

classe KVIDGridEditor

15 juin 2012

Table des matières

1 Démarrage	2
2 Fonctionnalités simples	2
2.1 Sélection de lignes	2
2.2 Ajout et suppression de lignes	2
2.3 Linéarisation et fit	3
2.4 Sauvegarde de la grille courante	3
3 Appliquer une transformation	3
3.1 Utilisation du pivot et du modulateur	3
3.2 Procédure	3
3.3 Description des transformations	3
3.3.1 Translations	3
3.3.2 Rotation	4
3.3.3 Scalings	4
3.3.4 Correction de la courbure d'une ligne	4
4 Spider identification	5
5 Personnalisation et ajout de nouvelles fonctionnalités	5
6 Raccourcis clavier	5

1 Démarrage

Pour utiliser l'éditeur de grille, il faut que le pointeur *gIDGridManager* soit correctement positionné. Vous pouvez par exemple placer les lignes suivantes dans votre *rootlogon.C* :

```
{
  new KVDataRepositoryManager;
  gDataRepositoryManager->Init();

  gDataSetManager->GetDataSet("INDRA_e613")->cd();
  gDataSet->BuildMultiDetector();
}
```

Lancez l'interface graphique du manager de grille depuis l'interpréteur de root :

```
gIDGridManager->StartViewer()
```

Ouvrez un fichier root contenant les histogrammes puis double cliquez sur la grille à modifier. Un canvas vas alors s'afficher ainsi qu'une boîte de dialogue vous proposant la liste des histogramme dans *gFile*. Si un histogramme porte le même nom que la grille, il est proposé en choix par défaut. Si vous souhaitez afficher seulement la grille, choisissez *Dummy* dans la liste proposée. L'histogramme et la grille s'affiche alors dans le canvas.

Si aucun *DataSet* n'est défini, vous pouvez accéder directement à l'éditeur avec les commande suivante :

```
root [0] new KVIDGridEditor
root [1] gIDGridEditor->StartViewer()
root [2] gIDGridEditor->SetGrid(myGrid)
root [3] gIDGridEditor->SetHisto(myHisto)
```

2 Fonctionnalités simples

2.1 Sélection de lignes

Il existe plusieurs possibilités pour sélectionner des lignes :

- le bouton *All* permet de sélectionner toutes les lignes de la grille (y compris les cuts),
- le bouton *Select* permet de passer en mode 'select', il suffit alors de cliquer sur chaque ligne,
- le bouton *More* puis *SelectLinesByZ* permet de selectionner une liste de lignes de Z.

Les lignes sélectionnées changent de couleur (orange pas défaut). Cette couleur peut être modifier via *More* puis *SetSelectedColor*.

2.2 Ajout et suppression de lignes

Pour ajouter des lignes, contours ou cuts, utilisez les boutons *Line* et *Cut*. L'éditeur vous proposera alors une liste de classe disponibles pour la grille courante. Si une seule classe est disponible elle est sélectionnée automatiquement.

Pour supprimer une ou plusieurs lignes il faut cliquer sur le bouton *Delete* qui passe alors en rouge. Si vous cliquez ensuite sur une ligne, l'éditeur vous proposera de la supprimer. Le mode 'delete' est actif tant que le bouton *Delete* est rouge.

2.3 Linéarisation et fit

Le bouton *Test* permet de linéariser l'histogramme avec la grille courante. Il est possible d'ajuster une fonctionnelle de Tassan-Got sur la grille via le bouton *Fit*.

2.4 Sauvegarde de la grille courante

Il est possible de sauvegarder la grille dans un fichier texte (.dat) en utilisant le bouton *More* puis *SaveCurrentGrid*. Un boîte de dialogue va alors s'ouvrir. Elle vous permet de naviguer dans l'arborescence et de définir le nom du fichier de sauvegarde. Si le nom que vous définissez ne contient pas d'extension, l'extension *.dat* est ajoutée par défaut. La chaîne de caractères *toto* est automatiquement remplacées par le nom de la grille.

Exemple 1 : *toto* est remplacé par `[TheGrid->GetName()].dat`.

Exemple 2 : *toto_GG.truck* est remplacé par `[TheGrid->GetName()]._GG.truck`.

3 Appliquer une transformation

3.1 Utilisation du pivot et du modulateur

Le pivot est symbolisé par une croix rouge positionnée à l'origine (0,0) lors du démarrage de l'éditeur. Les coordonnées du pivot sont (X_0, Y_0) . Ces deux grandeurs sont utilisées comme origine pour les différentes transformations. Le pivot se positionne de deux manières :

- double cliquer sur l'histogramme,
- le déplacer simplement avec la souris.

Exemple : si $(X_0, Y_0) = (100, 200)$, alors une rotation se fera autours du point $(100, 200)$.

Le modulateur permet de régler l'amplitude des transformations. La valeur -entière et positive- indiqué est noté i_m . Pour modifier la valeur de i_m , il faut utiliser les boutons $+$ et $-$:

- simple clic ± 1 ,
- double clic ± 10 ,
- *Shift*-clic ± 100 .

3.2 Procédure

Afin d'appliquer une transformation sur une ligne ou sur un groupe de lignes, il faut suivre les étapes suivantes :

1. sélectionner une ou plusieurs lignes,
2. sélectionner une transformation en cliquant sur le bouton qui lui est associé,
3. régler le modulateur,
4. utiliser la molette de la souris pour appliquer la transformation, vers le haut $Sign = +1$, vers le bas $Sign = -1$.

