

Infographie pour la Science des Données

1 Introduction : Infographie et Science des Données

L'informatique graphique, ou infographie, est la discipline qui s'intéresse à la création, au traitement et à l'exploitation des images numériques à l'aide de l'informatique. Elle joue un rôle croissant dans la science des données, notamment pour la visualisation et l'interprétation de grands ensembles de données complexes.

Definition 1.1. Une **Scène 3D** est une représentation virtuelle d'un environnement tridimensionnel. Elle est composée de plusieurs éléments essentiels :

- **Objets 3D** : Les éléments géométriques qui peuplent la scène (personnages, bâtiments, objets divers).
- **Matériaux** : Les propriétés de surface des objets 3D qui déterminent comment ils interagissent avec la lumière (couleur, texture, brillance, transparence).
- **Lumières** : Les sources lumineuses qui éclairent la scène (directionnelles, ponctuelles, ambiantes).
- **Caméra** : Le point de vue virtuel à partir duquel la scène est observée et rendue en une image 2D.

2 Historique de l'Infographie

L'infographie a connu une évolution rapide depuis ses débuts.

2.1 Sketchpad (1963)

Ivan Sutherland développe le **Sketchpad**, considéré comme le premier système graphique interactif complet. Il permettait de :

- Sélectionner, pointer, dessiner, éditer des formes géométriques directement sur un écran.
- Utiliser des structures de données et des algorithmes spécifiques pour la manipulation graphique.
- Mettre en œuvre une modélisation hiérarchique des objets.
- Interagir via des menus contextuels (pop-up).

Lien vidéo : https://youtu.be/6orsmFndx_o

2.2 Affichage

L'évolution des technologies d'affichage a été cruciale :

- **Écran à affichage vectoriel** :
 - 1963 : Oscilloscope modifié utilisé pour les premiers affichages graphiques.
 - 1974 : "Picture System" d'Evans et Sutherland, un système d'affichage vectoriel avancé.
- **Écran à affichage raster** :
 - 1975 : Buffer de frames (mémoire tampon d'image) introduit par Evans et Sutherland, permettant l'affichage point par point (pixel).
 - Années 1980 : Ordinateurs personnels basés sur les bitmaps (images matricielles). L'Apple Macintosh (1984) et les moniteurs CRT (Cathode Ray Tube) en sont des exemples.
 - Années 1990 : Écrans LCD (Liquid-Crystal Displays) sur les ordinateurs portables.
 - Années 2000 : Appareils photo numériques et projecteurs à micro-miroirs.
- **Autres technologies** : Stéréo, casques de Réalité Virtuelle, affichages tactiles, haptiques, audio 3D.

2.3 Entrées (Inputs)

Les dispositifs permettant d'interagir avec les systèmes graphiques ont également évolué :

- **2D** : Stylo lumineux, tablette graphique, souris, joystick, trackball, écran tactile.
- **3D** : Traqueurs de position 3D, systèmes à multiples caméras, télémètres actifs (mesure de distance).
- **Autres** : Gants de données (tactiles), reconnaissance vocale, reconnaissance gestuelle.

2.4 Rendu (Rendering)

Le rendu est le processus de génération d'une image 2D à partir d'une scène 3D. Les techniques ont évolué pour plus de réalisme :

