

UNIVERSITÉ DE FRIBOURG, SUISSE  
CENTRE DE DIDACTIQUE UNIVERSITAIRE

# RENOUVEAU D'UN COURS UNIVERSITAIRE SUR LE TRAITEMENT DE L'IMAGE NUMÉRIQUE

TRAVAIL DE FIN D'ÉTUDES  
EN VUE DE L'OBTENTION DU DIPLÔME EN ENSEIGNEMENT SUPÉRIEUR ET  
TECHNOLOGIE DE L'ÉDUCATION  
Did@CTIC

Sous la direction du Prof. Bernadette Charlier

**Jean-Luc Bloechle**  
Département d'Informatique, Université de Fribourg

Mai 2008

## Remerciements

S'il fallait résumer mon état d'esprit durant cette formation *Did@cTIC* par un seul mot, je dirais "incertitude". Beaucoup de doutes ont parsemé mon chemin, en particulier au début de la formation. Heureusement, de nombreuses personnes m'ont soutenu durant cette longue période, par leurs collaborations, leurs conseils et leurs encouragements. Ainsi, je tiens à remercier chaleureusement les personnes suivantes :

- Mes collègues assistants : Asmaa El Hannani, Catherine Pugin ainsi que Florian Evéquoiz ont toujours été là pour me soutenir, partager leurs expériences.
- Rolf Ingold, mon directeur de thèse, qui m'a donné le feu vert pour cette formation didactique et ainsi permis de consacrer quelques heures de mon temps hebdomadaire à sa réalisation.
- François Jimenez, compagnon d'étude et ami qui m'a directement incité à suivre la formation *Did@cTIC*, toujours présent lorsque j'avais besoin d'un coup de main.
- Sans oublier Virginie Fornerod, pour avoir attentivement relu et corrigé ce document... et me soutenir au quotidien dans notre vie commune.

Et bien sûr toute l'équipe "présente et passée" de didactique universitaire pour son encadrement et sa disponibilité : Bernadette Charlier, Ana Moura, Annick Rossier-Morel, Marie Lambert et Hervé Platteaux.

A vous tous, MERCI

## Table des matières

1	Introduction.....	4
1.1	Préambule.....	4
2	Mort et renaissance d'un cours.....	5
2.1	TIRF 2001, l'odyssée de l'apprentissage.....	5
2.1.1	Transparents rétro projetés.....	6
2.1.2	Espace de partage en ligne minimaliste.....	7
2.1.3	Séries d'exercices inadaptées.....	8
2.2	ImaPro 2007, le nouveau cours est arrivé à maturité.....	9
2.2.1	Transparents numériques en couleurs.....	9
2.2.2	Espace de partage en ligne géré par Moodle.....	10
2.2.3	Séries d'exercices pédagogiques.....	12
3	La métamorphose.....	13
3.1	Renouveau des transparents .....	13
3.2	Migration vers une nouvelle philosophie de partage internet.....	14
3.2.1	Site internet dynamique en PHP.....	14
3.2.2	Moodle, plateforme de partage internationale.....	17
3.3	Un framework Java pour une nouvelle philosophie d'apprentissage.....	19
3.3.1	Existe-t-il réellement un besoin sous-jacent ?.....	20
3.3.2	Trois ans de développement itératif.....	21
3.3.3	ImaPro, un support à l'apprentissage.....	23
4	Quelques exemples concrets en images.....	30
4.1	Opérations sur des images en noir/blanc.....	30
4.2	Opérations sur des images en niveaux de gris.....	31
4.3	Opérations sur des images en couleurs.....	32
5	Conclusions.....	35
5.1	Quelques parallèles avec les modèles vus aux cours.....	35
5.2	Le mot de la fin.....	37
6	Bibliographie.....	38

# 1 Introduction

Ma formation *Did@cTIC* arrive aujourd'hui à son terme. J'ai suivi cette formation durant trois années consécutives, en parallèle de mon poste d'assistant-doctorant dans le groupe de recherche *DIVA* du département d'informatique de l'université de Fribourg.

Lors de mon engagement comme assistant-doctorant, mon professeur, également directeur de notre groupe de recherche, m'a directement affecté au cours *TIRF : traitement d'image et reconnaissance de formes*. Ce cours était à l'époque un cours annuel divisé en deux parties logiques : au premier semestre le sujet *traitement d'image* était abordé, tandis qu'au deuxième le cours exposait les méthodes de *reconnaissance de formes*. Lors de mon engagement le système universitaire était en total changement. Une réforme de fond était en marche afin d'homogénéiser les cursus universitaires en suivant le processus de Bologne. Le cours annuel *TIRF* a dès lors été scindé en deux modules semestriels totalement indépendants : *ImaPro* pour *Image Processing* (traitement d'image) et *PatRec* pour *Pattern Recognition* (reconnaissance de formes). Ces cours d'approfondissement sont dispensés chaque année aux étudiants suivant le cursus de master en informatique à l'université de Fribourg.

Suite à mon engagement, mon professeur a profité d'utiliser cette nouvelle main d'oeuvre à disposition pour renouveler en profondeur ses deux cours d'approfondissement. Ce travail de diplôme va mettre en relief les étapes de ce renouveau, les choix pédagogiques effectués ainsi que les nouvelles technologies introduites.

## 1.1 Préambule

Avant de débiter la rédaction de ce mémoire, je tiens à clarifier la sémantique du mot didactique. En effet, la terminologie de ce mot a toujours été vague dans mon esprit, je pense qu'il est grand temps d'assouvir cette curiosité. Pour ce faire, je suis allé à la pêche aux informations virtuelles et suis tombé sur cette définition [WIK07a] :

"La didactique, terminologie essentiellement francophone, est l'étude des questions posées par l'enseignement et l'acquisition des connaissances dans les différentes disciplines scolaires. Se sont ainsi développées depuis une trentaine d'années environ, des didactiques des mathématiques, des sciences, du français, des langues, etc."

La rédaction de ce diplôme n'est que la partie émergée de l'iceberg. Un travail de longue haleine a été effectué durant ces trois dernières années afin de renouveler le cours *traitement d'image*. Ma formation didactique en parallèle a grandement profité au renouvellement de ce cours, tant au niveau pédagogique qu'au niveau technologique pur. Encore un point important à relever avant d'entrer dans le vif du sujet... Ma formation scientifique influence fortement ma façon de rédiger, je n'ai pas pour habitude d'écrire des épitaphes "sans queue ni tête", je privilégie naturellement des phrases précises et concises. De plus, lorsque cela est possible, je préfère exposer mes idées grâce à des schémas ou des images. En résumé, voici un proverbe bien connu attribué à Confucius [WIK07b] :

"A picture is worth a thousand words."

## 2 Mort et renaissance d'un cours

"L'informatique est un domaine récent et dynamique, qui poursuit son développement au rythme de l'évolution rapide de la technologie et des avancées théoriques de la science de l'information. L'informatique est une technologie très récente en constante évolution" [UNI06a]. Cet extrait tiré du site internet du département d'informatique de l'université de Fribourg expose bien la réalité à laquelle j'étais confronté lors de mon engagement en tant qu'assistant-doctorant. Le cours *traitement d'image et reconnaissance de formes*, enseigné par mon directeur de thèse, existait déjà depuis quelques années, malheureusement aucune mise à jour n'avait été faite jusqu'alors. La puissance des ordinateurs ainsi que les équipements technologiques des universités en général ont fait un bond gigantesque en une décennie. Depuis maintenant quelques années, une transition, non sans mal, s'effectue au niveau des cours universitaires : le passage des versions papiers à celles numériques des cours. Cela implique un engagement non négligeable de la part des professeurs et assistants qui ont à leurs charges la mise en place de ces changements. Ce renouveau est d'autant plus accentué dans le domaine des études en informatique, qui plus est en traitement de l'image, où l'enseignement peut profiter au maximum des dernières technologies, et de ce fait, devenir hautement interactif. Cette composante orientée vers une pédagogie repensée était tout simplement irréalisable il y a de cela seulement quelques années : technologie de projection numérique inexistante, plateformes d'enseignement en ligne balbutiantes, puissance des ordinateurs insuffisante pour développer des cours informatiques interactifs orientés vers la pédagogie.

Les sous-sections suivantes vont présenter la première partie du cours *TIRF* tel qu'il était en 2001, puis le même cours depuis lors renommé *ImaPro* pour *Image Processing* tel qu'il est aujourd'hui. Quant à la prochaine section, elle apportera un éclairage détaillé sur les acteurs de cette transition. L'utilisation réfléchies des TICs (acronyme de technologies de l'information et de la communication) dans un milieu éducatif sera également mise en valeur (cf. module B4 du cursus Did@cTIC [CHA06b]).

### 2.1 TIRF 2001, l'odyssée de l'apprentissage

En 2001, alors que j'arrivais à la fin de ma formation universitaire, j'ai suivi le cours annuel *TIRF*, acronyme de *traitement d'image et reconnaissance de formes*. Le professeur Rolf Ingold, superviseur du cours, y enseignait des méthodes de traitement d'image au premier semestre puis de reconnaissance de formes au deuxième. Pour ses tâches d'enseignement, le professeur était secondé par un assistant expert dans le domaine, mon prédécesseur.

Je me souviens encore distinctement du premier cours donné dans cette petite salle du bâtiment de physique de l'université de Fribourg, au fond du boulevard de Pérolles. Nous étions une dizaine d'étudiants, dispersés dans un auditoire d'une quarantaine de places. Le professeur avait commencé son cours par quelques points administratifs, puis il avait présenté une introduction au traitement d'image sur des transparents plastiques rétro projetés. Son assistant avait ensuite pris le relais pour notre première séance d'exercices.

### 2.1.1 Transparents rétro projetés

De nos jours, chaque auditoire de Pérolles 2 est équipé d'un projecteur numérique. Cela n'était pas du tout le cas dans les anciennes salles de cours, il y a seulement quelques années de cela. Les transparents de mon professeur étaient alors en noir et blanc, édités sur un traitement de texte usuel. Un cours sur l'image numérique présente un certain nombre de méthodes de traitements informatiques appliquées à des images en couleur. Des transparents monochromes ne permettent pas d'exposer intuitivement des concepts liés à la représentation des images numériques en couleur. Ci-dessous, voici trois transparents tirés du cours *TIRF*.

**PERCEPTION D'IMAGES**

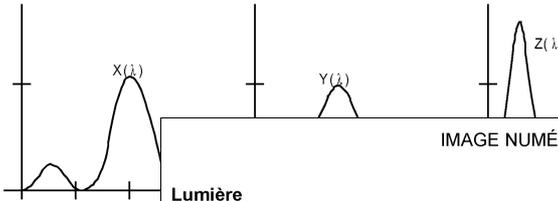
**Espace XYZ**

La CIE a défini un espace abstrait, plus pratique pour la manipulation des couleurs.

Le système XYZ est obtenu à partir de RGB par une transformation linéaire:

$$X = 2.77 R + 1.75 G + 1.13 B$$

$$Y = 1.00 R + 4.59 G + 0.06 B$$

$$Z = 0.00 R + 0.05 G + 5.59 B$$


IIUF - Prof. R.Ingold

**IMAGE NUMÉRIQUE**

**Lumière**

La lumière est la forme visible d'un rayonnement électromagnétique.

- elle se propage dans l'espace,
- dans des directions déterminées;
- elle varie en fonction du temps.

La partie visible du spectre électromagnétique correspond à des longueurs d'ondes comprises entre 380 et 760 nanomètres.



La perception des couleurs

- la lumière émise par
- la lumière réfléchi

IIUF - Prof. R.Ingold

**REPRÉSENTATION DES COULEURS**

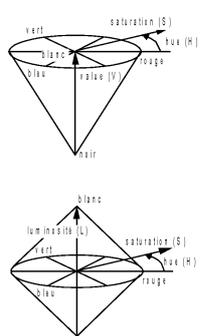
**Modèles HSV et HSL**

Le modèle HLV représente la couleur au moyen d'un triplet représentant respectivement:

- la teinte (hue)
- la saturation (saturation)
- la valeur (value)

Le modèle HLS représente la couleur au moyen d'un triplet décrivant respectivement:

- la teinte (hue)
- la luminosité (lightness)
- la saturation (saturation)



IIUF - Prof. R.Ingold      Traitement d'image      20.03.02 - 40/43

On constate que ces transparents présentent la théorie des espaces de couleur... sans couleur ! Pas facile pour un étudiant de se représenter divers dégradés mentalement, cela uniquement en interprétant des mots clefs tels que "bleu", "rouge", "vert".

## 2.1.2 Espace de partage en ligne minimaliste

A l'époque, le site internet du cours *TIRF* était très sommaire. Une page internet statique affichait certaines informations relatives au cours, secondée par un répertoire accessible à distance qui énumérait une liste de documents : une archive contenant les transparents du cours, une liste de séries distribuées lors des séances d'exercices et encore un document *PDF* d'introduction à un programme de retouche d'image nommé *GIMP*. Ci-dessous, deux captures d'écran présentent le site internet du cours *TIRF* tel qu'il était en 2001. Sur la première capture d'écran on constate que l'information disponible était réduite à son minimum : aucun élément concernant le contenu du cours ou les sujets abordés n'y figure. La deuxième capture d'écran n'affiche rien d'autre qu'un répertoire contenant des fichiers et des dossiers en vrac.

The top screenshot shows a Mozilla Firefox browser window displaying the main page of the TIRF course. The page title is "TIRF.html - Mozilla Firefox" and the URL is "http://diuf.unifr.ch/courses/01-02/tirf/Tirf.html". The page content includes the Department of Informatics logo, the course title "Traitement d'images et Reconnaissance de Formes", the year "Année 2001-2002, Prof. R. Ingold", and a brief description of the course. Below the description, there are sections for "Encadrement:" (Responsible: Prof. Rolf Ingold, Assistant: Karim HADJAR) and "Informations générales:" (Course location and time). At the bottom, there is a link for "Documentation en ligne: Séries d'exercices" and the name "KHADJAR, IIUF".

