# FAQ-2 du manuel d’extraction des trajectoires / Stabiliser des images

Logiciel : ImageTracker.jar – version du 12 juin 2012

Manuels à disposition : PrepareData.docx & Image Tracker Manuel

***remarque concernant les applications java***

Il est nécessaire d’avoir une machine virtuelle JAVA installée sur l’ordinateur

Si besoin la télécharger à partir du site java et choisir la configuration de l’OS (pour ordis de l’ENTPE – 64 bits : jre-8u91-windows-x64)

*Attention, il ne faut pas télécharger l’application pour les navigateurs mais celle de la MV.*

Deux possibilités pour faire tourner les applications Java

1/ exécuter le fichier .bat

Ce fichier doit contenir la ligne de commande qui lance l’application java

java -jar PrepareData.jar

java -Xmx1024m -jar ImageTracker.jar

si le nom du fichier .jar est modifié, changer son nom dans le fichier .bat

2/ utiliser l’invite de commande d’une fenêtre Ms-Dos (MicroSoft Disk Operating System)

Ouvrir la fenêtre Ms-Dos : drapeau Windows / rechercher programmes / taper cmd

Dans la fenêtre cmd.exe

i. Se placer dans la bonne partition : au départ le répertoire courant est C : \ >

pour accéder à un autre disque, taper simplement sa lettre de référence --- > F :

Nouvelle ligne F : \ >

ii. saisir le chemin d’accès au fichier .jar de PrepareData ou ImageTracker

rappel de quelques commandes Ms-Dos

cd .. pour redescendre dans l’arborescence ; cd nom\_dossier pour avancer ; dir pour lister le contenu du dossier ; cd\ pour revenir à la racine

penser à utiliser la touche tabulation pour ne pas saisir la totalité du nom du dossier

iii. une fois placé dans le bon répertoire, taper la commande java telle que contenue dans le fichier .bat

# Phase 1 / Préparation (PrepareData.jar)

Manuel : PrepareData\_Manual.docx

## Contexte

Les images étant prises par hélicoptère, elles ne sont pas stables. Il faut les stabiliser grâce au programme de Peter Knoppers de TUDelft nommé ImageTracker. Cependant, lancer ce programme sur 1h de video HD (2448x1000 pixels) à 25 images par secondes (soit 25\*3600 = 90 000 images) sur un PC bi-cœur demande environ 42 jours de calcul avec la version d’ImageTracker du 12 juin 2012. Plus exactement, sur 59 574 images, sur les PCs des salles informatiques de l’ENTPE, cela prend 668.5h, soit environ 27.85 jours ! Le temps de calcul pour chaque image est donc de :

Le programme parallélise les calculs sur tous les processeurs/cœurs qu’il trouve dans l’ordinateur. C’est par exemple deux fois plus rapide sur un quadri-cœur que sur un bi-cœur.

Pour éviter d’utiliser des machines ultra-performantes, le programme peut être lancé sur plusieurs ordinateurs, chaque ordinateur stabilisant une séquence de quelques minutes de vidéo. Par exemple, si on veut lancer des machines pendant un week-end, de vendredi 18h à lundi 7h, soit 61h, il faut   machines bi-cœur, où E représente la partie entière. Pour que toute la vidéo soit stabilisée de la même façon, il faut spécifier la même image de référence pour chaque séquence.

L’étape de préparation sert à séparer les images en autant de répertoires que nécessaire, chaque répertoire correspondant à un PC sur lequel la stabilisation sera lancée.

## Points à vérifier

Pour une explication du fonctionnement du programme **PrepareData**, voir le manuel sous le nom « PrepareData\_Manual.docx » dans le répertoire « 0 – PrepareData ». Cette partie va uniquement énumérer les points auxquels il faut faire particulièrement attention.

Avant de lire cette partie, il est bien d’avoir consulté le manuel de **ImageTracker** (dans le répertoire « 1 – ImageTracker ».