- **Années 1960 - Problème de visibilité :**
 - Algorithmes de lignes cachées : Roberts (1963), Appel (1967).
 - Algorithmes de surfaces cachées : Warnock (1969), Watkins (1970).
 - Tri par visibilité : Sutherland (1974).
- **Années 1970 - Graphiques en Raster :**
 - Lumière diffuse : Gouraud (1971).
 - Lumière spéculaire : Phong (1974).
 - Texture, surface courbe : Blinn (1974).
 - Z-buffer (tampon de profondeur) : Catmull (1974).
 - Anti-aliasing / antirénelage : Crow (1977).
- **Début Années 1980 - Illumination globale :**
 - Ray tracing (lancer de rayons) : Whitted (1980).
 - Radiosité : Goral et al. (1984), Cohen (1985).
 - Équation de rendu : Kajiya (1986).
- **Fin Années 1980 - Photoréalisme :**
 - Arbres d'ombrage : Cook (1984).
 - Langage de shading (programmation des effets de surface) : Perlin (1985).
 - RenderMan : Hanrahan & Lawson (1990).
- **Début Années 1990 - Rendu de non-photoréalisme :**
 - Rendu des volumes : Drebin et al. (1988), Levoy (1988).
 - Peinture impressionniste : Haeberli (1990).
 - Illustration automatique à l'encre et stylo : Salesin et al. (1994-).
 - Rendu de peinture : Meier (1996).
- **Fin Années 1990 - Rendu basé sur images :**
 - Interpolation de points de vue : Chen & Williams (1993).
 - Modélisation plénoptique : McMillan & Bishop (1995).
 - Rendu des champs lumineux : Levoy & Hanrahan (1996).

2.5 Programmation

L'évolution matérielle et logicielle a permis des avancées majeures :

- **Depuis début Années 1980 - Cartes Graphiques :**
 - 1979 : Premier processeur programmable dédié au graphisme 3D (Clark).
 - 1982 : Création de Silicon Graphics, Adobe et AutoDesk.
 - 1987 : Première carte graphique grand public.
 - 1992 : OpenGL 1.0 (API graphique standard).
 - 1993 : Création de NVIDIA.
 - 1994 : VRML (Virtual Reality Modeling Language).
 - 1996 : DirectX de Microsoft.
 - 1997 : Java3D de Sun.
 - 2007 : OpenGL 3.0.

3 Applications de l'Infographie

L'infographie trouve des applications dans de nombreux domaines.

3.1 Divertissement

- **Films d'animation :** Création de mondes et personnages virtuels (ex: Shrek).
- **Effets spéciaux (VFX) :** Intégration d'éléments générés par ordinateur dans des prises de vues réelles.
- **Jeux vidéo :** Création d'environnements interactifs en temps réel.

3.2 Design industriel et Ingénierie assistée par ordinateur (CAE)

Modélisation, simulation et visualisation de produits avant leur fabrication (ex: conception automobile).

3.3 Supervision & Téléopération (online)

Contrôle à distance de systèmes (robots, véhicules) avec retour visuel graphique.

3.4 Simulateurs (offline)

- **Conduite/Pilotage :** Entraînement réaliste pour les pilotes et conducteurs.
- **Entraînement :** Simulation de procédures complexes (chirurgie, maintenance).
- **Jeux :** Simulateurs de vol, de course, etc.
- **Téléopération :** Préparation et simulation de missions à distance.

3.5 Visualisation de données

Représentation graphique de données complexes pour faciliter leur compréhension :

- **Médicales :** Imagerie 3D (scanner, IRM).
- **Géophysiques :** Modélisation du sous-sol.
- **Biologiques :** Visualisation de molécules, de structures cellulaires.
- **Dynamique des fluides (CFD) :** Simulation d'écoulements.

3.6 Navigation

Systèmes de cartographie 3D, aide à la navigation (ex: vues aériennes dans les GPS).

3.7 Art & Design

Outils de création numérique pour les artistes et designers.

3.8 Communications

Présentations visuelles, illustrations techniques, publicité.

4 Définitions Fondamentales

Definition 4.1. L'infographie est l'application de l'informatique à la création, au traitement, et à l'exploitation des images numériques .