3.3 Description des transformations

3.3.1 Translations

Les translations sont associées aux boutons T_X et T_Y de l'interface. Les fonction appliquées sur l'ensemble des points des lignes sélectionnées à chaque pas sont les suivantes :

$$X' = X + Sign \times i_m \times BinWidthX \times 0,2 \quad \text{pour } T_X \quad (1)$$

$$Y' = Y + Sign \times i_m \times BinWidthY \times 0,2 \quad \text{pour } T_Y \quad (2)$$

où, $BinWidthX$ et $BinWidthY$ sont les largeurs des bins sur chaque axe.

3.3.2 Rotation

La rotation autour de l'axe Oz est associée au bouton R_Z de l'interface. Les fonction appliquées sur l'ensemble des points des lignes sélectionnées à chaque pas sont les suivantes :

$$X' = (X - X_0) \times \cos(\theta) - (Y - Y_0) \times \sin(\theta) + X_0 \quad (3)$$

$$Y' = (X - X_0) \times \sin(\theta) + (Y - Y_0) \times \cos(\theta) + Y_0 \quad (4)$$

où, $\theta = Sign \times 0,01 \times i_m$ (en °) et (X_0, Y_0) sont les coordonnées du pivot. Cette opération revient donc à faire une rotation de θ autour du pivot.

3.3.3 Scalings

Les scalings sont associées aux boutons S_X , S_Y et S_{XY} de l'interface. Les fonction appliquées sur l'ensemble des points des lignes sélectionnées à chaque pas sont les suivantes :

$$X' = (X - X_0) \times (1 + Sign \times i_m \times 5.10^{-4}) + X_0 \quad \text{pour } S_X \quad (5)$$

$$Y' = (Y - Y_0) \times (1 + Sign \times i_m \times 5.10^{-4}) + Y_0 \quad \text{pour } S_Y \quad (6)$$

où, (X_0, Y_0) sont les coordonnées du pivot. La transformation S_{XY} applique S_X et S_Y simultanément et avec la même amplitude.

3.3.4 Correction de la courbure d'une ligne

La correction de courbure est associée au bouton S_C de l'interface. La fonction appliquée sur l'ensemble des points des lignes sélectionnées à chaque pas est la suivante :

$$R' = R \times (1 + Sign \times i_m \times 5.10^{-4}) \quad (7)$$

où, R est la distance définit sur la figure 1.

Pour simplifier l'implémentation, cette opération est appliquée de la manière suivante :

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = R_Z(\theta) \cdot S_Y \cdot R_Z(-\theta) \cdot \begin{pmatrix} X \\ Y \end{pmatrix} \quad (8)$$

où, θ est l'angle définit sur la figure 1.

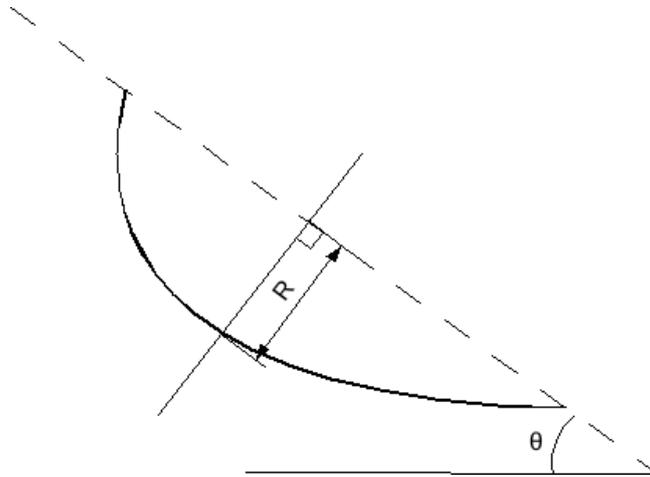


FIG. 1 – Shémat de la correction de courbure d'une ligne

4 Spider identification

L'éditeur de grille fourni aussi un outil semi-automatique de génération de lignes de Z. Cet outil utilise la classe `KVSpiderIdentifier` et ne fonctionne qu'avec les matrices Si-CsI. Pour l'utiliser, il faut suivre les étapes suivantes :

1. ouvrir l'éditeur avec une grille (vierge ou non) et un histogramme,
2. positionner le pivot sur une ligne de Z élevé,
3. cliquer sur le bouton *Spider*, une boîte de dialogue vas s'ouvrir,
4. indiquer le Z de la ligne sur laquelle le pivot se situe (environ).

La génération des lignes prend un peu de temps et le résultat est affiché dans l'éditeur. Si la grille contenait déjà des lignes, elles seront supprimées. Si le résultat de l'identification ne vous convient pas, recliquez sur *Spider*. Dans la boîte de dialogue, l'éditeur indique la valeur du paramètre calculé à partir de la position du pivot. Vous pouvez alors essayer de l'ajuster à la main.

5 Personnalisation et ajout de nouvelles fonctionnalités

Il est possible de personnaliser son éditeur de grille en ajoutant des fonctionnalités dans le bouton *More*. Pour cela, il suffit d'écrire une méthode *public* dans la classe `KVIDGridEditor` et d'ajouter son nom dans la liste de choix située dans la méthode *SuggestMoreActions()*.

6 Raccourcis clavier

Des raccourcis claviers sont maintenant accessibles grâce à l'implémentation de la classe `KVKeyHandler`. Ils sont accessibles uniquement si la touche *Verr.Num* est inactive.

F1-12	boutons du haut de l'interface
a	<i>All</i>
z	<i>Select</i>
e	<i>Edit</i>
s	<i>SaveCurrentGrid</i>
d	<i>Delete Mode on/off</i>
l	new <i>Line</i>
w	<i>Age of Empire Mode on/off</i>
x	<i>Unzoom</i>
c	<i>SetLogz</i>
v	<i>Vénèr Mode on/off</i>

Le pavé directionnel permet également de se déplacer sur l'histogramme.