

# 7. Reprezentarea geometrică a percepției

---

---

## Reprezentări geometrice

---

---

În cursurile precedente s-au descris metode pentru raționamentul despre sistemele de referință spațiale ale unui robot, dar încă nu s-a luat în considerare conținutul geometric al acestor sisteme, cum ar fi forma și mărimea structurii fizice a fiecărui element. Este nevoie să luăm în considerare aceste geometrii atunci când dezvoltăm mișcări și alegem poziții, deoarece un robot ar trebui să evite coliziunile neintenționate cu mediul, respectiv să intre în contact atunci când este necesar. Geometria este, de asemenea, importantă în proiectarea roboților, deoarece forma unui robot dictează cât de bine se poate încadra în locații strâmte și la cât din spațiul înconjurător poate ajunge, fără a se ciocni cu propriile părți.

Reprezentările matematice și computaționale ale geometriei au fost studiate mult timp, și un set bogat de tehnici sunt în uz activ astăzi. Ele provin dintr-o varietate de domenii, inclusiv proiectarea asistată de calculator (CAD), geometria computațională, grafica computerizată și vizualizarea științifică. Prin urmare, acest studiu atinge doar suprafața vastei varietăți de reprezentări și calcul.

### 7.1 Exemple de aplicații

Geometria este utilizată pe scară largă în robotică, inclusiv în:

- Proiectarea mecanismelor
- Simularea proceselor fizice
- Evitarea obstacolelor
- Detectarea proximității
- Planificarea mișcării
- Calibrarea
- Maparea 3D
- Recunoașterea de obiecte
- Vizualizarea și evaluarea performanțelor

Aceste aplicații vor necesita efectuarea unor *operații geometrice* asupra unuia sau mai multor obiecte. Câteva exemple includ: vizualizare, detectarea coliziunilor, calcul de distanțe, trasarea razelor (en. ray-casting), simplificare sau potrivirea formelor.

## 7.2 Reprezentări geometrice

O geometrie  $G$  este considerată ca o regiune a unui spațiu  $\mathbb{R}^n$  unde  $n$  este 2 pentru roboți planari, respectiv 3 pentru roboți 3D. Reprezentările geometrice pot fi categorizate în funcție de structura care este reprezentată:

- *Primitive* precum puncte, linii, sfere și triunghiuri ce reprezintă o formă  $G$  cu un număr fix de parametri.
- *Suprafețe* ce aproximează doar marginea lui  $G$ , notată  $\partial G$ .
- *Volume* ce aproximează întreaga regiune: interior, frontieră și exterior.
- *Spații de puncte* ce stochează un număr finit de puncte din suprafața  $\partial G$ .

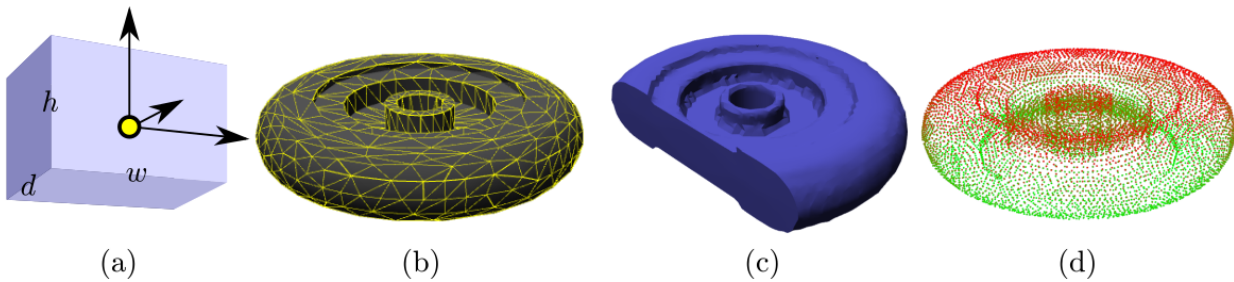


Fig. 7.1 Tipuri de reprezentări: (a) *Primitive*, în acest caz este reprezentat un spațiu de o dimensiune, poziție și orientare fixă. (b) *Suprafață*, un exemplu fiind o rețea din triunghiuri. (c) *Volume* fiind reprezentat ca un spațiu tăiat. (d) *Spațiu de puncte*, observându-se faptul că este practic o suprafață discretă.