Remark 4.2. Quelques points clés :

- Le mot "Infographie" est formé à partir de "INFormatique" et "GRAPHIque". C'est une appellation déposée par la société Benson en 1974.
- C'est un domaine interdisciplinaire faisant appel à la physique (lumière, optique), aux mathématiques (géométrie, algèbre linéaire), à la perception humaine, à l'interaction homme-machine, à l'ingénierie, à la conception graphique et à l'art.
- Distinction avec la **Vision par Ordinateur (Computer Vision)** : l'infographie génère des images à partir de descriptions (Input: description de scène -> Output: Image), tandis que la vision par ordinateur analyse des images pour en extraire des informations (Input: Image -> Output: description/interprétation).
- La **géométrie** est l'élément constitutif essentiel de toute image de synthèse. L'organisation spatiale des objets forme un **modèle géométrique** complet de la scène virtuelle.
- L'image de synthèse résulte de l'**interaction** entre les différentes sources de lumière de la scène et les objets qui la composent.

Definition 4.3. En infographie, le terme "**modèle**" peut se référer à :

- **Modèle géométrique** : Modèle d'un objet à rendre dans une image, enrichi par divers attributs (couleur, texture, réflectance, etc.).
- **Modèle mathématique** : Modèle d'un processus physique ou informatique (par exemple, un modèle de calcul de la lumière qui se reflète sur des surfaces brillantes).

5 Pipeline Graphique

Le pipeline graphique est une séquence d'étapes conceptuelles permettant de transformer une description de scène 3D en une image 2D affichable à l'écran. Chaque primitive géométrique (triangle, point, ligne) de la scène passe successivement par ces étapes. Voici les étapes principales du pipeline graphique classique :

1. **Modèles 3D** : Description initiale des objets dans leurs propres systèmes de coordonnées.
2. **Transformations de modélisation** :

- Application des transformations (translation, rotation, mise à l'échelle) pour positionner et orienter les objets les uns par rapport aux autres.
- Passage du système de coordonnées local de chaque objet (Object Space) vers un repère global unique (World Space).

3. Illumination (Shading) :

- Calcul de la couleur de chaque point visible des objets en fonction des propriétés de leurs matériaux, des sources de lumière et de la position de l'observateur.
- Les modèles d'illumination sont souvent locaux (calcul par primitive, sans gestion des ombres portées initialement) : modèles diffus, ambiants, spéculaires (Gouraud, Phong, etc.).

4. Transformation d'affichage (Viewing Transformation) :

- Passage des coordonnées du monde (World Space) aux coordonnées du point de vue (Eye Space ou Camera Space).
- La scène est décrite par rapport à la position et à l'orientation de la caméra. Le repère est souvent aligné avec l'axe Z.

5. Clipping (Découpage) :

- Élimination des parties de la scène qui se trouvent en dehors du volume de vision de la caméra (View Frustum).
- Passage en coordonnées normalisées (NDC - Normalized Device Coordinates), généralement un cube où les coordonnées varient entre -1 et 1 ou 0 et 1.
- Suppression des parties hors du volume de vision.

6. Transformation écran (Projection) :

- Projection des primitives 3D (après clipping) sur un plan 2D, simulant ce que la caméra "voit".
- Passage des coordonnées NDC vers l'espace image 2D (Screen Space), en pixels. Les coordonnées (left, right, bottom, top, near, far) du volume de vision sont mappées sur les dimensions (width, height) de la fenêtre d'affichage.

7. Pixelisation (Rasterization) :

- Découpage des primitives 2D (projetées) en fragments, correspondant aux pixels de l'écran.
- Interpolation des valeurs connues aux sommets (couleur, profondeur, coordonnées de texture, etc.) pour chaque fragment généré.

8. Visibilité / Rendu Final :

- Détermination des fragments visibles pour chaque pixel (élimination des parties cachées, souvent via un Z-buffer).
- Remplissage du frame buffer (mémoire image) avec la couleur finale des fragments visibles pour former l'image finale.

5.1 Systèmes de coordonnées

Le passage à travers le pipeline implique plusieurs changements de systèmes de coordonnées :

- **Repère objet (Object Space) :** Coordonnées locales à chaque objet.
- **Repère scène (World Space) :** Coordonnées globales communes à tous les objets.
- **Repère caméra (View/Eye Space) :** Coordonnées relatives à la caméra.
- **Repère caméra normalisé (NDC) :** Coordonnées normalisées après clipping (-1 à 1 ou 0 à 1).
- **Espace écran (Screen Space) :** Coordonnées en pixels sur l'écran.

