

TEXTO 1

Texto de apoio à disciplina Modelação Tridimensional

FAUTL – Ano lectivo 2012/2013 – 2º semestre

Professor Luís Mateus (lmmateus@fa.utl.pt)

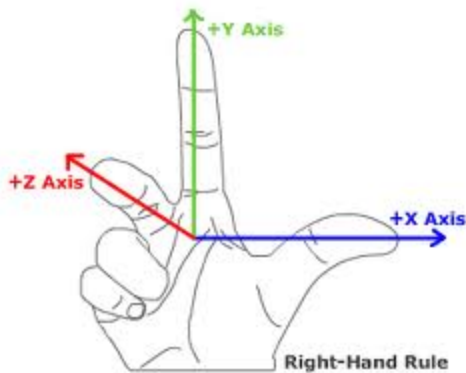
TEXTO 1

1. Introdução à modelação 3D

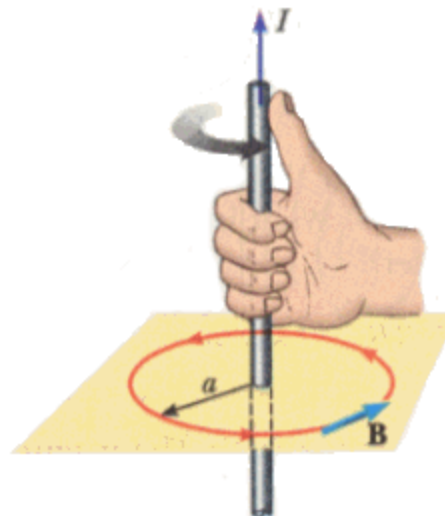
Este conjunto de textos visa apoiar a leccionação da disciplina de modelação 3D. Pressupõe-se que os alunos já possuem conhecimento de modelação 2D e que dominam alguns conceitos e detêm alguma experiência básicos relativos à interacção com software de desenho assistido por computador.

1.1 Sistemas de coordenadas e planos de trabalho

O espaço de trabalho de um modelador 3D é, antes de mais, um espaço de coordenadas tridimensionais. Em geral é considerada a **regra da “mão direita”** para a definição do **referencial cartesiano** e para a definição do **sentido positivo das rotações**.



sistema de coordenadas cartesiano de “mão direita”

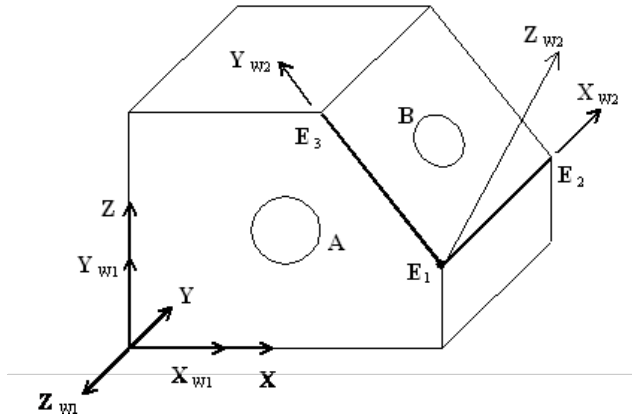


regra da “mão direita” aplicada à rotação

Em geral, e por omissão, é disponibilizado um **sistema de coordenadas cartesiano absoluto** que também se pode designar por **sistema de coordenadas mundo**. Normalmente o utilizador pode definir outros sistemas de coordenadas, que podem assumir a designação de **sistema de coordenadas do utilizador**. A definição destes sistemas de coordenadas está relacionada com a necessidade de mudar o **plano de trabalho**, num processo de modelação de modo a poder construir elementos em posições e orientações variadas. Esta mudança pode ser implícita ou explícita. Se for implícita, o sistema pode inferir a que plano de trabalho o utilizador pretende aceder e permite-lhe o acesso sem nenhuma operação intermédia. Se for explícita, o utilizador tem de declarar, através da inserção de parâmetros apropriados, o sistema de coordenadas que pretende definir.

Formas típicas de definir sistemas de coordenadas do utilizador (ou planos de trabalho) são:

- selecção de três pontos,
- translação e/ou rotação de um plano de trabalho previamente definido,
- alinhamento por uma entidade do modelo,
- definição do eixo z.



