

N'D, une interface logicielle pour découvrir et expérimenter le traitement des images et de la vidéo

M. Donias

¹ENSEIRB, 1 av. du Dr. A. Schweitzer, B.P. 99, F-33402 Talence Cedex

RESUME Initier des étudiants au traitement du signal et leur permettre d'acquérir des compétences réelles en implémentation d'algorithmes nécessite le choix d'un outil pédagogique adapté. *N'D* est une interface logicielle, centrée sur le traitement des images et des vidéos, qui répond à de telles préoccupations : elle permet de se familiariser avec les notions de base, notamment à l'aide d'outils interactifs, tout en expérimentant ses propres développements implémentés en Langage C. Dans cet article, plusieurs aspects de l'interface sont brièvement présentés. Un exemple de réalisation ainsi qu'un bilan de retour d'expérience réalisé auprès d'étudiants de l'ENSEIRB est également exposé avant quelques perspectives d'évolution.

Mots clés : traitement du signal, image, vidéo, interface logicielle, langage C

1 INTRODUCTION

L'enseignement du traitement du signal au sens large repose habituellement sur des cours fondamentaux complétés de travaux dirigés. Dans un contexte de plus en plus numérique et d'omniprésence de ce qui est à présent désigné par le terme multimédia, des travaux pratiques ou encore des projets s'ajoutent nécessairement à ces enseignements théoriques. Il semble en effet inconcevable aujourd'hui de sensibiliser fortement des étudiants et les familiariser aux concepts de base du traitement du signal sans leur faire expérimenter quelques d'algorithmes à l'aide d'un ordinateur. Or, toute implémentation, dans un cadre pédagogique, suppose l'utilisation d'une interface logicielle. Ceci est naturellement encore plus vrai dans le cas très particulier des images ou de la vidéo. Il se pose alors à l'enseignant le choix de l'outil le plus adapté.

Parmi toutes les solutions disponibles, *N'D* est une interface logicielle qui se caractérise par une forte interactivité et qui permet d'illustrer aisément des notions de base ou d'expérimenter des traitements plus complexes. Initialement développée dans un contexte de recherche et pour des besoins personnels, *N'D* est à présent utilisée dans le cadre de quelques enseignements que je dispense à l'ENSEIRB et également pour des projets ou des stages. Les publics concernés sont des étudiants de 2^{ème} et 3^{ème} année s'étant orientés dans une spécialisation dans le traitement du signal auxquels s'ajoutent quelques étudiants du Master 2 EEA APSI option signal et image de l'université Bordeaux 1. L'objectif pédagogique clairement affiché par ce choix est double : sensibiliser (aspect découverte) et acquérir des compétences d'implémentation en Langage C requises par les entreprises (aspect expérimentation).

Il est important de noter que *N'D* n'est pas spécifiquement une interface d'apprentissage. Il s'agit plus généralement d'un outil de recherche et de transfert vers des partenaires industriels. Utilisé par plusieurs

membres (stagiaires, doctorants ou enseignant-chercheurs) du groupe Signal du LAPS, l'interface permet, en effet, de visualiser des données, de tester, de valider et de diffuser des algorithmes.

Cet article se poursuit par une brève description de *N'D* : les deux volets, découverte et expérimentation, sont présentés ainsi qu'une comparaison avec quelques logiciels habituellement utilisés pour l'enseignement du traitement du signal. Suit un exemple de réalisation d'étudiants ainsi que le bilan d'un retour d'expérience. Quelques perspectives d'évolution de l'interface sont enfin présentées dans la dernière partie.

2 L'INTERFACE LOGICIELLE N'D

2.1 Présentation générale

N'D (voir figure 1) est une interface logicielle de visualisation et de traitement d'images (2D/3D) ou de vidéos (flux enregistrés ou temps-réel) sous Windows.

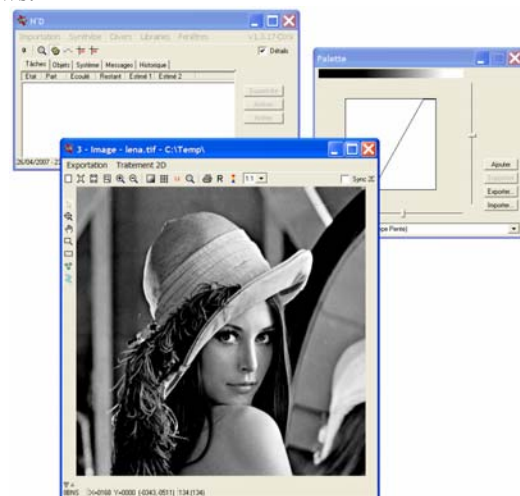


Fig 1 : *N'D*.

