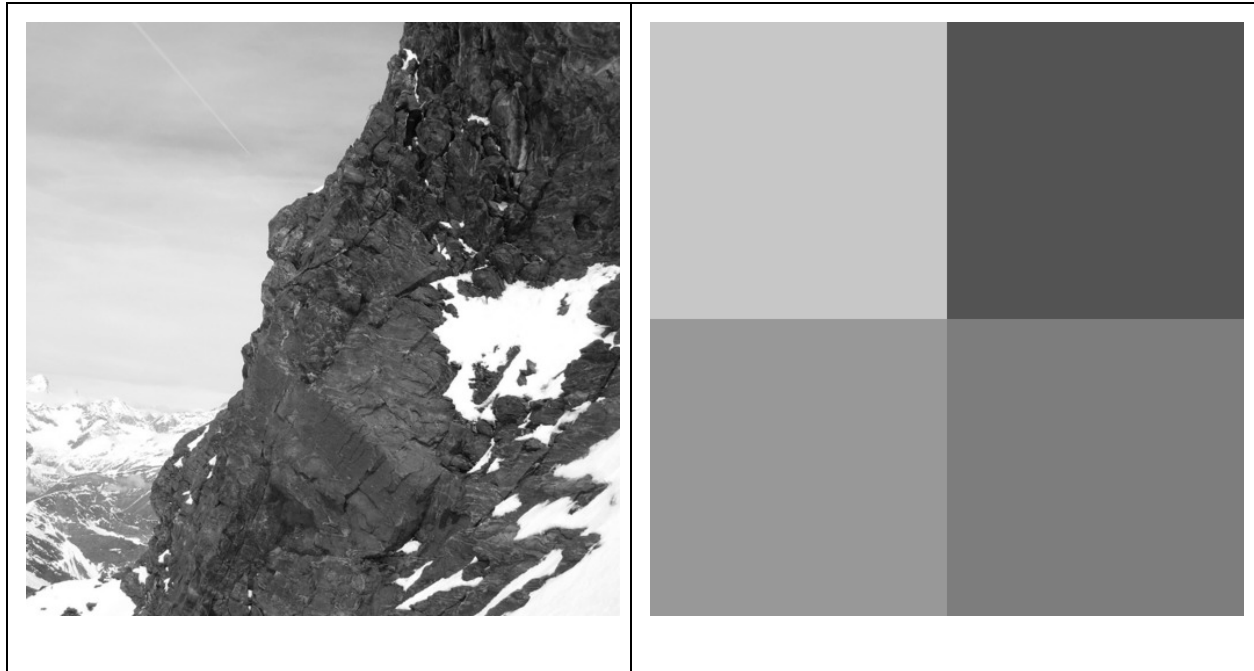


# TP5 Segmentation image par Split and Merge

*Stéphane Wouters 20110928*

## Division d'une image en 4 régions



Valeurs des gris : 199, 83, 152, 125

## Etape de division récursive

Pour la division, j'avais produit un premier algorithme qui découpé "physiquement" les images, c'est à dire qu'à chaque division une nouvelle image au format PGM était générée. C'est le suivant :

### Algorithme

```
ImageBase* divRecuratif (ImageBase* imIn, int seuil) {
    ImageBase** divide = divideImage(*imIn);
    for (int i = 0; i < 4; ++i) {
        int* variance = getVariance(*(divide[i]));
        if (variance[0] > seuil && im->getWidth() > 2)
            divide[i] = divRecuratif(divide[i], n, seuil);
        else {
            int* avg = getAvg(*(divide[i]));
            colorImage(divide[i], avg);
        }
    }
}
```