Exemplos de vários sistemas de coordenadas do utilizador e respectivos planos de trabalho.

Adicionalmente pode se associada uma localização geográfica a um modelo tridimensional. Essa localização pode ser expressa em coordenadas geográficas (latitude e longitude) e também em termos de definição da direcção do Norte. A localização geográfica é importante se pretendermos simular, por exemplo, o comportamento da luz solar num dia ou hora específicos. É óbvia a utilidade desta possibilidade no contexto do desenvolvimento de um projecto de Arquitectura.

1.2 Tipos de câmaras e tipos de vista

Os tipos de câmara que se utilizam condicionam o tipo de vistas que se podem obter. No essencial uma câmara fica definida por:

- um centro de projecções (ponto de vista ou posição da câmara),
- uma direcção e sentido principal do “olhar” (por exemplo dirigido a um objecto alvo),
- um campo de visão (em inglês “field of view” - FOV),
- uma superfície de projecção,
- um procedimento para mapear a superfície de projecção numa superfície plana (que em última instância é mapeada num ecrã de computador).

Normalmente a superfície de projecção é uma superfície plana.

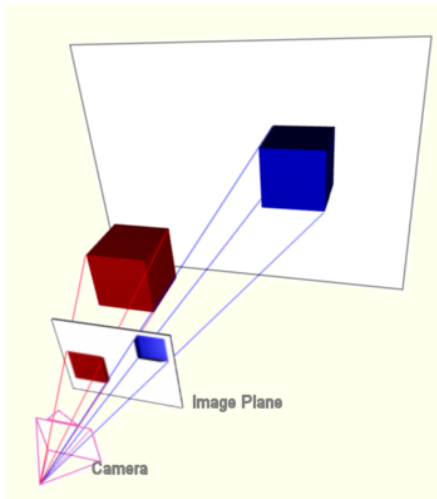
Quanto aos pontos de vista, estes podem estar colocados a distância finita ou a distância infinita.

Se o ponto de vista estiver colocado no infinito, tem-se a projecção paralela (em geral ortográfica). Se o ponto de vista estiver a uma distância finita, tem-se a projecção cónica ou perspectiva.

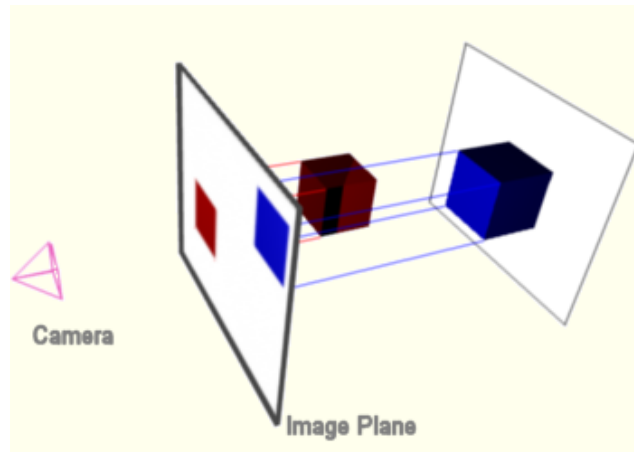
A **projecção ortográfica** permite as vistas em axonometria e as vistas em alçado e planta.

A **projecção perspectiva** permite simular, por exemplo, pontos de vista de peão, em que é possível experimentar maior imersão na cena representada.

Normalmente, qualquer software de modelação 3D permite **múltiplas vistas** (viewports) para visualização e interacção com o modelo. Por exemplo, uma vista pode corresponder a uma planta, outra pode corresponder a um alçado outra a uma visualização em perspectiva.



Projecção perspectiva



Projecção paralela



Múltiplas vistas (viewports)

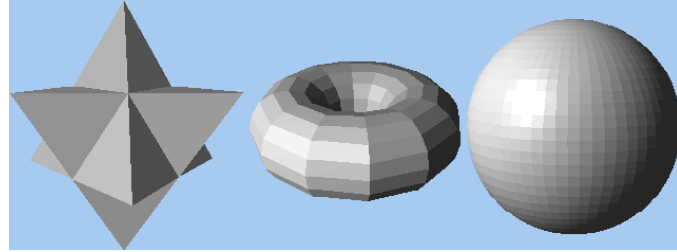
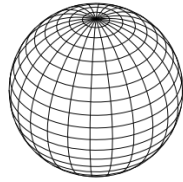
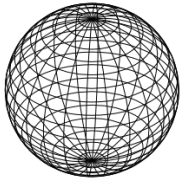
1.3 Estilos de visualização

