Geometria Computacional

Claudio Esperança
Paulo Roma Cavalcanti

Estrutura do Curso

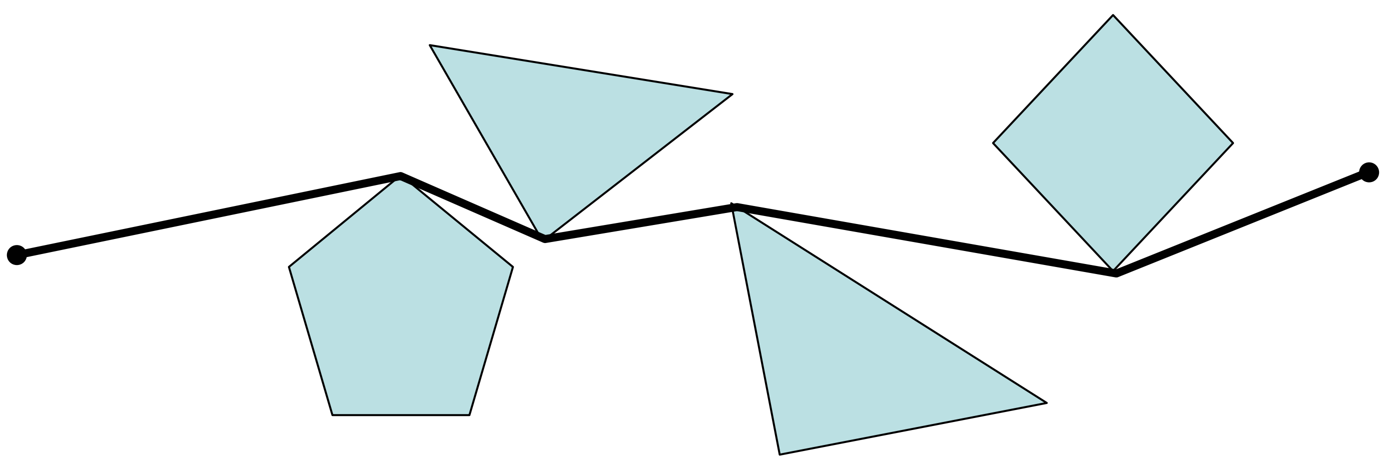
- Aspectos teóricos e práticos
  - Construção e análise de algoritmos e estruturas de dados para a solucionar problemas geométricos
  - Implementação
    - Programas em C++ / STL
    - Uso da biblioteca CGAL (www.cgal.org)

O que é Geometria Computacional?

- Origem do termo
  - (?) Livro “Perceptron”de Marvin Minsky
    - Usado para denotar algoritmos de modelagem de sólidos
- Campo da teoria de algoritmos
  - Entradas são coleções de objetos geométricos
    - Normalmente, objetos “planos” tais como pontos, retas, polígonos, poliedros
  - Saídas são estruturas de dados geométricos
- Surgiu do campo dos algoritmos discretos
  - Até hoje há ênfase em problemas de matemática discreta (conjuntos de objetos, grafos)
  - Componente geométrica pode oferecer subsídios para soluções mais eficientes

Exemplo: Caminho mais curto

- Pode ser reduzido ao problema de encontrar o caminho mais curto em um grafo (grafo de visibilidade)
  - Resolve-se com algoritmos não geométricos
    - Ex.: Algoritmo de Dijkstra
- Algoritmos geométricos podem dar uma solução mais eficiente



Eficiência dos Algoritmos

- Complexidade assintótica de pior caso
  - Problema do Caminho mais curto
    - Algoritmo simples $O\,(n^2 \log n)$
    - Algoritmo complexo $O\,(n \log n)$
- Casos médios
  - Difíceis de se caracterizar
  - Requerem que se estipule uma distribuição "típica"
- Muitas estruturas de dados e algoritmos que se conjectura serem eficientes para casos típicos
  - Quadtrees em geral
  - BSP trees

Limitações da Geometria Computacional

- Dados discretos
  - Aproximações de fenômenos contínuos
    - Funções quantizadas ao invés de funções contínuas (e.g. imagens)
- Objetos geométricos "planos"
  - Aproximações de geometrias "curvas"
- Dimensionalidade
  - Normalmente, 2D e um pouco de 3D
  - Problemas *n*-dimensionais são pouco abordados