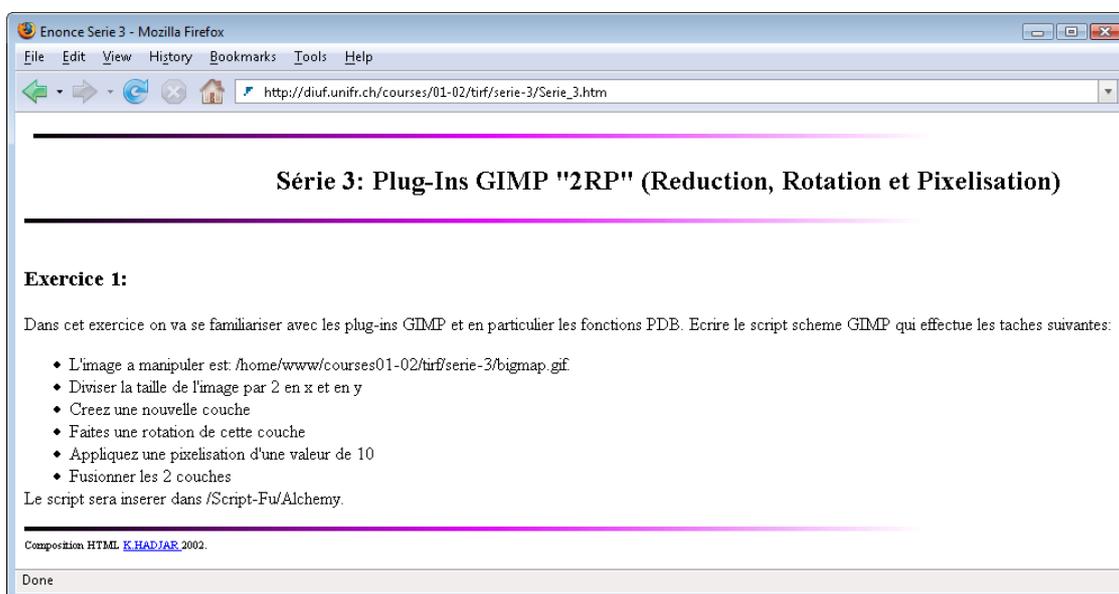
The bottom screenshot shows a directory listing for "Index of /courses/01-02/tirf/". The listing includes the following files and folders:

Name	Last modified	Size	Description
Parent Directory		-	
Intro GIMP.pdf	22-Mar-2002 17:15	55K	
Tirf.html	16-Apr-2002 15:50	3.2K	
TranspTIRF2002.zip	04-Jul-2002 12:20	4.0M	
logo-iuf-small.gif	16-Apr-2002 15:50	252	
purple line.gif	16-Apr-2002 15:51	2.4K	
purple line.jpg	16-Apr-2002 15:51	854	
serie-1/	15-Apr-2002 10:07	-	
serie-2/	15-Apr-2002 10:05	-	
serie-3/	01-May-2002 14:09	-	
serie-4/	08-May-2002 11:27	-	
serie-5/	16-May-2002 15:23	-	
serie-6/	15-May-2002 11:17	-	
serie-7/	22-May-2002 10:20	-	
serie-8/	29-May-2002 10:25	-	
serie-9/	04-Jun-2002 14:44	-	
serie-10.directory	10-Jun-2002 18:10	365	
serie-10/	10-Jun-2002 18:09	-	

Pour résumer, l'espace en-ligne du cours était simplement composé d'une page *HTML* statique ainsi que d'un répertoire sur un serveur distant accessible de l'extérieur.

### 2.1.3 Séries d'exercices inadaptées

Les séries d'exercices du cours *TIRF* étaient toutes disponibles sur le site internet au format *HTML*. A cette époque, la donnée des séries était toujours très succincte, la présentation ainsi que la correction des séries par l'assistant durant les séances d'exercices étaient superficielles, 10 à 15 minutes au maximum. Aucune correction "pédagogique" n'était distribuée ni même mise à disposition sur le site internet, seuls des fichiers de code informatique non commentés étaient parfois déposés sur le serveur, en guise de solution. De plus, les premières séries se contentaient de donner une approche de l'utilisation de *GIMP*, un programme de traitement d'image, alors que ce cours était justement censé nous enseigner comment créer les méthodes de traitement d'image implémentées dans un tel programme informatique. Les séries plus avancées demandaient ensuite de créer des scripts pour l'environnement de traitement d'image *GIMP* (cf. série d'exercices ci-dessous).



Le programme *GIMP* avait été choisi parce qu'il permettait de développer des algorithmes de traitement d'image dans un environnement déjà existant, ceci avait certainement le mérite d'épargner une quantité de travail considérable à l'assistant. Cependant, implémenter un script *GIMP* est une tâche complexe, il faut au préalable implémenter toute une panoplie de fonctions obscures afin de pouvoir coder les quelques lignes utiles. De plus le langage de programmation utilisé est le langage *C*, un langage vieillissant laissé de côté aujourd'hui au profit d'autres langages résolument orientés objets, bien plus aboutis, propres et performants.

Je tiens encore à relever un point non négligeable me concernant particulièrement en tant qu'étudiant : lorsque je suivais ce cours, il y a 5 ans de cela, je me souviens que rares étaient les fois où des étudiants rendaient leur série d'exercices terminée et fonctionnelle. Pour ma part je ne crois pas avoir rendu une seule fois une implémentation utilisable d'un script *GIMP*. En effet, le but des séries d'exercices était d'apprendre à programmer des méthodes de traitement d'image, alors que nous passions la majorité de notre temps à essayer de faire fonctionner ces scripts incompréhensibles. L'assistant ne donnait aucune explication, ni lors de la distribution des séries d'exercices, ni lors de leurs corrections, bref, nous étions seuls face à de nombreuses difficultés.

## 2.2 ImaPro 2007, le nouveau cours est arrivé à maturité

Dans cette sous-section, je vais présenter le cours *ImaPro* au jour d'aujourd'hui. Je vais également relever les évolutions et bien sûr, comparer les vecteurs d'information du cours actuel avec ceux du cours de 2001. Quant à l'étude de l'évolution du cours de sa version "antique" à sa version "moderne", celle-ci n'apparaîtra que dans la section 3 de ce travail de diplôme.

### 2.2.1 Transparents numériques en couleurs

Et oui, le cours *ImaPro* est une version totalement réécrite de l'ancien cours *TIRF*. *ImaPro* est destiné aux étudiants en master de l'université de Fribourg. Selon les accords de Bologne, les étudiants universitaires en bachelor/master peuvent maintenant diviser leur cursus entre diverses universités. Les supports de cours ont ainsi été traduits en anglais afin de permettre à un maximum d'étudiants d'effectuer un cursus multi-universitaire.

#### XYZ Colour Space

- CIE ("Commission Internationale d'Eclairage") has defined three new hypothetical light sources, x, y, and z which yield positive matching curves:

$$X = 0.49 R + 0.31 G + 0.20 B$$

$$Y = 0.17697 R + 0.8124 G + 0.01063 B$$

$$Z = 0.01 G + 0.99 B$$

- The

#### Visible Light

- Visible light consists of electromagnetic energy having
  - wavelengths in the range 380 (violet) to 780 nm (red)
  - frequencies in the range 380 THz (red) to 780 THz (violet)

#### HSV Color Model

- The HSV (Hue Saturation Value) color model is often preferred because it is often more natural to think about a color in terms of hue and saturation than in terms of additive or subtractive color

- H (from 0 to 360) represents the color type (such as red, blue, or yellow):
- S (from 0 to 100) represent the "purity" of the color:
- V (from 0 to 100) represent the brightness

Conversions between RGB and HSV will be studied in the exercises ! <sup>1</sup>

Les objectifs du cours n'ont pas changé depuis 2001, seule la manière de les atteindre a été totalement remise en question. Ainsi, le renouvellement du cours de traitement d'image s'est en premier lieu focalisé sur la mise à jour du contenu théorique relatif à ce cours. Ensuite, il s'est également appliqué à la refonte totale de la mise en forme des transparents. En effet, l'aspect de ces transparents a considérablement évolué, la couleur a fait son apparition, les lignes graphiques ont été revues, des séquences animées ou interactives ont été ajoutées, l'esthétique générale des transparents a ainsi été affinée. Les nouveaux documents répondent maintenant aux normes de la nouvelle charte graphique de l'université. Le département d'informatique a également tenu à homogénéiser les présentations numériques. Dans ce sens, un gabarit numérique a été imposé à tous les collaborateurs du département d'informatique.

## 2.2.2 Espace de partage en ligne géré par Moodle

Depuis déjà quelques années, un certain nombre de plateformes technologiques, pour le partage de l'information, ont fait leur apparition sur le web. Moodle en est une parmi d'autres. Cette plateforme a l'avantage d'être performante, libre et surtout utilisée par un large éventail d'utilisateurs à travers le monde. De plus, cette plateforme est mise à disposition et gérée par le centre NTE (Nouvelles Technologies et Enseignement) de l'université de Fribourg, il suffit simplement de demander un compte au centre NTE pour qu'un nouveau cours Moodle soit créé !

The screenshot shows a web browser window displaying a Moodle course page. The browser's address bar shows the URL: <http://moodle.unifr.ch/course/view.php?id=1635>. The page content is organized into several sections:

- Encadrement:** Lists the responsible professor (Prof. Rolf Ingold) and assistant (Jean-Luc Bloechle) with their contact information.
- Informations générales:** Provides details about the course schedule, including theoretical and practical sessions, and the target audience (students in the computer science master's program).
- Exercices pratiques:** Describes the practical exercises designed to complement the theoretical approach, mentioning the use of fundamental image processing algorithms.
- Sujets traités:** Lists the topics covered, such as Visual Perception and Digital Image Representation, Point Operators, Local Operators, Introduction to Fourier Transforms, Global Operators, Image Compression, Discrete Geometry, Segmentation, and Morphological Operators.
- Livres conseillés:** Recommends several books on image processing, including 'Digital Image Processing' by Rafael C. Gonzalez and 'The Image Processing Handbook' by John C. Russ.
- Forum des nouvelles:** A section for course news, showing recent posts from September 26 to October 3, including announcements for presentations and frameworks.

The browser's status bar at the bottom indicates 'Done'.

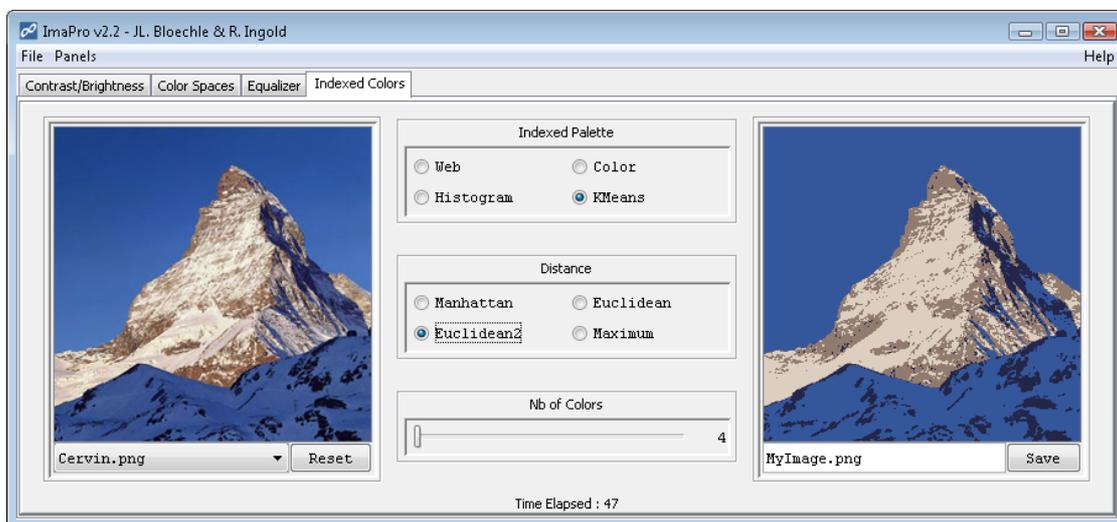


### 2.2.3 Séries d'exercices pédagogiques

Les fondements mêmes du cours *ImaPro* ont été repensés afin de permettre une nouvelle formule résolument orientée vers la pratique. De ce fait, les exercices reflètent concrètement la théorie enseignée au cours, ou alors, d'un autre point de vue : le cours présente toute la théorie nécessaire à la compréhension des exercices traités aux séances pratiques. Les cours théoriques et pratiques sont devenus complémentaires, ce qu'ils étaient d'ailleurs censés être depuis le départ...

Aujourd'hui, les séries d'exercices sont rédigées avec un outil de traitement de texte et mises à disposition en version *PDF* sur la plateforme *Moodle*. Elles sont illustrées, comportent des exemples simples et leurs objectifs sont clairement définis. A chaque série distribuée correspond également une correction détaillée, l'étudiant peut ainsi comprendre ses erreurs en comparant son implémentation avec celle distribuée et commentée par l'assistant.

Voici maintenant l'élément central dans la renaissance du cours *ImaPro* : un framework de traitement d'image a été spécialement développé à l'attention des étudiants (cf. capture d'écran ci-dessous). Concrètement, ce framework est un programme offrant à l'étudiant la possibilité de développer des algorithmes de traitement d'image en ne se focalisant que sur la partie algorithmique du problème. Ceci évite à l'étudiant une perte considérable de temps et d'efforts sur des détails secondaires n'ayant aucun rapport avec le traitement d'image. L'étudiant peut ainsi apprendre et comprendre la programmation des méthodes de traitement d'image dans un environnement résolument pédagogique, l'interface graphique lui permettant de tester interactivement le résultat de son implémentation, d'y déceler les erreurs et de les corriger d'une façon très intuitive .



Le framework est réellement une révolution dans l'enseignement du cours traitement d'image. Il est aussi bien utilisé comme outil de démonstration dans la partie théorique du cours, par le professeur Rolf Ingold, que comme outil de travail dans la partie pratique du cours. Les élèves utilisent l'environnement *ImaPro* pour effectuer leurs séries de programmation, mais ils l'utilisent également précieusement pour tester les méthodes de traitement d'image expliquées au cours. En effet, celles-ci sont déjà toutes disponibles dans le framework. L'interface interactive permet une appréciation précise de l'effet de chaque paramètre d'une méthode de traitement d'image.

### 3 La métamorphose

La transformation du cours *TIRF* en cours *ImaPro* n'a pas eu lieu du jour au lendemain, bien au contraire ! Diverses étapes plus ou moins longues se sont succédées sur trois ans, pour aboutir au résultat actuel. Le cours n'est d'ailleurs jamais réellement terminé, chaque année apportant son lot de changements. Mais, contrairement à l'évolution naturelle du cours, une métamorphose complète a été entreprise ces trois dernières années, un passage obligé pour un cours vieillissant, dont la matière enseignée se base sur une technologie de pointe.

#### 3.1 Renouveau des transparents

Je me souviens toujours de mes débuts en tant qu'assistant-doctorant dans le groupe de recherche *DIVA*, la première tâche qui me fut attribuée par mon superviseur Rolf Ingold, était de transférer l'information contenue dans les transparents de son cours *TIRF* à partir d'un document texte en *Word* vers un format de présentation en *Powerpoint*. Ceci fut la toute première étape du long processus de changement qui était en train de se mettre en place.

Ce travail m'avait paru, à première vue, peu volumineux et de surcroît extrêmement simple. Et bien ce fut un total manque d'appréciation. J'ai finalement passé plus d'un mois à copier puis coller manuellement le texte d'une page à l'autre. Le cours contenait une dizaine de fichiers au format *Word*, chacun de ces fichiers comportait plus de 30 pages. Il fallait à chaque page recréer la mise en forme du texte puis l'adapter au format *Powerpoint*. Les nouvelles pages contenaient moins de texte et la fonte était plus grande. A cause de cela, il était souvent nécessaire de diviser une page d'origine en deux transparents. Beaucoup de formules étaient utilisées dans le cours, malheureusement, le transfert des formules du format *Word* au format *Powerpoint* posait d'importants problèmes de compatibilité entre les versions... Si bien qu'il a fallu rééditer toutes les formules une à une, à la main.