A forma como as entidades criadas e modeladas aparecem visualmente ao utilizador pode ser variada. Os modos mais comuns de visualização são:

- a) **wireframe**
- b) **wireframe-hidden**
- c) **shaded**

Para além destes modos básicos há muitos outros, com inúmeras combinações de parâmetros. Cada sistema de modelação implementa um conjunto variado destas formas de visualização e permite ao utilizador seleccionar os mais apropriados em cada contexto.

No interface de um software de modelação 3D, cada vista pode ter o seu estilo de visualização de acordo com as opções do utilizador.



Wireframe model com e sem invisibilidades (hidden lines)

Shaded models.

1.4 Modos de navegação

Ao contrário dos sistemas tradicionais de representação, em que a vista sobre o desenho é estática, em que o ponto de vista está cristalizado, em qualquer software de modelação 3D o utilizador pode mudar os pontos de vista sobre a cena modelada. Essa alteração pode ser dinâmica (com inputs do tipo arrastar o rato ou rodar a roda do rato) ou pode ser estática através da inserção de parâmetros (do tipo definição da posição da câmara e ponto alvo).

Em geral é possível navegar dinamicamente através dos seguintes modos:

- a) **pan** (equivale a deslocar lateralmente a câmara).
- b) **zoom** (equivale a aproximar a câmara do objecto)
- c) **orbit** (equivale a rodar a câmara em torno do objecto ou de um ponto definido como pivot)

Existem outros modos de navegação mais ou menos sofisticados que podem ser disponibilizados nas ferramentas de software.

1.5 Criação de entidades 3D

A forma mais directa de criar entidades 3D é através da inserção de **primitivas geométricas** (linhas, figuras planas, figuras 3D), normalmente acessíveis através de menus ou comandos específicos do software que se esteja a utilizar.

Note-se que, uma vez que o espaço de trabalho é conceptualmente tridimensional, qualquer conjunto de entidades colocadas no espaço, ainda que unidimensionais (linhas) ou bidimensionais (faces planas), define, na prática um modelo tridimensional.

O processo de criação de uma entidade num modelo implica a definição de uma **posição**, de um **tamanho** e de uma **orientação**.

Qualquer um destes parâmetros pode ser definido arbitrariamente, sendo a entidade ajustada posteriormente. Alternativamente, o controlo da posição pode ser feito através da inserção de coordenadas (em modo de **coordenadas absolutas** ou **coordenadas relativas / cartesianas** ou **polares**) ou através da inferência de posição por interacção com entidades previamente definidas (por exemplo o extremo de uma linha ou o centro de um círculo).

O tamanho e a orientação podem ser controlados pela inserção numérica de parâmetros ou utilizando entidades do modelo como constrição.

A orientação também pode ser controlada de modo eficiente através da definição prévia de sistemas de coordenadas do utilizador e respectivos planos de trabalho, conforme referido anteriormente.

1.6 Interacção básica com as entidades do modelo 3D

As formas básicas de interacção com as entidades de um modelo são:

- a) alteração do tamanho (**escala**)
- b) alteração da posição (**mover**)
- c) alteração da orientação (**rotação**)
- d) clonagem (**cópia** ou **reflexo**)

2. Figuras e primitivas geométricas e arquitectónicas

Por **figura geométrica** entende-se qualquer entidade que se possa caracterizar por ter posição, e/ou extensão. Exemplos de figuras geométricas são: pontos, rectas, segmentos de recta, círculos, esferas, cubos, prismas, etc. Num software de modelação 3D estas figuras geométricas designam-se por **primitivas geométricas** e podem ser instanciadas através de comandos específicos (por exemplo um comando pode permitir desenhar uma linha, ou uma esfera ou um toro).

