

**Département de génie logiciel et des T.I.**

Rapport de Laboratoire

|  |  |
| --- | --- |
| **Numéro du laboratoire** | Laboratoire 4 |
| **Nom du laboratoire** | Courbes et transformations Affines |
| **Étudiant(s)** | Gildor Makesa Mvuemba  Olivier, Granger- Hotte |
| **Code(s) permanent(s)** | MAKM87260201  GRAO89120006 |
| **Numéro d’équipe** | 06 |
| **Cours** | GTI 411 |
| **Session** | Hiver 2025 |
| **Groupe** | S20251-GTI411-01 |
| **Chargé(e) de laboratoire** | Lucas Mercier |
| **Date** | 13 avril 2025 |

Table des matières

[Introduction 2](#_Toc837023986)

[Outils et concepts 3](#_Toc2140231116)

[Partie 1 : Courbes paramétriques 3](#_Toc621816131)

[Droites linéaires 3](#_Toc1696111877)

[Courbes de Bézier 4](#_Toc2098453058)

[Courbes d’Hermite 4](#_Toc1920017661)

[Courbes B-Spline 4](#_Toc1133907329)

[Partie 2 : Transformation affines 5](#_Toc1388448203)

[Implémentation 7](#_Toc1775686738)

[Résultats et discussion 7](#_Toc1518134116)

[Conclusion 12](#_Toc634790741)

[Annexe : Manuel d’utilisateur 13](#_Toc1787802117)

[Exécution du code 13](#_Toc1542160471)

# Introduction

L'imagerie numérique joue un rôle essentiel dans de nombreux domaines, allant de l'animation et la modélisation 3D à l'analyse biomédicale et la vision par ordinateur. Parmi les outils fondamentaux de cette discipline, les courbes paramétriques permettent de modéliser des trajectoires, d'interpoler des données ou encore de générer des formes complexes. De même, les transformations affines sont essentielles pour manipuler et aligner des images, notamment dans le traitement d’images médicales ou la simulation graphique. Ce laboratoire vise à explorer ces concepts à travers l’implémentation de courbes paramétriques et d’opérations de transformation sur une image et des points de repère articulaires.

Dans un premier temps, nous travaillerons sur la génération de différentes courbes paramétriques, dont les droites linéaires, les courbes de Bézier, les courbes d’Hermite et les courbes B-spline, en veillant à respecter les propriétés mathématiques de chacune. Ensuite, nous nous pencherons sur les transformations affines, qui nous permettront de modifier la position, l’échelle et l’orientation d’une image et de ses points associés. L’objectif est d’assurer une bonne compréhension de ces outils fondamentaux en les implémentant manuellement.

Finalement, dans ce rapport, nous détaillerons d’abord l’approche adoptée pour dessiner les différentes courbes paramétriques, puis nous expliquerons les méthodes utilisées pour appliquer les transformations affines sur une image et un ensemble de points articulaires. Chaque section comprendra une description des algorithmes implémentés ainsi que les résultats obtenus.

# Outils et concepts

## Partie 1 : Courbes paramétriques

### Droites linéaires

Les droites linéaires sont les formes les plus simples d’interpolation entre deux points. En effet, elle relie deux points par une interpolation linéaire, suivant une évolution proportionnelle au paramètre *t*. Cette forme est souvent utilisée pour connecter deux points dans un graphique ou modéliser des trajectoires simples et rectilignes.

### Courbes de Bézier

Les courbes de Bézier, quant à elles, offrent une plus grande flexibilité. Définies à partir de points de contrôle, elles sont construites à l’aide d’un processus d’interpolation récursif connu sous le nom d’algorithme de De Casteljau. Selon le nombre de points, on obtient une courbe linéaire (2 points), quadratique (3 points) ou cubique (4 points).

Elles sont omniprésentes dans les logiciels de dessin vectoriel, l’animation, la conception de polices de caractères ou la modélisation 3D, car elles permettent de créer des courbes lisses et précises à partir d’un petit nombre de paramètres.

### **Courbes d’Hermite**

Les courbes d’Hermite se définissent par des points de position et des vecteurs tangents associés, permettant ainsi d’intégrer l’orientation locale dans la définition de la courbe. Contrairement aux courbes de Bézier, où les tangentes sont implicites, les courbes d’Hermite offrent une manipulation directe des dérivées, ce qui permet un meilleur contrôle de la forme et de la direction. Chaque segment est ainsi déterminé par deux points et leurs vecteurs tangents, indiquant la direction à l’entrée et à la sortie du segment. Cette caractéristique est particulièrement avantageuse pour des applications techniques et physiques, bien que moins intuitive en conception graphique pure, et se traduit par des mouvements plus naturels en animation et simulation.