5.2 Vue d'ensemble et Implémentation

Le pipeline graphique traite des :

- **Modèles géométriques** : Objets, surfaces, etc.
- **Modèle d'illumination** : Calcul des interactions lumineuses.
- **Caméra** : Point de vue et ouverture (frustum).
- **Fenêtre (viewport)** : Grille de pixels sur laquelle l'image est plaquée.
- **Couleurs** : Intensités convenant à l'afficheur (ex: 24 bits RVB).

Le pipeline peut être implémenté de diverses manières :

- Entièrement en **software** (logique configurable), sans carte graphique dédiée (lent).
- Avec des **cartes graphiques** (hardware) qui accélèrent certaines étapes :
 - *Première génération* : Hardware pour la rasterisation/pixelisation. Le reste en software configurable.
 - *Deuxième génération* : Hardware configurable pour la transformation/clipping et la rasterisation.
 - *Troisième génération* : Hardware programmable (GPU) pour la plupart des étapes, offrant une grande flexibilité (vertex/pixel shaders).
- À certaines étapes, des **outils de programmation** (ex: vertex program, pixel program/shader) permettent de personnaliser le traitement.

6 GPU (Graphics Processing Unit)

Le GPU est un processeur massivement parallèle spécialisé dans les calculs graphiques, mais de plus en plus utilisé pour des calculs généraux (GPGPU).

- **Matériel spécialisé** : Très performant pour certaines opérations graphiques (transformations de matrices, traitement de pixels/vertices). Son catalogue d'opérations natives est plus réduit que celui d'un CPU.
- **Parallélisme massif** : Conçu pour exécuter la même opération (le même code/shader) sur des millions de points de données (vertices, pixels) simultanément (architecture SIMD - Single Instruction, Multiple Data).
- **Efficacité** : Très efficace lorsque le même code est exécuté sur de grandes quantités de données homogènes.
- **Divergence** : Moins performant lorsque le flot d'exécution diverge (par exemple, si des pixels voisins doivent exécuter des branches différentes d'un 'if' dans un shader), car certaines unités de calcul deviennent inactives.

Un GPU typique contient de nombreuses unités d'exécution (Exec) organisées en groupes (SIMD/SIMT), des caches mémoire, des unités spécialisées (Texture, Tessellation, Clip/Cull, Rasterize, Z-buffer/Blend) et un ordonnanceur (Scheduler/Work Distributor) pour distribuer le travail, le tout connecté à une mémoire dédiée (GPU Memory).

7 Rappels Mathématiques Essentiels

7.1 Pourquoi les mathématiques ?

Remark 7.1. De nombreux concepts graphiques font appel aux mathématiques :

- Systèmes de coordonnées
- Transformations (rotation, translation, mise à l'échelle)
- Rayonnement (Ray-casting, lancer de rayons)
- Conversion des couleurs
- Tests d'intersection
- Requêtes géométriques
- Simulations physiques
- Et bien d'autres...

L'algèbre linéaire et l'analyse vectorielle sont particulièrement fondamentales.

7.2 scalaire

Definition 7.2. Un **scalaire** est une grandeur totalement définie par un nombre (magnitude) et éventuellement une unité. Il n'a pas d'orientation dans l'espace.

Remark 7.3. • **Exemples :** Masse, distance, température, volume, densité.

- **Propriétés :** Les scalaires obéissent aux lois de l'algèbre ordinaire.
- **Opérations élémentaires :** Addition et multiplication.
- **Propriétés des opérations :**
 - Commutativité : $\alpha + \beta = \beta + \alpha$; $\alpha \cdot \beta = \beta \cdot \alpha$
 - Associativité : $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$; $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$
 - Distributivité : $\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$
- **Identité :**
 - Addition : $\alpha + 0 = 0 + \alpha = \alpha$ (0 est l'élément neutre)
 - Multiplication : $\alpha \cdot 1 = 1 \cdot \alpha = \alpha$ (1 est l'élément neutre)

7.3 Vecteur

Definition 7.4. Un **vecteur** est une entité mathématique définie par n valeurs numériques (composantes) extraites du même ensemble E (par exemple $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$). Ces valeurs décrivent le module (longueur) et l'orientation du vecteur.