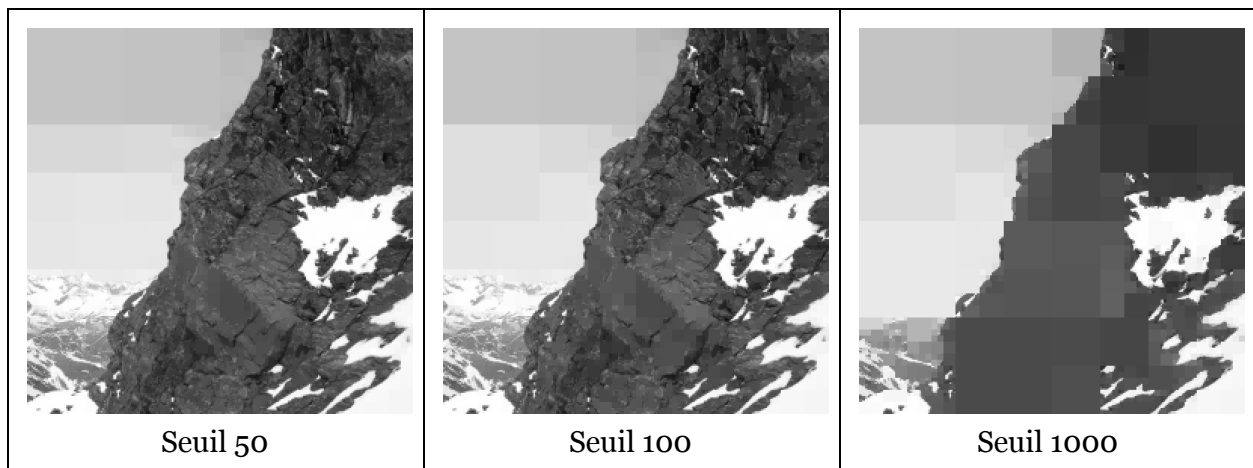
```

    ImageBase* merged = mergeImage(divide);
    return merged;
}

```

Pour des raisons de performances, je l'ai recommencé en utilisant un système de classe de Region qui contiennent chacun les nouvelles positions de l'image de référence. Le résultat est le même, mais cette fois on peut construire un arbre au cours des modifications (utiles pour la fusion).

## Résultat



## Fusion

Dans l'algorithme de décomposition, on stock à chaque fois les régions parentes et fils pour former un arbre. On parcourt ensuite l'arbre pour tester la ressemblance et fusionner les parties de l'images qui ont les mêmes couleurs.

```

void colorParents(Region* r, int* color) {
    r->colorRegion(color);
    if (r->parent != NULL) {
        colorParents(r->parent, color);
    }
}

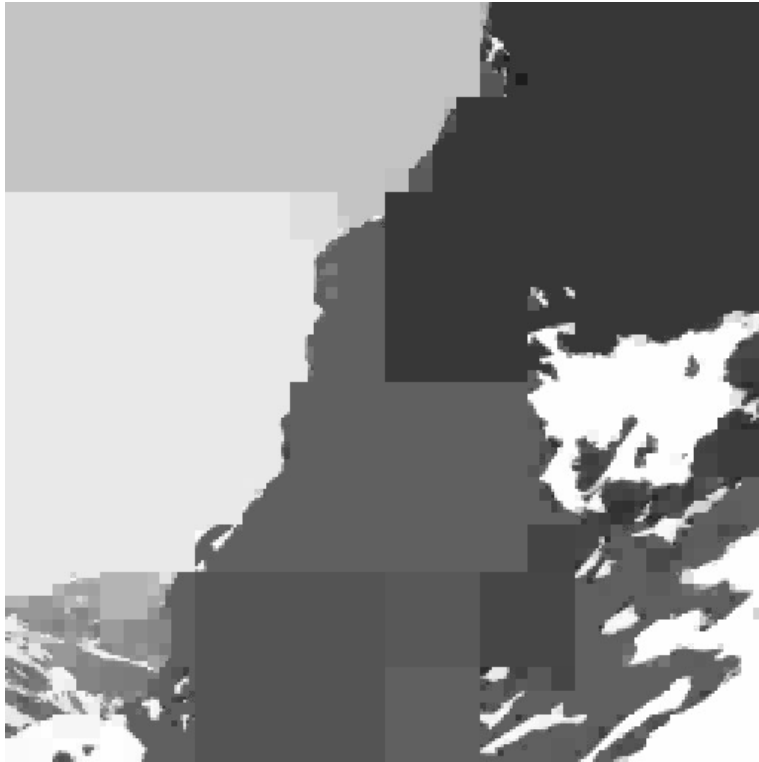
void fusion(Region* r) {
    if (r->fils != NULL) {
        for (int i = 0; i < 4; ++i) {
            if (r->isSameThan(r->fils[i])) {
                colorParents(r, r->fils[i]->getColor());
            }
            fusion(r->fils[i]);
        }
    }
}

```



Seuil utilisé : 400

## Résultat



On voit que les parties de les images qui avaient une couleur proche ont fusionés.

## Images couleurs