Une fois ce travail effectué, Rolf a repris le flambeau en mettant à jour le contenu même du cours, il a profité de cette occasion pour le traduire en anglais, puis y ajouter de nouvelles illustrations. En réalité, ce travail s'est déroulé sur plus de deux ans : la première année, les transparents étaient écrits 50% en anglais, 50% en français, puis l'année suivante la traduction fut terminée. Il en est de même pour le contenu et les illustrations. D'ailleurs Rolf révisé encore chaque année ses transparents, il en profite pour améliorer leurs contenus ainsi que leurs formes.

REPRÉSENTATION DES COULEURS

**Modèles HSV et HSL**

Le modèle HLV représente la couleur au moyen d'un triplet représentant respectivement:

- la teinte (hue)
- la saturation (saturation)
- la valeur (value)

Le modèle HLS représente la couleur au moyen d'un triplet décrivant respectivement:

- la teinte (hue)
- la luminosité (lightness)
- la saturation (saturation)

IIIUF - Prof. R. Ingold      Traitement d'image

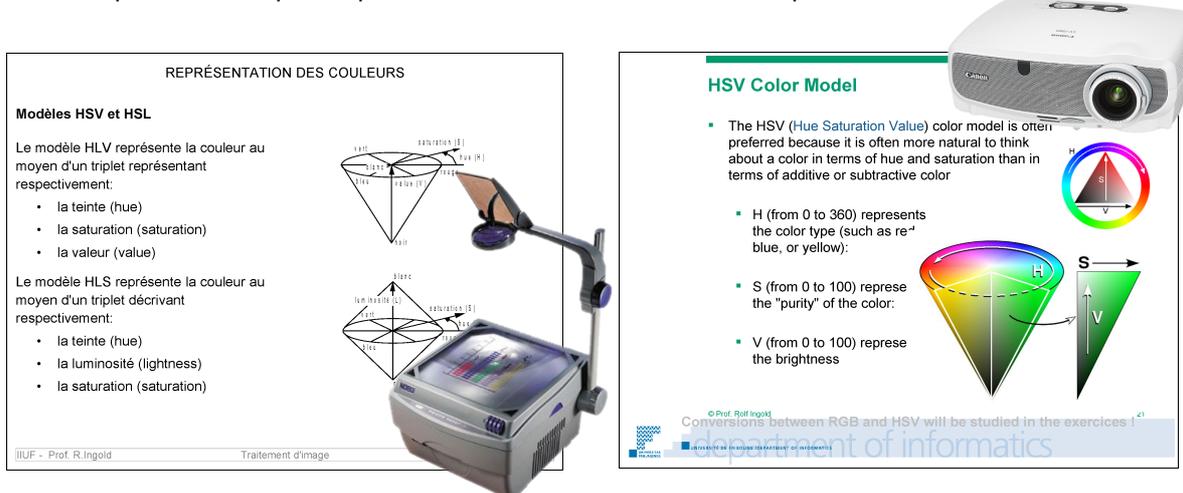
**HSV Color Model**

• The HSV (Hue Saturation Value) color model is often preferred because it is often more natural to think about a color in terms of hue and saturation than in terms of additive or subtractive color

- H (from 0 to 360) represents the color type (such as red, blue, or yellow):
- S (from 0 to 100) represent the "purity" of the color.
- V (from 0 to 100) represent the brightness

© Prof. Rolf Ingold  
Conversions between RGB and HSV will be studied in the exercises.<sup>14</sup>

Department of Informatics



L'illustration précédente donne un aperçu des changements réalisés en deux ans : à gauche la version *TIRF 2001* du cours, à droite *ImaPro 2007*. Le nouveau cours comporte un grand nombre d'illustrations couleur, le contenu a été mis à jour et la forme est maintenant aux normes du département d'informatique; le département impose en effet le respect d'une charte graphique. Les transparents profitent des fonctionnalités avancées de *PowerPoint*, comme le contrôle d'événements séquentiels ainsi que l'insertion d'animations.

## 3.2 Migration vers une nouvelle philosophie de partage internet

Lorsque j'ai repris le flambeau du cours *traitement d'image*, la philosophie de partage internet était limitée à son plus strict minimum : une brève page de présentation statique ainsi qu'un accès à un répertoire distant. J'ai alors décidé de mettre à jour le partage de l'information en ligne.

Aujourd'hui, nos cours sont tous disponibles au travers de plateformes *Moodle*, mais cela ne fut pas le cas lors de la première mise à jour de nos espaces de partage en ligne. En effet, au préalable, une mini-plateforme en langage *PHP* avait été développée pour répondre à nos besoins, elle fut d'abord utilisée par notre groupe de recherche puis par d'autres. C'est alors que la popularité croissante de *Moodle* nous a motivés à effectuer une seconde mise à jour de notre espace de partage en ligne.

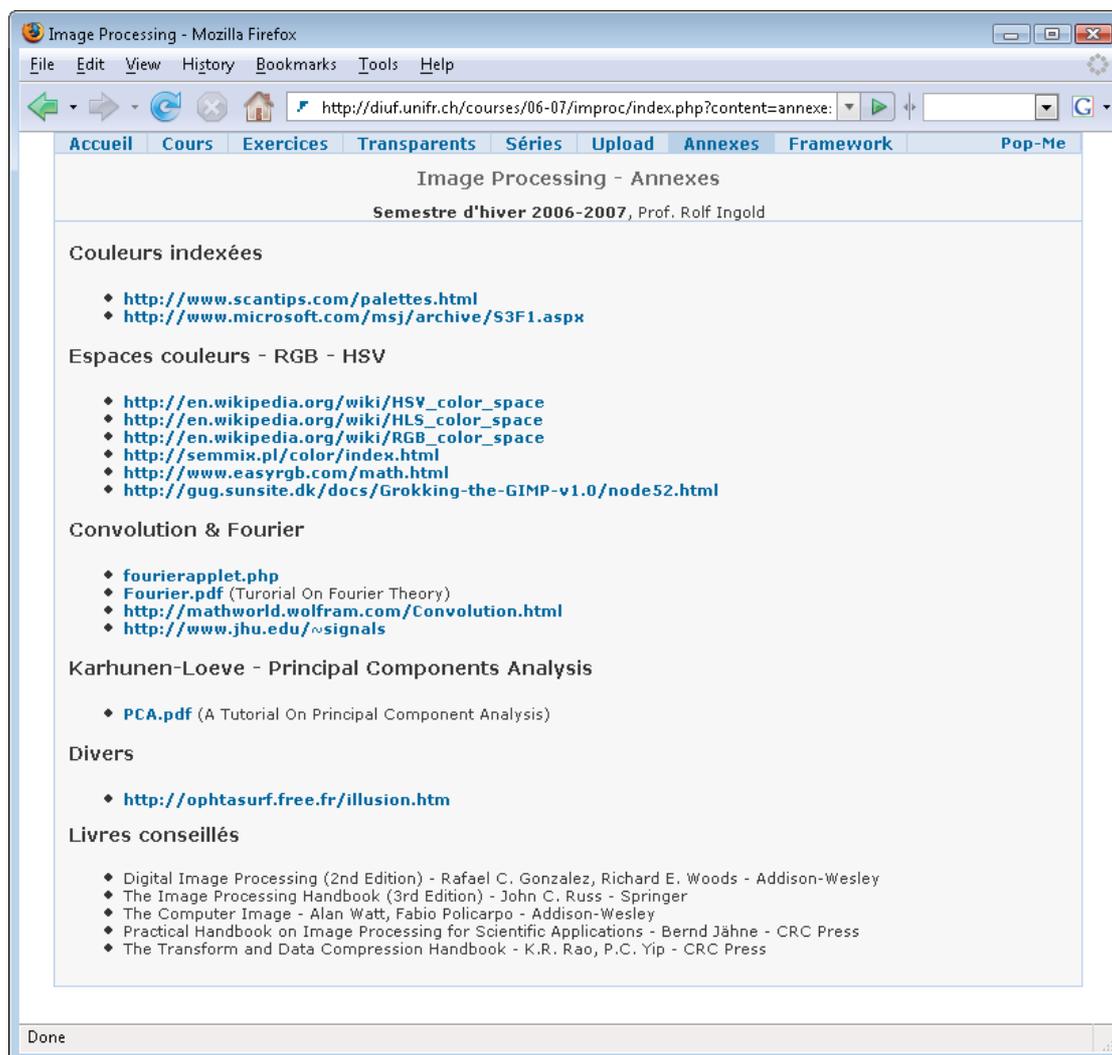
### 3.2.1 Site internet dynamique en *PHP*

A l'époque, je ne connaissais pas encore le potentiel des plateformes d'échange en ligne, encore moins l'existence de *Moodle*. J'ai donc décidé, avec l'aval de mon professeur, de développer personnellement une mini-plateforme internet dynamique, permettant aux étudiants d'accéder aux informations d'une manière plus conviviale. Ce nouvel espace de partage devait également permettre un ajout automatisé des ressources afin de simplifier le travail de l'assistant.

C'est ainsi que notre "ex" nouveau site internet dynamique a été développé en *PHP*, ce langage informatique scripté permettait en effet d'automatiser la création de pages web à partir de ressources déposées sur le serveur. Ce choix facilitait grandement la génération des pages internet, en affichant automatiquement les séries d'exercices ainsi que les transparents du cours, et cela sans aucune édition de la part de l'assistant. Il suffisait de déposer un nouveau document dans un répertoire, sur le serveur, pour que celui-ci soit affiché et mis en page lors de la prochaine requête de la page.

La mini-plateforme *PHP* proposait encore d'autres nouvelles fonctionnalités, comme une page offrant à l'étudiant la possibilité de déposer son devoir en ligne ainsi qu'un espace *Annexes* affichant un ensemble de ressources accessibles par un simple clic (documents *PDF*, liens internet). La force de cette plateforme maison résidait dans le fait que l'ajout d'informations ou de documents ne nécessitait aucune modification du code *HTML* de la page web, puisque ce code était justement automatiquement généré à partir des ressources disponibles dans les divers répertoires existants. Grâce à ce paradigme de déploiement, une personne ne connaissant rien à la programmation internet pouvait sans problème ajouter une ressource en la déposant dans le répertoire adéquat, celle-ci apparaissant alors instantanément, mise en forme sur le site web. De plus, le téléchargement en ligne des séries d'exercices permettait de vérifier automatiquement la validité du fichier téléchargé, de noter l'action effectuée dans un fichier *log*, puis de classer ledit

fichier dans un répertoire correspondant. Un archivage pratique et centralisé, si on le compare au système précédemment en vigueur : la réception des séries d'exercices par courrier électronique.

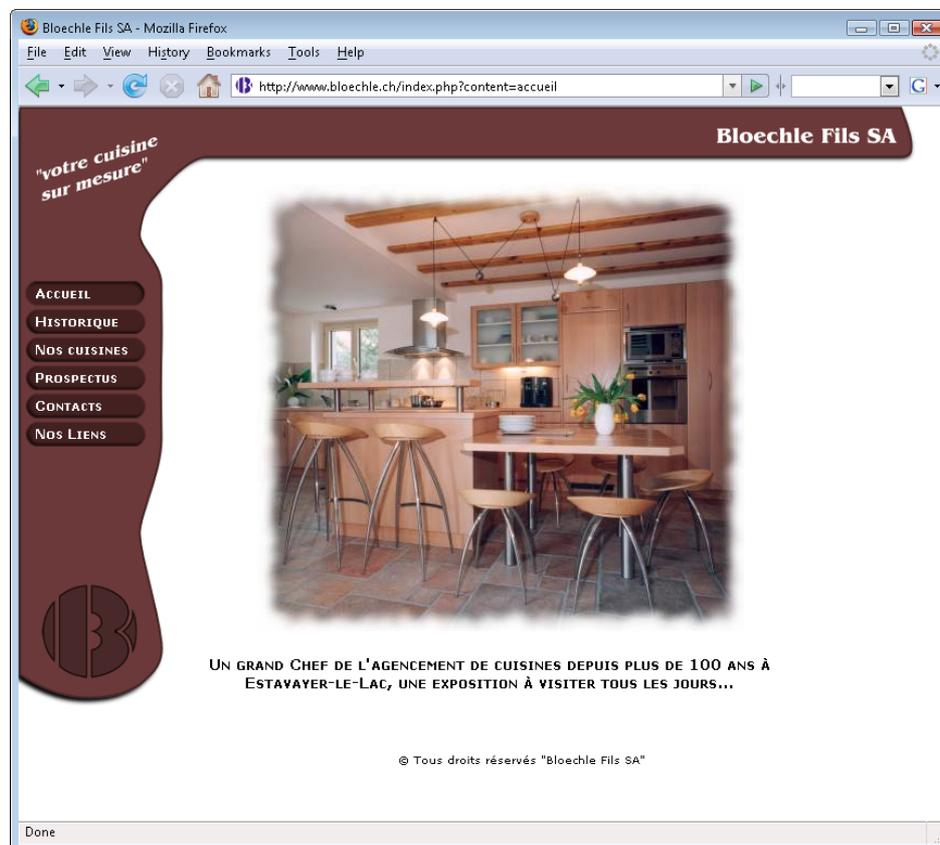
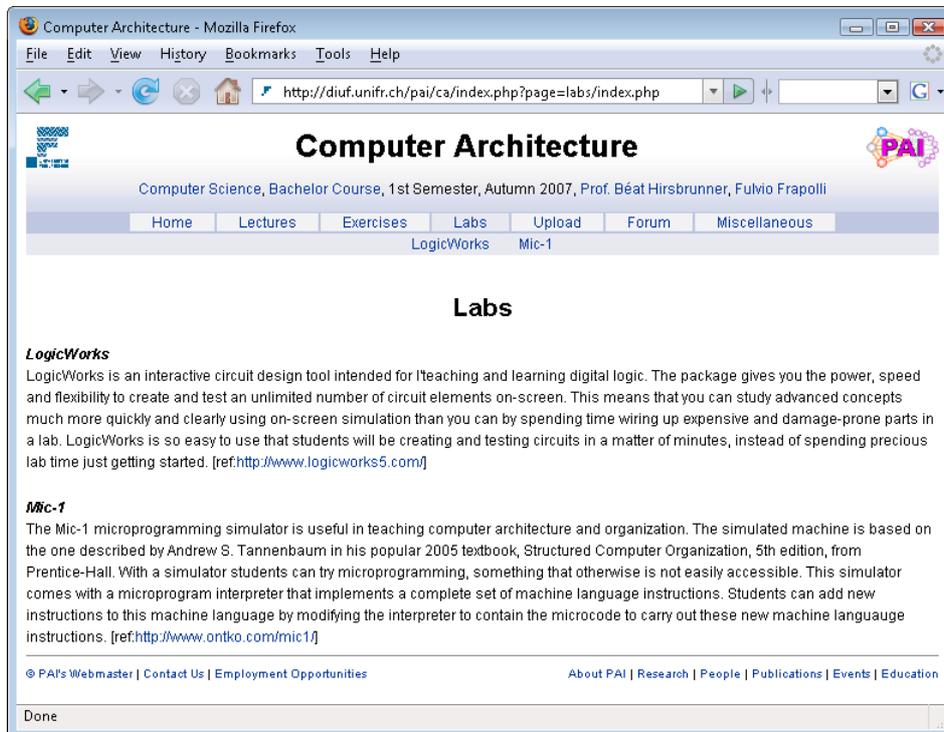


Une capture d'écran de la page web *Annexes* du cours *Traitement d'Image 2006* est présentée ci-dessus. Un ensemble d'annexes y est listé par thèmes, un clic sur l'une d'elles redirige l'utilisateur sur l'annexe demandée.