Remark 7.5. • n est appelé la **dimension** du vecteur. On dit que le vecteur est défini dans E^n , qui est un espace vectoriel de dimension n .

- **Exemple :** Un vecteur dans \mathbb{R}^3 est noté $\vec{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ ou (x, y, z) .

- **Exemples physiques :** Déplacement, vitesse, accélération, force.
- Les vecteurs obéissent aux lois de l'**algèbre vectorielle**.

7.3.1 Vecteurs unitaires et base

Remark 7.6. • Dans un repère, les vecteurs sont décrits dans une **base**, généralement composée de **vecteurs unitaires** (norme 1) orthogonaux entre eux.

- Dans \mathbb{R}^3 , la base canonique est $(\vec{i}, \vec{j}, \vec{k})$, où $\vec{i} = (1, 0, 0)$, $\vec{j} = (0, 1, 0)$, $\vec{k} = (0, 0, 1)$.
- Un vecteur \vec{V} est représenté par l'addition des vecteurs de base multipliés par ses **projections** (composantes) sur chaque axe :

$$\vec{V} = x'\vec{i} + y'\vec{j} + z'\vec{k}$$

où x', y', z' sont les composantes de \vec{V} dans la base $(\vec{i}, \vec{j}, \vec{k})$.

7.3.2 Opérations sur les vecteurs

Remark 7.7. • Opérations élémentaires

- Produit scalaire
- Produit vectoriel
- Addition de vecteurs
- Produit vecteur-scalaire
- Normalisation

7.4 Addition de deux vecteurs

Remark 7.8. • **Géométriquement :** L'addition de deux vecteurs \vec{A} et \vec{B} obéit à la règle du parallélogramme (ou règle du triangle). Le vecteur résultant $\vec{A} + \vec{B}$ va de l'origine de \vec{A} à l'extrémité de \vec{B} lorsque \vec{B} est placé à l'extrémité de \vec{A} .

- **Algébriquement :** Cela se fait en additionnant les composantes correspondantes. Si $\vec{A} = (A_x, A_y, A_z) = A_x\vec{i} + A_y\vec{j} + A_z\vec{k}$ et $\vec{B} = (B_x, B_y, B_z) = B_x\vec{i} + B_y\vec{j} + B_z\vec{k}$, alors :

$$\vec{A} + \vec{B} = (A_x + B_x)\vec{i} + (A_y + B_y)\vec{j} + (A_z + B_z)\vec{k}$$

$$\vec{A} + \vec{B} = (A_x + B_x, A_y + B_y, A_z + B_z)$$

Example 7.9 (Addition de composantes). Un vecteur \vec{A} peut être décomposé en ses composantes rectangulaires $A_x = A \cos \theta_A$ et $A_y = A \sin \theta_A$ (en 2D). Pour additionner $\vec{R} = \vec{A} + \vec{B}$: $R_x = A_x + B_x = A \cos \theta_A + B \cos \theta_B$ $R_y = A_y + B_y = A \sin \theta_A + B \sin \theta_B$ Le module de \vec{R} est $R = \sqrt{R_x^2 + R_y^2}$ et son angle θ_R est tel que $\tan \theta_R = R_y / R_x$.

7.5 Norme d'un vecteur

Definition 7.10. La **norme** (ou module, longueur) d'un vecteur \vec{V} , notée $||\vec{V}||$ ou $|\vec{V}|$, est sa taille.

Remark 7.11. • Elle est calculée par le théorème de Pythagore en utilisant ses composantes.