Técnicas usadas em GC

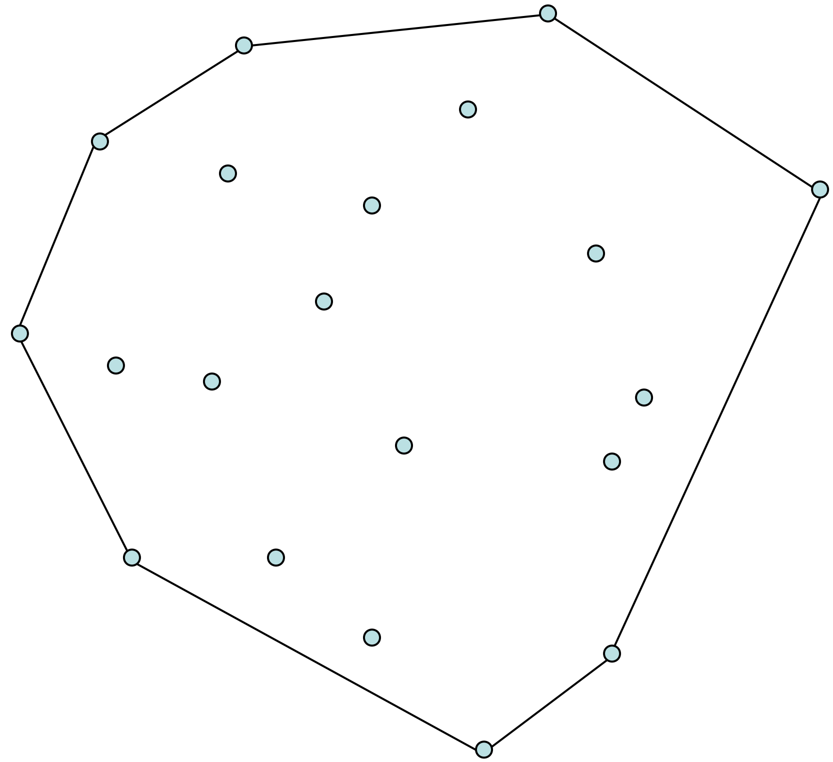
- Técnicas convencionais de desenho de algoritmos
  - Dividir para conquistar
  - Programação dinâmica
- Técnicas próprias para algoritmos geométricos
  - Varredura (*plane sweep*)
  - Construções randomizadas incrementais
  - Transformações duais
  - *Fractional Cascading*

Tendências

- Muitas soluções “ótimas” foram obtidas mas as implementações ...
  - Muito complicadas
  - Muito sensíveis a casos degenerados
  - Problemas de precisão
  - Complexidade inaceitável para problemas pequenos
- Foco em obter soluções práticas
  - Algoritmos simples
    - Freqüentemente randomizados
  - Tratamento de casos degenerados
  - Engenharia de software