* Lancer **PrepareData.jar** : la dernière version date du 06 févier 2012. Il faut que Java soit installé sur l’ordinateur (pour charger java : [www.java.com/fr/download.manual.jsp /](http://www.java.com/fr/download.manual.jsp%20/) choisir 32 bits ou 64 bits). Le programme peut être lancé depuis une clé USB.
* Bien vérifier le contenu du fichier BAT (attention, il faut l’ouvrir dans un éditeur de texte pour qu’il ne s’exécute pas). Ce fichier sera créé dans chaque répertoire, donc s’il y a une erreur, il faudra la corriger partout.
* Il ne faut pas ajouter le paramètre *LastImage*, il sera ajouté automatiquement.
* Il faut spécifier la bonne quantité de mémoire RAM à allouer au lancement du programme ImageTracker (-Xmx1000m veut dire qu’on alloue 1 000Mo. La valeur maximum possible dépend du PC, mais cette valeur marche bien sur les PCs de salle info).
* Il faut spécifier la bonne image de référence (la même pour tous les films d’une zone)

**/ !\** Si l’hélicoptère était dans l’autre sens, il vaut mieux tourner l’image de référence de 180°, et utiliser l’image pivotée comme référence (sinon ImageTracker bugue)

**/ !\** Vérifier que les dimensions (hauteur - largeur) de l’image de référence sont les mêmes que celles des images à stabiliser

* Spécifier le format des images de sortie (OutputWidth, OutputHeight, OutputXOffset, OutputYOffset, OutputRotation).
	+ Pour cela, lancer **ImageTracker**, aller dans l’onglet **Select input image files** puis cliquer sur **Select first image file** et choisir l’image de référence.
	+ Aller dans l’onglet **Select output directory and image properties** et régler les paramètres pour avoir l’image souhaitée (les images de sotie auront le même format). Reporter les valeurs des paramètres dans le BAT de **PrepareData** (OutputWidth, OutputHeight, OutputXOffset, OutputYOffset, OutputRotation).
	+ Garder ces réglages quelque part pour pouvoir les retrouver facilement.
* Ne pas toucher à ReferenceImage ni à StartImage
* Sélectionner l’image de référence (bouton **Select reference image**). Le paramètre FirstImage est alors mis à jour avec le nom de l’image précédé de « 000000\_REF\_ ».
* Sélectionner l’image de calibration (photo du damier). Il faut que dans le même répertoire que l’image de calibration, il y ait le fichier \*.ele contentant l’élevation (voir le manuel de **ImageTracker** pour plus de précisions). Pour l’image de calibration, il faut prendre celle où la mise au point est 500m, et où l’ouverture est de f/1.4. Ce sont les mêmes réglages que ceux utilisés sur l’hélicoptère. Comme le damier était à 6m environ de la caméra, l’image de calibration a un flou qui s’étend sur 6 pixels, mais ça n’as pas d’importance puisque **ImageTracker** fait ensuite une convolution avec un masque de 12x12 pixels.
* Sélectionner l’exécutable (met à jour le BAT)
* Définir le nombre d’ordinateurs à utiliser. Sur les PCs de salle info (processeurs dual-core), il faut environ 0.01122h par image. Sachant qu’il y a 61h dans un week-end (voir plus haut), le nombre de PCs à utiliser s’exprime de la façon suivante :

Où E[] est la partie entière et le deuxième terme « 1 » sert de sécurité.

* Laisser les valeurs par défaut de InitialXOffset, InitialYOffset, InitialRotation, InitialEnlargement, InitialWedgeX, InitialWedgeY, elles seront à régler à la main pour chaque PC avant de lancer ImageTracker sur chaque PC voir ci-dessous le paragraphe 2.4 page 6) .

Après le lancement de **PrepareData**, il n’y a pas de barre de progression (je n’ai pas réussi à la faire marcher dans un temps raisonnable). Il y a seulement un pop-up à la fin qui indique que c’est fini. Pour se rendre compte de la progression, il suffit d’ouvrir l’explorateur de fichier et de voir combien de fichiers ont déjà été déplacés. Le programme met environ 1h pour déplacer 90000 images (1h de video).

# Stabilisation (ImageTracker.jar)

Une fois que les images sont réparties dans les différents répertoires PC##, il peut être bien de les renommer en ajoutant quelque chose à la fin du nom qui permette d’identifier la séquence (surtout si des séquences différentes vont être stabilisées pendant le même week-end). Par exemple « \_Z1\_13\_F1 » pour la zone 1 le 13 septembre et le film 1 (éviter de choisir des noms trop longs parce qu’il faudra les écrire à la main après !)