### **Courbes B-Spline**

Enfin, les courbes B-Spline (Basis Spline) généralisent les courbes de Bézier en répartissant l’influence des points de contrôle sur des portions locales de la courbe. Elles assurent une continuité de haut niveau (C1, C2...) et permettent de manipuler des formes complexes de façon stable. Elles sont largement utilisées en CAO/FAO pour la conception de pièces industrielles, dans la modélisation de surfaces, ou encore pour représenter des données biométriques ou des structures anatomiques en traitement d’image.

## Partie 2 : Transformation affines

**Matrices de transformations affines 2D**

Les transformations affines sont des opérations géométriques permettant de modifier la position, la taille, l’orientation et même la forme d’un objet dans le plan. Ces transformations sont exprimées au moyen de matrices 3×3 en coordonnées homogènes, ce qui facilite la combinaison séquentielle de plusieurs opérations en une unique transformation.

**Matrices de transformations affines 2D**

Les transformations affines sont des opérations géométriques qui permettent de modifier la position, la taille, l’orientation et la forme d’un objet dans un plan. Elles sont représentées sous forme de matrices 3×3 en coordonnées homogènes, ce qui permet de combiner plusieurs transformations de manière élégante et efficace.

**Matrice de translation**

La translation permet de déplacer un point dans le plan selon un vecteur de déplacement. Elle modifie uniquement la position du point sans changer sa forme, son orientation ou sa taille.

**Matrice d’agrandissement**

Le redimensionnement consiste à modifier la taille d’un objet en multipliant ses coordonnées par des facteurs d’échelle sx et sy pour les directions horizontale et verticale respectivement. Cette transformation conserve la forme de l’objet tout en modifiant ses dimensions.

**Matrice de transvection**

La transvection, aussi appelée cisaillement, déforme l’objet en inclinant ses axes parallèles. Dans le cas du cisaillement horizontal, par exemple, chaque point voit sa coordonnée x modifiée proportionnellement à sa coordonnée y, ce qui entraîne un étirement ou une inclinaison de l’objet sans modifier sa surface globale.

**Matrice de rotation**

La rotation fait tourner un objet autour d’un point (souvent l’origine) d’un certain angle θ. Cette transformation conserve la forme et la taille de l’objet. Pour une rotation autour de l’origine, les nouvelles coordonnées sont calculées en fonction des fonctions trigonométriques de θ.

**Pourquoi les matrices de transformation homogènes sont utiles ?**

Les matrices de transformation homogènes simplifient la manipulation et la combinaison de diverses transformations géométriques en offrant un cadre mathématique unifié et efficace pour le traitement des images et des objets en 2D (et en 3D).

# Implémentation

**Fonction Bézier linéaire**

Elle prend en entrée deux points, un point de départ et un point d’arrivée, et un paramètre t compris entre 0 et 1. Elle va alors calculer un point situé quelque part sur le segment droit reliant les deux, en fonction de la valeur de t. Par exemple, si t vaut 0, le point retourné est le point de départ, si t vaut 1, c’est le point d’arrivée, et si t vaut 0.5, c’est le point au milieu du segment. Cette méthode est donc utilisée pour tracer une ligne droite ou pour animer un objet se déplaçant entre deux positions