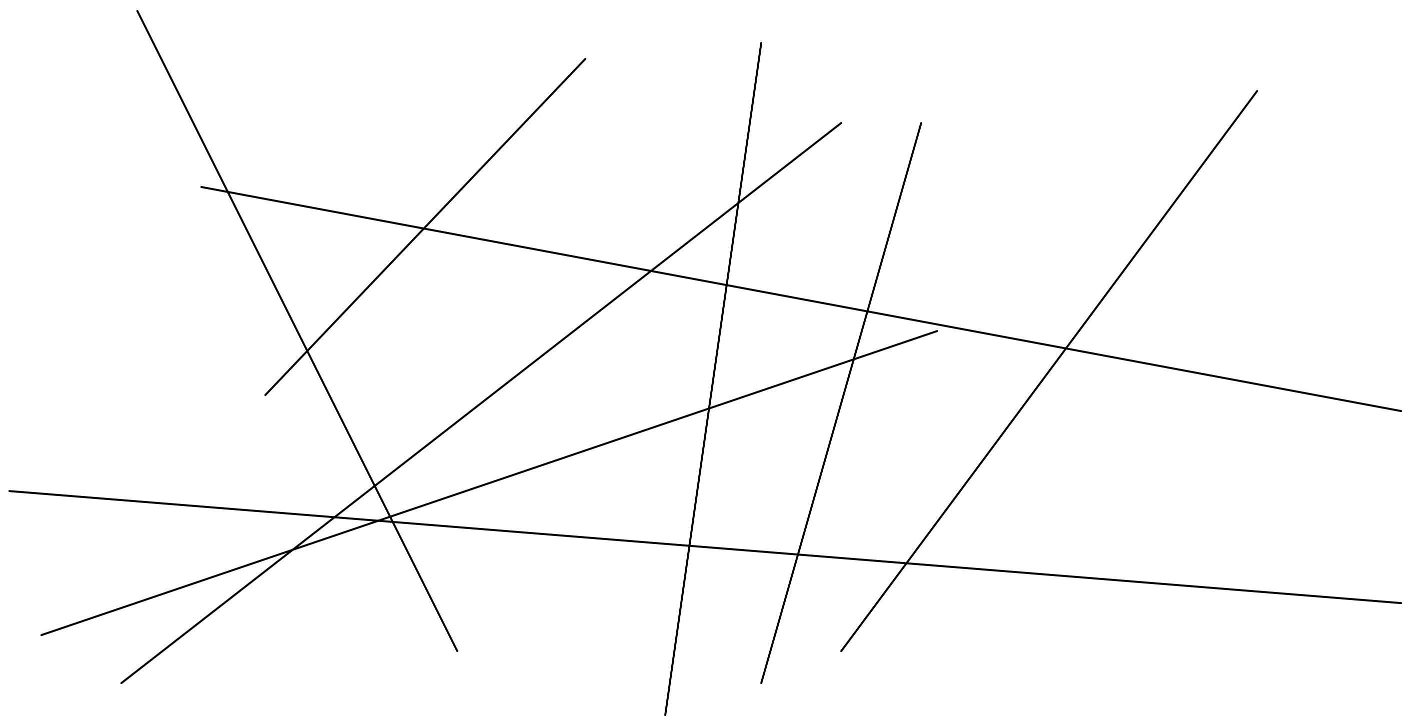
Problemas – Fecho Convexo

- Menor polígono (poliedro) convexo que contém uma coleção de objetos (pontos)



