

# Introduction à MegaWave2

L. Moisan — septembre 2005.

MegaWave2 est un logiciel public développé depuis une dizaine d'années par des chercheurs à des fins d'enseignement et de recherche. C'est à la fois un environnement de développement (surcouche de C) et une bibliothèque d'algorithmes. Dans le cadre des séances de travaux pratiques, nous utiliserons essentiellement cette deuxième fonction. Pour obtenir plus d'informations, consulter en ligne les manuels ou télécharger MegaWave2, voir le site

<http://www.cmla.ens-cachan.fr/~megawave>

## 1 Installation

Cette opération sera décrite lors du premier TP. Elle n'est requise que lors de la première utilisation de MegaWave2.

## 2 Les commandes de MegaWave2

Toute fenêtre de type terminal vous donne accès aux commandes de MegaWave2. Par exemple, la commande

```
cvview lena.img &
```

doit afficher à l'écran l'image de Léna (cette image sert souvent de référence en traitement d'images).

Analysons rapidement la ligne précédente :

- `cvview` est le nom de la commande MegaWave que nous avons utilisé : cette commande permet de visualiser telle quelle une Cimage (image où chaque niveau de gris est codé par un entier entre 0 et 255). **Pour connaître la syntaxe d'une commande MegaWave, il suffit de taper son nom seul.** En tapant `cvview`, on a donc une description de l'utilisation de cette commande. On découvre qu'elle prend une image en argument, et qu'elle accepte plusieurs arguments optionnels. **Pour obtenir une documentation plus précise, tapez `mwdoc` suivi du nom de la commande** (ici, `mwdoc cvview`).

- `lena.img` est bien entendu le nom de l'image que nous avons visualisée. C'est un fichier qui devrait se trouver dans le répertoire courant, mais MegaWave2 étend automatiquement la recherche à tous les sous-répertoires de `$MEGAWAVE2/data` lorsqu'il ne trouve pas une donnée. En tapant

```
ls $MEGAWAVE2/data/tp1
```

vous pouvez constater que ce répertoire contient bien le fichier `lena.img`.

- le caractère `&` en fin de ligne de commande signale que la commande en cours doit être exécutée en tâche de fond (et permettre ainsi à l'utilisateur d'entrer d'autres commandes sans attendre que cette dernière soit terminée). De manière générale, on utilise le caractère `&` en fin de ligne **uniquement pour les commandes de visualisation** (`cvview`, `cmview`, `splot`, etc.). En effet son utilisation lors d'une suite de calculs, en plus d'être inutile, ne garantit pas l'exécution séquentielle des commandes, ce qui peut être source de résultats erronés.

Maintenant que nous avons découvert la syntaxe de `cvview` en entrant le nom de cette commande seul, nous pouvons essayer ses options, par exemple

```
cvview -z 2 lena.img &
```

qui affiche l'image de Léna avec un zoom par duplication d'un facteur 2. Pour changer la méthode de zoom, on pourrait par exemple essayer

```
cvview -z 2 -o 3 lena.img &
```

Comme tous les modules de MegaWave2 dédiés à l'affichage, `cvview` dispose de quelques commandes interactives. Placez le pointeur de la souris dans la fenêtre de `cvview` et tapez `'h'` pour obtenir une description de ces commandes. Essayez ensuite certaines d'entre elles (boutons de la souris, `'l'`, etc.).

Toutes les commandes MegaWave s'utilisent de la même façon : un nom de commande, suivi éventuellement d'options puis d'arguments obligatoires (fichiers en entrée ou en sortie, valeurs numériques). Certaines options nécessitent un argument, d'autres non. Vous pouvez obtenir la liste des commandes en tapant

```
ls $MEGAWAVE2/bin/ix86
```

Pour une commande donnée, en plus de sa syntaxe en ligne (`<commande>`) et de sa documentation (`mwdoc <commande>`), vous pouvez trouver le fichier contenant le code source (C) à l'aide de la commande `mwwhere <commande>.c`. Pour une organisation plus thématique des commandes, vous pouvez aussi explorer les sous-répertoires de `$MEGAWAVE2/src`.

### 3 Quelques types de données dans MegaWave2

MegaWave définit un certain nombre de types de données. Sans les décrire tous en détails, en voici quelques uns parmi les plus importants.

- **Cimage** : une image en niveaux de gris codée comme un tableau d'entiers entre 0 et 255. C'est ce format qui est utilisé par un écran pour la visualisation. L'affichage à l'écran se fait à l'aide de la commande `cvview`, l'affichage de valeurs particulières à l'aide de la commande `cprintasc`.

- **Fimage** : une image codée comme un tableau de réels (float). L'affichage à l'écran se fait avec la commande `fvview`, **qui ramène la dynamique de l'image entre 0 et 255**. L'examen des valeurs est possible soit avec `fvview` (bouton gauche), soit avec la commande `fprintasc`. Essayez

```
fderiv -x gradient lena.img
cvview gradient &
fvview gradient &
```

puis expliquez les différences obtenues (utiliser le bouton de gauche de la souris dans chaque fenêtre).

- `Ccimage` : image couleur (R,V,B) codée en 3 tableaux d'entiers entre 0 et 255. Visualisation avec `ccview`. Essayez par exemple

```
ccview ccimage &  
cview ccimage &
```

Les trois plans couleur peuvent être décomposés/recomposés grâce aux commandes `cfgetchannels` et `cfputchannels`. A titre d'exercice, échangez les canaux rouges et bleus de `ccimage`.

- `Cmovie` : un film constitué de plusieurs `Cimage`. Un fichier `Cmovie` est un fichier texte contenant le nombre d'images et la liste des fichiers images. Visualisation avec `cmview`. Essayez par exemple

```
cmview -l cmovie &
```

et testez l'effet des touches 'h', '+', '-', espace, 'f', et 'b'.

- `Fsignal` : un signal constitué d'un tableau monodimensionnel de valeurs. Un fichier `Fsignal` est un fichier texte contenant quelques paramètres (nombre d'échantillons, échelle, ...) et la liste des valeurs. La visualisation d'un `Fsignal` se fait grâce à la commande `splot`. Essayez par exemple

```
splot fsignal &
```

puis testez les commandes interactives de `splot` (tapez 'h' pour en avoir la liste).

Lors de l'utilisation des commandes `MegaWave2`, certaines conversions de type se font automatiquement. Par exemple, si vous utilisez la commande `cview` avec une `Fimage`, les valeurs de cette `Fimage` seront automatiquement tronquées (et non rééchelonnées comme avec `fvview`) en des entiers entre 0 et 255. Réciproquement, l'utilisation de la commande `fvview` avec un `Cimage` ne pose pas de problème.

Les formats de fichiers sont généralement spécifiques à `MegaWave`, mais certains formats d'images standard (GIF, TIFF, etc.) sont reconnus automatiquement en lecture. Il est également possible de forcer le type et le format de fichier en sortie d'un module en utilisant l'option `-ftype`. Exemple :

```
ccopy -ftype TIFF lena.img lena.tif
```

L'utilisation d'un type inconnu entraîne l'affichage des types/formats possibles.