**Fonction Bézier quadratique**

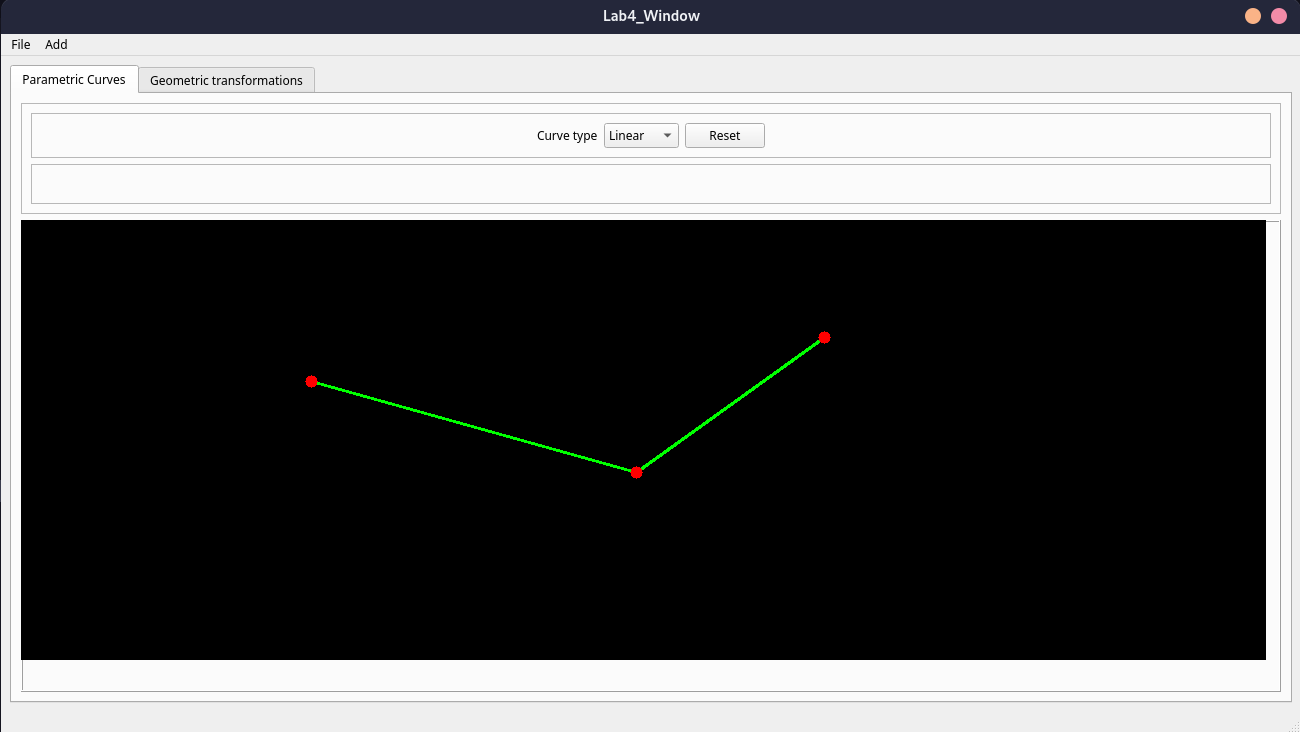
La fonction Bézier quadratique introduit un troisième point, appelé point de contrôle, qui permet de courber la trajectoire. Elle commence toujours au premier point (p0), se termine au dernier (p2), mais elle est "tirée" vers le point de contrôle (p1), ce qui donne une courbe douce et arquée. Le calcul se fait en deux étapes d’interpolation successives entre les points. D’abord entre les points initiaux, puis entre les résultats obtenus, toujours en fonction du paramètre t. Le résultat est une courbe qui suit une trajectoire plus fluide entre les deux extrémités, influencée par la position du point de contrôle.