- En 2D, pour $\vec{V} = x'\vec{i} + y'\vec{j}$, $\|\vec{V}\| = \sqrt{(x')^2 + (y')^2}$.
- En 3D, pour $\vec{V} = x'\vec{i} + y'\vec{j} + z'\vec{k}$, $\|\vec{V}\| = \sqrt{(x')^2 + (y')^2 + (z')^2}$.
- En dimension n , on applique le même principe récursivement.
- La norme du vecteur \vec{AB} (vecteur allant du point A au point B) est la distance entre A et B.

7.6 Normalisation d'un vecteur

Definition 7.12. La **normalisation** est le processus qui transforme un vecteur non nul en un vecteur unitaire (de norme 1) ayant la même direction et le même sens.

Remark 7.13. • Pour normaliser un vecteur \vec{V} , il suffit de diviser chacune de ses composantes par sa norme $\|\vec{V}\|$.

- Le vecteur normalisé \vec{v} est : $\vec{v} = \frac{\vec{V}}{\|\vec{V}\|}$.
- Si $\vec{V} = (x', y', z')$, alors $\vec{v} = \left(\frac{x'}{\|\vec{V}\|}, \frac{y'}{\|\vec{V}\|}, \frac{z'}{\|\vec{V}\|} \right)$, avec $\|\vec{V}\| = \sqrt{(x')^2 + (y')^2 + (z')^2}$.

7.7 Produit scalaire (Dot Product)

Definition 7.14. Le **produit scalaire** de deux vecteurs \vec{A} et \vec{B} , noté $\vec{A} \cdot \vec{B}$, est un scalaire.

Remark 7.15. • **Définition géométrique :** C'est le produit des modules des deux vecteurs par le cosinus de l'angle θ entre eux :

$$\vec{A} \cdot \vec{B} = \|\vec{A}\| \|\vec{B}\| \cos \theta$$

Il correspond aussi au produit du module de l'un par la projection du second sur le premier.

- **Définition par composantes :** C'est la somme des produits des composantes correspondantes :

$$\vec{A} \cdot \vec{B} = A_x B_x + A_y B_y + A_z B_z$$

- **Propriétés :**

- Commutativité : $\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{u}$
- Distributivité par l'addition : $(\vec{u} + \vec{v}) \cdot \vec{w} = \vec{u} \cdot \vec{w} + \vec{v} \cdot \vec{w}$
- Distributivité par un scalaire k : $\vec{u} \cdot (k\vec{v}) = k(\vec{u} \cdot \vec{v})$

- **Signe et angle :** Le signe du produit scalaire renseigne sur l'angle θ entre les vecteurs (supposés non nuls) :

- $\vec{A} \cdot \vec{B} > 0 \implies \cos \theta > 0 \implies -90^\circ < \theta < 90^\circ$ (angle aigu)
- $\vec{A} \cdot \vec{B} = 0 \implies \cos \theta = 0 \implies \theta = \pm 90^\circ$ (vecteurs orthogonaux)
- $\vec{A} \cdot \vec{B} < 0 \implies \cos \theta < 0 \implies 90^\circ < \theta < 180^\circ$ ou $-180^\circ < \theta < -90^\circ$ (angle obtus)

On peut calculer l'angle entre deux vecteurs via : $\cos \theta = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$

- **Applications :**

- Projection d'un vecteur sur un autre.
- Calcul d'angle entre deux vecteurs.
- Élimination des faces cachées (back-face culling) : si la normale à une face pointe à l'opposé de la direction de vue (produit scalaire négatif), la face est cachée.
- Calcul de la quantité de lumière perçue (loi de Lambert : intensité proportionnelle au cosinus de l'angle entre la normale et la direction de la lumière, i.e., au produit scalaire de leurs vecteurs unitaires).
- Ombres simples.

7.8 Produit vectoriel (Cross Product)

Definition 7.16. Le **produit vectoriel** de deux vecteurs \vec{A} et \vec{B} dans \mathbb{R}^3 , noté $\vec{A} \times \vec{B}$ ou $\vec{A} \wedge \vec{B}$, est un vecteur.