L'architecture développée en *PHP* avait comme premier but de permettre une mise à jour aisée de son propre cours en ligne, de surcroît, cette architecture permettait de rapidement déployer un nouveau site en ligne. La modularité du système était étudiée de manière à permettre l'ajout de boutons ou la modification de l'apparence graphique très facilement. C'est ainsi que tous les cours de notre groupe de recherche ont été portés sur cette mini-plateforme, tout cela, bien évidemment, avec l'aval de notre professeur.

C'est alors que le groupe de recherche voisin du notre, nommé *PAI* (Pervasive & Artificial Intelligence), informé de mon travail, décida de transférer ses cours sur ce nouveau système. L'illustration suivante montre l'utilisation de notre mini-plateforme par le groupe *PAI*, comme nouvel espace de partage en ligne, pour le cours *Computer Architecture*.

Par la suite, j'ai grandement profité de ce travail effectué en milieu académique et lui offert un nouveau cycle de vie. En effet, le site internet de l'entreprise familiale, *Bloechle Cuisines*, a été développé en utilisant cette mini-plateforme *PHP* (cf. 2ème capture d'écran).



### 3.2.2 Moodle, plateforme de partage internationale

Depuis quelques années déjà, diverses plateformes *LMS* (Learning Management Systems) ont fait leur apparition sur le web : *Claroline*, *WebCT* ou *Moodle*. Ce n'est que dernièrement qu'un engouement généralisé a propulsé certaines d'entre elles au hit parade des systèmes de partage de l'information en ligne. *Moodle* est ainsi devenu une référence en la matière, cette plateforme est utilisée par un grand nombre d'institutions à travers le monde. Une communauté d'utilisateurs et de développeurs s'est même formée autour de cette plateforme, lui apportant sans cesse de nouvelles fonctionnalités.

L'université de Fribourg promeut également cette plateforme par l'intermédiaire du centre *NTE*. Toute la partie installation, mise à jour, sauvegarde et gestion de la plateforme est ainsi totalement transparente pour le corps enseignant.

Pour ces diverses raisons, notre professeur nous a récemment orientés vers l'utilisation de *Moodle* comme plateforme d'échange en ligne. C'est une bonne décision en soit, qui a cependant demandé un travail de migration conséquent en arrière-plan. Il s'agissait de transférer, encore une fois, un cours en ligne d'un système obsolète vers un autre d'actualité. Heureusement, l'utilisation de *Moodle* s'est avérée simple et efficace, le nouveau cours en ligne a finalement pris forme en moins d'une semaine. En effet, le contenu existait déjà, il ne restait plus qu'à le transférer sur la nouvelle plateforme d'échange, puis parfois l'adapter (cf. capture d'écran ci-dessous).

The screenshot shows a web browser window displaying a Moodle course page. The browser's address bar shows the URL: `http://moodle.unifr.ch/course/view.php?id=1635`. The page header includes the University of Fribourg logo and the text 'UNIVERSITÉ DE FRIBOURG / UNIVERSITÄT FREIBURG'. The user is logged in as an invited guest.

The main content area is titled 'Centre NTE' and 'MOODLE UNIFR >> 07\_120'. It features a sidebar with navigation links for 'Personnes' (Participants), 'Activités' (Devoirs, Forums, Ressources), and 'Aperçu des semaines' (Weekly Overview). The weekly overview shows the course 'Traitement d'Image - Automne 07' with a description: 'Ce cours constitue une base théorique pour le traitement d'image. Il présente un ensemble important d'algorithmes fondamentaux de traitement d'image.' The overview lists several weeks with their respective topics and activities, such as 'Image Processing - Visual Perception and Digital Image Representation' and 'Image Processing - Point Operators'.

On the right side, there is a 'Dernières nouvelles' (Recent News) section with three entries, each dated 18 Oct, 12:24, 18 Oct, 12:23, and 3 Oct, 15:45, respectively. Each entry includes the name of the user (Jean-Luc Bloechle) and a brief description of the news item, such as 'Série 04 - Convolutions et filtres plus...' and 'Série 02 - Espaces de couleur plus...'.

La plateforme *Moodle* offre tout de même un certain nombre d'avantages non négligeables par rapport à l'ancienne mini-plateforme *PHP*, en voici quelques-uns :

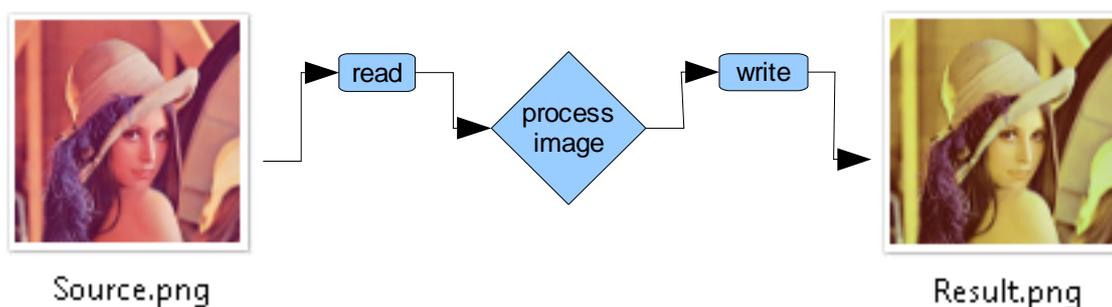
- Les étudiants utilisent une seule et même plateforme pour tous leurs cours durant leur cursus universitaire.
- La plateforme *Moodle* offre un système d'authentification, de plus, celui-ci est partagé avec celui de l'université, il n'y a donc pas besoin de créer de nouveaux comptes.
- Les ressources peuvent être organisées par semaine, offrant à l'étudiant une vision hebdomadaire du cours.
- Les ressources s'ajoutent simplement avec une interface graphique : fichiers, liens sur d'autres pages web, etc.
- Les cours peuvent être préparés à l'avance, il est en effet possible de choisir quand et comment les ressources sont rendues visibles pour les étudiants.
- Des activités peuvent être ajoutées. Parmi celles-ci, l'activité *devoir* s'avère extrêmement pratique : un travail doit être déposé sur le serveur dans un intervalle de dates prédéfinies.
- *Moodle* offre divers systèmes de notation pour les devoirs ainsi qu'une notification automatique des résultats aux utilisateurs.
- Les cours *Moodle*, par l'intermédiaire du centre *NTE*, sont archivés annuellement. De plus, le centre *NTE* offre la possibilité de déployer un nouveau cours en récupérant totalement ou partiellement celui de l'année écoulée.

### 3.3 Un framework *Java* pour une nouvelle philosophie d'apprentissage

Le cours *ImaPro* est une révision complète d'un cours vieillissant, la clef de voûte de son renouveau pédagogique réside dans la refonte des séries d'exercices ainsi qu'une nouvelle approche adoptée pour l'apprentissage des méthodes de traitement d'image.

Un cours tel que le traitement d'image ne peut être enseigné à un apprenant sans une application pratique des concepts théoriques vus aux cours. Dans notre cas, l'application pratique consiste à implémenter en langage informatique divers algorithmes pour le traitement d'image. Cette approche offre à l'étudiant une vision plus pragmatique du problème et lui permet ainsi de mieux comprendre le fonctionnement des différentes méthodes enseignées aux cours théoriques.

Malheureusement, tout n'est pas si simple lorsque l'on désire travailler à bas niveau et communiquer directement avec les ressources d'un ordinateur : son processeur, sa mémoire et ses périphériques... En effet, l'objectif des séances pratiques (et également des cours) est bien l'apprentissage de méthodes pour le traitement informatique de l'image numérique et non pas l'étude de la communication interne avec les diverses ressources et routines d'un ordinateur. Pour tendre vers un tel objectif, encore faut-il pouvoir présenter des séries d'exercices pertinentes aux étudiants afin que ceux-ci puissent effectuer un travail d'implémentation concret, concis, ciblé, et surtout pédagogique. De plus, un programme informatique peut vite devenir extrêmement compliqué à décrypter, comment juger du bon fonctionnement des travaux rendus par les étudiants dans de tels cas de figure ?



Le schéma ci-dessus présente un processus minimal permettant d'effectuer un traitement d'image quelconque, un tel processus nécessite divers éléments :

- une image source, par exemple "Source.png"
- un processus appliquant une transformation à l'image
- une image résultant de la transformation, par exemple "Result.png"

Dans la configuration présentée ci-dessus, les images source et résultante sont stockées sur une mémoire persistante : un disque dur, un disque compact ou une mémoire flash. Lorsque l'on travaille avec de tels supports physiques d'information, il est tout d'abord nécessaire de lire l'image source à partir dudit support, de la décompresser en mémoire tout en respectant son format (gif, jpg, png,... ), d'appliquer la transformation désirée, puis de recompresser l'image et de la réécrire sur le support de stockage. Ce mode opératoire demande un travail de programmation non négligeable et n'offre malheureusement aucune interactivité dans le processus de traitement numérique de l'image : choix d'une image unique, paramètres prédéfinis, etc.

### 3.3.1 Existe-t-il réellement un besoin sous-jacent ?

Apporter ou non des changements à un cours universitaire établi présuppose un temps de réflexion. L'évolution d'un système d'enseignement, pour en améliorer son caractère pédagogique, se base généralement sur des fondements explicites. Il est donc nécessaire, avant de partir tête baissée, de se poser quelques questions pertinentes :

- Quels sont les besoins des étudiants/enseignants ?
  1. Apprentissage par la pratique : les étudiants doivent implémenter (coder en langage de programmation) des algorithmes de traitement d'image.
  2. Compréhension : les étudiants doivent comprendre ces algorithmes de traitement d'image et pouvoir apprécier l'influence de divers paramètres sur le résultat obtenu.
  3. Support à la programmation : les étudiants ne doivent pas perdre leur temps pour autre chose que la programmation des algorithmes, l'implémentation doit être claire et concise.
  4. Eveille de la curiosité : les séries d'exercices doivent en tout temps garder un esprit pédagogique et susciter au mieux la curiosité des étudiants.
  5. Démonstrations concrètes : les enseignants doivent expliquer les méthodes de traitement d'image aux étudiants, si possible à l'aide d'une projection interactive, l'utilisation adéquate des TICEs semble ici évidente.
  6. Correction aisée : la correction des travaux d'étudiants par l'assistant doit rester dans une fourchette de temps raisonnable, tout en évitant d'être négligée.

- Le système en vigueur répond-il à nos besoins ?

Voici la question de base, pourquoi changer un système qui répond à nos attentes ? Dans notre cas, la réponse est bien évidemment négative. Par rapport à la liste de nos besoins présentée ci-dessus, seul le premier est satisfait par l'ancien "système en vigueur", tandis que les besoins 2, 3, 4, 5 et 6 ne répondent clairement pas à nos attentes.

- Le jeu en vaut-il la chandelle ?

Par là, j'entends la question suivante : mettre à jour le système est-il vraiment la bonne décision à prendre ? En effet, premièrement il faut être capable de pouvoir répondre aux besoins cités précédemment et deuxièmement, il faut que l'effort de mise à jour fourni en vaille la peine : passer de longs mois à mettre à jour un système qui n'aurait qu'un impact négligeable sur l'enseignement n'est pas forcément la meilleure façon d'aborder le problème !

Personnellement, il m'a paru évident que la mise à jour de notre système pédagogique était inévitable, et que celle-ci allait grandement améliorer l'enseignement. Mais je dois avouer qu'au préalable, Rolf Ingold et moi n'avions pas réellement analysé notre ancien système afin d'en extraire les points positifs ou négatifs, nous étions simplement partis de la constatation suivante : notre cours ne répond pas à nos attentes et celles de nos étudiants, nous allons donc y remédier.

### 3.3.2 Trois ans de développement itératif

Une fois notre décision prise, nous n'avons pas perdu notre temps. Mon professeur et moi avons tout d'abord commencé par une longue discussion sur le développement de notre futur outil d'apprentissage. Pour ce premier entretien, Rolf n'est pas venu les mains vides, il avait déjà élaboré les prémisses du framework *ImaPro*, et me présenta quelques lignes de code informatique offrant l'affichage d'une image dans un environnement fenêtré. A partir de cette ébauche, nous avons défini les grandes lignes du cahier des charges. Je suis ensuite retourné dans mon bureau pour y débiter le développement du projet, et ainsi consacrer mon temps au travail de programmation nécessaire.

Dans l'idéal, ce framework aurait du être finalisé il y a deux ans de cela, les cours commençaient à l'époque vers la fin du mois d'octobre 2005. Pour ma part, j'avais commencé l'implémentation du framework au début du mois de septembre. Je pensais pouvoir terminer sa programmation en deux mois. En réalité, une première version est effectivement sortie après deux mois, mais celle-ci s'est vite avérée très imparfaite. Les étudiants du cours ont tous téléchargé l'application et commencé à faire leurs séries d'exercices avec celle-ci. Dès la première semaine, j'ai reçu de nombreux *feedbacks* de la part de mes étudiants :

- Certains étudiants ont des difficultés avec l'installation et la configuration du framework.
- Il manque une documentation détaillée sur le framework et les fonctions disponibles.
- Il manque des exemples simples montrant l'utilisation et l'extension du framework.
- Certaines fonctionnalités sont inexistantes ou alors trop complexes.
- De nombreux bugs et lacunes pénalisent l'utilisation du framework.
- Des remarques très positives et encourageantes sont heureusement présentes.

Dans un premier temps, j'étais un peu sous pression, il s'agissait de délivrer un produit fonctionnel en un temps restreint, en ne sachant pas réellement les fonctionnalités qu'il fallait y incorporer. En effet, je connaissais le contenu des deux premières séries d'exercices, mais n'avais pas encore été informé de la suite... ceci a eu pour fâcheuse conséquence qu'à partir de la troisième série, je délivrais une nouvelle version du framework toutes les deux semaines ! Ces mises à jour intempestives n'étaient évidemment pas du tout agréables pour les étudiants. Ceux-ci devaient à chaque fois retélécharger le framework, migrer leurs données d'une version à l'autre et même corriger certaines implémentations antérieures suite à des incompatibilités.