Elle prend en charge les formats de données (bmp, jpeg, tiff, png, avi, etc) les plus courants.

Il s'agit d'une interface au caractère WIMP (Window, Icon, Menu, Pointing device) très prononcée dans laquelle chaque action est déclenchée au moyen d'une sélection de fenêtre, d'un clic, d'un choix d'une entrée d'un menu, etc. Son principe de fonctionnement est «contextuel» : un traitement est exécuté à partir d'un menu situé dans la fenêtre de l'image ou de la vidéo concernée. Les différents outils disponibles, présentés dans la suite de l'article, procurent une forte interactivité.

N'D est modulaire : l'interface peut être enrichie de nouveaux traitements implémentés en Langage C et développés sous la forme de bibliothèques. La gestion des modules ainsi créés est dynamique : les nouvelles fonctions peuvent en effet être ajoutées mais également enlevées « à la volée ». Elles sont de plus exécutées dans un mode multitâche : l'interface reste utilisable durant l'attente d'un résultat ce qui présente un intérêt certain pour les algorithmes coûteux en temps de calcul.

2.2 Une interface de découverte

N'D se compose d'un certain nombre d'outils interactifs à caractère fortement pédagogique. Il n'est naturellement pas possible dans cet article de les décrire de manière exhaustive. A titre d'exemple, en dehors d'un outil de sélection de palette classique, on peut citer la fenêtre de correction d'intensité (voir figure 2) qui permet de régler précisément les paramètres d'une transformation linéaire appliquée à l'image ou de les contrôler en déplaçant des curseurs directement sur son histogramme.

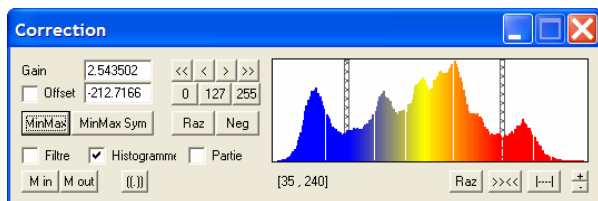


Fig 2 : Outil de correction d'intensité.

La loupe (voir figure 3), dans une forme numérique, adaptative ou encore 3D est un autre outil qui permet de se rendre compte de manière optimale du contenu d'une image.

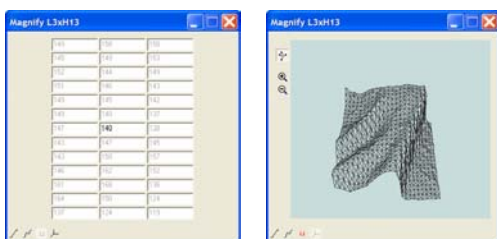


Fig 3 : Loupe.

Enfin, un curseur 2D, allié à un déplacement synchronisé de la partie visualisée de plusieurs images, permet de confronter efficacement le résultat d'un algorithme de traitement à l'image d'origine. Cette fonctionnalité (voir figure 4) n'existe, à ma connaissance, dans aucune autre interface.

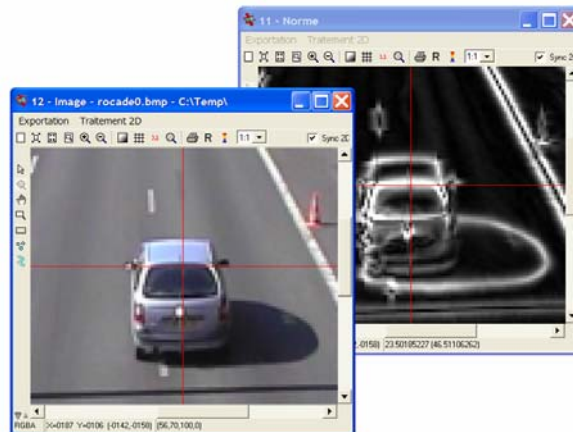


Fig 4 : Curseur 2D et déplacement synchronisé.