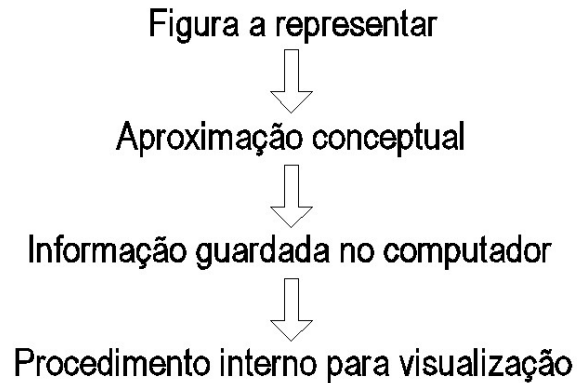
Analogamente, é possível falar de **primitivas arquitectónicas**. Tratar-se-á de elementos, com significado do ponto de vista da arquitectura (por exemplo um bloco de uma janela, um bloco de um sanitário, um bloco de uma coluna,...). Estas primitivas são compostas de figuras geométricas articuladas entre si ou resultantes de variadas operações geométricas.

Para efeitos deste texto, considerar-se-ão primitivas geométricas do tipo **linhas, superfícies e sólidos**.

Ao invocar um comando para a criação de uma primitiva geométrica é necessário ter em conta a sua natureza geométrica. Essa natureza geométrica traduz-se no tipo de **parâmetros e opções de input** que são solicitados ao utilizador. Este é um argumento no sentido da necessidade de compreensão da natureza geométrica (e não só) do repertório de primitivas geométricas disponibilizadas num software, se se pretender ter um controlo efectivo sobre as suas funcionalidades e se se pretender tirar o maior partido possível da ferramenta.

Uma primitiva geométrica é guardada como uma listagem de elementos e internamente o software dispõe de procedimentos que tratam essa informação e a tornam visível ao utilizador. Isto significa que, embora conceptualmente uma primitiva possa ter uma determinada natureza, pode acontecer que esta seja aproximada visualmente através de figuras distintas. Por exemplo uma superfície esférica pode ser representada como uma superfície poliédrica (facetada), e neste caso, pode ser apenas guardado como informação a localização do centro e do raio, sendo a visualização resolvida pelo procedimento interno ao software. Ou uma linha matematicamente complexa ou sem expressão matemática definida (linhas de forma livre) pode ser aproximada por arranjos de linhas mais simples, por exemplo polínómios do 2º grau (cónicas) ou do 3º grau (cúbicas) a que geralmente se dá o nome de splines, e que na prática são casos particulares de Nurbs (Non-uniform rational B-splines).

Resumindo, pode-se considerar a seguinte sequência:



Workflow de representação de uma figura num software de modelação.

A figura a representar pode ser por exemplo uma linha de forma livre resultante de um gesto de desenho, a aproximação conceptual pode ser a decisão de representar aquela linha através de arcos de circunferência, a informação guardada no computador poderá ser o centro de cada arco e os seus extremos, e o procedimento interno a segmentação dos arcos em linhas poligonais de modo a satisfazer um critério de qualidade visual.

2.1 Linhas

Conceptualmente, uma linha pode ser considerada como o resultado do movimento de um ponto no espaço de acordo com uma determinada lei ou regra.

Exemplos de linhas são: segmentos de recta, os arcos de circunferência, elipses, Nurbs, ciclóides, tóricas, etc.

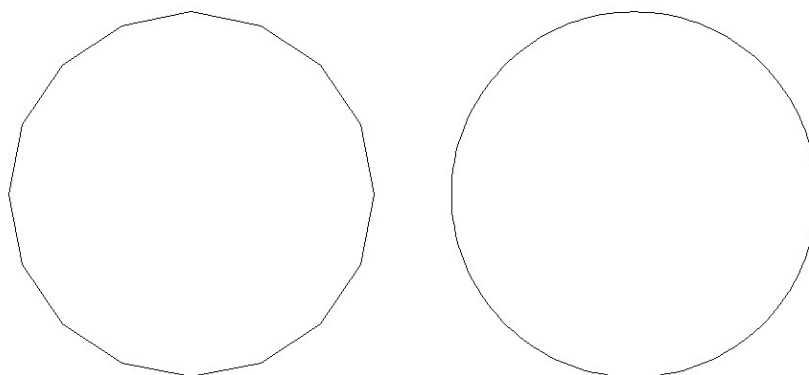
A criação de um segmento de recta implica a definição das coordenadas dos seus extremos, a criação de uma linha Nurbs implica a definição de pontos por onde a linha passa ou a definição de pontos de controlo, uma ciclóide implica considerar o movimento de um ponto de uma circunferência quando esta está sujeita ao rolamento sobre uma recta, uma tórica é uma linha equidistante de uma elipse, etc.