Tout informaticien consciencieux est censé offrir de nouvelles versions de ses produits en respectant une compatibilité "en arrière", afin que ses clients et utilisateurs n'aient pas de mauvaises surprises lors des mises à jour. Ce paradigme est en théorie valable pour mon framework *ImaPro*. Malheureusement, une telle compatibilité en arrière limite grandement la mise à jour du framework lors des grandes révisions inévitables durant les premières phases de développement. J'ai donc opté pour la manière forte en expliquant aux étudiants les modifications apportées à chaque mise à jour, ce qu'ils devaient changer dans leur code et comment. Ceci m'a permis de renouveler certains concepts trop lourds, voire totalement inefficaces, de reprogrammer des parties entières du framework, puis de l'étendre sans me soucier de la compatibilité, en ayant comme seuls objectifs : simplicité et efficacité.

Comme dit précédemment, la première année fut très mouvementée. Chaque semaine il fallait préparer une nouvelle série d'exercices pour le prochain cours de traitement d'image, puis adapter le framework pour qu'il inclue les nouvelles fonctionnalités correspondantes. Je passais les trois quarts de mes journées à préparer le cours et travaillais régulièrement le soir, à la maison. Les mises à jour du framework étaient très courantes, trop courantes même. Un jour, mon professeur me convoqua dans son bureau, il m'exposa en quelques mots son point de vue sur ma méthode de travail qui n'était selon lui pas du tout adéquate :

- Les mises à jour étaient beaucoup trop nombreuses, l'étudiant était saturé par cette façon d'agir, il perdait du temps à installer chaque version et n'en comprenait pas l'utilité.
- Je devais adopter un système de versions clair et responsable, distinguer chaque mise à jour par une politique de nommage séquentielle.
- Il fallait modifier la structure même du framework pour permettre des mises à jour incrémentales sans que l'étudiant ne doive à chaque fois tout réinstaller.
- Je devais absolument écrire une documentation complète du framework.
- Le framework était trop complexe à utiliser, il fallait repenser son architecture.
- Il fallait encore modifier les fonctionnalités existantes pour les rendre beaucoup plus simples, plus intuitives.

Après une telle entrevue, je dois dire que j'étais plutôt désappointé. J'avais passé tellement de temps à développer ce framework pour finalement m'entendre dire que je n'y étais pas du tout, qu'il fallait que je revoie totalement mes méthodes de travail. J'étais vraiment déçu, touché dans ma propre estime. Je pensais sincèrement avoir bien fait, alors qu'on me rétorquait que tout était trop compliqué, obscur et non pédagogique. J'ai tout d'abord eu une réaction de négation, je n'étais pas d'accord avec mon professeur, je pensais qu'il avait complètement tort, qu'il se faisait son petit cinéma, qu'il n'avait pas compris ce que j'avais développé. La déception passée, j'y ai réfléchi plus objectivement, en repensant aux problèmes précités, je devais avouer que certains points soulevés étaient tout à fait justifiés, tandis que d'autres l'étaient moins.

Avant d'entamer la deuxième année d'enseignement, en octobre 2006, j'ai profité des vacances académiques pour retravailler le framework durant plusieurs semaines, dans l'optique de le rendre plus propre, plus simple. *ImaPro* a vécu une refonte totale en tenant compte des remarques de Rolf, des étudiants et mon expérience acquise l'année précédente. Voilà que débute la deuxième année de cours, et malgré mes efforts, tout ne roule pas comme prévu, je suis à nouveau obligé d'apporter des modifications au framework en cours d'année. Certaines séries d'exercices évoluent et je constate que des fonctionnalités indispensables manquent encore à *ImaPro*.

Il y a maintenant trois mois de cela, mon professeur me convoque à nouveau dans son bureau, il a téléchargé la dernière version d'*ImaPro*, il trouve que la documentation n'est pas du tout satisfaisante et que le fonctionnement du framework est encore trop compliqué, il aimerait rendre tout cela plus simple, plus transparent pour l'étudiant. A cet instant, je me demande si les critiques du professeur sont fondées, ai-je été négligeant ? Il m'explique son point de vue, que mon framework est très bien, qu'il est reconnaissant du temps que je consacre à développer cet outil, mais que malgré cela, je n'ai pas encore réussi à présenter un travail simple et pédagogique.

Partant de ses remarques, et avec un peu de recul, je devais bien reconnaître que malgré tous mes efforts, je n'avais effectivement pas encore rempli le cahier des charges demandé :

- La documentation de mon framework était toujours incomplète.
- Les fonctionnalités proposées par le framework étaient trop nombreuses.
- L'architecture du framework était encore trop *monolithique*.

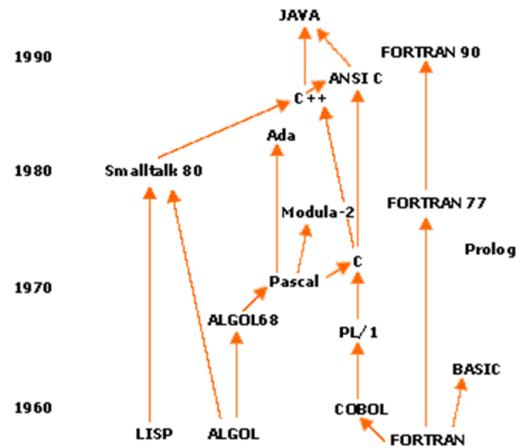
J'ai à nouveau mis la main à la pâte et reprogrammé le framework une dernière fois, de fond en comble. En connaissant sur le bout des doigts les points faibles d'*ImaPro*, je l'ai réorganisé d'une manière modulaire. Un certain nombre de fonctions inutiles ont simplement été supprimées, tandis que d'autres trop complexes ont été simplifiées. L'architecture du programme est devenue très hiérarchisée avec des modules clairement définis. De plus, beaucoup de fonctions ont été renommées, afin de rendre leur utilisation plus intuitive, grâce à une nomenclature plus logique respectant les standards du domaine.

Et voilà, le framework *ImaPro* en est actuellement à sa version 2.2. A ce jour, mon professeur semble très satisfait du résultat obtenu et je pense pouvoir dire que nos objectifs sont atteints, après trois ans d'un travail itératif de longue haleine. Le développement du framework ainsi que son amélioration constante ont été possibles grâce à une remise en question continue, donnant lieu à des périodes de mises à jour, voire même de refonte complète de l'outil. Les acteurs de ces remises en question furent tout autant le professeur que les étudiants, ceux-ci ont en effet apporté une collaboration précieuse au travers de leurs remarques très pragmatiques : découverte de bugs, demande de fonctionnalités supplémentaires, et bien sûr leurs nombreuses questions. Je n'avais jusqu'alors pas soupçonné l'information que pouvait contenir de telles interrogations : celles-ci reflétaient régulièrement une incompréhension sur l'utilisation du framework, souvent due à une documentation lacunaire, une utilisation non intuitive ou trop complexe.

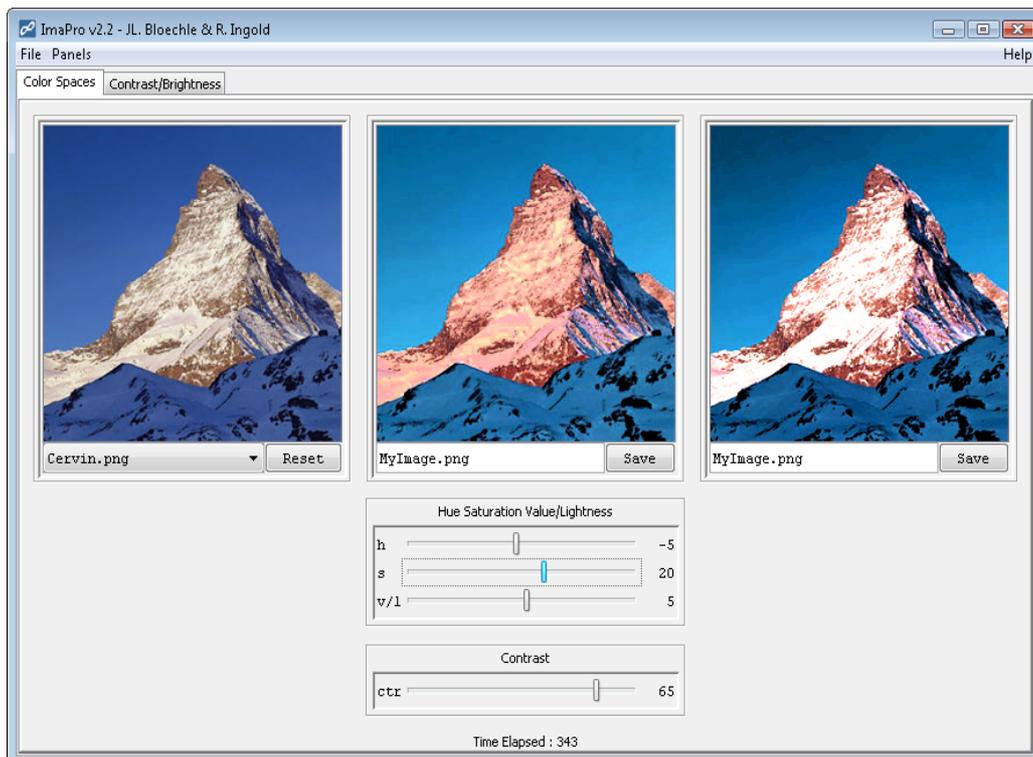
### 3.3.3 *ImaPro*, un support à l'apprentissage

Cette sous-section va présenter le framework *ImaPro* en détail, son fonctionnement, ses qualités, mais également ses défauts. Je vais bien sûr essayer d'être le plus objectif possible afin de ne pas discréditer mon analyse.

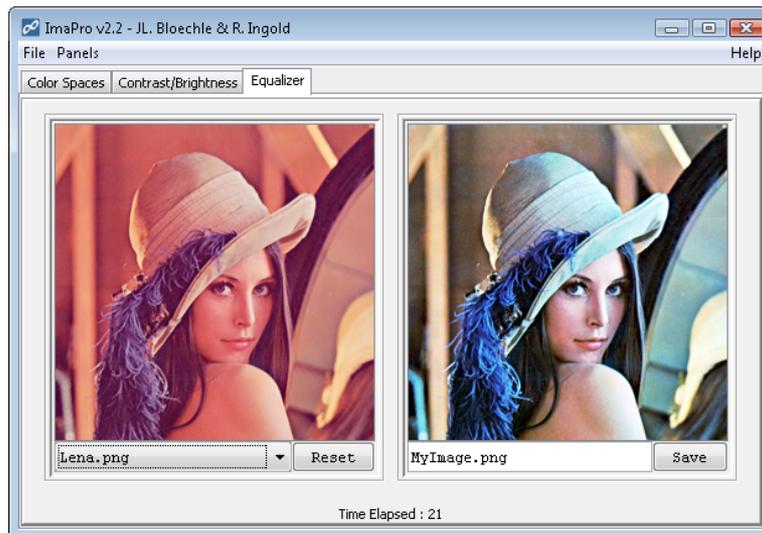
Le traitement d'image est une discipline informatique en constante évolution. Bien que les théories mathématiques sous-jacentes ainsi que les algorithmes fondamentaux aient été développés il y a déjà plus de vingt ans, leurs applications concrètes continuent de progresser. En effet, la puissance des ordinateurs est aujourd'hui telle, qu'un programmeur n'est plus obligé de sacrifier l'élégance au profit de la performance. Créer un algorithme de traitement d'image n'est plus uniquement qu'une question d'optimisation mais plutôt une réflexion plus large sur l'interaction propre et efficace de divers modules, un découpage du traitement en unités logiques clairement définies. *Java* est un langage de programmation que l'on définit comme étant *orienté objet*, ceci implique qu'il a justement été développé dans un but précis : offrir un langage informatique de haut niveau, performant, propre et modulaire. Ce langage n'est pas autant performant au sens de la rapidité que d'autres langages vieillissants de plus bas niveau comme le langage *C* ou l'assembleur, mais il est incomparable sur le plan du développement et de l'architecture logicielle, des atouts qui le classent aujourd'hui au sommet des langages de programmation.



Le schéma ci-dessus expose une taxonomie des principaux langages de programmation, de leurs débuts à nos jours [WEB07a]. *Java* y est affiché comme un langage récent, un atout considérable, puisqu'il tire profit des meilleurs paradigmes des anciens langages tout en évitant leurs lacunes et en corrigeant leurs imperfections.

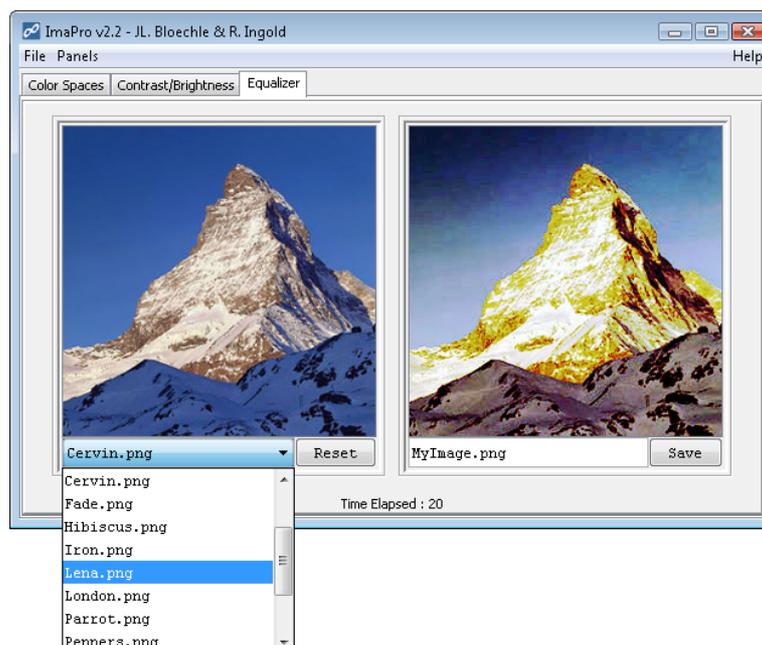


Le framework *ImaPro* a justement été développé en langage *Java* pour toutes ces raisons. La capture d'écran ci-dessus présente un exemple concret d'interface interactive développée avec *ImaPro*. Un contexte fenêtré dynamique pour le test ainsi qu'un ensemble d'outils de programmation performants pour le traitement d'image sont maintenant disponibles pour l'étudiant. Celui-ci peut alors facilement développer ses algorithmes de traitement d'image dans un environnement assisté répondant aux standards de programmation actuels. Supposons qu'un étudiant veuille programmer une *égalisation automatique* du contraste sur une image, une interface graphique développée avec *ImaPro* pourrait afficher la fenêtre suivante (cf. ci-suit).



Dans ce cas, l'interface développée est très simple, elle ne comporte que deux composants : une image source et une image résultante. Cela suffit déjà pour l'implémentation d'algorithmes de traitement d'image ne demandant pas de paramètres en entrée, ou lorsque les paramètres sont prédéfinis une fois pour toute avant le traitement.

