



# Projet Logiciel en C Décodeur JPEG

L. Allemand-Giorgis, F. Bouchez, F. Broquedis,  
**M. Chabanas**, T. Dias Alves, JF Méhaut

28 mai 2015

```
ff d8 ff e0 00 10 4a 46
49 46 00 01 01 01 00 60
00 60 00 00 ff fe 00 12
53 68 61 75 6e 20 74 68
65 20 73 68 65 65 70 00
ff db 00 43 00 0a 07 07
09 07 06 0a 09 08 09 0b
0b 0a 0c 0f 19 10 0f 0e
0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28
2d 39 30 28 2a 36 2b 22
23 32 44 32 36 3b 3d ...
```

```
ff d8 ff e0 00 10 4a 46
49 46 00 01 01 01 00 60
00 60 00 00 ff fe 00 12
53 68 61 75 6e 20 74 68
65 20 73 68 65 65 70 00
ff db 00 43 00 0a 07 07
09 07 06 0a 09 08 09 0b
0b 0a 0c 0f 19 10 0f 0e
0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28
2d 39 30 28 2a 36 2b 22
23 32 44 32 36 3b 3d ...
```



```
ff d8 ff e0 00 10 4a 46
49 46 00 01 01 01 00 60
00 60 00 00 ff fe 00 12
53 68 61 75 6e 20 74 68
65 20 73 68 65 65 70 00
ff db 00 43 00 0a 07 07
09 07 06 0a 09 08 09 0b
0b 0a 0c 0f 19 10 0f 0e
0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28
2d 39 30 28 2a 36 2b 22
23 32 44 32 36 3b 3d ...
```

■ 12 jours





# Vos profs des 15 prochains jours...



Léo Allemand-Giorgis



Florent Bouchez



François Broquedis



Matthieu Chabanas



Thomas Dias Alves



Jean-François Méhaut

- Présenter succinctement les objectifs du projet C en général
- Expliquer le déroulement du projet pour le sujet JPEG
- Répondre aux questions éventuelles liées au projet ou au sujet

Apprentissage d'un langage de référence : le C

Mise en pratique des connaissances acquises dans d'autres modules

- Algo/Prog, Conception et exploitation des processeurs, Théorie de l'information, etc.
- Outils de développement : `gdb`, `ddd`, `valgrind`, `gprof`, etc.

Premier aperçu des contraintes d'un "gros" projet logiciel

- Travail en équipe : répartition des tâches, mise en commun
- Gestion du temps et de la difficulté des différentes étapes
- Respect d'un cahier des charges
- Importance de la phase de validation (tests, debug, ...)
- Opportunité d'utiliser des outils de gestion de versions (Git, SVN)



Amphi de présentation et lancement du projet : **Jeudi 28 mai**

Libre-service encadré : **du vendredi 29 mai au mercredi 10 juin**

- 15 demies journées d'encadrement + 3 demies journées d'autonomie
- Salles machines 1er étage (E101, E102, **E103**, E104)
- 2 ou 3 encadrants JPEG par demie journée (pas le week-end !, NO SPAM)

Date de rendu : **mercredi 10 juin, 19h**

Soutenances : **jeudi 11 juin après midi et vendredi 12 matin**

## A quoi sert un encadrant ?

- Les encadrants sont là pour :
  - vous aider sur la programmation en C en général
  - vous aider sur les points techniques du projet
  - vous conseiller sur l'aspect conception (comment structurer son projet)
  - vous apporter des conseils sur l'organisation (le travail en équipe n'apporte pas que des avantages !)
- Les encadrants NE SONT PAS LÀ pour :
  - faire le projet à votre place
  - vous donner les solutions à tous les problèmes
  - remplacer les logiciels de debug
  - faire la hotline 24h/24, 7j/7
- Dans tous les cas, réfléchissez à vos questions avant de les poser et discutez avec vos binômes, souvent la réponse est dans la question, et soyez précis

## Petit rappel sur la fraude

- Il est interdit de copier ou de s'inspirer de fichiers concernant le projet, à l'exception des fichiers fournis par les encadrants. Ceci inclus :
  - les fichiers des années précédentes
  - les fichiers d'étudiants n'appartenant pas au trinôme
- De même, il est interdit de transmettre des fichiers à des étudiants extérieurs au trinôme. Ceci implique que vous protégez vos fichiers en lecture (contrôle effectué périodiquement). Des informations sur la sécurisation des fichiers sont disponibles sur le Wiki
- Des outils de détection automatique de triche sont utilisés
- En cas de fraude avérée, la sanction est le 0 au projet, en plus des sanctions prévues dans le règlement de l'école

- L'évaluation du projet compte pour 80% de la note finale du Projet Logiciel en C (les autres 20% étant le cours de préparation au C)
- Votre projet sera évalué lors de la soutenance de 30 minutes des 11 et 12 juin. Tout le trinôme assiste à la soutenance
- La soutenance se déroule comme suit
  1. 10 minutes de démonstration du code produit  
→ ne pas prévoir une démonstration, mais plusieurs qui démontrent les atouts de votre programme final
  2. les 20 dernières minutes sont consacrées aux questions et tests de l'examineur.