Există mai multe compromisuri implicate în alegerea reprezentărilor. În general, reprezentările de tip suprafață sunt cele mai folosite pentru vizualizare și CAD, în timp ce reprezentările volumetrice sunt cele mai utile pentru reprezentarea hărților 3D și pentru clasificarea punctelor în puncte interioare, respectiv exterioare. Reprezentările bazate pe puncte sunt larg utilizate pentru a reprezenta date brute din senzori 3D precum lidar și camere de adâncime. În schimb, lucrul cu primitive geometrice este ales în cazurile în care viteza este mai importantă decât acuratețea sau completitudinea datelor.

### 7.2.1 Primitive geometrice

Primitivele reprezintă cea mai simplă formă de reprezentare geometrică. Ele sunt compacte, convenabile matematic și se pretează la operații geometrice rapide deoarece calculele exacte pot fi efectuate în timp constant. Cu toate acestea, ele reprezintă cea mai puțin flexibilă formă de reprezentare și nu redau cu acuratețe ridicată majoritatea proceselor.

Primitivele uzuale includ puncte, segmente și forme 2D sau 3D. Pentru fiecare tip de primitivă, forma este reprezentată printr-un număr fix de parametri.

- Un **punct** este caracterizat de coordonatele acestuia:  $(x, y)$  sau  $(x, y, z)$ .
- Un **segment** este caracterizat de coordonatele punctelor  $a$  și  $b$  ale capetelor acestuia.
- Un **cerc** sau o **sferă** se definește prin punctul centrului  $c$  și raza  $r$  pozitivă.

- Un **triunghi** se caracterizează prin cele 3 segmente din care este compus:  $a$ ,  $b$  și  $c$ .
- Un **cilindru** este definit de o bază  $b$ , o axă principală  $a$ , o înălțime  $h$  pozitivă și o rază  $r$  pozitivă.
- Un **dreptunghi** sau un **paralelipiped dreptunghic** reprezentat printr-o axă de coordonate  $T$  cu originea la colțul său și dimensiuni atribuite celor 2 axe în 2D (lățime / înălțime), respectiv celor 3 axe în 3D (lățime / înălțime / adâncime).
- Un **dreptunghi** sau un **paralelipiped dreptunghic** aliniat pe un sistem de coordonate inferior  $l$  și unul superior  $u$  ce conține toate punctele  $\mathbf{x} = (x_1, x_2, x_3)$ , astfel  $x_i \in [l_i, u_i]$  cu  $i \in \{1, 2, 3\}$ .

### 7.2.2 Reprezentări prin suprafață

Reprezentările spațiilor de tip suprafață (cunoscute și sub numele de reprezentări ale marginilor) stochează frontiera unei geometrii  $\partial G$  fără a reprezenta explicit interiorul / exteriorul acesteia.

Cea mai comună reprezentare 3D a suprafeței este o rețea de triunghiuri  $\{(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_N, b_N, c_N)\}$  care se îmbină pentru a aproxima suprafața geometriei. Echivalentul în 2D îl reprezintă lanțul poligonal. Rețelele de triunghiuri au două mari avantaje față de alte reprezentări:

1. Cu un număr suficient de triunghiuri, se poate aproxima orice suprafață.
2. Echipamentele grafice sunt puternic optimizate pentru a vizualiza rețele mari de triunghiuri

Este important de menționat că, pentru a reprezenta o suprafață solidă, laturile triunghiurilor vecine în rețea trebuie să coincidă, astfel încât majoritatea laturilor să fie reprezentate în mai multe triunghiuri. Pentru a economisi spațiu de stocare / complexitate de reprezentare, rețelele de triunghiuri sunt adesea reprezentate printr-o listă de vârfuri  $\{v_1, v_2, \dots, v_M\}$  și o listă de indici pentru triunghiuri  $\{(i_{a1}, i_{b1}, i_{c1}), (i_{a2}, i_{b2}, i_{c2}), \dots, (i_{aN}, i_{bN}, i_{cN})\}$ , unde indicii  $(i_{ak}, i_{bk}, i_{ck})$  indică faptul că triunghiul  $k$  este compus din vârfurile  $v_{i_{ak}}, v_{i_{bk}}, v_{i_{ck}}$ . De exemplu, Figura 7.2 conține o suprafață solidă a unui cub, reprezentată cu ajutorul unei rețele de triunghiuri.