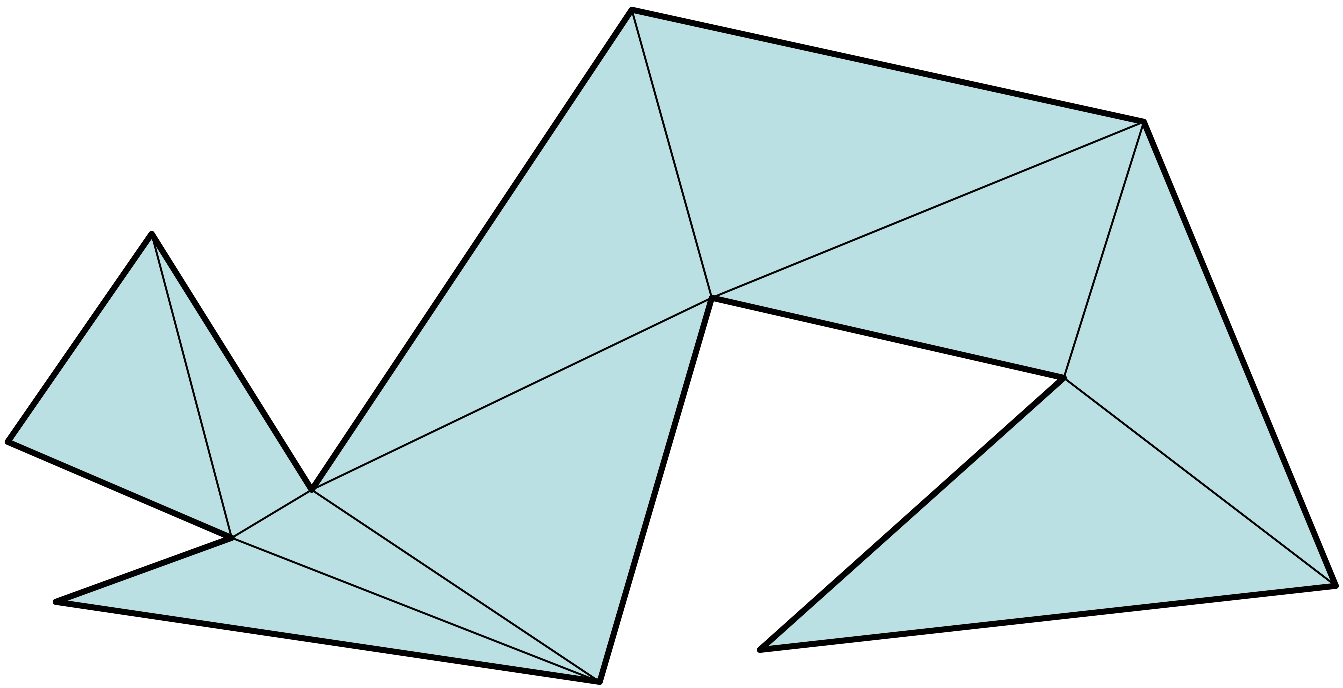
Problemas - Interseções

- Determinar interseções entre coleções de objetos



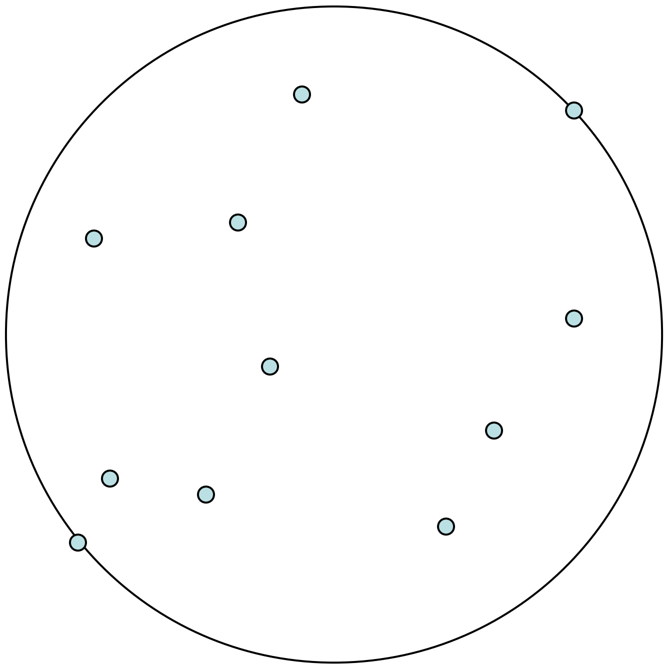
Problemas – Triangulações

- Dividir domínios complexos em coleções de objetos simples (*simplexes*)