Il est à noter que l'interface dispose, toujours de manière contextuelle, de traitements basiques, bien utiles, opérant sur une (valeur absolue, transformations géométriques, etc) ou plusieurs (addition, extrémum, etc) images.

2.3 Une interface d'expérimentation

N'D est modulaire. Il est en effet possible de lui ajouter très simplement de nouveaux traitements en glissant-déplaçant, par exemple, des bibliothèques (fichiers binaires d'extension .dll) spécifiquement générées à cet effet. Ceci permet à la fois de bénéficier de bibliothèques d'algorithmes développés par d'autres ou encore d'expérimenter ses propres algorithmes. Les modules intégrés ajoutent de nouvelles entrées dans les menus de l'interface principale ou des fenêtres spécifiques (images, vidéos, etc). Leur gestion est dynamique : les bibliothèques sont chargeables et déchargeables à volonté sans nécessiter un arrêt de l'interface.

D'un point de vue de l'implémentation, il s'agit d'écrire, en Langage C, le code d'une bibliothèque (d'extension .dll) qui peut être générée par un outil classique tel que *Microsoft Visual Studio*, *Dev-Cpp*, *CodeBlocks*, etc. Un canevas complet (fichiers source et de compilation) à compléter est fourni pour simplifier l'opération et comporte notamment un fichier qui liste toutes les fonctions reconnues par l'interface.

Le principe de fonctionnement de l'interface étant contextuel, l'ajout d'un nouveau traitement s'obtient par quelques lignes de code qui s'organisent en deux parties distinctes :

- la spécification d'une entrée supplémentaire dans le menu concerné (image, vidéo, etc).

- la définition d'une fonction exécutée lors de la sélection du traitement et qui met en oeuvre l'algorithme proprement dit.

Un court exemple d'implémentation est le suivant :

```
menu = ndMenuCreate("Filtres");
ndMenuItem(menu, "Moyenneur", On_Menu);
ndLibraryAddMenu(id, "MENU_2D", menu);
```

Ces trois lignes permettent de créer et d'ajouter une nouvelle rubrique (ou menu), ici intitulée « Filtres » dans le menu de traitement d'une image, identifié par la constante MENU_2D. La sous-rubrique intitulée « Moyenneur » est associée au pointeur d'une fonction On_Menu qui est exécutée lors du lancement du traitement dans l'interface. Plusieurs sous-rubriques peuvent être insérées au sein d'une même rubrique en utilisant autant de fois que nécessaire la fonction ndMenuItem.

Un exemple de code permettant l'exécution d'un traitement simple tel qu'un moyeneur 2D est le suivant :

```
id = ndImageGetIn();
ndImageGetSize(id, &width, &height);
in = (double *)ndImageGetData(id,
    TYPE_DOUBLE);
id = ndImageNew();
ndImageSetSize(id, width, height);
ndImageSetTypeData(id, TYPE_DOUBLE);
ndImageShow(id);
out = (double *)ndImageGetPtrData(id);
moyeneur(in, width, height, out, size);
```

Il apparaît tout d'abord la déclaration des entrées (une unique image dans cet exemple) qui sont dépilées une à une (ndImageGetIn) et représentées par un identifiant. Leurs caractéristiques (dimensions et pointeurs de donnée au format numérique souhaité) sont extraites par les fonctions ndImageGetSize et ndImageGetData. Une image de sortie est ensuite créée à l'aide de ndImageNew suivie de ndImageShow dès lors que les dimensions et le format numérique de codage ont été définis. Une fois le pointeur de donnée de sortie obtenu, le traitement proprement dit est exécuté.

Le traitement de vidéos est tout aussi aisé comme l'illustre l'extrait de code suivant :

```
id = ndVideoGetId();
ndVideoSetFormatIn(id, TYPE_RGBA);
ndVideoSetFormatOut(id, TYPE_RGBA);
ndVideoStart(id, videoOn, videoOff);
```

videoOn et videoOff sont ici respectivement la fonction de traitement de chaque image de la vidéo et la fonction exécutée une seule fois lors de l'arrêt complet du traitement.

Une caractéristique importante de N'D est qu'il est possible de définir plusieurs fonctions de traitement au sein d'une même librairie. Quelques tutoriels mettant en oeuvre les principales fonctionnalités sont disponibles.

2.4 Alternatives

L'enseignement pratique du traitement du signal dans les écoles d'ingénieur, à l'université, ou encore dans les IUT, est le plus souvent dispensé à l'aide des logiciels *Matlab* [2] ou *Labview* [3].