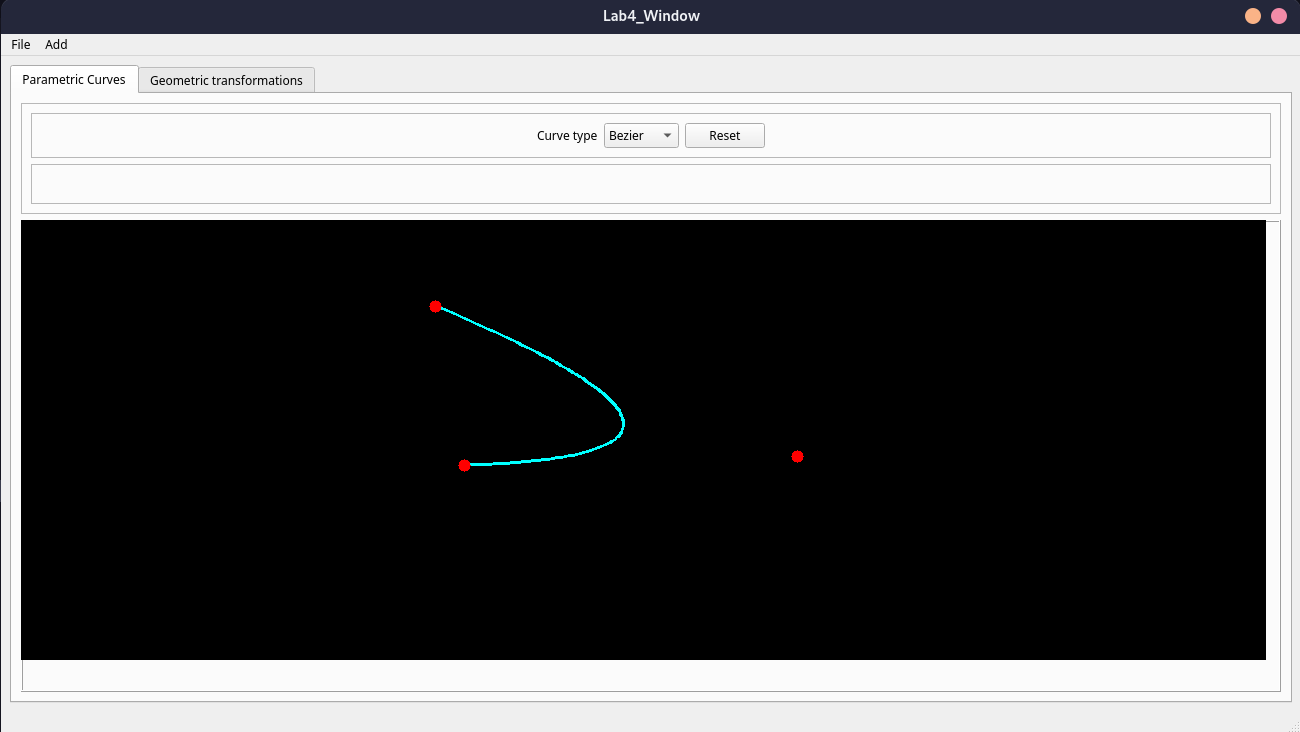
**Fonction Bézier cubique**

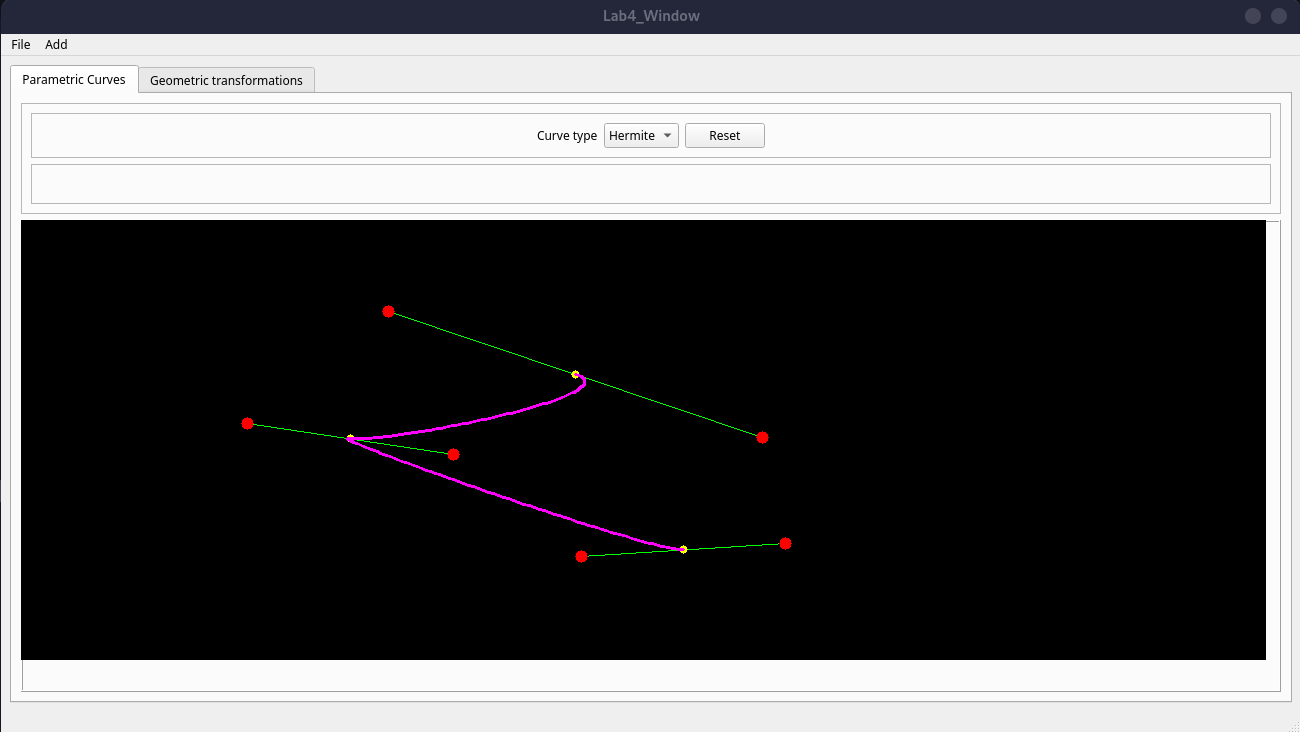
La fonction Bézier cubique est une version plus avancée qui utilise quatre points : un point de départ, un point d’arrivée, et deux points de contrôle. Ces deux points intermédiaires permettent de créer des formes encore plus complexes et précises. Le principe reste le même : il s’agit d’une série d’interpolations entre les points, à plusieurs niveaux, toutes basées sur le même paramètre t. Cette méthode permet de dessiner des courbes très fluides, avec des changements de direction, ce qui la rend idéale pour les animations ou les dessins vectoriels.