Remark 7.17. • **Direction :** Le vecteur résultant $\vec{A} \times \vec{B}$ est perpendiculaire au plan formé par \vec{A} et \vec{B} . Son sens est donné par la règle de la main droite (si les doigts de la main droite se courbent de \vec{A} vers \vec{B} , le pouce indique la direction de $\vec{A} \times \vec{B}$).

- **Module :** Le module du produit vectoriel est égal au produit des modules des deux vecteurs par le sinus de l'angle θ entre eux :

$$\|\vec{A} \times \vec{B}\| = \|\vec{A}\| \|\vec{B}\| \sin \theta$$

Ceci correspond à l'aire du parallélogramme formé par \vec{A} et \vec{B} . Le module est nul si les vecteurs sont parallèles ($\theta = 0$ ou $\theta = 180^\circ$) et maximal s'ils sont perpendiculaires ($\theta = 90^\circ$).

- **Calcul par composantes :**

$$\begin{aligned} \vec{A} \times \vec{B} &= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} \\ &= (A_y B_z - A_z B_y) \vec{i} - (A_x B_z - A_z B_x) \vec{j} + (A_x B_y - A_y B_x) \vec{k} \\ &= (A_y B_z - A_z B_y, A_z B_x - A_x B_z, A_x B_y - A_y B_x) \end{aligned}$$

- **Propriétés :**

- Anticommutativité : $\vec{u} \times \vec{v} = -(\vec{v} \times \vec{u})$
- Distributivité sur l'addition : $(\vec{u} + \vec{v}) \times \vec{w} = \vec{u} \times \vec{w} + \vec{v} \times \vec{w}$
- Distributivité par un scalaire k : $(k\vec{u}) \times \vec{v} = \vec{u} \times (k\vec{v}) = k(\vec{u} \times \vec{v})$
- Non-associativité : $(\vec{u} \times \vec{v}) \times \vec{w} \neq \vec{u} \times (\vec{v} \times \vec{w})$ en général.

- **Application principale :** Calcul de la normale à un plan ou à une surface (par exemple, la normale à un triangle défini par les sommets P1, P2, P3 peut être calculée par $(\vec{P2} - \vec{P1}) \times (\vec{P3} - \vec{P1})$).

Exemple 7.18 (Calcul de produit vectoriel). Soit $\vec{A} = (1, 2, -4)$ et $\vec{B} = (3, -1, 5)$.

$$\begin{aligned}\vec{A} \times \vec{B} &= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 1 & 2 & -4 \\ 3 & -1 & 5 \end{vmatrix} \\ &= \vec{i}(2 \times 5 - (-4) \times (-1)) - \vec{j}(1 \times 5 - (-4) \times 3) + \vec{k}(1 \times (-1) - 2 \times 3) \\ &= \vec{i}(10 - 4) - \vec{j}(5 + 12) + \vec{k}(-1 - 6) \\ &= 6\vec{i} - 17\vec{j} - 7\vec{k} \\ &= (6, -17, -7)\end{aligned}$$

7.9 Matrice

Définition 7.19. On appelle **matrice** M un tableau à deux indices de $n \times m$ valeurs numériques extraites du même ensemble E . n est le nombre de lignes et m est le nombre de colonnes. On note $M = (m_{ij})$ où $1 \leq i \leq n$ et $1 \leq j \leq m$.

Remark 7.20. • **Exemple :** Une matrice 4×5 :

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} \end{pmatrix}$$

- Si $n = m$, la matrice est dite **carrée**.

7.9.1 Opérations sur les matrices

Remark 7.21. • **Addition (Matrice-matrice, Scalaire-matrice) :** Se fait terme à terme pour des matrices de même dimension.

