

Divers modules en Traitement d'Image  
Patrick Bonnin  
Année 2016 - 2017

## Comment Rédiger un Rapport de TP en Traitement d'Image :

### 1 Introduction

Le but de cette note est de préciser les règles essentielles à respecter lors de la rédaction d'un rapport technique comportant une importante composante informatique, comme par exemple en *Traitement d'Image* : le développements d'opérateurs.

### 2 Structure Globale du Rapport

Le rapport doit comporter au minimum :

- une introduction ;
- un développement ;
- une conclusion ;
- et éventuellement des annexes.

Partant d'une idée générale personnelle (par exemple ce qu'est le Traitement d'Image pour vous), et en guidant le lecteur selon le principe de l'entonnoir, l'introduction permet de présenter l'objet du rapport.

La conclusion vous permet de récapituler les compétences acquises au cours du projet, et de donner votre avis sur l'exercice.

Le développement vous permet de montrer vos compétences techniques aussi bien sur les plans :

- du Traitement d'Image : Conception d'Opérateurs ;
- de l'implantation en Langage Informatique ;
- que de la correcte utilisation des opérateurs, et notamment au niveau du réglage de leurs paramètres.

C'est également l'occasion de prouver vos compétences de communiquant à l'écrit. Ainsi, une partie *Documentation Commerciale* présentera les divers opérateurs implantés, non pas listés sous forme de catalogue<sup>1</sup>, mais avec une classification des fonctionnalités. Plusieurs opérateurs peuvent réaliser la même fonctionnalité : par exemple un *Lissage d'Image* peut être réalisé par les opérateurs moyen, gaussien et médian. Certaines fonctionnalités sont complémentaires : l'érosion, la dilatation, l'ouverture et la fermeture. Dans ces deux cas précis, les opérateurs seront regroupés, et leurs résultats comparés.

**En effet, au travers du rapport vous devez prouver vos compétences techniques, aussi bien de réflexion, de conception que de réalisation.**

#### 2.1 Eléments Essentiels du Rapport

Les éléments essentiels du rapport sont :

---

1. Nous ne sommes pas aux 3 Suisses ou à la Redoute !

- le *Cahier des Charges* : c'est le sujet du projet, mais également la demande du client, dans le cadre d'un réel projet industriel ;
- l'analyse permettant s'arriver aux *Spécifications* détaillées, à partir d'un cahier des charges "léger". C'est un élément essentiel du *Cycle en "V"*.
- pour la composante informatique :
  - le *Principe* de l'algorithme : c'est-à-dire en quelques mots les caractéristiques de la solution proposée,
  - l'*Algorithme* : en *Langage Naturel*, c'est-à-dire en français, incluant les principales structures algorithmiques : tests, boucles etc ... Dans certains cas, l'algorithme peut être présenté selon une formalisation mathématique,
  - *Extraits de Code Source* : le code source réalisé sera présenté étape par étape ou élément essentiel par élément essentiel, en suivant le découpage de l'algorithme. Cette partie est composée d'éléments rédactionnels illustrés de code source.
  - *Code Source* en globalité : en Annexe du rapport, si demandé, sinon dans une archive informatique.
- la *Présentation des Résultats* : les images des résultats des opérateurs implantés. Il ne faut pas oublier de présenter l'image à traiter avant de présenter les résultats de traitement ! Comme préalablement mentionné, les résultats de plusieurs opérateurs réalisant soit la même fonctionnalité (par exemple le lissage), soit des fonctionnalités complémentaires (par exemple les seuillages classique, hat et multiple, ainsi que les opérateurs de morphologie mathématique : érosion, dilatation, ouverture, fermeture) seront présentés en même temps, de manière comparative ;
- les *Commentaires* sur les résultats obtenus. Ceux-ci sont ils conformes à votre attente. Cette partie, illustrant la remontée du Cycle en "V" de la conception d'un produit, y compris un logiciel informatique, est nécessaire.

## 3 Exemple pour un Opérateur Simple de Traitement d'Image : le Seuillage Classique

### 3.1 Cahier des Charges

Le but de l'opérateur de *Seuillage Classique* est de séparer :

- les objets d'intérêt ;
- du fond ;

sachant qu'un objet d'intérêt se distingue notablement du fond, de par sa luminance, plus claire ou plus foncée.

### 3.2 Principe

L'histogramme de ce type d'image est généralement bi-modal. Le mode le plus important correspond généralement au fond, le plus petit aux objets à détecter.

Il faut rechercher expérimentalement la valeur du paramètre de contrôle appelé *Seuil* permettant de séparer les deux modes de l'histogramme.

Le principe de l'opérateur de seuillage consiste à affecter la valeur logique :

- "1" (ou la valeur 255) aux pixels dont la valeur de la luminance est supérieure ou égale à la valeur du seuil ;
- "0" (ou la valeur 0) sinon.

Ceci est valable dans le cas le plus probable où les objets sont plus clairs que le fond. Dans le cas inverse, il faudra, au préalable faire une *inversion vidéo*.

### 3.3 Algorithme

L'opérateur est composé du balayage vidéo simple, et du traitement du pixel courant.