- *Joint Photographic Experts Group* : comité de normalisation pour la compression d'images
- A l'origine de la norme ISO/IEC IS 10918-1 — ITU-T Recommendation T.81, plus communément appelé JPEG d'après le nom du comité
- Norme très complexe, proposant de nombreux algorithmes de compression
- Plusieurs formats d'échange supportent le JPEG, le JFIF (JPEG File Interchange Format) est le plus courant

- Format incontournable pour le stockage d'image, malgré son âge respectable (début des années 90)
- Puissance requise pour l'encodage/décodage très faible  
→ pratique à utiliser dans les appareils photos numériques par exemple, internet, ...

## Image brute non compressée

- $n \times m$  pixels en RGB

## Compression des données de l'image

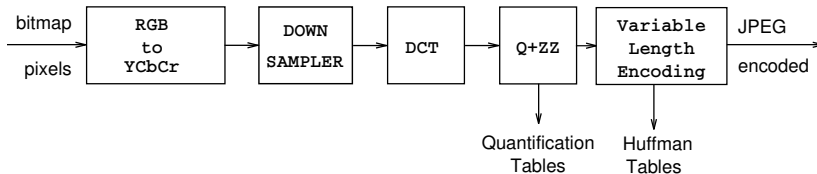
- Changement de représentation : RGB  $\rightarrow$  YCbCr
- Sous échantillonnage de certaines composantes
- Changement de domaine : spatial  $\rightarrow$  fréquentiel (DCT)
- Quantification des valeurs

## Stockage efficace de l'information

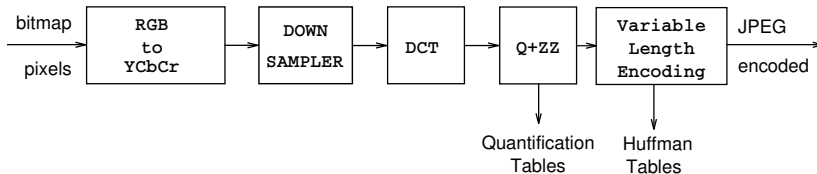
- Codage de Huffman
- Codage RLE : répétition de valeurs (0)
- Flux bit à bit
- Format de fichier JFIF : marqueurs, données



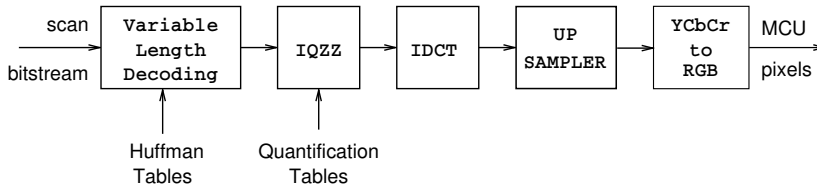
## Encodage



## Encodage



## Décodage



# Objectifs techniques du projet JPEG

## Décodeur

- Réaliser intégralement, en C, un décodeur d'images JPEG
- Format de fichier: JFIF uniquement
- Mode `baseline`, 3 types de sous-échantillonnage
- Écrire l'image résultante dans un format TIFF non compressé

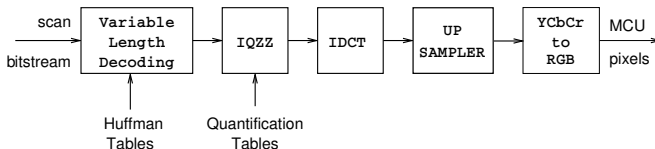
## Extensions

- Optimisations
- Réaliser un encodeur à partir du décodeur
- ...



## Découpage en modules

Le décodeur est découpé en **modules**, de difficulté hétérogène, réalisant des parties "atomiques" du décodage



```
bitstream.o huffman.o unpack.o iqzz.o idct.o upsampler.o  
conv.o tiff.o main.o
```

## Fourni :

Un fichier objet par module, sans les sources, et un Makefile pour la construction de l'application

## Travail minimum: décodeur complet

- Implanter TOUS les modules du décodeur JPEG
- En respectant l'API fournie
- Validation

## Extensions

- Optimiser le décodeur (iDCT efficace - algorithme fourni)
- Réaliser un encodeur à partir du décodeur
- Des tas d'autres choses possibles selon vos envies...

## Fichiers à rendre (TEIDE)

Une unique archive contenant :

- Vos fichiers sources \*.c des différents modules
- Un Makefile suffisant pour compiler et nettoyer l'application
- Des fichiers tests dans la limite de l'espace disponible sur Teide

Projet prévu sur les machines de l'Ensimag

- Pas d'ordinateurs personnels pendant la soutenance
- Un projet qui ne fonctionne pas à l'Ensimag est sanctionné comme un projet non terminé

Respect du coding style proposé (cf. Wiki)

## Quelques conseils

- Gardez le Makefile à jour tout le long du projet
- Vérifiez régulièrement le fonctionnement à l'Ensimag
- Prévoyez une base de test
- Figez le projet bien avant la date de rendu
- Pour la soutenance :
  - 10/15 mn pour présenter 2 semaines de travail!
  - Pour la partie obligatoire : prouvez nous que ça marche sans insister
  - Pour le reste : cherchez à montrer à quel point vous avez été plus loin. N'hésitez pas à "vendre" votre travail (améliorations des performances, démonstration des avantages, ...)
- La qualité prime sur la quantité !  
Mieux vaut du code lisible, bien conçu et testé que plus de choses non validées...





Bon courage !