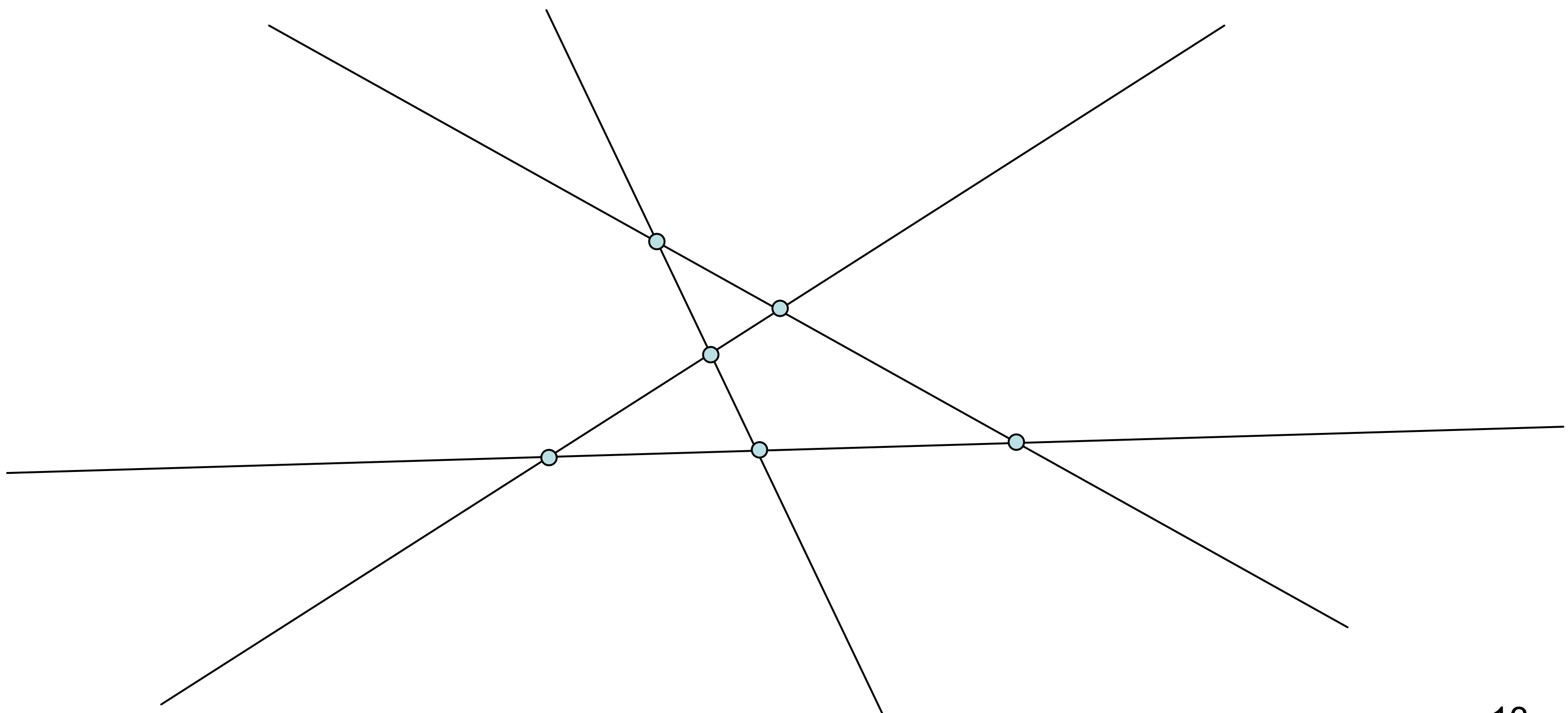
Problemas – Prog. Linear em 2d e 3d

- Problemas de otimização
  - Ex.: menor disco que contém um conjunto de pontos



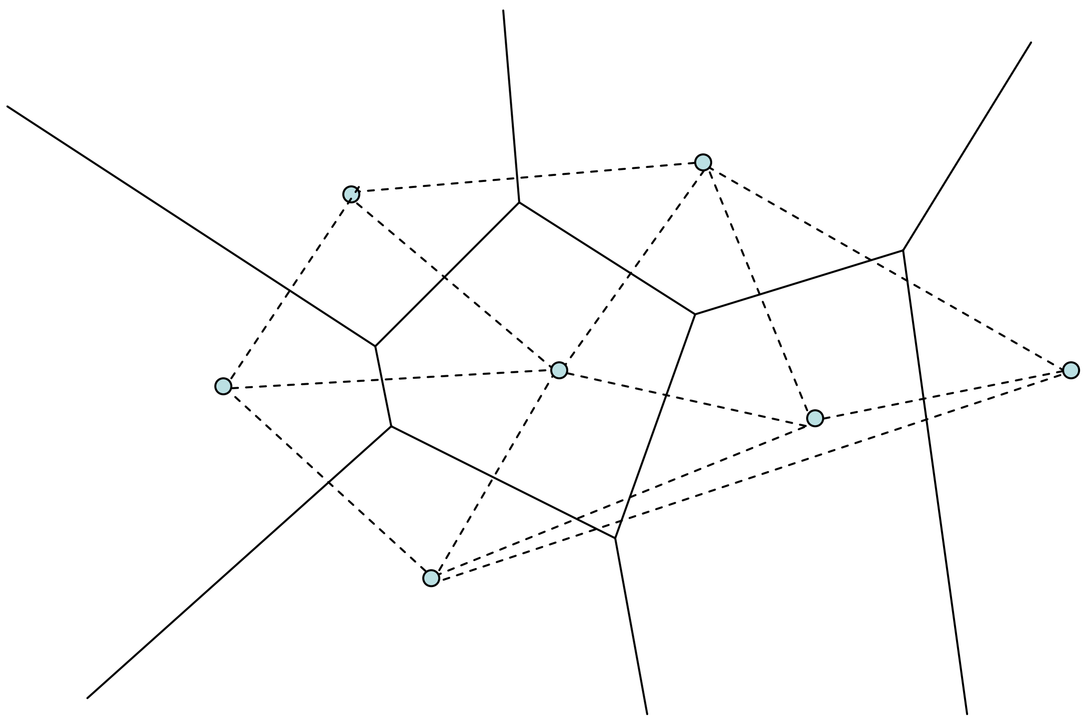
Problemas – Arranjos de Retas

- Dada uma coleção de retas, é o grafo formado pelos pontos de interseção e segmentos de reta entre eles
  - Problemas sobre pontos podem ser transformados em problemas sobre retas (dualidade)



