

Exercice 0 - Mise en place

En utilisant ce que vous avez appris dans les deux TP précédents, l'objectif est de développer un programme exploitant des images stéréo. Votre programme final devra effectuer trois tâches :

- Afficher une carte de disparité.
- Afficher un Wiggle GIF.
- Afficher un anaglyphe.

Vous téléchargerez vos images sur la base de données MiddleBury disponible ici : <http://vision.middlebury.edu/stereo/data/>



(a) Image gauche: im0.png



(b) Image droite: im1.png

Figure 1: Exemple de paire d'images stéréo.

Exercice 1 - Carte de disparité

La carte de disparité est une image contenant l'information sur les correspondances de chaque pixel entre deux images 'vues' d'une même scène. La valeur d'un pixel sur une carte de disparité correspond à la distance de ce même point sur l'ordre image : son décalage en nombre de pixels. Plus le décalage est important, plus le point dans la scène 3D est proche de la caméra.

Vous utiliserez la classe StereoBM ou StereoSGBM incluse dans opencv_contrib afin de produire cette carte de disparité à partir des deux images.

Exercice 2 - Wiggle GIF

Vous créez un "Wiggle GIF" affichant les deux images stéréo alternativement permet de donner une impression de 3D. Vous utiliserez un l'objet QTimer pour afficher une simple animation des deux images.

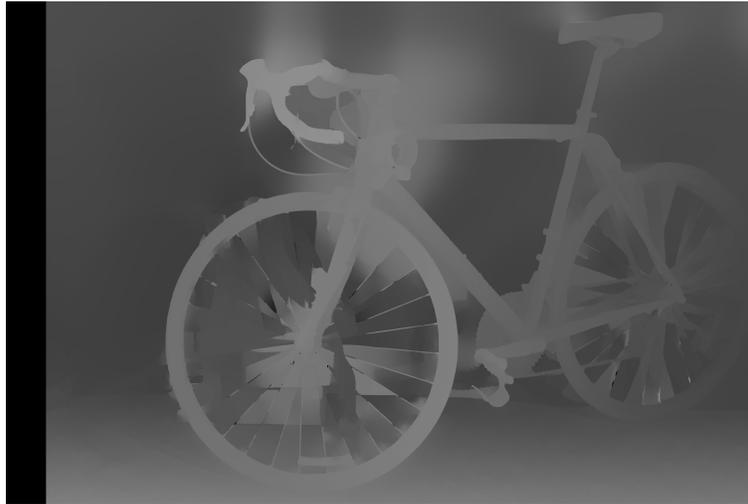


Figure 2: Exemple de carte de disparité.

Exercice 3 - Anaglyphe

Un anaglyphe est constitué de deux images superposées (appelées homologues) de couleurs complémentaires représentant la même scène mais vue de points légèrement décalés : le plus souvent la vue gauche en rouge et la vue droite en cyan. Ces images homologues ne sont donc pas identiques : le décalage (appelé parallaxe ou disparité) n'est pas le même pour tous les éléments de l'image ; il est d'autant plus grand, dans un sens ou dans l'autre, que les éléments sont situés près du plan de l'image physique, par exemple imprimée, projetée sur un écran ou affichée sur l'écran d'un ordinateur ou d'un téléviseur.

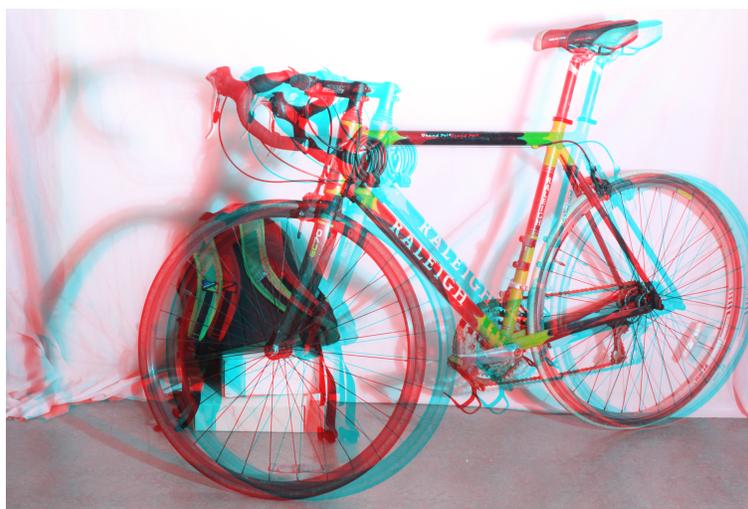


Figure 3: Exemple d'anaglyphe.

Exercice 5 - Consignes de rendu

Votre programme final devra ressembler à ceci :

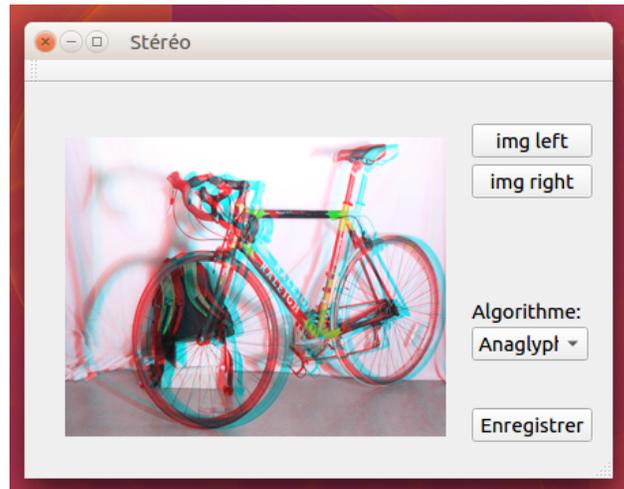


Figure 4: Rendu attendu.

- Deux boutons pour charger les images gauche / droite.
- Une combo box pour choisir le type de traitement à effectuer (disparité, GIF, anaglyphe).
- Un bouton pour enregistrer l'image une fois traitée.

La deadline doit être rigoureusement respectée.

- Le travail doit être rendu à antoine.billy@labri.fr ET lhofer@labri.fr
- Le titre du mail doit être préfixé par [ProjetsTechnologiques]
- La pièce jointe doit être nommée nom_prenom.tar.gz et avoir les propriétés suivantes
- L'extraction avec `tar -xzf nom_prenom.tar.gz` doit s'exécuter sans erreurs
- L'archive doit contenir un dossier nommé nom_prenom dans lequel sont présents tous
- Les fichiers nécessaires à la compilation, mais aucun fichier produit lors de la compilation, ni les images utilisées pour tester votre programme.
- Dans le dossier nom_prenom, la commande :

```
mkdir build cd build qmake -qt=qt5 .. make
```


Doit s'effectuer avec succès et créer un exécutable projet_technologique qui est votre programme.