A criação de uma linha pode ser feita através de um comando específico, através da clonagem e modificação de uma linha existente, através da intersecção de superfícies, através da extracção de arestas de um sólido, através da declaração de uma função matemática, etc. Normalmente, qualquer software de modelação 3D disponibiliza as seguintes primitivas geométricas do tipo linha:

- a) **segmento de recta e polilinha**
- b) **circunferência e arco de circunferência**
- c) **elipse e arco de elipse**
- d) **spline** (em geral polinómios do 2º e 3º grau)

Na prática, o procedimento que qualquer software utiliza para tornar visível uma linha ao utilizador, por exemplo no ecrã de um computador, é através de uma aproximação da mesma através de uma linha poligonal. Em função do nível de zoom, essa aproximação procura ser de

tal ordem que a percepção da linha aparente ser suave. Porém, se o software não actualizar automaticamente o procedimento de visualização, essa aproximação pode ser visível, como se ilustra na próxima figura, à esquerda, em que estão representados dois círculos.



Visualização de círculos.

No entanto, há software que guarda determinado tipo de linhas (por exemplo arcos e circunferências) como linhas poligonais, com maior ou menor número de segmentos. Nestes casos é preciso cautela na utilização destas linhas para efeitos de determinação de pontos de construção de uma figura. Por exemplo, a intersecção de uma linha com um polígono de 36 lados não coincide com a intersecção de uma linha com uma circunferência.

2.2 Superfícies

Conceptualmente, uma superfície pode ser considerada como o resultado do movimento de uma linha no espaço.

Exemplos de superfícies são: superfícies poliédricas (faces planas), superfícies cónicas, cilíndricas, esféricas, tóricas, superfícies Nurbs, superfícies topográficas ou de forma livre (sem uma regra matemática definidora), etc.

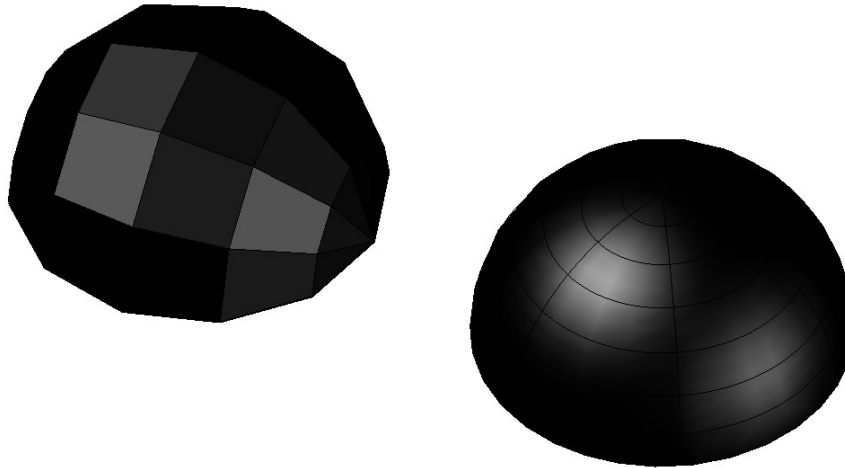
A criação de uma superfície pode ser feita através de um comando específico, através da clonagem e modificação de uma superfície existente, através da combinação de superfícies definidas, através da interacção entre superfícies e sólidos, através da declaração de uma função matemática, etc.

Normalmente, qualquer software de modelação 3D disponibiliza as seguintes primitivas geométricas do tipo superfície:

- a) **face poligonal** (porção de plano delimitada por um polígono).
- b) **superfície facetada (mesh)**
- c) **superfície curvas (nurbs)**

Uma vez que a figura a representar pode ser distinta da aproximação conceptual, na prática, cada uma das opções anteriormente referidas pode, e em geral é, subdividida em várias opções que são disponibilizadas ao utilizador e que traduzem diversas lógicas e princípios de geração de superfícies. Antes de detalharmos este aspecto vamos focar-nos no exemplo da

figura seguinte e perceber como o organigrama dado acima pode ser interpretado. A figura a representar é a superfície de uma semi-esfera. As aproximações conceptuais traduziram-se numa representação facetada (MESH) e numa representação por superfície curva (NURBS). Por fim, a visualização no ecrã traduz-se numa vista 2D de uma representação facetada (embora no caso da NURBS isso não seja perceptível ao utilizador) incluindo uma modelação da luz (sobre este aspecto falaremos mais tarde).



