

ESIR, 2^e année

TP 1 - Introduction au codage d'images : EQM, PSNR, entropie, histogramme

O. Le Meur

Mots clés

OpenCV, histogramme, erreur quadratique moyenne, PSNR, entropie.

1 Introduction

1.1 Objectifs du TP

Ce TP a plusieurs objectifs :

- la prise en main de la librairie OpenCV ;
- la prise en main et la manipulation des outils permettant d'évaluer l'efficacité de codage. On travaillera sur la norme de compression JPEG principalement.

1.2 Notation

Le tp sera noté en fonction des critères suivants :

- La qualité du compte-rendu au format pdf (**le nom du pdf sera composé des noms des auteurs**). Un bon compte-rendu comporte une introduction, une conclusion (voire une discussion) et les réponses aux questions. Un tableau de valeur sans commentaire est inutile, et pas noté. La capacité à critiquer et à analyser les résultats est fondamentale. Faites preuve de sens critique (à bon escient) ;
- La qualité de l'écriture des programmes et la facilité à les lire-comprendre ;
- Le compte rendu doit être rendu à la date indiquée.

1.3 Matériel fourni

Le matériel fourni est le suivant :

- un exemple de Makefile et un fichier main.cpp afin de faciliter la prise en main d'OpenCV ;
- quelques fonctions utiles pour vous aider à démarrer votre TP ;
- une image non dégradée et trois versions de cette image compressée avec différents niveaux de qualité.

2 Une très brève introduction au standard de compression JPEG

Le standard JPEG définit une norme pour la compression des images fixes couleurs.

Cette norme utilise la DCT (transformée en cosinus discrète) suivie d'une quantification des coefficients. En pratique, on utilise une DCT par blocs de taille 8×8 et une quantification uniforme des coefficients DCT. Les coefficients, en vue de la transmission, sont disposés dans un ordre dit *zig-zag*. On obtient une suite mono-dimensionnelle de coefficients, rangée selon un ordre tel que, en moyenne, cette suite est décroissante. Ceci permet d'effectuer de nombreux regroupements de coefficients nuls, et ainsi de coder efficacement chaque bloc par codage de plages (*run-length*). Ce schéma de compression est à peu près symétrique c'est à dire que les temps de compression et de décompression sont semblables.

3 Votre travail

3.1 Une vue générale

Votre travail consiste à étudier l'impact de la compression JPEG sur les images fournies. Cette étude consiste à :

- calculer l'erreur quadratique moyenne (EQM) entre deux images données en entrée de votre programme ;
- calculer et sauvegarder les cartes de distorsion ;
- calculer l'entropie des images avant et après compression ;
- calculer le PSNR des images ;
- visualiser les histogrammes.

3.2 PSNR et cartes d'erreur

Question 3.1. Mettre en place un programme lisant deux images et les convertir en YCrCb. Afficher les images. Afficher les différentes composantes et les sauvegarder dans un fichier pour le compte rendu.

Pour la suite, nous vous invitons à transformer votre variable de type Mat composée de 3 canaux en un vector < Mat > en utilisant la commande split(...).

Question 3.2. Calculer le PSNR sur chaque canal (une valeur de PSNR par canal).

Soient I_o et I_r l'image originale et reconstruite, de même taille $M \times N$, respectivement. La valeur de l'intensité lumineuse associée au pixel de coordonnées (i, j) d'une image I est noté $I(i, j)$. Le calcul du PSNR fait intervenir le calcul de l'erreur quadratique moyenne, l'EQM (ou MSE pour Mean Square Error) :

$$EQM(I_o, I_r) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_o(i, j) - I_r(i, j)]^2.$$

Le PSNR est donné par l'expression $PSNR(I_o, I_r) = 10 \log_{10} \frac{D^2}{EQM(I_o, I_r)}$, où D est la dynamique couverte par les variations de l'intensité lumineuse. Les pixels des images à traiter sont codés sur 8 bits, donc $D = 2^8 - 1 = 255$. L'unité du PSNR est le décibel et est noté dB. Plus celui-ci est important, plus les images comparées sont semblables.

Question 3.3. Visualiser la carte d'erreur calculée entre les deux images en entrée de votre programme. L'erreur est ici la simple différence entre les deux images. Enregistrer la carte dans un fichier afin de le mettre dans votre compte-rendu.

Attention, l'erreur est signée. Ne pas oublier de la recentrer sur 128 avant de l'enregistrer et de la visualiser. Par ailleurs, la carte d'erreur sera constituée des trois canaux. Utiliser la fonction merge pour fusionner des canaux, par exemple.

Question 3.4. A partir des 3 images dégradées :

1. calculer la valeur de PSNR ;
2. Tracer la courbe $\text{tailleFichierEnBit} = f(\text{PSNR})$. Commentez.
3. lorsque la compression devient très forte, qu'observez-vous sur les images ?

3.3 Entropie

Question 3.5. Calculez l'entropie des images dégradées. On utilisera uniquement le canal Y pour faire le calcul de l'entropie.

Vous pouvez utiliser la fonction computeHistogram donnée dans le fichier main pour effectuer le calcul.

Question 3.6. Tracer la courbe $\text{entropie} = f(\text{niveau de compression})$. Commentez l'allure de la courbe.

3.4 Histogramme

Question 3.7. A partir des fonctions mises à votre disposition, tracer les histogrammes de :

1. de l'image originale ;
2. des images dégradées

on le fera uniquement sur le canal Y

Question 3.8. Tracer les histogrammes des cartes d'erreur et calculer leur entropie. Commentez les valeurs et la forme des histogrammes (comparer avec les histogrammes de la question précédente).

Ne pas oublier de recentrer les cartes d'erreur sur 128....

Question 3.9. Calculer le kurtosis des 3 histogrammes des cartes d'erreur afin de caractériser l'aplatissement de la distribution des erreurs.

Étant donnée une variable aléatoire réelle X d'Espérance mathématique μ et d'écart type σ , on définit son kurtosis non normalisé comme le moment d'ordre quatre de la variable centrée réduite : $\beta_2 = E \left[\left(\frac{X-\mu}{\sigma} \right)^4 \right]$.