L'image source, à gauche, correspond à la photo originale non traitée, tandis que l'image résultante, à droite, correspond à sa version finale, une fois l'algorithme de traitement d'image appliqué. Si l'on s'attarde un peu sur la capture d'écran, on peut observer qu'une barre de menu horizontale est affichée sur le haut de la fenêtre. Celle-ci contient trois entrées donnant chacune accès à des sous-menus correspondants : *File*, *Panels*, *Help*. Ces sous-menus permettent, entre autres, d'ouvrir ou fermer des panneaux de traitement d'image programmés par l'étudiant, ou préexistants. De plus, le menu *Help* offre la possibilité d'accéder à la documentation interne du framework.



Lors du développement de méthodes de traitement d'image, il est très appréciable de pouvoir tester un algorithme en cours d'implémentation sur un ensemble d'images types afin d'en apprécier visuellement le bon fonctionnement. La modification interactive des paramètres de la transformation est également une fonction cruciale pour contrôler le résultat de son implémentation, comprendre l'algorithme et détecter au mieux les erreurs. Le framework offre justement de telles fonctionnalités, pour cela, il supporte un ensemble pertinents de *widgets* interactifs (détaillés plus tard) et dispose d'une petite base de données contenant des images en couleurs, en niveaux de gris ainsi que des images noir/blanc. L'accès aux images est facilité par une liste de sélection située directement au-dessous de chaque image source. Ce composant s'avère très vite indispensable pour observer la réaction des algorithmes sur diverses images, et cela simplement en cliquant sur le nom d'une image dans une liste déroulante. L'image résultante est alors automatiquement recalculée puis mise à jour.

Voici un exemple concret du code informatique générant l'interface présentée précédemment (celle contenant l'image du Cervin). Le programme *Java* présenté ci-dessous effectue une égalisation d'histogramme, il utilise les fonctionnalités du framework *ImaPro* :

```
import diuf.thread.ProcessThread;
import diuf.imapro.image.ImageRGB;
import diuf.imapro.core.ImaProPanel;
import diuf.imapro.image.Image;
import diuf.imapro.widget.WidgetAction;

public class EqualizerPanel extends ImaProPanel
{
    private Image src = new ImageRGB();
    private Image res = new ImageRGB();

    public EqualizerPanel()
    {
        addSource(src);
        newColumn();
        addResult(res);
    }

    protected void process(WidgetAction action, ProcessThread thread)
    {
        int width = src.width();
        int height = src.height();
        int area = src.area();
        int[][] map=new int[src.depth()][];

        for (int b=0; b<map.length; b++)
        {
            map[b]=src.getPlane(b).histogram(256);

            for (int i=1; i<map[b].length; i++)
                map[b][i]+=map[b][i-1];
            for (int i=0; i<map[b].length; i++)
                map[b][i]=(int)((map[b].length-1)*map[b][i]/(double)area);
        }

        for (int y=0; y<height; y++)
        {
            for (int x=0; x<width; x++)
            {
                int[] p=src.getPixelValues(x,y);

                for (int b=0; b<p.length; b++)
                    p[b]=map[b][p[b]];

                res.setPixelValues(x,y,p);
            }
        }
    }
}
```

Je conviens volontiers que ce petit exemple ne doit pas être très parlant pour un profane. Néanmoins, un programmeur *Java* vous avouera qu'effectivement, la quantité de lignes de code informatique est minime pour une interface graphique effectuant l'égalisation d'histogramme d'une image. En réalité toute la puissance de ce code *Java* réside dans l'instruction *extends ImaProPanel*. Cette instruction signifie que tout ce qui est créé dans cet espace de programmation est géré en arrière-plan par le framework *ImaPro*. Cette assistance est totalement transparente pour le programmeur qui n'a plus qu'à élaborer son algorithme de traitement d'image dans une fonction prédéfinie nommée *process*. La création de l'interface graphique, la gestion des interactions avec l'utilisateur, la mise à jour des images, l'utilisation partagée du processeur, etc., tout est géré par le framework. Sans une telle architecture, un étudiant ne pourrait tout simplement pas programmer un environnement graphique interactif.

Des milliers de lignes de code informatique ont été développées pour parvenir au résultat actuel, mais l'objectif est atteint : *ImaPro* est un framework modulaire offrant un environnement interactif pour le développement d'algorithmes de traitement d'image. Grâce à lui, l'étudiant peut utiliser son temps de travail intelligemment, en se consacrant à son principal problème qu'est l'implémentation d'un algorithme de traitement d'image.

En plus de la gestion de l'environnement graphique, le framework offre encore une multitude d'outils et de fonctionnalités supplémentaires au programmeur, et non des moindres :

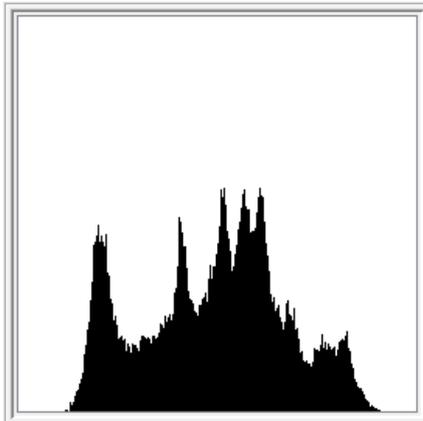
- divers composants graphiques interactifs et paramétrables nommés *widgets*,
- une structure de haut-niveau pouvant contenir toute l'information relative à une image,
- des fonctions avancées permettant l'accès et la transformation des valeurs d'une image,
- un système de filtre permettant l'imbrication de plusieurs traitements successifs, etc.

Les composants graphiques disponibles donnent à l'étudiant la possibilité de développer rapidement et facilement une interface graphique complexe et interactive. Ils sont d'une efficacité redoutable lorsqu'il s'agit d'implémenter des algorithmes dont la configuration des paramètres est dynamique. Parmi les composants développés, voici les *widgets* (acronyme de *windows gadget*) les plus utilisés :

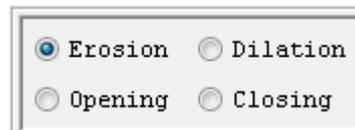
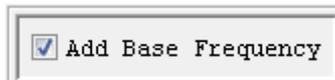
- Images sources et résultantes



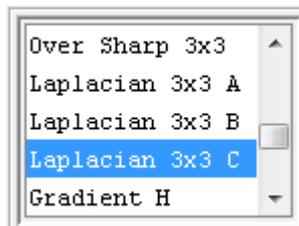
- Histogrammes et images



- Boutons à choix multiples ou uniques

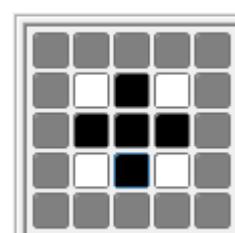
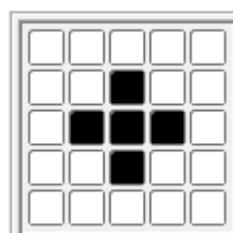


- Liste à choix et ensemble de boutons

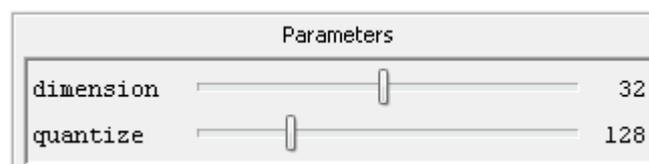


- Masques d'entiers, binaires et ternaires

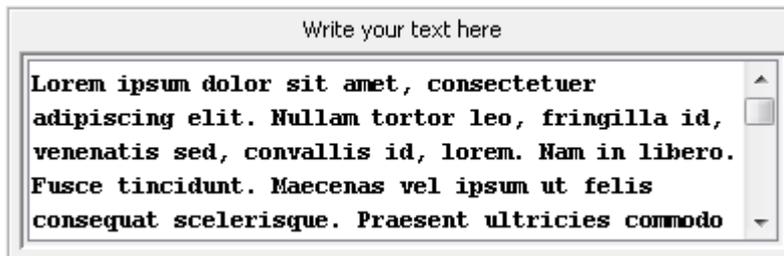
0	0	0	0	0	1
0	-1	-1	-1	0	
0	-1	8	-1	0	127
0	-1	-1	-1	0	
0	0	0	0	0	



- Glisseurs de valeurs dans un intervalle discret



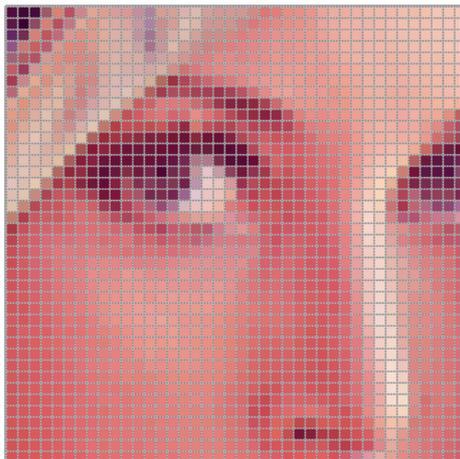
- Champs de texte éditable avec ascenseur vertical



- Champs de texte non éditable

psnr	29.88 dB
space entropy	4.90 bit
encoding size	13.19 vs 13.27 ko

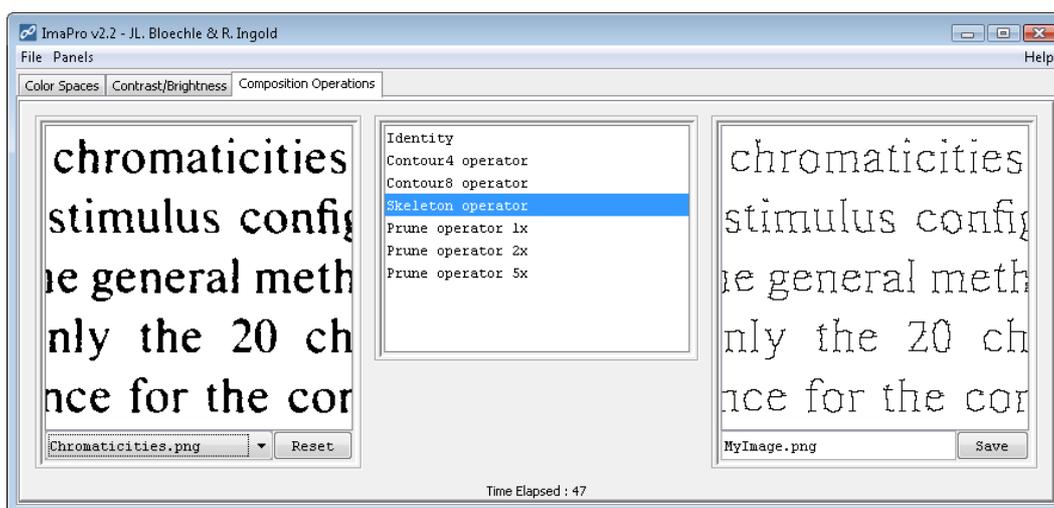
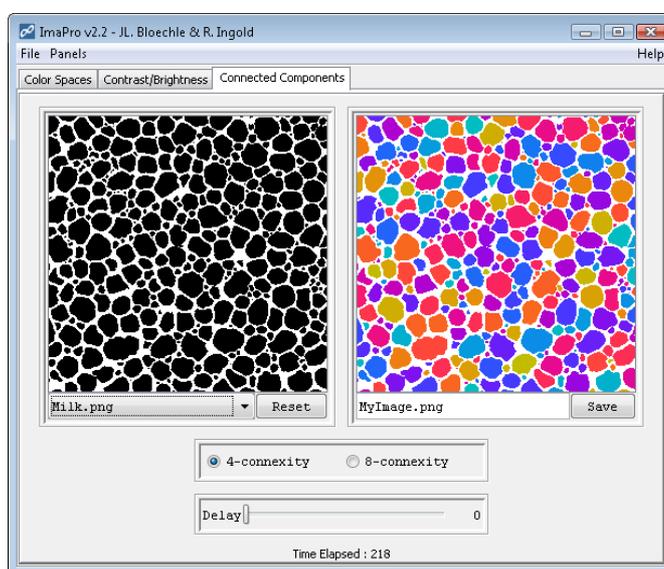
Encore un petit mot quant aux structures numériques contenant les images et leurs transformations. Dans le cas d'*ImaPro*, les images sont toutes définies par des matrices bidimensionnelles de *pixels* (acronyme de *picture element*) comme illustré sur l'image ci-dessous. Chaque élément de la matrice est lui-même composé de une ou plusieurs valeurs, suivant le modèle de représentation de l'image. Ces valeurs sont accessibles par diverses fonctions. Ainsi l'application d'une transformation à une image consiste principalement à parcourir tous les éléments de la matrice de l'image source, à récupérer les valeurs correspondantes, à les modifier suivant des méthodes plus ou moins complexes, puis à assigner les nouvelles valeurs calculées dans la matrice de l'image résultante... Et voilà, le tour est joué :-).