## Initialisation

Maintenant pour chaque répertoire :

* Double-cliquer sur le .bat (**ImageTracker** doit se lancer avec tous les paramètres définis précédemment)
* En parallèle, ouvrir le .bat (**clic droit** puis **Modifier**)
* Dans **ImageTracker** vérifier que tout s’est bien chargé (Image de calibration, fichiers d’entrée, format de sortie)
* Puis aller dans l’onglet **Run**. Deux images sont superposées : l’image de référence (N°0) et la première image de la séquence à stabiliser (N°1). Il régler les 6 premiers paramètres (X-Offset, Y-Offset, Rotation, Enlargement, Wedge X, Wedge Y) de façon à aligner ces deux images. L’image affichée doit être plus ou moins grise. Plus l’image sera grise, plus ImageTracker aura initialement des réglage lui permettant de trouver à coup sûr et dans un temps faible les bons paramètres de fonctionnement. Si le calage de ces 6 paramètres n’est pas parfait, il sera effectué par le code par tâtonnements successifs.
* Reporter ces 6 valeurs dans les paramètres correspondants du .bat (InitialXOffset, InitialYOffset, InitialRotation, InitialEnlargement, InitialWedgeX, InitialWedgeY)
* Ajouter le paramètre Run à la fin de la ligne (de cette façon la stabilisation démarrera en double-cliquant sur le .bat)
* Enregistrer le .bat et fermer

Répartir les répertoires PC## sur plusieurs disques externes (utiliser de préférence les disques SSD qui n’ont pas besoin d’alimentation et qui sont aussi plus rapides) pour transférer les données sur les PCs de salles info (on est obligés de le faire sur chaque PC par USB car on ne peut pas le faire à distance). Le nombre optimal de disques à utiliser est entre 4 et 5 (chaque répertoire pèse environ 10Go et il faut 10min pour le copier sur un PC). De cette façon on peut lancer des transferts sur plusieurs PCs à la fois (le temps de lancer les transferts sur 4 ou 5 PCs, le premier transfert se termine). Il faut environ une heure pour faire les transferts dans une salle de 20 PCs.

## Points administratifs :

* Prévenir le responsable réseau du service informatique (Christian Gardet : christian.gardet@entpe.fr) que l’on veut utiliser les PCs des salles info pendant tel week-end (les salles utilisables sont les E005, E006b et E006c). Voir les fichiers « gestion salle E005.docx », « gestion salle E006b.docx » et « gestion salle E006c.docx » pour connaître le nombre de PCs dans chaque salle (à noter que les E006b et E006c communiquent entre elles et qu’il faut fermer à clés ces deux salles en même temps).
* Récupérer la (les) clef(s) auprès du LOG (Jean-Pierre Dupont) des salles que l’on veut (s’y prendre une semaine à l’avance).
* Vérifier quand les salles seront libres (voir à l’accueil). Si les salles sont libres le vendredi après-midi, on peut éventuellement les réserver pour faire les transferts tranquillement puis lancer les calculs. Sinon il faut faire les transferts plus tôt dans la semaine, et dès que les salles se libèrent, lancer les calculs. Il faut aussi demander à partir de quelle heure elles sont prises le lundi matin (horaires disponibles à l’accueil à partir de vendredi après-midi, ou avant sur Hélisa)
* Le vendredi matin avant le week-end où les calculs seront lancés, prévenir le LOG (LOG.SG@entpe.fr) de ne pas éteindre les PCs des salles que l’on va utiliser (pour l’indiquer au gardien qui ferme les salles).
* Imprimer 2 fois le nombre de salles le fichier « Expérimentation en cours.doc » en mettant à jour la date.
* Imprimer les fichiers « gestion salle E00#.docx » des salles qui seront utilisées.

## Transfert des fichiers

* Se loguer sur le PC avec son login/mot de passe ENTPE (on peut se loguer sur autant de machines que l’on veut). S’il n’y a personne dans la salle, on peut commencer par se loguer sur tous les PCs.
* Copier le répertoire PC## qui est sur le disque dur externe.
* Le coller dans **C:\temp**
* Noter le nom exact du répertoire (ex : PC05\_Z1\_13\_F1) sur la fiche de gestion de la salle dans l’emplacement correspondant au PC sur lequel on vient de lancer le transfert.
* Cocher la case Données transférées

## Lancement de la stabilisation

* Se loguer sur le PC (si ce n’est pas déjà fait).
* Aller dans le répertoire **C:\temp**
* Double-cliquer sur **ImageTracker\_launch.bat** (l’onglet **Run** doit s’afficher et la stabilisation doit commencer)
* Cocher la case **programme lancé** sur la feuille.
* Une fois que tous les PCs ont été lancés, vérifier chaque PC : l’image affichée doit être parfaitement grise et alignée.
* Changer le **Refresh rate** à **100 s** (ça rafraichira moins souvent et préservera les ressources du processeur uniquement au calcul)
* Poser l’affiche « Expérimentation en cours » deux fois dans chaque salle (une sur la porte d’entrée et une sur la table ou le tableau)
* Eteindre la lumière et fermer la salle à clef.