• **Multiplication :**

- *Scalaire-matrice :* On multiplie chaque élément de la matrice par le scalaire.
- *Matrice-vecteur :* Le produit d'une matrice M ($m \times n$) par un vecteur colonne V ($n \times 1$) est un vecteur colonne W ($m \times 1$). La i -ème composante de W est le produit scalaire de la i -ème ligne de M par le vecteur V .

$$w_i = \sum_{k=1}^n m_{ik} v_k \quad (1 \leq i \leq m)$$

Exemple 7.22 (Produit Matrice-Vecteur).

$$\begin{pmatrix} 2 & 4 \\ 6 & 8 \\ 10 & 12 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \times 1 + 4 \times 3 \\ 6 \times 1 + 8 \times 3 \\ 10 \times 1 + 12 \times 3 \end{pmatrix} = \begin{pmatrix} 2 + 12 \\ 6 + 24 \\ 10 + 36 \end{pmatrix} = \begin{pmatrix} 14 \\ 30 \\ 46 \end{pmatrix}$$

- *Matrice-matrice :* Le produit d'une matrice M_1 ($n \times m$) par une matrice M_2 ($m \times p$) est une matrice M ($n \times p$). L'élément m_{ij} de M est le produit scalaire de la i -ème ligne de M_1 par

la j -ème colonne de M_2 .

$$m_{ij} = \sum_{k=1}^m (m_1)_{ik} (m_2)_{kj} \quad (1 \leq i \leq n, 1 \leq j \leq p)$$

Example 7.23 (Produit Matrice-Matrice).

$$\begin{pmatrix} 1 & 3 \\ 7 & 9 \end{pmatrix} \times \begin{pmatrix} 2 & 4 \\ 6 & 8 \\ 10 & 12 \end{pmatrix} \quad (\text{impossible: dimensions incompatibles})$$

$$\begin{aligned} \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{pmatrix} \times \begin{pmatrix} 2 & 4 \\ 6 & 8 \\ 10 & 12 \end{pmatrix} &= \begin{pmatrix} 1(2) + 3(6) + 5(10) & 1(4) + 3(8) + 5(12) \\ 7(2) + 9(6) + 11(10) & 7(4) + 9(8) + 11(12) \end{pmatrix} \\ &= \begin{pmatrix} 2 + 18 + 50 & 4 + 24 + 60 \\ 14 + 54 + 110 & 28 + 72 + 132 \end{pmatrix} = \begin{pmatrix} 70 & 88 \\ 178 & 232 \end{pmatrix} \end{aligned}$$

- **Inversion** : Pour certaines matrices carrées, il existe une matrice inverse M^{-1} telle que $MM^{-1} = M^{-1}M = I$ (matrice identité).
- **Transposition** : La transposée M^T d'une matrice M est obtenue en échangeant les lignes et les colonnes ($(M^T)_{ij} = M_{ji}$).

Les matrices sont fondamentales en infographie pour représenter les transformations géométriques (rotation, translation, mise à l'échelle, projection).

8 Espace Vectoriel et Espace Affine

Remark 8.1. • **Espace vectoriel** : C'est l'espace où vivent les **vecteurs** (représentant des déplacements ou des directions). Ses propriétés principales sont l'existence d'un vecteur nul (origine) et la stabilité par combinaison linéaire (toute combinaison linéaire de vecteurs de l'espace appartient à l'espace).

- **Espace affine** : C'est l'espace où vivent les **points**. Il est construit à partir d'un point de référence (l'origine) et d'un espace vectoriel associé (qui définit les déplacements autorisés à partir de ce point). On peut y définir des objets géométriques comme les droites, les plans, etc. La différence entre deux points d'un espace affine est un vecteur de l'espace vectoriel associé. L'addition d'un point et d'un vecteur donne un point.

Remark 8.2. En infographie, on travaille constamment avec ces deux types d'espaces. Les sommets des objets sont des points (espace affine), tandis que les normales, les directions de lumière ou les déplacements sont des vecteurs (espace vectoriel). Les transformations sont souvent représentées par des matrices qui agissent différemment sur les points et les vecteurs (notamment via l'utilisation de coordonnées homogènes).