Alte reprezentări uzuale ale suprafeței includ:

- **Poligoane convexe** în 2D sau **politopuri convexe** în 3D, ce sunt caracterizate de o listă de fețe, margini și vârfuri conectate. Acestea au, de asemenea, o reprezentare volumetrică în termeni de jumătăți de plan  $(a_i, b_i)$  cu  $i \in \{1, 2, \dots, N\}$  astfel încât geometria poate fi determinată de mulțimea  $G = \{x | a_i \cdot x \leq b_i, i \in \{1, 2, \dots, N\}\}$ .
- **Curbe parametrice** în 2D sau **Suprafețe parametrice** în 3D, ce sunt exprimate printr-o funcție  $f(u) : D \rightarrow \mathbb{R}^2$ , respectiv  $f(u, v) : D \rightarrow \mathbb{R}^3$ . Aceste funcții generează o curbă / suprafață pe măsură ce parametrul  $D$  variază pe  $\mathbb{R}$ .

- **Suprafețe de subdiviziune**, ce definesc suprafața ca limită a operațiilor de netezire a unei rețele poligonale. Această reprezentare este des întâlnită în grafica computerizată.

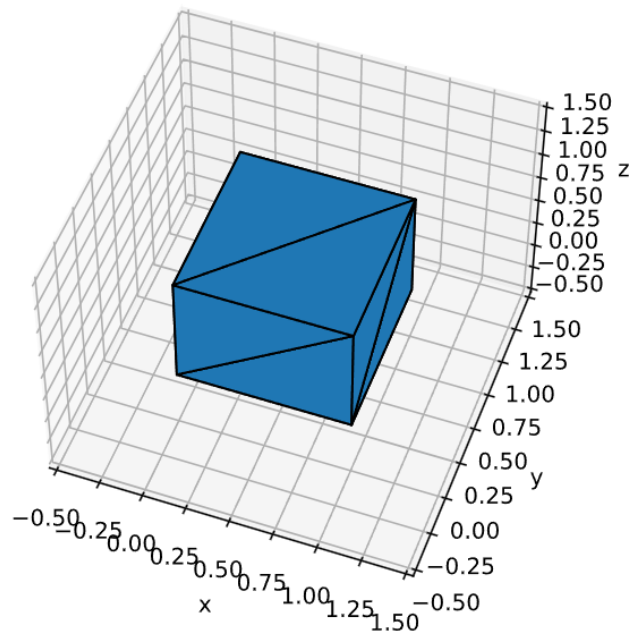


Fig. 7.2 Reprezentarea suprafeței unui cub cu ajutorul a 8 laturi și 12 triunghiuri.

Un aspect important al reprezentărilor suprafeței este că acestea pot să nu închidă corect un volum. De exemplu, un singur triunghi sau două triunghiuri care se intersectează sunt o rețea validă de triunghiuri, dar niciuna dintre ele nu reprezintă o suprafață a unui volum închis. O rețea de triunghiuri care înconjoară cu adevărat un volum este cunoscută sub numele de rețea etanșă. Rețelele care nu sunt etanșe sunt, de asemenea, cunoscute sub numele de *supă* (en. polygon soup).

### 7.2.3 Reprezentări volumetriche

Reprezentările volumetriche au avantajul principal de a putea efectua teste de interior/exterior rapid și de a putea modifica rapid hărți 2D/3D cu informații noi. Toate aceste elemente stochează o funcție  $f$  peste un spațiu  $n$ -dimensional (unde  $n=2$  sau  $3$ ) care indică dacă un punct se află în interiorul sau în afara geometriei.