## Récupération des résultats

Le lundi matin, il est bien d’arriver un peu avant 8h (pour ne pas trop gêner les étudiants). Les salles auront déjà été ouvertes par les femmes de ménage, mais pas de panique, les calculs ont dû se terminer avant 7h et même si quelqu’un a éteint un PC par la force brute, les images stabilisées sont enregistrées sur le disque dur.

Il suffit d’utiliser 2 ou 3 disques externes pour récupérer les images stabilisées car il n’y a que 3-4 Go à transférer (les images stabilisées sont aussi rognées pour n’avoir que la route et sont donc moins grosses).

* Se loguer sur chaque PC
* Vérifier que le programme s’est bien arrêté (il doit indiquer **Finished** dans la barre de progression en bas)
* Cocher la case **Oui** s’il s’est bien arrêté. Sinon cocher **Non** et indiquer à quelle image le programme s’est arrêté (il faut aller dans C:\temp\PC##\output et regarder le numéro de la dernière image commençant par « s »)
* Fermer ImageTracker
* Copier le nom du répertoire PC##
* Ajouter ce nom au sous répertoire **output** (il doit maintenant s’appeler **outputPC##**)
* Copier le répertoire **outputPC##**
* Le coller sur le disque externe
* Cocher la case **Résultats récupérés sur…** et noter le nom du disque externe sur lequel les données ont été récupérées.
* Supprimer le répertoire PC## de C:\temp (**MAJ + suppr** pour le supprimer définitivement, sinon il faut en plus vider la corbeille)
* Cocher la case Données effacées
* Retirer le disque
* Se déloguer

Si pour une raison x ou y, on ne peut pas arriver suffisamment tôt pour récupérer les données, on peut seulement fermer les fenêtres et se déloguer (éventuellement en vérifiant au moins mentalement que le programme s’est arrêté normalement). On pourra revenir plus tard quand la salle se sera libérée pour récupérer les données.

## Fusion des résultats

Maintenant toutes les images de la séquence ont été stabilisées. Elles se trouvent sur plusieurs disques externes (ceux utilisés au paragraphe 2.5).

* Connecter tous les disques à un même PC.
* Rassembler les répertoires **outputPC##** sur un seul disque sous un seul répertoire nommé suivant la nomenclature suivante :

sCam\_1\_Zone\_I\_JJMMAA\_CB

le « s » faisant référence à « stabilisé ».

* Vérifier qu’il y a autant de répertoires **outputPC##** qu’il y avait de répertoires **PC##** au paragraphe 2.1.
* Dans le répertoire « sCam\_1\_Zone\_I\_JJMMAA\_CB », créer un dossier « images »
* Pour chaque répertoire **outputPC##**, sélectionner toutes les images commençant par un « s » et les déplacer vers le dossier **images**
* Vérifier qu’il y a le bon nombre d’images (autant d’image stabilisées qu’il y avait d’images à stabiliser)
* Créer un répertoire « fichiers generes » dans le répertoire « sCam\_1\_Zone\_I\_JJMMAA\_CB »
* Déplacer tous les répertoires **outputPC##** (qui maintenant ne contiennent que des fichiers générés par **Imagetracker** pour son fonctionnement interne, mais plus d’images stabilisées) vers le dossier **fichiers generes** (ça peut toujours servir !)

## Phase 3 / traitement des arrêts

Une stabilisation peut s’arrêter pour plusieurs raisons :

1. Coupure de courant
2. Extinction forcée d’un ordinateur
3. Problème intrinsèque aux images (passage d’un nuage qui bouche tout, sursaturation de l’image qui devient toute blanche)
4. Problème d’initialisation
5. Les images à stabiliser sont à 180° de l’image de référence

Dans tous les cas, l’arrêt de la stabilisation arrive rarement (1 PC sur 40). Il vaut donc mieux relancer la stabilisation sur un PC de bureau disponible pour ne pas monopoliser une salle entière juste pour un PC.

