INF555 TD3
Géométrie projection, filtre bilatéral (et appariement de nuages de points)

Frank NIELSEN
nielsen@lix.polytechnique.fr

28 Septembre 2011

1 Géométrie projective et la dualité points/droites

Écrire une applet qui affiche le point d'intersection $p = l_1 \times l_2$ (produit vectoriel) de deux droites l_1 et l_2 représentées par des coordonnées **homogènes** (les droites seront tirées aléatoirement). Visualiser par *dualité* la droite p^* qui passe par les deux points l_1^* et l_2^* (duaux des droites). En déduire, une procédure (fonction statique en Java) pour tester si deux segments de droite se coupent ou pas.

2 Le filtre bilatéral

En se basant sur le squelette du programme `BilateralFilteringSkeleton.java`, implanter une fonction statique qui prend en argument une image, l'écart type σ_s de la Gaussienne spatiale, l'écart type σ_i de la Gaussienne pour les intensités et k le nombre d'iterations du filtre bilatéral à effectuer, et retourne en sortie l'image lissée qui préserve les arêtes. On fixera la taille des noyaux à $2\sigma_s + 1$ et $2\sigma_i + 1$. Mesurer le temps de calcul avec la fonction `getTimeInMillis()` du package `java.util.Date`. Affichez enfin l'image résiduelle (image originale moins celle filtrée). (Cf. `http://www.shellandslate.com/fastmedian.html` pour une méthode bien plus rapide...)

```
static int [] BilateralFiltering(int [] raster, double sigmas, double sigmai, int k)
{...}
```

3 ** Iterative closest point (ICP)

En utilisant la bibliothèque JAMA (décomposition SVD), calculer la meilleure transformation rigide (R, t) (au sens de la somme des distances carrées) qui apparie les points:

p_i	q_i
$(2, 2)$	$(-0.8, -1)$
$(6, 3)$	$(-3.7, 1.9)$
$(5, 1)$	$(-1.5, 2.1)$

On rappelle qu'une fois les centroides $\bar{p}$ et $\bar{q}$ appariés, on considére les ensembles $p_i \to p_i'$ et $q_i \to q_i'$. La meilleure rotation est ensuite obtenue comme $R = UV^T$ où U et V proviennent de la décomposition de la matrice "scatter" $W = \sum_i p_i' q_i'^T$. La translation est alors $t = \bar{q} - R\bar{p}$. Affichez la transformation rigide complète sous la forme d'une matrice 3×3. Indiquez l'angle en degré de la rotation et visualiser l'appariemment trouvé (par exemple, on visualisera en bleu les points p et en rouge les points q). Créez des transformations aléatoires (R, t) et choisissez un bruit aléatoire ϵ afin de générer deux ensembles de points p_i et q_i qui s'apparient grâce a votre transformation. Testez votre méthode qui permet de retrouver une transformation proche de celle tirée aléatoirement.

```
...
SingularValueDecomposition svd=W.svd();
Matrix R=svd.getU().times(svd.getV().transpose());
...
```