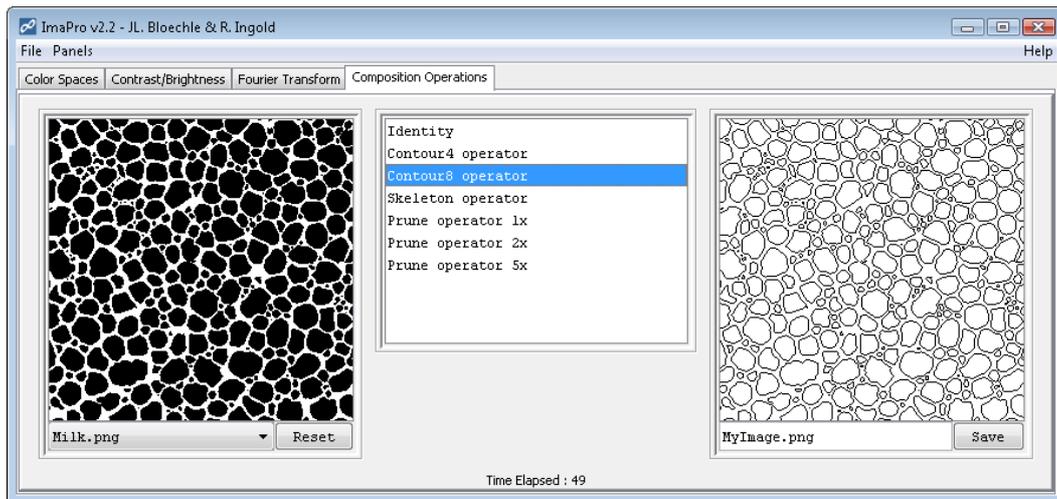


## 4 Quelques exemples concrets en images...

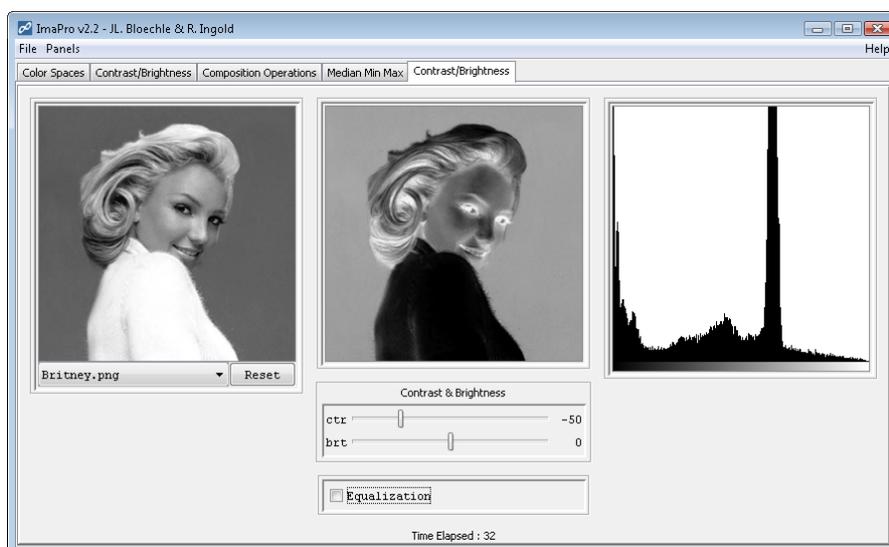
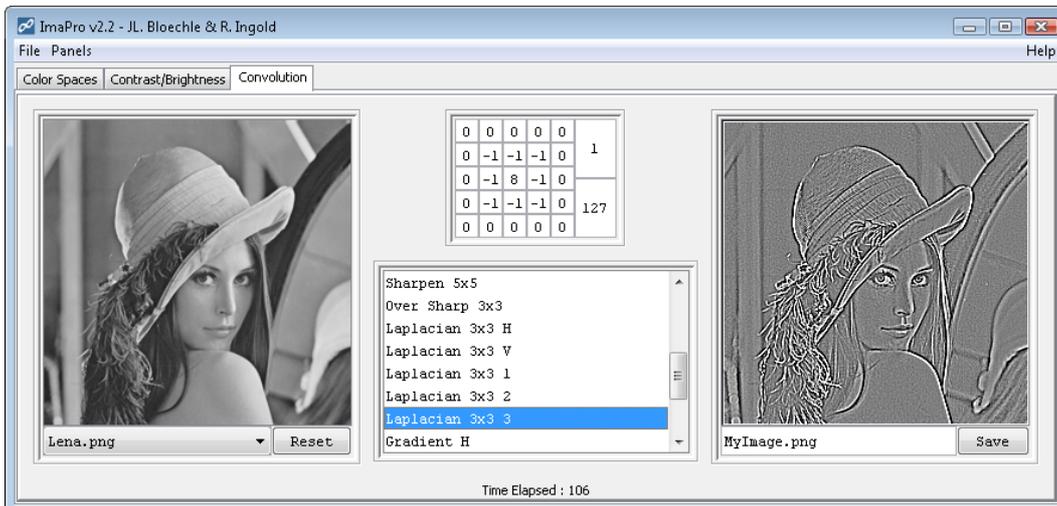
Cette section expose des panneaux de traitement d'image développés dans *ImaPro*. Tous ces exemples sont déjà implémentés et directement accessibles dans le menu *Panels* du framework. Les transformations d'image ont été scindées en trois parties, selon l'image source traitée : noir/blanc, niveaux de gris, couleurs. Attention, une transformation d'une image noir/blanc peut donner comme résultat une image couleur, la distinction ne prenant en compte que l'image d'origine. Aucun commentaire n'est donné sur les images suivantes, il s'agit simplement de pouvoir apprécier visuellement l'étendue des possibilités du framework !

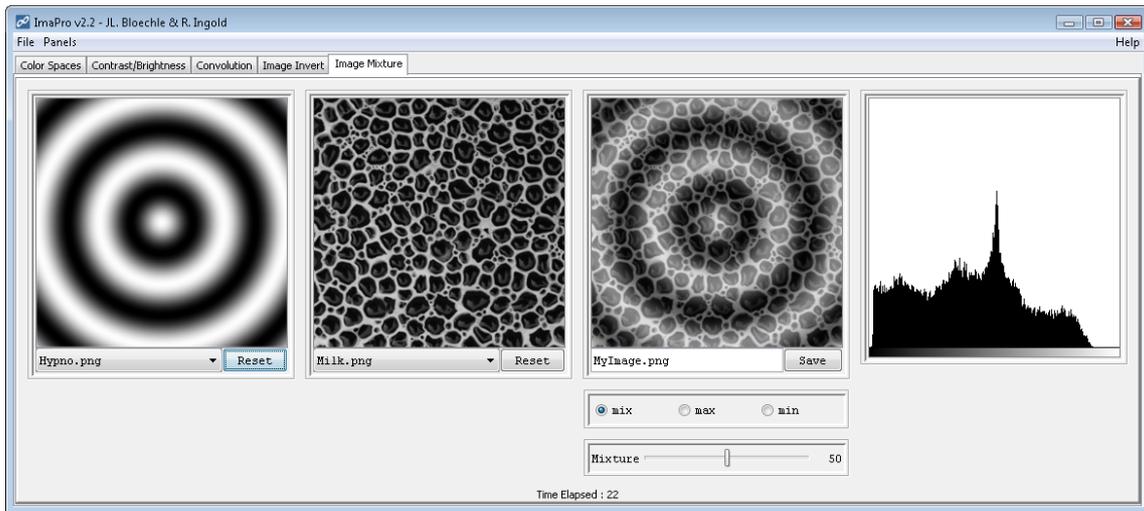
### 4.1 Opérations sur des images en noir/blanc



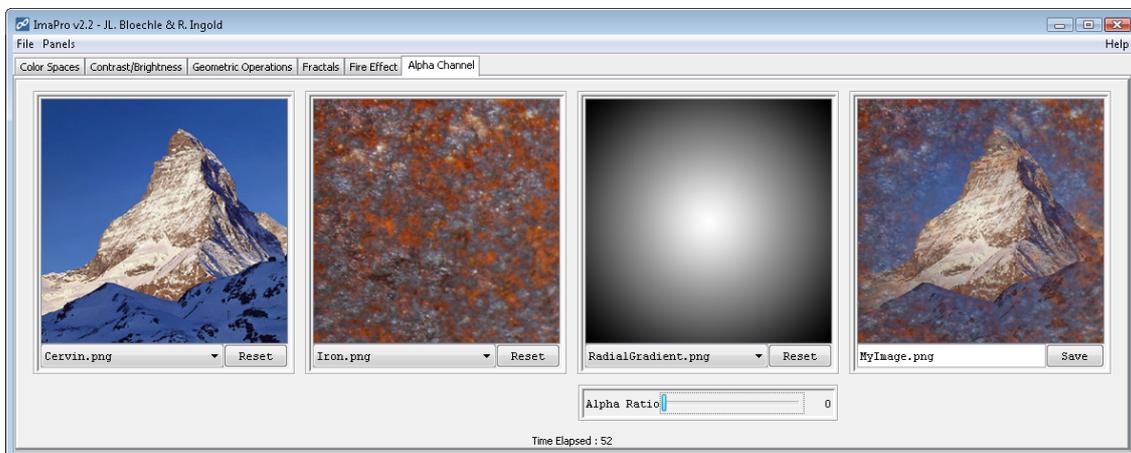
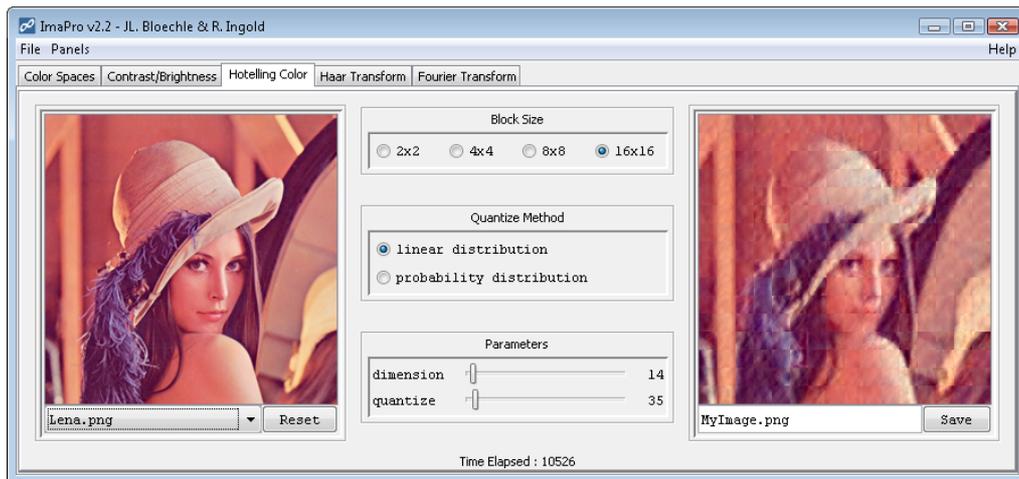


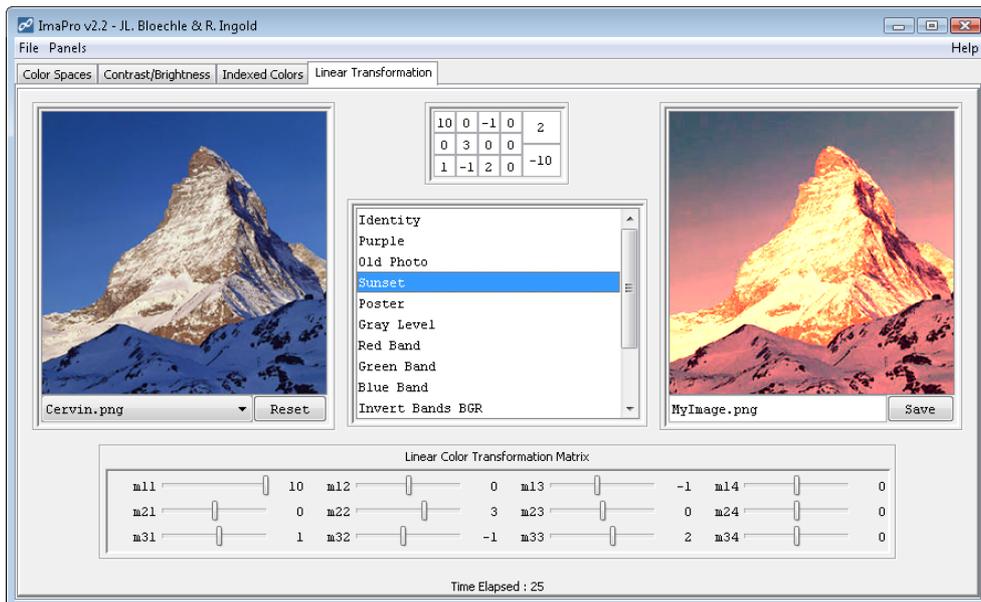
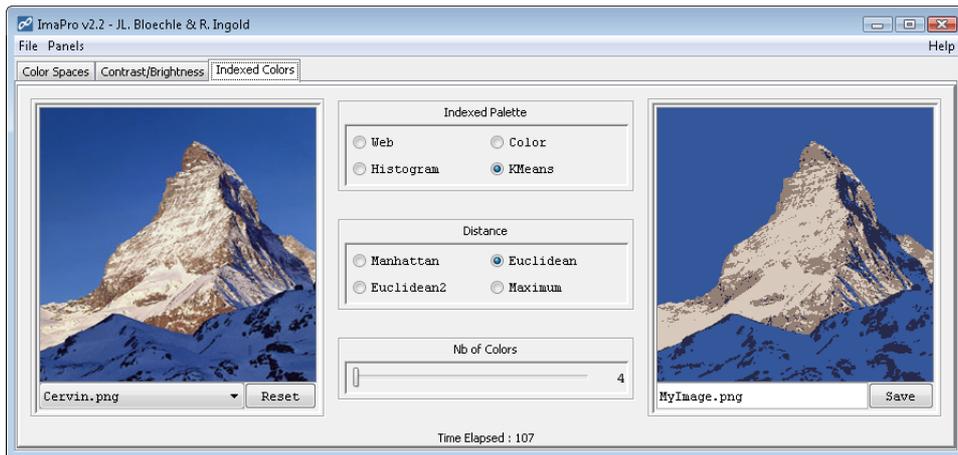
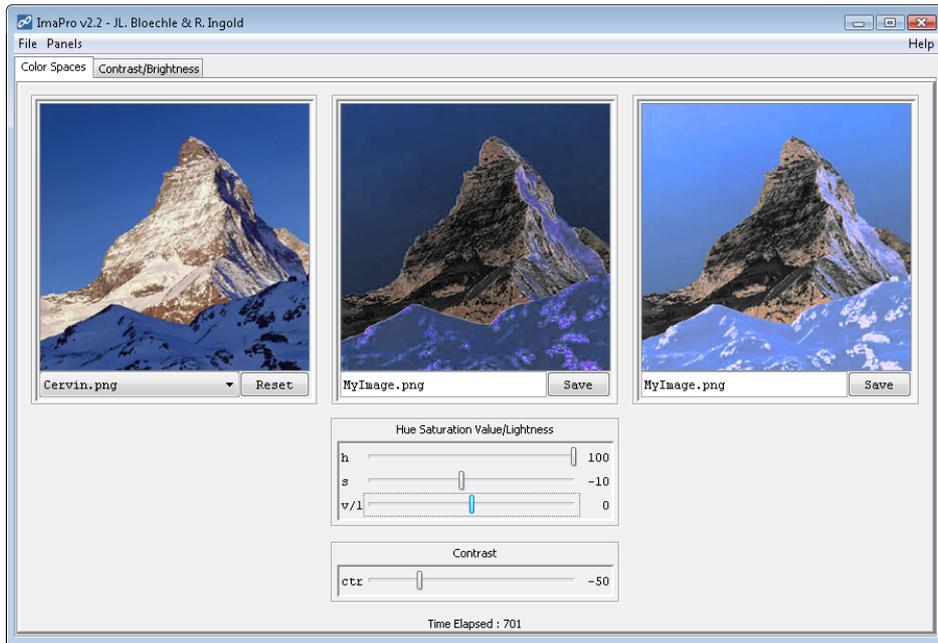
## 4.2 Opérations sur des images en niveaux de gris