Si la stabilisation s’est arrêtée pour la raison 1 ou 2 :

* Il suffit de relancer **ImageTracker** à partir de la dernière image correctement stabilisée (pour s’assurer qu’elle soit bien stabilisée) en utilisant exactement les mêmes réglages (image de calibration, liste d’entrée, répertoire de sortie, propriétés des images de sortie, image de référence).
* Pour l’initialisation, ouvrir le fichier **results.txt** qui se trouve dans le répertoire **output** (ouvrir avec un autre éditeur de texte que le bloc-notes)
* Aller tout en bas du fichier : la dernière ligne commence par le nom du fichier de l’image de référence, suivi de celui de la dernière image stabilisée. Il y a ensuite 7 nombres. Les 6 premiers correspondent aux paramètres d’alignement. Reporter ces 6 valeurs dans les 6 premiers champs de l’onglet **Run.**

 Si la stabilisation s’est arrêtée pour la raison 3 :

* Relancer **ImageTracker** à partir d’une image postérieure au problème (nuage, sursaturation…)
* Comme pour les raisons 1 et 2, on peut utiliser les paramètres d’alignement du fichier **results.txt** pour s’aider, mais il faudra sûrement les corriger un peu pour réaligner les images (l’hélicoptère a pu bouger de façon significative pendant le phénomène).

Si la stabilisation s’est arrêtée pour la raison 4 :

* Approximer plus précisément les paramètres d’alignement avant de lancer le programme.

Si la stabilisation s’est arrêtée pour la raison 5 :

* Il faut pivoter l’image de référence de 180°

# Annexe - ImageTracker - Image stabilization software developed at TU Delft

(texte écrit par Peter Knoppers, TU Delft, NL)

ImageTracker is software that can be used to stabilize a series of color or gray scale images that were taken using a non-stable (vibrating or moving) camera. ImageTracker can perform the following operations:

1. Correct lens distortion (requires an image of a checkerboard pattern and knowledge of the viewing angle of the camera; the camera hardware and lens settings must be the same as used for the images that will be lens-corrected and stabilized).
2. Search for the image transformation that matches each image of a series as close as possible onto a reference image.
3. Apply an optional output transformation to each stabilized image and write (part of) the observed scene to a numbered file in a user specified output directory. The output transformation can perform [orthorectification](http://en.wikipedia.org/wiki/Orthophoto); correct the image in case the camera was not pointing straight down onto the scene.

ImageTracker cannot remove motion blur.

The image transformations that ImageTracker applies to match images onto the reference image assume that the observed scene is flat. This assumption is usually not quite true (even in the Netherlands). A non-flat scenery combined with a moving camera location makes perfect registration impossible and may cause several local optimal sets of parameter values to exist. In practice the results can be quite impressive for non flat scenery and a moving camera location.

The stabilization operates on the entire image. This requires that most of the objects visible in the scene are fixed. For traffic observations (the primary purpose of this software), only vehicles and pedestrians are supposed to move and most other objects in the scene are stationary. The ImageTracker software allows the camera to move along the scene. This is done by extending the internal copy of the reference image as new scenery comes into view.

ImageTracker is written in Java and should run on any computer for which there is a [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine). The software can take advantage of multiple core and multiple CPU computers; in fact it should be able to scale nicely to hundreds of cores when such hardware becomes available. Image stabilization is a very CPU-intensive operation. On a 2008 quad-core desktop computer, stabilization of 5 mega-pixel images can take one minute per image. Larger jobs can be split into multiple smaller ones and divided over several computers. In this case approximate initial values for the search parameters must be provided by the user.

## Input files

ImageTracker can read almost any image file type. If the images were acquired with a video camera, it may be necessary to [de-interlace](http://www.100fps.com/) them first. De-interlacing should be done by interpolating the skipped lines or doubling the non-skipped lines. Simply leaving out the skipped lines will not work as the remaining image would have non-sqpare pixels (resulting in problems when the camera rotates during the image sequence).

## Output files

The stabilized images are written in [PNG format](http://en.wikipedia.org/wiki/Portable_Network_Graphics). The output file names are the same as the input ones, but prefixed with an "s" (for stabilized). The extension is changed to "png" if the input one was different.