Valorile funcției pot fi interpretate în mai multe moduri, printre care:

- **Hartă de ocupanță**: o funcție binară  $f : \mathbb{R}^n \rightarrow \{0, 1\}$  unde 0 indică spațiu gol iar 1 spațiu ocupat.
- **Hartă de ocupanță probabilistică**: o funcție  $f : \mathbb{R}^n \rightarrow [0, 1]$  unde valorile reflectă nivelul de certitudine că spațiul este ocupat, 1 fiind ocupat iar 0 fiind gol.
- **Suprafață implicită**: o funcție  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  unde  $f(x) < 0$  reprezintă interiorul geometriei,  $f(x) = 0$  suprafața iar  $f(x) > 0$  exteriorul.
- **Câmp de distanță** (en. SDF - Signed distance field): o suprafață implicită în

care  $f$  indică distanța euclidiană până la suprafață dacă este în afara geometriei sau adâncimea de pătrundere negativă dacă este în interiorul geometriei.

- **Câmp de distanță trunchiat** (en. TSDF - Truncated signed distance field): o variantă uzuală în cartografierea 3D, care acceptă o distanță de trunchare  $\tau$ . Valorile câmpului de distanță cu valoare absolută mai mare decât  $\tau$  sunt trunchate.
- **Hartă de atenuare / reflectanță**: o funcție  $f : \mathbb{R}^n \rightarrow [0, 1]$  ce indică cât de mult un punct din spațiu interferă cu lumina, sunetul sau alte semnale.
- **Hartă de înălțime / hartă topografică**: o combinație între o reprezentare a suprafeței și a volumului. Acestea stochează o funcție 2D  $f(x, y)$  care indică înălțimea suprafeței la punctul  $(x, y)$ . Acest lucru definește în 3D un volum peste  $(x, y, z)$  astfel încât  $z > f(x, y)$  indică spațiu gol,  $z = f(x, y)$  suprafață iar  $z < f(x, y)$  interiorul. Această metodă este folosită în special de roboții mobili pentru a reprezenta terenul.

Reprezentarea computațională a funcției în sine poate varia. Câteva exemple includ:

- **Combinație de funcții de bază**:  $f(x)$  este o combinație liniară a funcțiilor de bază  $f(x) = \sum_i^m c_i b_i(x)$ , unde  $b_i$  sunt funcții de bază fixe. Evaluarea are o complexitate  $O(m)$ .
- **Grid dens de pixeli** (en. Dense pixel / voxel grid): funcția  $f$  este stocată pe un grid 2D sau 3D, iar evaluarea lui  $f(x)$  presupune o căutare în tabel. Pentru suprafețe implicite,  $f(x)$  este de obicei evaluată folosind interpolare biliniară / triliniară a valorilor la nodurile gridului. Evaluarea are o complexitate de  $O(1)$ .
- **Grid ierarhic**: poate fi 2D (en. quadtree) sau 3D (en. octree) care stochează funcția la niveluri progresiv mai fine, până la o rezoluție dată. Evaluarea are o complexitate de  $O(\log_n)$ .
- **Grid rar** (en. Sparse / hash grid): stochează doar celule sau blocuri importante de date volumetrice. De exemplu, celulele ocupate ale unei hărți de ocupanță sau celulele netrunchate ale unui TSDF. Evaluarea are o complexitate de  $O(1)$ .

Diverse combinații de tipuri și reprezentări ale valorilor sunt utile pentru anumite sarcini.

Reprezentările populare includ:

- Hărți de ocupanță pe un grid dens de pixeli sunt folosite în jocul video popular Minecraft pentru a stoca o hartă a unei lumi 3D care poate fi modificată în mod nelimitat.
- Hărți de ocupanță probabilistică pe un grid dens de pixeli sunt folosite în mulți algoritmi 2D de localizare și mapare simultană (ex: SLAM).
- Hartă de ocupanță pe un octree, utilizată în algoritmul popular de mapare 3D OctoMap.
- TSDF pe griduri dense, utilizate în algoritmele de mapare 3D la scară mică (obiect sau cameră unică).
- TSDF pe griduri rare, utilizate în algoritmele de mapare 3D la scară mare (clădiri).