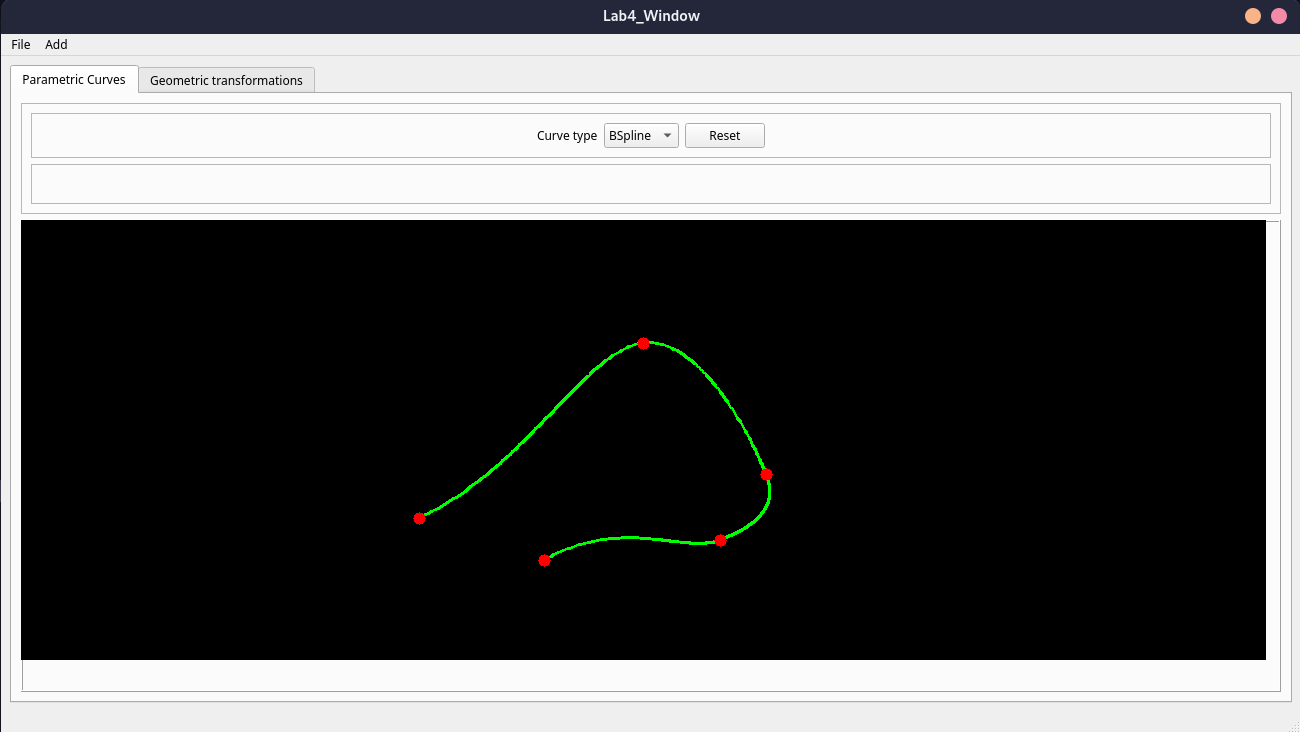
# Résultats et discussion

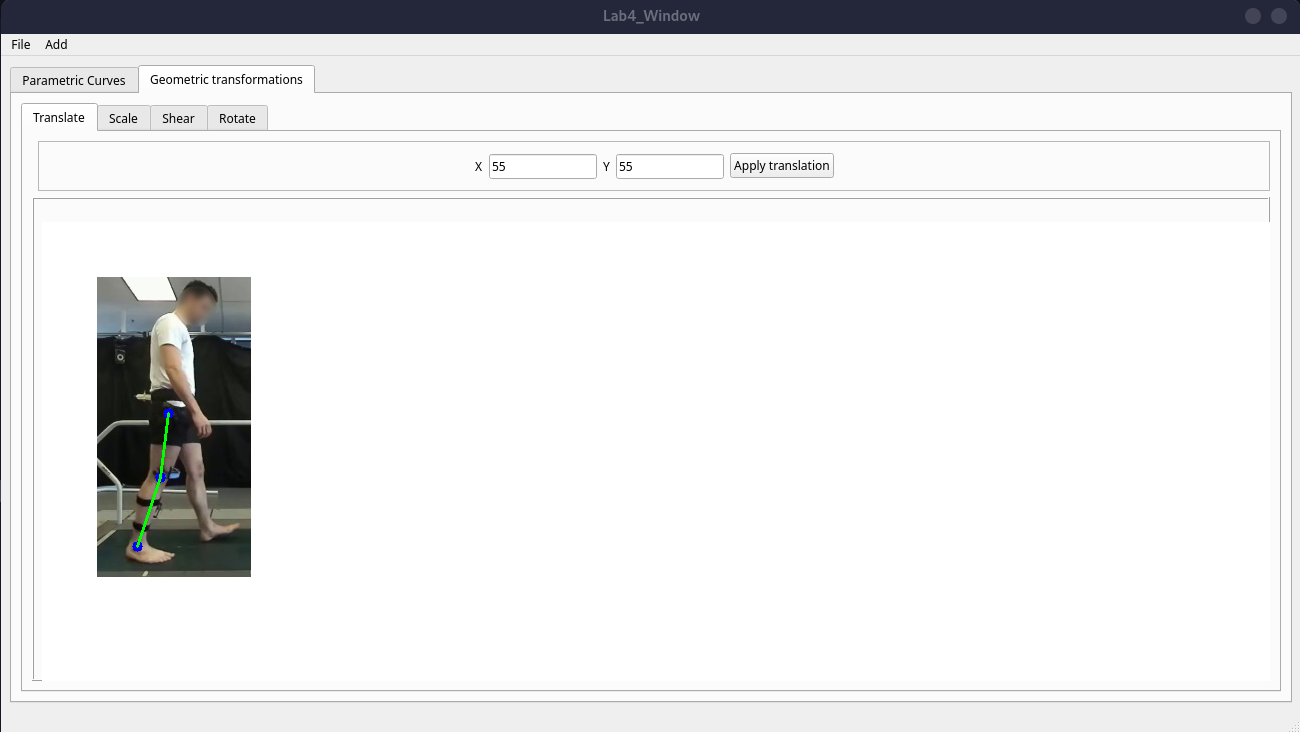