# Use of the ImageTracker program

On start-up the program displays a single window with 5 tabs:

1. Lens calibration
2. Select input image files
3. Select output directory and image properties
4. Alignment settings
5. Run

Normally the user would visit these tabs in this order.

## Lens calibration

This tab contains all the tools to compute the required lens correction. Without proper lens correction, stabilization can be problematic. Even the best lenses exhibit significant distortion near the edges of the image. The ImageTracker program can be used without lens correction (by not loading a calibration image in this tab), but the results may be disappointing.

The calibration image should be a checkerboard pattern. A sheet of vinyl floor covering is suitable for this purpose. The reference image must be taken with the camera pointing straight onto the checkerboard pattern (the sheet must be flat and the line from the camera to the center of the image must be perpendicular to the sheet. The checkerboard grid must run approximately parallel to the width and height directions of the camera. The image should enclose about 30 x 20 squares of the pattern. As you can see on the calibration image below, lightning need not be perfectly uniform. (This image was reduced in size to 25%; the original is 2448x2050 pixels.)


In the *Lens calibration* tab click on the *Open* button and select the lens calibration image. After loading the image the software will locate all the cross-points in the image and try to line those up in rows and columns. Once finished, the user can inspect the results by selecting any of the six radio buttons:

|  |  |
| --- | --- |
| Button | Displayed information |
| Raw image | The original image |
| Analized image | A gray-scale image with dark and light patches near the cross-points of the raw image The pixels with a locally extreme value are indicated with red and green cross-hair markers. |
| Image with calibration points | The original image with diagonal cross hair markers. |
| Corrected image | The straightened image. Usually some pixels in the straightened images do not reverse-map onto the raw image. These pixels are rendered in Red. |
| Statistics | Some indicators of the quality of the calibration; see below. |
| Table with calibration points | A table with the coordinates (in pixels) of the calibration points. |

The statistics information gives a short summary of the quality of the lens correction that was derived. The *Standard deviation of distance* should be less than 1% of the *Mean distance*. The difference between the *Longest distance* and the *Shortest distance* should be much smaller than the *Mean distance*. The number of incomplete pairs should be small relative to the number of point pairs.

### Camera viewing angle


Besides the lens correction image, ImageTracker needs to know the viewing angle of the camera. This is expressed in a peculiar way; the distance from the camera to the center of the image measured in pixels. This can be measured while taking the lens calibration image, but it can also be done in a separate setup. To measure the size of a pixel in mm, point the camera straight (perpendicular) to a flat surface. Now attach two sticky memo stickers on the surface near the extreme left and right of the area observed by the camera. Put those stickers roughly along the center line of the image. Use a tape measure to measure the length of the observed center line in mm (call this value *width*). Also use the tape measure to measure the distance of the camera to the middle of that center line (call this value *distance*).

Divide the *width* by the number of pixels in that direction of the image. This gives the size of a pixel in mm.
Divide the measured *distance* by the computed size of a pixel to obtain the distance from the camera to the surface in pixels (call this value *elevation*)
This *elevation* is required when ImageTracker computes the distortion due to movement of the camera location. We recommend that a text file is created with the same name as the calibration image, but extension ".ele". This file must be stored in the same directory as the calibration image. When ImageTracker loads the lens calibration image it will look for the corresponding ".ele" file and - if it exists - load that too. Otherwise the user must enter the value manually in the *Elevation* box in the *Lens calibration* tab.

## Select input image files

The image that will be used as reference image must be in the same directory as the images that will be stabilized.
Click on the button *Select first image file ...* and navigate to the directory containing the image files to be stabilized. Select the first image file to be stabilized. Then click *Select last image file ...* and select the last image file to be stabilized.
If the reference image is not within the series of images to be stabilized, you must select the reference image as the first (or last) image.

After selecting the first and last images the list under *Image files* will be populated. Clicking on a file name causes that file to be displayed. Checking the check box *Apply lens correction* causes the image to be displayed with lens correction. It is instructive to see how the image changes when this check box is (un-)checked. The effect resembles the "popping" of a color slide that heats up in a projector.

The status of the check box only affects the preview image in this tab.