- Griduri de atenuare / reflectanță sunt utilizate în imagistica medicală volumetrică, cum ar fi ecografia, scanările CT și MRI, precum și sonarul utilizat în vehicule subacvatice și pe suprafață.

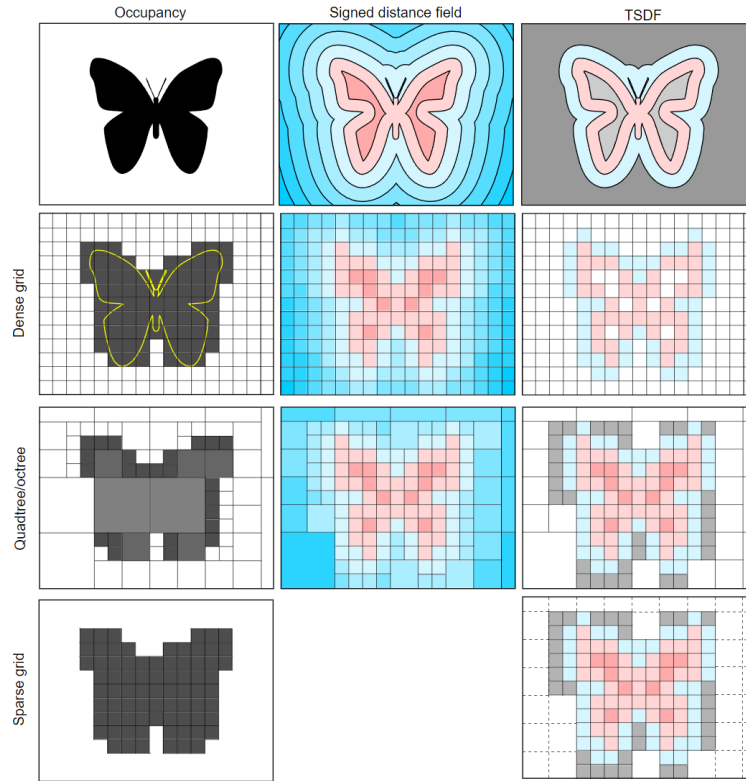


Fig. 7.3 Câteva reprezentări volumetrice populare, ilustrate în 2D.

Toate reprezentările bazate pe grid sunt definite într-un volum, de obicei într-o cutie cu margini aliniate pe axe, cu colțurile  $l$  și  $u$ . Gridul va avea dimensiunile  $m \times n \times p$  iar conversia între sistemul global și grid necesită o simplă deplasare și o scalare. În mod particular, indicii celulelor  $(i, j, k)$  ale unui punct  $(x_1, x_2, x_3)$  sunt determinați printr-o transformare.

$$\begin{aligned} i &= \lfloor m(x_1 - l_1)/(u_1 - l_1) \rfloor \\ j &= \lfloor n(x_2 - l_2)/(u_2 - l_2) \rfloor \\ k &= \lfloor p(x_3 - l_3)/(u_3 - l_3) \rfloor \end{aligned} \quad (7.1)$$

unde  $\lfloor \cdot \rfloor$  este operația de întreg inferior, care returnează cel mai mare număr întreg mai mic sau egal cu argumentul său.

#### 7.2.4 Spații de puncte

Reprezentările bazate pe puncte sunt larg utilizate în robotică pentru a reprezenta geometriile mediului, deoarece sunt tipul de date produs direct de senzorii de adâncime. Aceste reprezentări stochează o eșantionare de puncte  $p_1, \dots, p_N$  de pe suprafața geometriei  $\partial G$ . Punctele individuale pot fi asociate cu culori, normale și alte informații. De obicei, această eșantionare este foarte incompletă, prezentând o densitate neregulată sau goluri mari cauzate

de ocluzii (obiectul nu este vizibil din perspectiva senzorului).

Există două tipuri majore de reprezentări bazate pe puncte: nori de puncte și imagini de adâncime (cunoscute și sub numele de imagini de rază). Un nor de puncte este pur și simplu o listă nestructurată de puncte fără nicio presupunere de regularitate. Punctele nu sunt identificate a priori cu obiecte sau între ele. Acest tip de reprezentare rezultă din senzorii LiDAR care se deplasează și din scanările de adâncime unite în mod imperfect.