L'algorithme du balayage vidéo simple est le suivant :

```
* Pour y = 0 à y = nlig - 1
* Pour x = 0 à x = ncol - 1
  ° I(x,y) // pixel de l'image source
  ° B(x,y) // pixel de l'image binaire résultat
```

L'algorithme du Traitement du pixel courant est le suivant :

```
° Si I(x,y) >= S
  - B(x,y) = "1"
° Sinon
  - B(x,y) = "0"
```

La formulation mathématique correspondante est présentée figure n° 1.

Ainsi, l'algorithme complet est le suivant :

```
* Pour y = 0 à y = nlig - 1
* Pour x = 0 à x = ncol - 1
  ° Si I(x,y) >= S
    - B(x,y) = "1"
  ° Sinon
    - B(x,y) = "0"
```

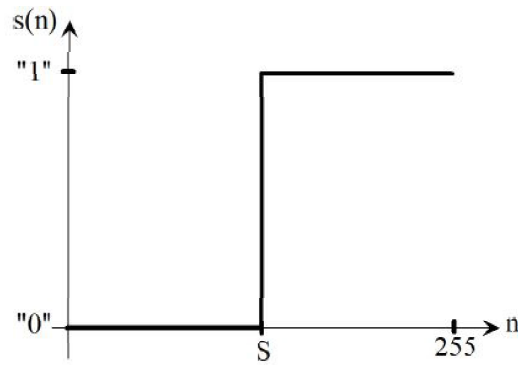


FIGURE 1 – Seuillage Classique.

### 3.4 Codes Source

Le code source du balayage vidéo simple est le suivant :

```
for(POINT_Y(point) = 0; POINT_Y(point) < NLIG(image);
    POINT_Y(point)++)
for(POINT_X(point) = 0; POINT_X(point) < NCOL(image);
    POINT_X(point)++)
{
    PIXEL(image, point) ... ; // Acces au Pixel de l'image Source
    PIXEL(imres, point) ... ; // Acces au Pixel de l'image resultat
}/* --- End of the Image Video Scan --- */
```

Le code source du traitement du pixel courant est le suivant :

```
if(PIXEL(image, point) >= Th)
    PIXEL(imres, point) = 255; // Level "1"
else
    PIXEL(imres, point) = 0;    // Level "0"
```

Enfin, le code source de l'opérateur est le suivant :

```
for(POINT_Y(point) = 0; POINT_Y(point) < NLIG(image);
    POINT_Y(point)++)
for(POINT_X(point) = 0; POINT_X(point) < NCOL(image);
    POINT_X(point)++)
{
    if(PIXEL(image, point) >= Th)
        PIXEL(imres, point) = 255; // Level "1"
    else
        PIXEL(imres, point) = 0;    // Level "0"
}/* --- End of the Image Video Scan --- */
```

### 3.5 Présentation des Résultats

Les résultats de l'opérateur de *Seuillage Classique* sont présentés sur une image industrielle, où les objets à détecter sont des ampoules de niveau de gris plus clair que le convoyeur plus foncé (cf figure n° 2).

Nous présentons l'image source, son histogramme, ainsi que l'image résultat.

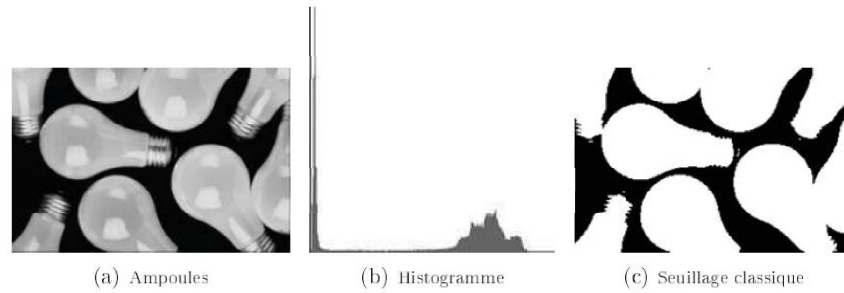


FIGURE 2 – Résultats du Seuillage Classique.

### 3.6 Commentaires sur les Résultats

L'histogramme de l'image source est parfaitement bi-modal, avec une franche séparation des deux modes. Cette image représente un cas parfait d'utilisation de cet opérateur. On peut remarquer une importante latitude de réglage de la valeur du seuil, tant que celle-ci ne se rapproche pas trop de l'un des deux modes de l'histogramme. Seuls, quelques pixels du bord des ampoules vont apparaître ou disparaître selon le réglage.

Les ampoules sont globalement correctement détectées. En revanche les compter est une opération plus complexes. En effet, celles-ci apparaissent collées. Deux solutions sont possibles :

- la première de type mécanique, où l'on suppose qu'un dispositif puisse permettre de les séparer. Dans ce cas, un simple opérateur de décomposition en composantes connexes (ou "blob detection") peut permettre leur comptage ;
- la seconde de type algorithmique : il faut adjoindre à la décomposition en composantes connexes précédente une procédure de reconnaissance de forme, permettant de compter les ampoules, même lorsque celles-ci sont en amas.