## Select output directory and image properties

This tab contains all the controls that affect how a stabilized image is saved. The user should use these settings to optimize the size of the stabilized images and select only the region of interest. The computation time of the subsequent processing (such as searching for moving objects, that can be achieved by the ObjectFinder software) is directly impacted by the size of the stabilized images. To set the parameters of the output images, the user should select the reference image in the *Select input image files* tab. Going back on the *Select output directory and image properties*, the reference image should be displayed. The user can now set the properties of the stabilized images. The appearance of the output image is updated once a parameter is changed. Please note that no stabilization is made at this point, so the user may be disappointed when selecting another image to display, because it may not display the region of interest he chose. But it does not matter as long as he set it on the reference image. The settings are divided in 5 groups (sometimes the window must be widened in order to show all the groups; this is a bug in Java FlowLayout).

### Output directory

This group contains only one button. Click it to select the directory where the stabilized images must be written. In addition to the stabilized images, the ImageTracker also creates a series of small images that, together, constitute the reference image and a text file *result.txt* that contains the values of the parameters used to correct each image. The current version of ImageTracker also writes some other files to this directory for debugging purposes. These can be ignored.

### Output image size

This group contains two boxes to set the *Width* and *Height* of the output images. If the camera was moved along the landscape, make sure that the output images will be large enough to contain the entire "flight". Pixels that were not observed in a particular image will be output as Red pixels (R,G,B = 255,0,0).

### Color correction

This group contains two boxes to change *Contrast* and *Brightness*. Contrast is a floating point value that is used as a multiplier for the pixel values. *Brightness* is added to the pixel values **after** multiplication with *Contrast*. Finally the values for R, G and B are clipped and rounded to obtain integer values in the range 0..255.

### Translate, rotate, enlargement

This group contains four boxes to set a translation (*X offset* and *Y offset* in pixels), *Rotation* (counter-clockwise in degrees) and *Enlargement.*

### Wedge distortion

This group contains two boxes for wedge (key stone) distortion (*Wedge X*, *Wedge Y*). The Wedge values can be used to correct if the pictures of the camera were not taken straight down. *Wedge X* is a rotation along the X-axis of the camera, *Wedge Y* is a rotation along the Y-axis of the camera. Orthorectification can be performed by setting these values. With the correct values, objects on the ground that were the same distance apart should be the same distance apart and objects on the ground that are at right angles should be at right angles. The elevation value entered in the *Lens calibration* tab must be correct for orthorectification to work. (It is, in fact, possible to find a reasonably close approximation of the elevation by trial and error.)

## Alignment settings

This tab allows the user to change the ranges and step-sizes of the parameters that are varied to match an image on the reference image. The program uses the values used for the preceding image as initial guess for the current image. So these maximum changes must be sufficient to handle the difference between two successive image. The parameters are changed in rounds. Each round, the step size of each parameter is halved. For each parameter there is a maximum precision at which it can reasonably be tuned. This precision corresponds to a change of about a tenth of the size of a pixel.
The parameters are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Step size in first round | First round | Last round | Maximum change | Smallest step size |
| X-Offset | 2.0 pixels | 0 | 5 | 120 pixels | 0.06 pixels |
| Y-Offset | 2.0 pixels | 0 | 5 | 120 pixels | 0.06 pixels |
| Rotation | 0.25° | 0 | 5 | 15° | 0.008° |
| Enlargement | 0.0002 | 4 | 5 | 0.01 (1%) | 0.0001 (0.01%) |
| X-Wedge | 0.05° | 4 | 5 | 2° | 0.025° |
| Y-Wedge | 0.05° | 4 | 5 | 2° | 0.025° |
| X-Speed | 0.00005 pixels/line-time | 4 | 5 | 0.02 pixels/line-time | 0.000025 pixels/line-time; |
| Y-Speed | 0.00005 pixels/line-time | 4 | 5 | 0.02 pixels/line-time | 0.000025 pixels/line-time; |

If these default settings perform badly (tracking is lost, or does not quickly enough catch up with changes of the camera view) the user can change these.

### X-Speed and Y-Speed

The last two parameters are needed to correct image distortion that happens in video cameras with CMOS sensors when the camera tilts or pans. If the images were not acquired with such a camera, change the value of *first round* for these parameters into a value that exceeds the value of *last round*. This prevents ImageTracker from attempting to optimize these parameters.