O imagine de adâncime este o imagine gri 2D care indică distanța de la senzorul camerei până la suprafața observată de fiecare pixel. (Unele valori de adâncime pot fi invalide, cum ar fi 0 sau  $\infty$ , indică că nu există date disponibile pentru acel pixel). Aceasta este forma brută a datelor returnate de camerele de adâncimecum se poate observa și în figura 7.4. Avantajul acestei forme de reprezentare față de norii de puncte este că proximitatea în imagine implică o anumită noțiune de proximitate spațială în spațiul tridimensional. În plus, linia dintre punctul focal al camerei și punctul citit trebuie să fie lipsită de alte geometrii. Algoritmii pot exploata această cunoaștere pentru operațiuni geometrice mai rapide, ceea ce vom vedea când ne întoarcem la subiectul mapării 3D. Este necesară cunoașterea poziției și orientării camerei în spațiu pentru a corela imaginea 2D cu punctele 3D într-un cadru de referință global.

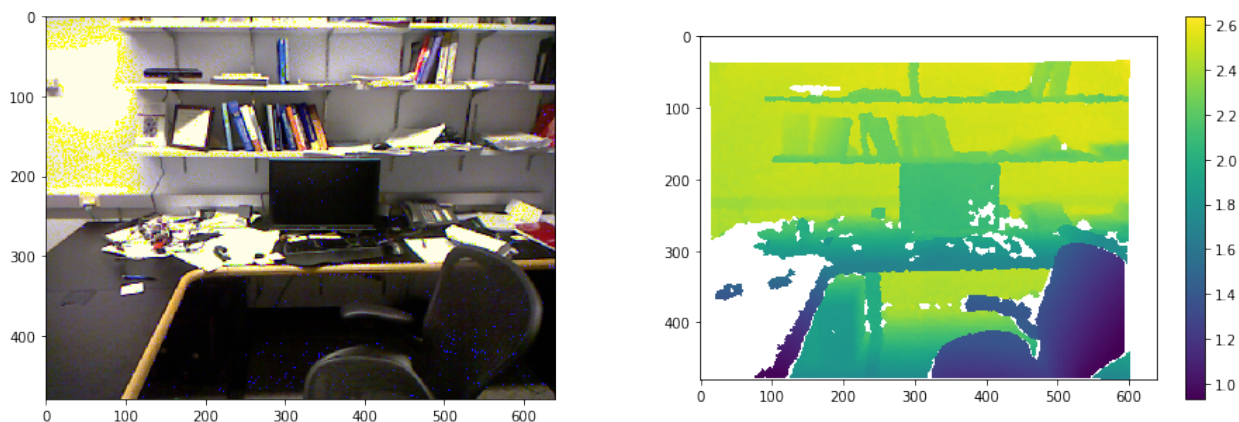


Fig. 7.4 O imagine color și imaginea gri echivalentă captată de o cameră de adâncime.

Atât norii de puncte, cât și imaginile de adâncime vor fi, în general, reprezentări imperfecte ale geometriei din cauza zgomotului senzorului, ocluziunii și a câmpului vizual limitat.

### 7.2.5 Acuratețea reprezentărilor

Cu excepția primitivelor geometrice, fiecare reprezentare poate reda geometria la diferite nivele de rezoluție, cum ar fi densitatea triunghiurilor într-o rețea de triunghiuri sau dimensiunea pixelilor într-un grid de ocupanță. Alegerea reprezentării și a rezoluției acestora ar trebui să fie făcută pentru a echilibra mai multe cerințe concurente:

- Fidelitatea față de geometria reală

- Coerența cu operațiile geometrice dorite
- Complexitatea computațională a operațiilor geometrice
- Dificultatea de achiziție / creare
- Complexitatea implementării
- Cerințele stocare și viteză de transmisie