**Résultats**

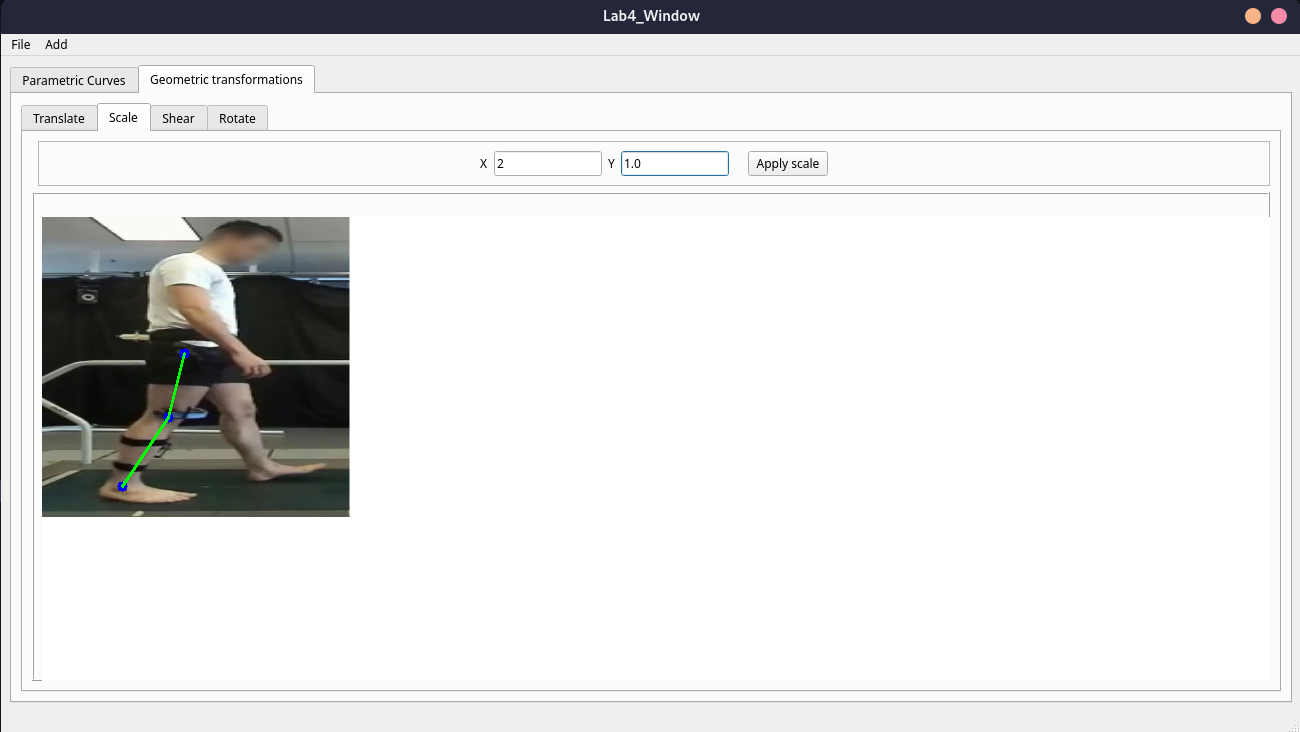
*Figure 1 – Courbe paramétrique linéaire*