Problemas – Diagramas de Voronoi e Triangulações de Delaunay

- Dada uma coleção de pontos S
  - Diagrama de Voronoi delimita as regiões de pontos mais próximos
  - Triangulação de Delaunay é o dual do D. V.



Problemas – Busca Geométrica

- Algoritmos e estruturas de dados para responder consultas geométricas. Ex.:
  - Todos os objetos que interceptam uma região
    - Polígono
    - Disco
  - Par de pontos mais próximos
  - Vizinho mais próximo
  - Caminho mais curto

Geometria Afim

- Composta dos elementos básicos
  - escalares
  - pontos - denotam posição
  - vetores - denotam deslocamento (direção e magnitude)
- Operações
  - escalar · vetor = vetor
  - vetor + vetor ou vetor – vetor = vetor
  - ponto – ponto = vetor
  - ponto + vetor ou ponto – vetor = ponto

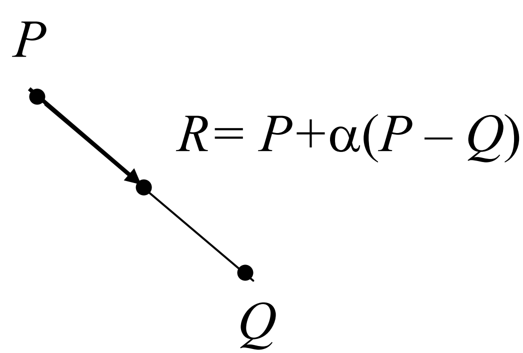
Combinações Afim

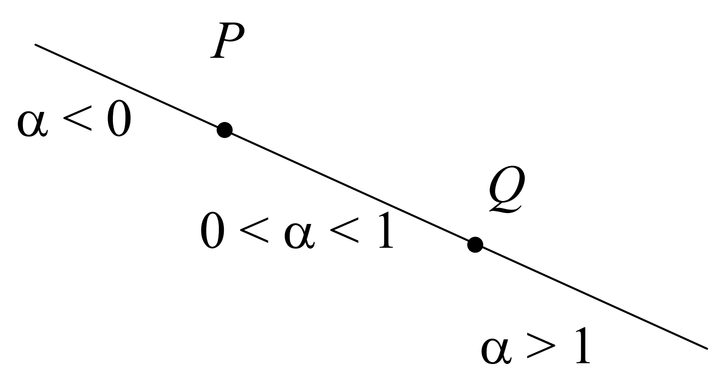
- Maneira especial de combinar pontos

$$\alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_n P_n$$

$$\text{onde } \sum_{i=1}^{n} \alpha_i = 1$$

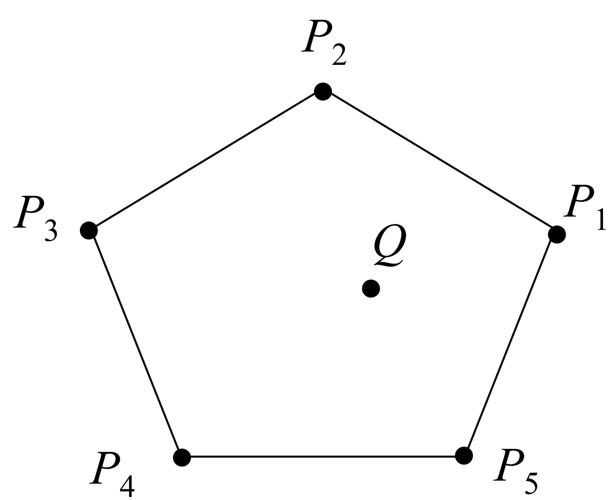
- Para 2 pontos P e Q poderíamos ter uma combinação afim $R = (1-\alpha)P + \alpha Q = P + \alpha(P - Q)$





