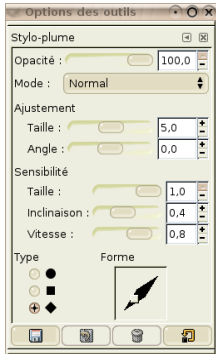
- une **image indexée** utilise une **palette spéciale**
- ici on a demandé la **génération automatique** de la palette
- le résultat permet de retoucher l'image sans ajouter de couleur

L'outil Pipette



- l'outil **pipette à couleurs** permet de **prendre la couleur d'un pixel**
- on peut l'utiliser simplement pour **obtenir les informations**
- on peut aussi s'en servir pour fixer l'une des **couleurs de premier ou arrière-plan**
- pour les outils de tracé, la touche **Ctrl** permet de **choisir la couleur de premier plan** juste avant un tracé

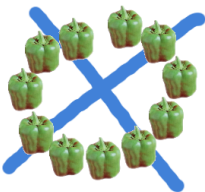
L'outil stylo-plume



- l'outil **stylo-plume** trace lui aussi des traits à **bords nets**
- il n'utilise pas le **dialogue des brosses**
- on peut choisir un **embout de plume**, puis le **déformer** et l'**orienter**
- l'outil permet donc des **effets de calligraphie**, surtout à la tablette graphique

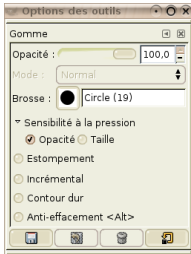


L'outil pinceau (suite)



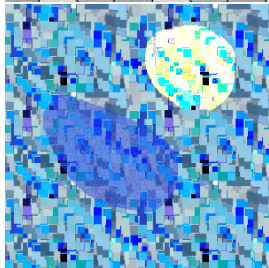
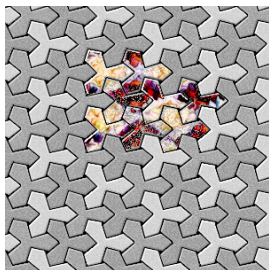
- les outils de tracé, et l'outil pinceau en particulier, ont deux **modes de mélange** supplémentaires, nécessitant tous deux la **transparence** :
 - le mode **derrière** ne peint que sur les **parties transparentes** : ici les poivrons ont été tracés en premier
 - le mode **effacer la couleur** retire du pixel inférieur le composant de couleur de la brosse : si les deux sont de même couleur, c'est l'équivalent de la gomme

L'outil gomme



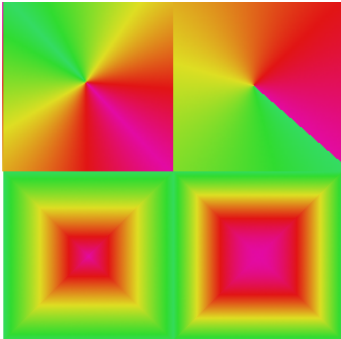
- l'outil **gomme** utilise la **brosse courante** pour **peindre avec la couleur d'arrière-plan**
- si le calque le permet, l'outil **rend transparentes** les parties effacées
- les utilisations restent peu nombreuses

L'outil de remplissage (suite)



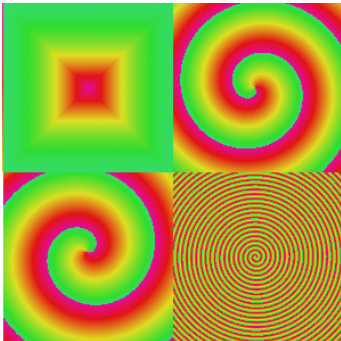
- les **motifs** sont très variés et l'ensemble est extensible
- la plupart ont la propriété de **pavage du plan**
- on peut aussi **remplir une sélection**
- les **modes de mélange** enrichissent les possibilités
- la **semi-transparence** aussi

L'outil de dégradé (suite)



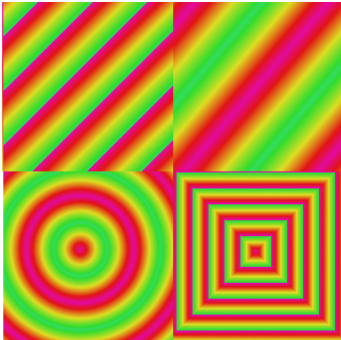
- formes suivantes de dégradé
 - **cônique (symétrique)** : le point de départ est le sommet du cône, la direction suivie place la couleur d'arrière-plan
 - **cônique (asymétrique)** : la direction suit l'angle entre la partie éclairée et la partie obscure
 - **suivant la forme (angulaire)** : le mouvement de la souris est sans signification
 - **suivant la forme (sphérique)** : même chose

L'outil de dégradé (suite)



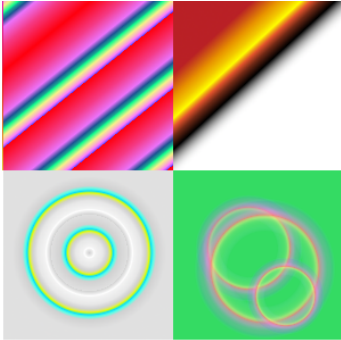
- formes suivantes de dégradé
 - suivant la forme (excroissances) : comme les deux précédents
 - spiral (sens anti-horaire) : le mouvement de la souris donne le sens et le pas de la spirale
 - spiral (sens horaire) : même chose
 - un mouvement très réduit produit des effets intéressants

L'outil de dégradé (suite)



- pour les dégradés autres que cône et suivant la forme on peut **répéter le dégradé** :
 - la répétition **en dents de scie** reprennent chaque fois le dégradé dans le même sens
 - la répétition **en onde triangulaire** le font alternativement dans un sens et dans l'autre
 - le mouvement initial détermine le **pas de la répétition**

L'outil de dégradé (fin)



- le **dialogue des dégradés** en propose une grande quantité
- certains sont plus faits pour le **mode radial**
- certains utilisent la **transparence** et permettent les **superpositions**
- on peut **construire un nouveau dégradé** grâce à l'**éditeur de dégradé**