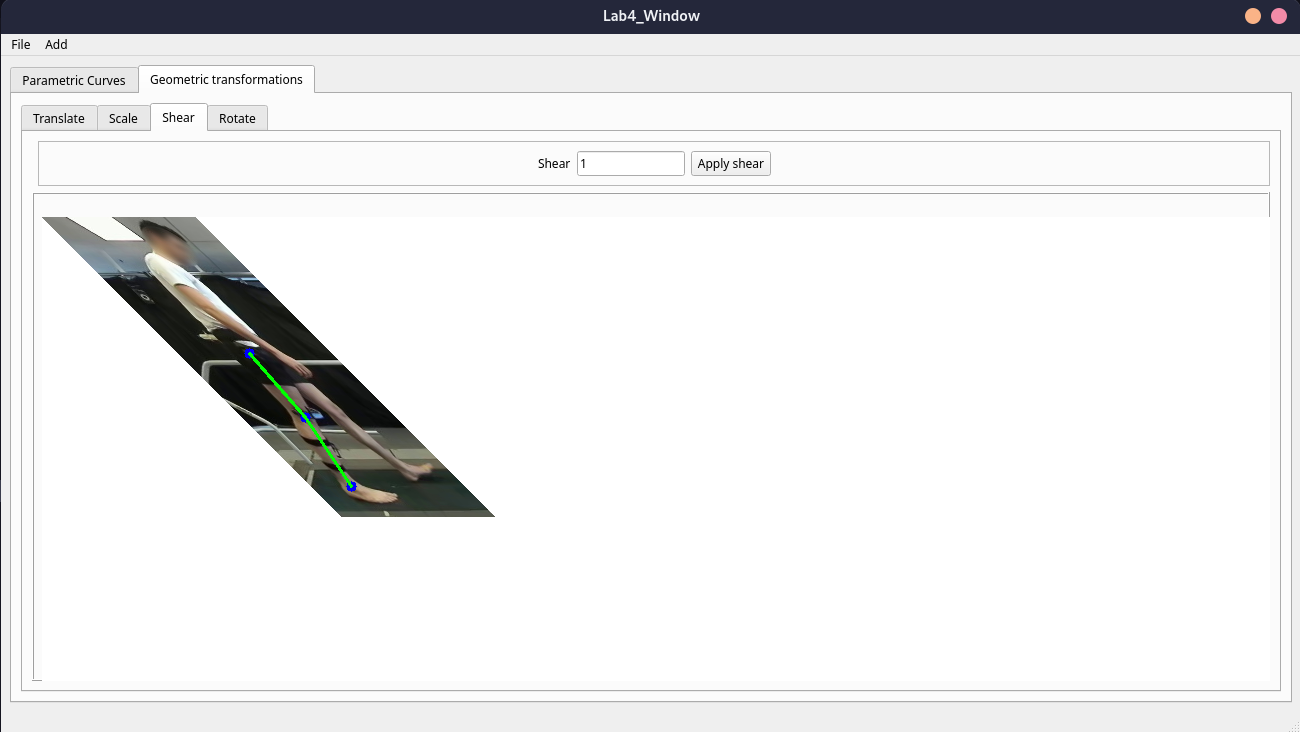
*Figure 2 – Courbe paramétrique de Bézier*