Combinações Convexas

- Combinações afim onde se garante que todos os coeficientes α_i são positivos (ou zero)
- Usa-se esse nome porque qualquer ponto que é uma combinação convexa de n outros pontos pertence à <u>envoltória convexa</u> desses pontos



Geometria Euclidiana

- Extensão da geometria afim pela adição de um operador chamado *produto interno*
- Produto interno é um operador que mapeia um par de vetores em um escalar. Tem as seguintes propriedades:
  - Positividade : $(u,u) \geq 0$ e $(u,u) = 0$ sse $u=0$
  - Simetria: $(u,v) = (v,u)$
  - Bilinearidade: $(u,v+w) = (u,v) + (u,w)$ e

$$(u,\alpha v) = \alpha(u,v)$$

Geometria Euclidiana

- Normalmente usamos o produto escalar como operador de produto interno:

$$\vec{u} \cdot \vec{v} = \sum_{i=1}^{d} u_i v_i$$

- Comprimento de um vetor é definido como:

$$\left|\vec{v}\right| = \sqrt{\vec{v} \cdot \vec{v}}$$

- Vetor unitário (normalizado):

$$\hat{v} = \frac{\vec{v}}{\left|\vec{v}\right|}$$

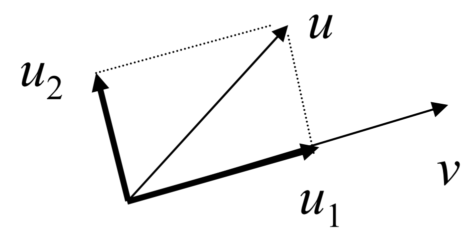
Geometria Euclidiana

- Distância entre dois pontos P e $Q = | P - Q |$
- O ângulo entre dois vetores pode ser determinado por

$$\hat{a}ngulo\,(\vec{u},\vec{v}) = \cos^{-1}\left(\frac{\vec{u}\cdot\vec{v}}{|\vec{u}||\vec{v}|}\right) = \cos^{-1}(\hat{u}\cdot\hat{v})$$

- Projeção ortogonal: dados dois vetores u e v, deseja-se decompor u na soma de dois vetores u_1 e u_2 tais que u_1 é paralelo a v e u_2 é perpendicular a v

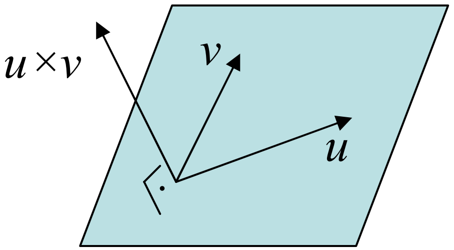
$$\vec{u}_1 = \frac{\vec{u}\cdot\vec{v}}{\vec{v}\cdot\vec{v}}\vec{v} \qquad \vec{u}_2 = \vec{u} - \vec{u}_1$$



Produto Vetorial (3D)

- Permite achar um vetor perpendicular a outros dois dados
- Útil na construção de sistemas de coordenadas

$$\vec{u} \times \vec{v} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{vmatrix}$$



- Propriedades (assume-se u, v linearmente independentes):
  - Antisimetria: $u \times v = - v \times u$
  - Bilinearidade: $u \times (\alpha v) = \alpha\,(u \times v)$ e $u \times (v + w) = (u \times v) + (u \times w)$
  - $u \times v$ é perpendicular tanto a u quanto a v
  - O comprimento de $u \times v$ é igual a área do paralelogramo definido por u e v, isto é, $|\, u \times v \,| = |\, u \,|\,|\, v \,|\, sin\ \theta$