Representação da superfície de uma semi-esfera como MESH (à esquerda) e como NURBS (à direita).

Na prática tanto as superfícies facetadas como as superfícies Nurbs podem ser geradas por procedimentos idênticos, por exemplo através da rotação de uma linha em torno de um eixo (por revolução), através da interpolação linear entre duas linhas curvas e/ou rectas (superfícies regradas), através da interpolação dada uma condição de fronteira (interpolação bi-linear, interpolação bi-quadrática, interpolação bi-cúbica, etc), através da deslocação de um perfil ao longo de uma trajectória (extrusão - “extrude”; ou varrimento - “sweep”), por ajuste a um conjunto de linhas (“loft”) ou pontos dados, por distorção ou quebra de entidades mais simples (“tweak”), etc. A diferença encontra-se na informação que é guardada após o processo de geração e nos procedimentos que são utilizados para tornar essa informação visível (veja-se o exemplo dado acima). Mais uma vez, note-se que, por exemplo, ao invocar um comando que deve gerar uma superfície de revolução, na prática, o resultado desse comando é apenas uma aproximação (mesh ou nurbs). O controlo da aproximação é um importante aspecto uma vez que se pretende que haja uma correspondência visual entre a figura a representar e aquela que de facto a representa.

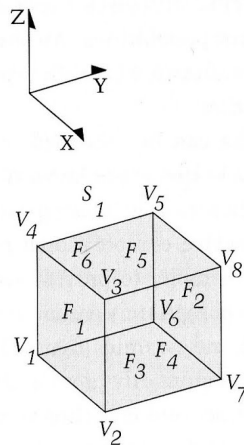
Mais uma vez, é importante notar que há software que guarda internamente as superfícies como malhas facetadas, ainda que aquelas sejam instanciadas por invocação de um comando que designa uma superfície complexa (por exemplo uma esfera pode ser efectivamente representada como um poliedro).

2.3 Sólidos

Conceptualmente, um sólido é um volume finito delimitado por superfícies.

Exemplos de sólidos são: esfera, toro, elipsóide, cubo, pirâmide, anti-prisma. Mas qualquer volume delimitado por superfícies (planas ou não) pode ser um sólido. Porém é importante notar que, do ponto de vista da estruturação da informação, o facto de um volume estar encerrado, não significa obrigatoriamente que a informação guardada defina um sólido.

Um modo de representar os sólidos numa estrutura de informação, num computador, é através da designada “boundary representation” (B-Rep). Veja-se o exemplo seguinte em que se ilustra a representação B-Rep de um cubo.



$V_1 (0,0,0)$ $E_1 (V_1, V_2)$
 $V_2 (1,0,0)$ $E_2 (V_2, V_3)$
 $V_3 (1,0,1)$ $E_3 (V_3, V_4)$
 $V_4 (0,0,1)$ $E_4 (V_4, V_1)$
 $V_5 (0,1,1)$ $E_5 (V_5, V_6)$
 $V_6 (0,1,0)$ $E_6 (V_6, V_7)$
 $V_7 (1,1,0)$ $E_7 (V_7, V_8)$
 $V_8 (1,1,1)$ $E_8 (V_8, V_5)$
 $E_9 (V_1, V_6)$
 $E_{10} (V_2, V_7)$
 $E_{11} (V_3, V_8)$
 $E_{12} (V_4, V_5)$

$F_1 (E_1, E_2, E_3, E_4)$
 $F_2 (E_5, E_6, E_7, E_8)$
 $F_3 (E_1, E_{10}, E_6, E_9)$
 $F_4 (E_2, E_{11}, E_7, E_{10})$
 $F_5 (E_3, E_{12}, E_8, E_{11})$
 $F_6 (E_4, E_9, E_5, E_{12})$

$S_1 (F_1, F_2, F_3, F_4, F_5, F_6)$

Representação B-Rep de um cubo (imagem retirada de Digital Design Media).

a representação B-Rep de um cubo.