Matlab est un environnement complet de développement : s'appuyant sur un langage interprété de haut niveau, il permet d'exécuter des calculs et des fonctions au moyen d'une ligne de commande ou de scripts, de visualiser l'état de variables, de réaliser des affichages évolués, etc. Axé principalement sur le calcul matriciel, *Matlab* voit ses possibilités fortement étendues par l'ajout de bibliothèques complémentaires spécialisées (« toolboxes ») qui concernent, notamment, le signal ou les images. Il existe un certain nombre de logiciels très similaires à *Matlab* tels que *Scilab* [4] (gratuit développé par l'INRIA et offrant une compatibilité importante avec *Matlab*), *Octave* [5], *Aphelion* [6] ou encore *IDL* [7].

Historiquement dédié à la mesure et à la commande, développé par National Instruments, *Labview* est très différent. Il s'agit en effet d'un logiciel de développement basé sur un langage de programmation graphique dénommé Langage G. Tout algorithme se présente alors visuellement sous la forme de modules reliés entre-eux par des fils et disposant chacun d'une interface d'accès aux paramètres. La réunion des différentes interfaces constitue au final une sorte de tableau de bord d'un instrument virtuel.

Ces outils sont aujourd'hui devenus des standards utilisés aussi bien dans l'enseignement que dans les entreprises. Ils partagent une différence marquante avec N'D : l'implémentation en Langage C, bien que permise, n'est pas native. C'est ainsi que, dans un objectif d'accélération d'algorithmes dont la nature matricielle est très peu marquée, l'implémentation en Langage C est possible sous *Matlab* sous la forme de librairies (« mex-files »). En pratique, leur réalisation est rendue difficile et fastidieuse, principalement en raison de l'impossibilité d'écrire plusieurs fonctions au sein de la même librairie. Sous *Aphelion*, il est de plus nécessaire de disposer de droits d'administration pour ajouter de nouveaux traitements au sein de l'interface.

Par ailleurs, de manière plus anecdotique, à l'exception de *Scilab* et *Octave*, les produits évoqués précédemment sont payants, même si leur coût est toujours moins élevé dans un contexte pédagogique.

Dans une démarche radicalement différente, le traitement du signal peut également être expérimenté à l'aide de bibliothèques de traitement, riches en algorithmes et peu axées sur les aspects interface. Il en est

ainsi de *OpenCV* [8], une librairie « libre », initialement développée par Intel, implémentée en Langage C et plus particulièrement dédiée au traitement temps-réel de vidéos. Ici, contrairement par exemple à *N'D*, l'application finale ne s'exécute pas au sein d'un ensemble d'outils graphiques préexistants. Elle consiste, en effet, en un exécutable autonome à l'interface réduite au strict minimum.

3 UTILISATION PAR LES ETUDIANTS

N'D est utilisé dans le cadre de projets avancés (semestre 4 ou 5) à l'ENSEIRB par des groupes de 4 étudiants. Parmi les différentes applications déjà développées, le projet intitulé « souris 3D » tire parti des possibilités d'intégration (entrées/sorties) et de traitement d'une source vidéo offertes par *N'D*.

L'objectif est, dans cet exemple, de réaliser une interface homme-machine 3D et plus précisément un périphérique de manipulation. L'application s'appuie sur un objet (voir figure 5), entièrement assemblé par un technicien de l'ENSEIRB, constitué de 3 axes orthogonaux et de quelques boules colorées. Manipulé par un opérateur selon les six axes de liberté possibles (translation et rotation), il est filmé et « instrumenté » par une caméra. La posture 3D (position et orientation) peut alors être utilisée en tant que contrôle 3D, pour piloter, par exemple, une scène 3D par déplacement du point d'observation ou des objets présents.

D'un point de vue algorithmique, il s'agit principalement de reconstruire et suivre un objet 3D dont le modèle est connu a priori. L'application permet ainsi de mettre en œuvre différentes notions liées au traitement des images et de la vidéo : calibration de caméra, détection de points remarquables, traitement de la couleur, calcul d'une transformation de type homographie, reconstruction 3D.