Sistemas de coordenadas

- Um sistema de coordenadas para $\mathbf{R}^n$ é definido por um ponto (origem) e n vetores
- Ex. Seja um sistema de coordenadas para $\mathbf{R}^2$ definido pelo ponto O e os vetores X e Y. Então,
  - Um ponto P é dado por coordenadas x_P e y_P tais que

$$P = x_P.X + y_P.Y + O$$

  - Um vetor V é dado por coordenadas x_V e y_V tais que

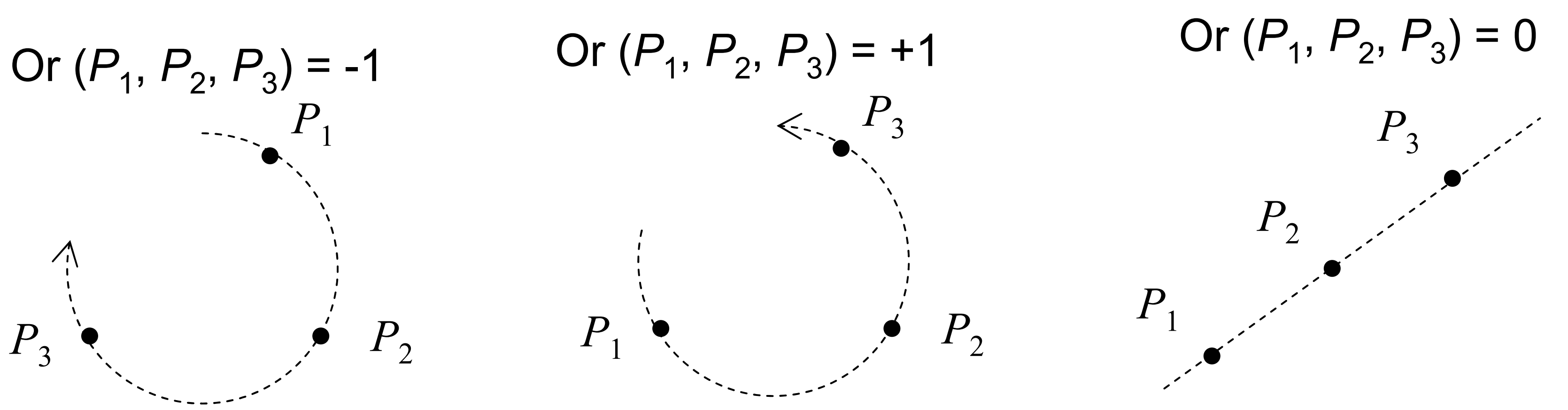
$$V = x_V.X + y_V.Y$$

Coordenadas Homogêneas

- Coordenadas homogêneas permitem unificar o tratamento de pontos e vetores
- Problema é levado para uma dimensão superior:
  - Coordenada extra w= 0 para vetores e =1 p/ pontos
  - O significado da coordenada extra é levar ou não em consideração a origem do sistema
- Coordenadas homogêneas têm diversas propriedades algébricas interessantes
  - Ex. Subtração de dois pontos naturalmente resulta em um vetor

Orientação

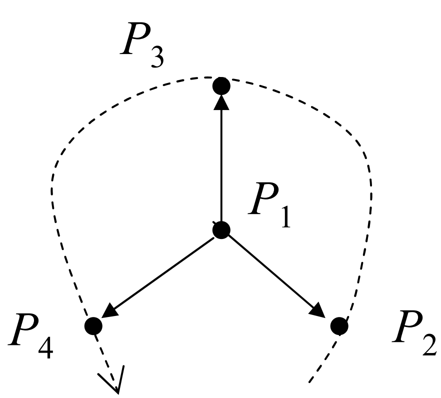
- Orientação de 2 pontos em 1D
  - $P_1 < P_2$, $P_1 = P_2$ ou $P_1 > P_2$
- Orientação de 3 pontos em 2D
  - O percurso P_1 , P_2 , P_3 é feito no sentido dos ponteiros do relógio, no sentido contrário ou são colineares



Orientação

- Orientação de 4 pontos em 3D
  - O percurso P_1 , P_2 , P_3 , P_4 define um parafuso segundo a regra da mão direita, mão esquerda ou são coplanares

Or (P_1, P_2, P_3, P_4) = +1



- *O conceito pode ser estendido a qualquer número de dimensões ...*

Computando Orientação

- A orientação de n+1 pontos em um espaço n-dimensional é dado pelo sinal do determinante da matriz cujas colunas são as coordenadas homogêneas dos pontos *com o 1 vindo primeiro*

$$\mathrm{Or}_2(P_1, P_2, P_3) = \mathrm{sign}\left(\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}\right) \qquad \mathrm{Or}_3(P_1, P_2, P_3, P_4) = \mathrm{sign}\left(\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix}\right)$$