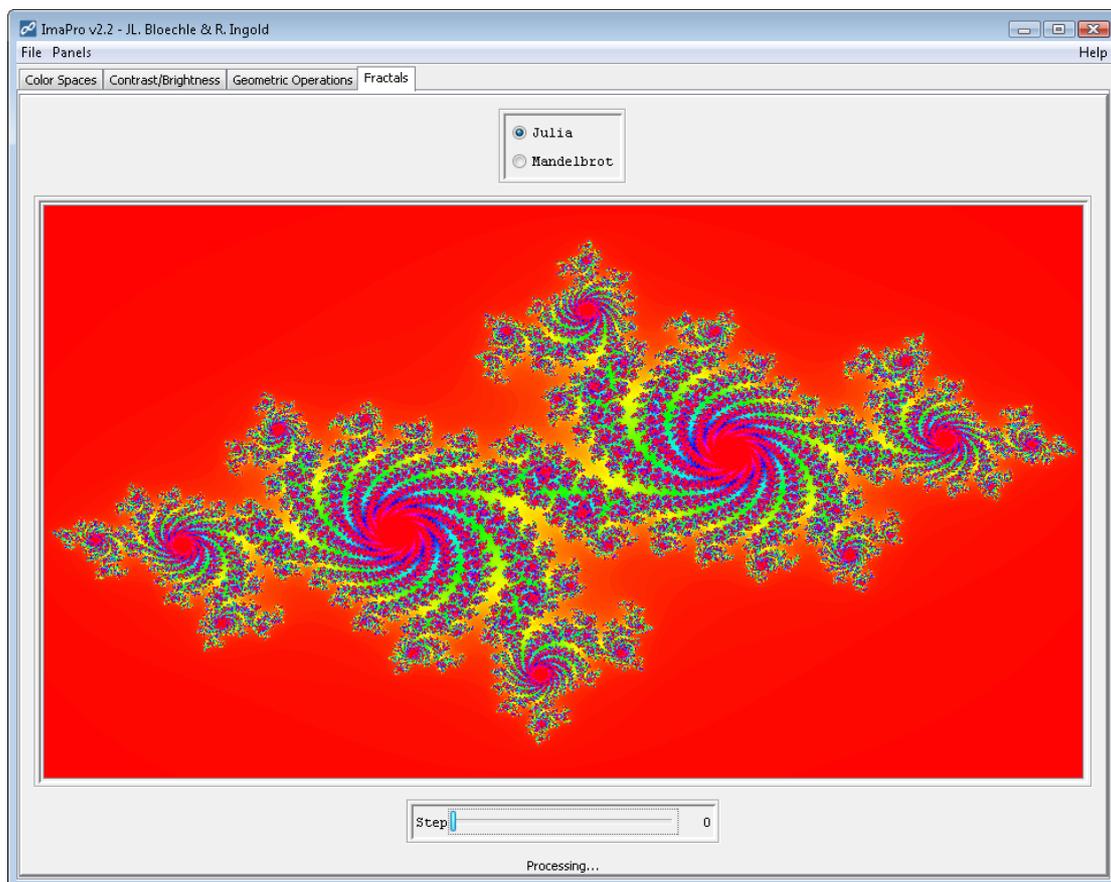
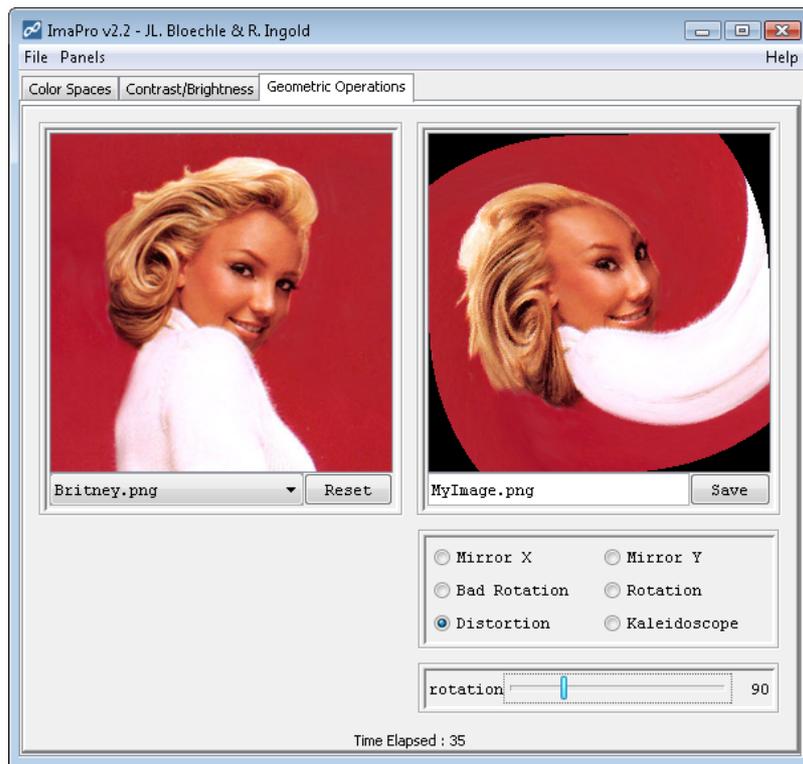




### 4.3 Opérations sur des images en couleurs







## 5 Conclusions

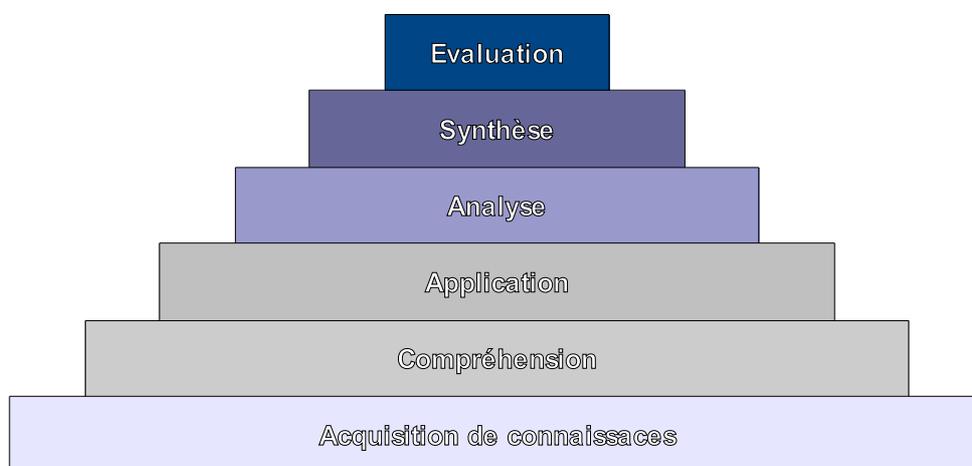
Un certain nombre de concepts enseignés lors des modules de didactique universitaire ont été implicitement intégrés dans ce travail. Afin d'éclairer le lecteur non averti sur certaines de ces intégrations, je vais en présenter quelques-unes dans la prochaine sous-section. Finalement, je vais terminer par des considérations plus personnelles.

### 5.1 Quelques parallèles avec les modèles vus aux cours

La refonte du cours *traitement d'image* bénéficie largement de mon expérience acquise durant le suivi de mon cursus de didactique universitaire. Les connaissances théoriques et pratiques accumulées durant ces trois années ont été mises à profit, plus ou moins consciemment, lors du développement du cours *ImaPro* ainsi que durant la rédaction de ce travail de diplôme. Deux approches complémentaires abordées dans le *module A* du cursus de didactique universitaire [CHA05a] ont particulièrement guidé mes pas et m'ont permis de réaliser l'objectif suivant : un apprentissage interactif, résolument orienté vers la pratique, le tout dans un environnement privilégiant un contexte pédagogique.

Premièrement, les travaux de Jean Piaget, spécialiste suisse de la psychologie du développement de renommée internationale. Ce psychologue préconise une démarche tournée vers la compréhension. Pour y parvenir, il conseille à l'enseignant d'utiliser les ressources et connaissances de chaque apprenant pour atteindre les objectifs d'apprentissage fixés. De plus, il privilégie un apprentissage par l'expérience plutôt que par la transmission. Et justement, le renouveau du cours *ImaPro* suit ces principes. En effet, la connaissance préexistante des étudiants est fortement mise à contribution lors de la programmation informatique, de même, l'apprentissage par l'expérience était le but recherché lors du développement de notre framework de traitement d'image.

Deuxièmement, je fais référence à la taxonomie élaborée par Benjamin Bloom, éminent chercheur dans le domaine de l'apprentissage et de la cognition. Cette taxonomie décompose la connaissance en six niveaux hiérarchiques successifs :



Benjamin Bloom décrit les six niveaux de la pyramide comme suit :

1. **Acquisition de connaissances** : être capable de se rappeler de mots, faits, de dates, de conventions, de classifications, de principes, de théories, etc.
2. **Compréhension** : être capable de transposer, d'interpréter et d'extrapoler à partir de certaines connaissances.
3. **Application** : être capable de se rappeler de connaissances ou de principes pour résoudre un problème.
4. **Analyse** : être capable d'identifier les éléments, les relations et les principes d'organisation d'une situation.
5. **Synthèse** : être capable de produire une oeuvre personnelle après avoir conçu un plan d'action.
6. **Évaluation** : être capable de porter un jugement critique fondé sur des critères internes ou externes.

Et voici leurs correspondances dans le cas du cours *ImaPro* :

1. **Acquisition de connaissances.** Ce point concerne principalement la partie théorique du cours, durant laquelle l'étudiant reçoit des informations en mode passif. L'acquisition de connaissances est également active lors de la révision du cours à la maison et durant les séances d'exercices.
2. **Compréhension.** L'acquisition n'est rien sans la compréhension. Les méthodes de traitement d'image nécessitent une étude approfondie et réfléchie afin d'être comprises, elles se basent généralement sur d'autres connaissances préalablement acquises. Le support d'*ImaPro* permet de faciliter cette compréhension de l'étudiant grâce à un environnement d'apprentissage interactif.
3. **Application.** C'est l'élément clef ouvrant la voie vers le renouveau du cours traitement d'image. Les étudiants bénéficient aujourd'hui d'un framework leur permettant d'appliquer concrètement leur savoir afin de résoudre un problème donné. Ainsi, à partir de ses connaissances préacquises, un élève peut implémenter diverses méthodes de traitement d'image grâce à un environnement de développement assisté.
4. **Analyse.** Lorsque le travail d'implémentation devient plus conséquent, ou que les connaissances théoriques deviennent vastes, l'analyse permet de cerner les éléments d'un problème et de les hiérarchiser. La connaissance gagne en niveau d'abstraction.
5. **Synthèse.** La synthèse naît dès l'instant où un étudiant décide de développer une nouvelle méthode de traitement d'image non acquise par l'apprentissage. Il devra mettre en oeuvre toutes ses connaissances concernant le domaine afin d'atteindre son objectif.
6. **Évaluation.** L'étape la plus élevée dans la connaissance est sans doute l'évaluation. Le chercheur, le professeur, l'assistant ou l'étudiant est devenu expert dans le domaine, il peut maintenant porter des jugements fondés sur un sujet qu'il maîtrise parfaitement.

Qu'en est-il de ma situation de formation aux niveaux du micro et du mésosystème ? Le modèle META (Matière, Environnement, Temps et Apprentissage), présenté dans le *module A6* du cursus de didactique universitaire [CHA06a], propose une telle lecture en définissant les deux systèmes concernés comme suit :

- Microsystème : il se trouve au niveau de la classe, du cours.
- Mésosystème : il fait référence aux éléments de la communauté éducative proche.

Dans ce contexte, le macrosystème n'entre pas en ligne de compte. En effet, "le but de ce modèle est d'aider le formateur à identifier les éléments qui composent les situations de formation, à les catégoriser, à identifier des relations entre ces éléments" (extrait de [CHA98a]).

Le modèle META, étudié en détail durant les cours, permet d'aborder un système sous différents angles. Comme la lumière décomposée par un prisme, le système est partitionné au travers du modèle META. L'analyse des diverses composantes prises séparément peut offrir une vision plus claire des entités du système, ainsi que leurs interactions. L'ajout de la dimension temporelle offre alors une vision systémique des micro- et mésosystèmes. Sans le citer explicitement, le modèle META a été au centre de mes réflexions lors des diverses étapes de renouvellement de mon cours, une approche implicite, naturellement ancrée dans mes pensées.

## 5.2 Le mot de la fin

Il y a encore quelques semaines de cela, je gardais en moi un sentiment d'incertitude quant à la rédaction de ce travail. Un mois a passé depuis lors et je constate avec grande satisfaction que je suis finalement arrivé au terme de ce diplôme de didactique universitaire. Pour atteindre cet objectif, il a été nécessaire de faire certaines concessions et mettre entre parenthèses mon doctorat quelques temps afin de pouvoir me consacrer entièrement à cette rédaction.

Après avoir vécu plus de trois années à renouveler un cours universitaire, l'avoir vu évoluer, grandir et devenir adulte, je dois admettre que le résultat final est très satisfaisant. Le cours *ImaPro* émerge enfin de sa nymphe, à l'image d'un papillon émergeant d'une chrysalide elle-même fruit d'une chenille. Une métamorphose somme toute intuitive dans laquelle mon professeur et moi avons consacré des centaines d'heures. Bien évidemment, ce processus ne s'est pas déroulé sans anicroche, il y a eu des hauts et des bas, certains désaccords sont survenus en cours de route entre mon professeur et moi. Par exemple, lorsqu'il exigeait un travail trop conséquent de ma part, ou qu'il insistait sur des détails futiles à mes yeux. Sans oublier les requêtes irréalistes, comme développer un système de *plug-ins* pour le framework; un tel système est généralement présent dans les produits informatiques à grande distribution.

S'il y avait un terme à retenir dans tout cela, le voici : apprentissage. Finalement, dans cette histoire, c'est moi qui ai le plus acquis de connaissances. J'ai appris à élaborer un cours complet, j'ai déployé un espace de partage en ligne, j'ai développé un framework pour le cours dans lequel j'y ai implémenté en langage *Java* toutes les méthodes de traitement d'image étudiées. Grâce à cette quantité de travail effectuée, mon savoir a considérablement augmenté. Je suis aujourd'hui capable d'évaluer la connaissance d'autrui, de porter un jugement critique sur des méthodes de traitement d'image développées par des pairs. En effet, j'ai acquis une expertise pointue dans le domaine, un savoir d'évaluation : le sixième niveau de la pyramide de Benjamin Bloom.

## 6 Bibliographie

WIK07a: Wikipedia, Terminologie du mot didactique, 2007, <http://fr.wikipedia.org/wiki/Didactique>

WIK07b: Wikipedia, A picture is worth a thousand words, 2007, [http://en.wikipedia.org/wiki/A\\_picture\\_is\\_worth\\_a\\_thousand\\_words](http://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words)

UNI06a: Université de Fribourg, L'informatique est un domaine récent, 2006, [http://www.unifr.ch/informatics/studium/info/einleitung/default\\_f.php](http://www.unifr.ch/informatics/studium/info/einleitung/default_f.php)

CHA06b: B. Charlier, H. Platteaux, Module B4 - Introduction des TIC dans un système de formation, 2006

WEB07a: Jambit, Family Tree of Important Programming Languages, 2007, <http://www.jambit.com/en/jambit/java.html>

CHA05a: B. Charlier, Module A - Conception de l'enseignement et de l'apprentissage, 2005

CHA06a: B. Charlier, H. Platteaux, Module A6 - Usage des technologies pour l'enseignement et les apprentissages, 2006

CHA98a: B. Charlier, Journée EAD-FASEG - Questionner un système de formation à distance avec le modèle META, 1998