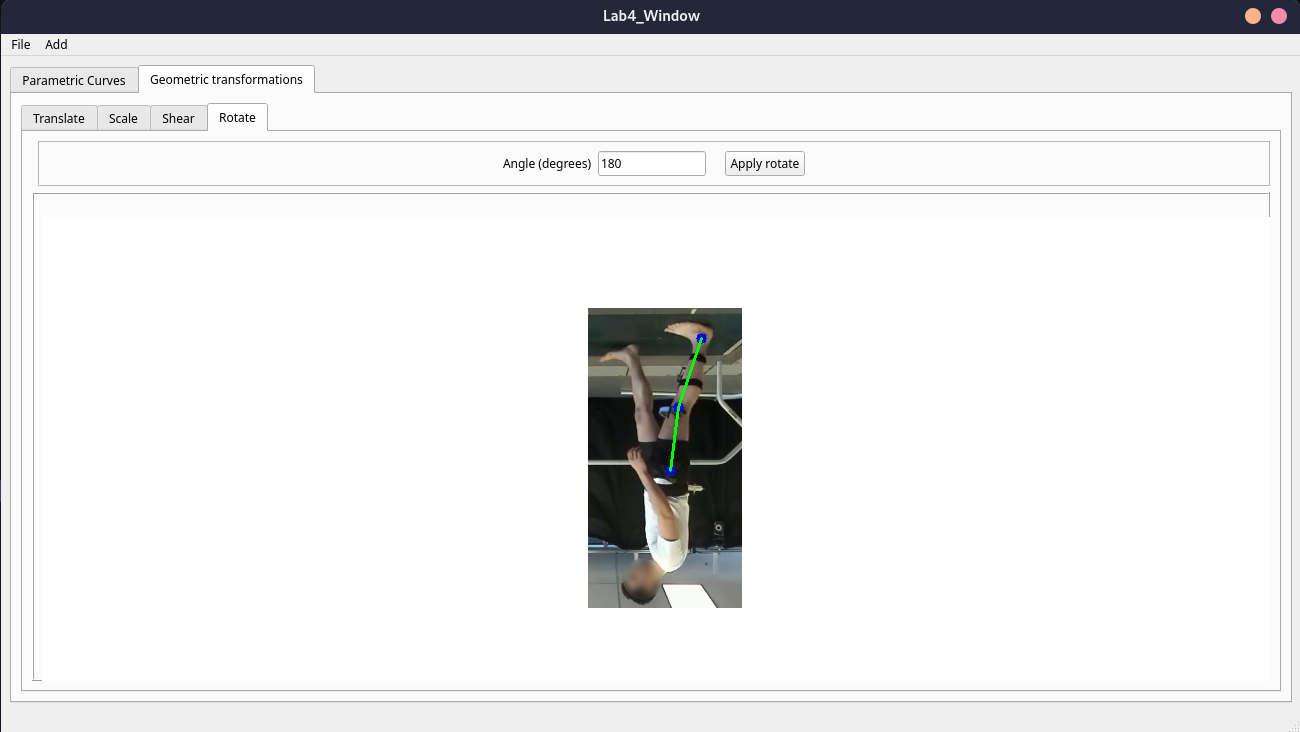
*Figure 3 – Courbe paramétrique d’Hermite*

*Figure 4 – Courbe paramétrique BSpline*

*Figure 5 – Transformation affine: Translation*

*Figure 6 – Transformation affine: Scale*

*Figure 7 – Transformation affine: Shear*

*Figure 8 – Transformation affine: Rotation*

**Problème rencontré avec la B-spline**

* Dans mon implémentation de la courbe B-spline, j’ai constaté que la courbe ne passait pas exactement dans l’ensemble des points de contrôle. En effet, la B-spline « classique » est avant tout une **courbe d’approximation : elle passe à proximité** de chacun des points, mais ne les traverse pas obligatoirement.

Une image contenant capture d’écran, diagramme, Graphique

Description générée automatiquement

**Propositions d’amélioration**

* **Utiliser une spline interpolante** : Pour forcer la courbe à passer exactement par tous les points, j’ai changé le code afin de recourir à une **Catmull-Rom spline** (spline-cubique (degrés 3)). Ainsi, la courbe traverse nécessairement chaque point de contrôle.

Une image contenant capture d’écran, diagramme

Description générée automatiquement

# Conclusion

Ce laboratoire nous a permis d’explorer deux piliers de l’imagerie numérique : les courbes paramétriques et les transformations affines. Grâce à l’implémentation manuelle des droites linéaires, des courbes de Bézier, d’Hermite et des B-Spline, nous avons pu constater comment différents modèles mathématiques peuvent être utilisés pour modéliser des trajectoires et générer des formes complexes avec une grande précision. Chaque type de courbe présente ses avantages spécifiques – de l’interpolation simple des droites linéaires à la flexibilité des courbes de Bézier, en passant par le contrôle direct des dérivées offert par les courbes d’Hermite et la stabilité des B-Spline dans la modélisation avancée.

Parallèlement, nous avons étudié l’impact des transformations affines sur l’image et les ensembles de points, en nous appuyant sur des matrices 3×3 en coordonnées homogènes. La translation, le scaling, la transvection et la rotation constituent autant de techniques permettant non seulement de repositionner et redimensionner des objets, mais aussi de combiner ces opérations de manière efficace et élégante. Ces outils sont essentiels dans de nombreux domaines, allant de la conception assistée par ordinateur à la simulation graphique, en passant par l’animation et l’analyse biomédicale.

En somme, l’acquisition de ces compétences nous a permis de mieux comprendre et manipuler les éléments fondamentaux de l’imagerie numérique, ouvrant ainsi la voie à des applications plus complexes et à l’optimisation des processus de traitement et de visualisation. Ce laboratoire constitue donc une étape clé pour la maîtrise des techniques de modélisation et de transformation, indispensables dans le monde contemporain de l’informatique graphique et du traitement d’images.

# Annexe : Manuel d’utilisateur

**Installation des dépendances**

Pour installer les différentes librairies, on peut utiliser la commande suivant (avec l’utilitaire pip):

* pip install numpy opencv-python PyQt5

## **Exécution du code**

On exécute dans le terminal la commande Python suivante:

* python Lab\_1\_imagerie/GTI411 - LAB 4/MainWindow.py