Se vor discuta mai multe aspecte ale acestor cerințe mai jos. În ceea ce privește fidelitatea, se pot observa câteva tendințe generale. În primul rând, reprezentările de suprafață 3D (de exemplu, rețelele de triunghiuri, norii de puncte) au nevoie în general de  $O(1/h^2)$  elemente pentru a obține o eroare de aproximare de  $h$  față de o anumită suprafață. Reprezentările volumetrice bazate pe un grid 3D vor necesita  $O(1/h^3)$  celule de grid, ceea ce poate fi o povară semnificativă: să presupunem că împărțim o cameră de  $5m \times 5m \times 5m$  în celule de un centimetru:  $500 \cdot 500 \cdot 500 = 125$  milioane de celule ar fi necesare! Pe de altă parte, presupunând că suprafața obiectului are aproximativ  $5m \times 5m$ , o reprezentare bazată pe suprafață ar avea nevoie doar de aproximativ 25.000 de elemente pentru a reprezenta același obiect.

Utilizarea unor structuri de date mai avansate, cum ar fi octree, poate comprima complexitatea spațială a reprezentării volumetrice la aproape  $O(1/h^2)$ , dar la costul creșterii complexității implementării operațiilor geometrice.

### 7.2.6 Reprezentarea pozițiilor rigide și a mișcării

Deoarece este atât de comun să se reprezinte obiectele ca și corpuri rigide, este mult mai ușor să se modeleze mișcarea unui obiect ca mișcarea unui sistem de referință al corpului rigid, decât să se modifice explicit geometria în sine. Această abordare de transformare implicită este folosită pe scară largă în robotică, vizualizare și grafică computerizată. În mod specific, se poate imagina obiectul ca având un sistem de referință local și o geometrie locală  $G_L \subset \mathbb{R}^3$  în coordonate locale. Poziția obiectului este o transformare  $T$  de la sistemul de referință la sistemul global (lume). Pe măsură ce obiectul se mișcă,  $T$  se schimbă, dar datele geometriei locale rămân statice.

Prin urmare, se poate reprezenta mulțimea de puncte conținute de obiectul  $G(t) \subset \mathbb{R}^3$  la momentul  $t$ , dându-se doar poziția  $T(t)$ . Cele două reprezentări sunt:

- Transformare explicită:  $G(t) = \{T(t)|x \in G_L\}$ . Practic, punctele geometriei locale se mișcă împreună cu transformarea obiectului.
- Transformare implicită:  $G(t) = \{x \in \mathbb{R}^3 | T(t)^{-1}x \in G_L\}$ . În acest caz, punctele în coordonatele lumii sunt proiectate înapoi în sistemul de referință.

Deoarece este mult mai eficient să se schimbe transformările față de datele geometriei, abordarea transformării implicite este preferată în cele mai multe cazuri, cu excepția primitivelor geometrice. De exemplu, dacă o rețea de triunghiuri cu 1.000.000 de puncte suferă o rotație și o translație, acest lucru poate fi calculat folosind 1.000.000 de înmulțiri matrice-



vector  $3 \times 3$  și aceeași cantitate de adunări a vectorilor 3D. Cu toate acestea, în reprezentarea implicită, se poate actualiza o transformare folosind doar o înmulțire a matricei  $4 \times 4$  (folosind coordonate omogene) sau o înmulțire a matricei  $3 \times 3$ , o înmulțire a matricei cu vectorul, și o adunare a vectorului 3D (folosind matricea de rotație și vectorii de translație). De acum înainte, se va descrie o geometrie  $G$  în forma implicită ca o pereche (geometrie locală / transformare)  $G = (G_L, T)$ .



# Bibliografie

---

---

- [1] K. Hauser, *Robotic Systems*. Illinois, IL, USA: university of illinois urbana-champaign, 2023. [Online]. Available: <https://motion.cs.illinois.edu/RoboticSystems/>
- [2] C. Powers, D. Mellinger and V. Kumar, *Quadrotor Kinematics and Dynamics*. Dordrecht: Springer Netherlands, 2015, pp. 307–328. [Online]. Available: [https://doi.org/10.1007/978-90-481-9707-1\\_71](https://doi.org/10.1007/978-90-481-9707-1_71)