La figure suivante fait apparaître les différents constituants du projet : une acquisition de flux vidéo issu d'une webcam (fenêtre Caméra In, en haut à droite), une détection des différents points colorés de la « souris 3D » (fenêtre Caméra Out, en bas à droite) ainsi que, à titre de validation, une reconstruction du modèle de l'objet observé (fenêtre Démo, à gauche).

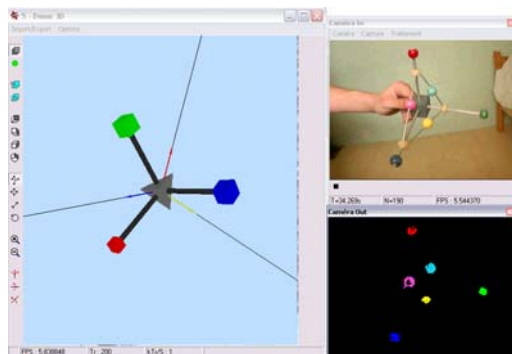


Fig 5 : Projet « souris 3D ».

Afin de bénéficier de la perception de *N'D* par les étudiants, un retour d'expérience a été demandé à ceux qui l'ont utilisé de manière « suffisamment intensive ». Parmi les points positifs identifiés, on peut citer :

- la « légèreté » de l'interface qui est constituée de seulement quelques fichiers déplaçables et ne nécessite aucune « installation ».
- la gratuité.
- l'aspect contextuel : les traitements sont exécutés à partir de menus situés dans les fenêtres concernées.
- la forte interactivité offerte par des outils riches et dynamiques.
- la complémentarité à *Matlab*.
- l'implémentation en Langage C qui, tout à la fois, permet de conforter ses connaissances dans un langage requis par les entreprises, conduit à des traitements efficaces, oblige à une compréhension certaine des algorithmes et apporte une coloration « moins pédagogique » que *Matlab* ce qui suscite en définitive davantage d'intérêt.
- la potentialité à s'enrichir et à devenir une bibliothèque d'algorithmes.

A contrario, le principal grief concerne justement ce dernier point c'est-à-dire l'absence de bibliothèques de traitements basiques, obligeant l'utilisateur à tout implémenter par ses propres moyens. Un autre point négatif est l'apprentissage court mais indispensable du mode de fonctionnement des librairies avant toute implémentation.

4 CONCLUSION

N'D est un logiciel qui permet de s'initier au traitement du signal et de réaliser, au prix d'un effort initial très raisonnable, ses propres expérimentations ou développements.

Parmi les caractéristiques particulièrement appréciées par les étudiants, figurent sa « légèreté », la diversité des outils interactifs disponibles, l'intégration de multiples sources (image, vidéo, etc), la modularité ainsi que l'efficacité des algorithmes développés du fait d'une implémentation obligatoire en Langage C.

Il est à noter que, dans un cadre pédagogique, *N'D* ne constitue en aucun cas un concurrent potentiel de produits existants tels que *Matlab*. Il souffre, par exemple, de l'absence d'un langage interprété qui offre une souplesse incomparable d'expérimentation. Il s'agit d'un outil complémentaire qui dispose d'interfaces particulièrement adaptées au traitement des images ou de la vidéo.

En constante évolution, *N'D* s'enrichit au fil du temps de nouveaux outils suscités par les besoins de ses utilisateurs. Une passerelle vers *Matlab*, permettant un échange dynamique de données, vient ainsi de lui être ajouté tout récemment. Il gagnerait naturellement,

à court terme, à être complété par un ensemble de bibliothèques d'algorithmes basiques de traitement d'images. Ceci permettrait la démonstration ou l'expérimentation de traitements plus complexes. Un point important à améliorer, dans un souci de pédagogie, concerne également la documentation qui se limite, pour l'instant, à quelques tutoriels trop peu commentés.

5 REFERENCES

Cette section regroupe quelques liens Internet vers le site Web de *N'D* et ceux de logiciels ou de bibliothèques utilisés pour le traitement d'images et de vidéos.

- [1] N'D - <http://perso.wanadoo.fr/m.donias>
- [2] Matlab – <http://www.mathworks.fr>
- [3] Labview – <http://www.ni.com/labview/f>
- [4] Scilab – <http://www.scilab.org>
- [5] Octave – <http://www.gnu.org/software/octave>
- [6] Aphelion - <http://www.adcis.net/index.html>
- [7] IDL - <http://www.ittvis.com/idl>
- [8] OpenCV -
<http://www.intel.com/technology/computing/opencv>