## Run

This tab contains the controls to set up tracking, start it and observe the results. On the right is displayed the difference between the reference image and the current processed image (or the start image before running the program). If the reference image is the same as the current image, the difference is a uniform gray panel. If they are different, the scene should be visible (because of lighting difference from one image to another), but it should appear in gray scale.
The ImageTracker can only track limited changes in camera position between two successive images. If the reference image is not the first image in the series, the user should enter the approximate initial values for the six parameters in the first six boxes in order to align the reference image with the start image. The alignment would be correct when most of the pixels in the difference image would be gray.
The last three boxes allow the user to change the *Reference image* (default 0), *Start image* (default 0) and the *End image* (default 99999).
Clicking on the *Run* button starts the stabilization process. Clicking on *Cancel* stops it (it can take a couple of seconds to terminate the operation).
The *Refresh rate* can be altered to limit the maximum number of intermediate difference images that are shown on the screen. A larger interval reduces the CPU time that is consumed to display these difference images.
If the ImageTracking operation is restarted with a different reference image, the user **must** remove the tile files in the *Output directory* (or change the output directory under the *Select output directory and image properties* tab to a new, empty, directory).

### Restarting the ImageTracker

The ImageTracker writes the found values for the six alignment parameters in a file *results.txt* in the *Output directory*. These values can be considered checkpoints for restarting (after e.g. a power failure).
Set up the ImageTracker exactly like it was before the run was aborted. (Same lens calibration image, input image list, output directory, output image properties, same reference image.) In the *Run* tab enter the rank number of the last successfully stabilized image as *Start image*. Copy the six values for *X-Offset* etc. from the *results.txt* file. This should result in good registration of that image. Finally click *Run* to resume the image stabilization process.

## Command line parameters

The ImageTracker program can almost entirely be set up and started using command line parameters. The only exception are the values under *Alignment settings*.
The following program arguments are understood:

|  |  |
| --- | --- |
| Program argument | Description |
| **FirstImage=**full\_path\_and\_file\_name | Set the first image file name |
| **LastImage=**full\_path\_and\_file\_name | Set the first image file name |
| OutputDirectory=*path\_name* | Set the output directory |
| **OutputWidth=**width\_in\_pixels | Set the width of the output images |
| **OutputHeight=**height\_in\_pixels | Set the height of the output images |
| OutputContrast=*contrast\_value* | Set the contrast factor of the output images |
| OutputBrightness=*brightness\_value* | Set the brightness value of the output images |
| **OutputXOffset=**X\_offset\_in\_pixels | Set the X-Offset of the output images |
| **OutputYOffset=**Y\_offset\_in\_pixels | Set the Y-Offset of the output images |
| **OutputRotation=**rotation\_in\_degrees | Set the rotation value of the output images |
| OutputEnlargement=*factor* | Set the enlargement factor of the output images |
| **OutputWedgeX=**rotation\_in\_degrees | Set the X-Wedge value of the output images |
| **OutputWedgeY=**rotation\_in\_degrees | Set the Y-Wedge value of the output images |
| **Elevation=**elevation\_in\_pixels | Set the elevation value. **Warning:** this value is ignored if a lens correction image is loaded that has it's own elevation in an accompanying *.ele* file. |
| **InitialXOffset=**X\_offset\_in\_pixels | Set the initial X-offset value (*Run* tab) |
| **InitialYOffset=**Y\_offset\_in\_pixels | Set the initial Y-offset value (*Run* tab) |
| **InitialRotation=**rotation\_in\_degrees | Set the initial rotation value (*Run* tab) |
| InitialEnlargement=*factor* | Set the initial enlargement value (*Run* tab) |
| **InitialWedgeX=**wedge\_X\_in\_degrees | Set the initial wedge-X value (*Run* tab) |
| **InitialWedgeY=**wedge\_Y\_in\_degrees | Set the initial wedge-Y value (*Run* tab) |
| **ReferenceImage=**rank\_of\_reference\_image | Set the reference image |
| **StartImage=**rank\_of\_start\_image | Set the start image |
| **EndImage=**rank\_of\_end\_image | Set the end image |
| **LensCalibration=**full\_path\_and\_file\_name | Load the lens calibration image (and, if present, the elevation from the accompanying *.ele* file) |
| **MaxThreads=**number\_of\_threads | Set the maximum number of threads to use in the internal image processing functions. This value cannot be changed in the graphical user interface. Currently the default is 20 which seems to work well for quad and hex core CPUs. |
| Run | Start the stabilization process. There should be no *=* at the end of this program argument. |