

Geometria Computacional

Retas: Algoritmos

Prof. Edson Alves

2018

Faculdade UnB Gama

1. Classificação de retas

Classificação de retas

Retas paralelas, concorrentes e coincidentes

- Em relação às possíveis interseções entre duas retas, há três cenários possíveis:
 1. nenhum ponto em comum (retas paralelas)
 2. um único ponto em comum (retas concorrentes)
 3. todos os pontos em comum (retas coincidentes)
- O coeficiente angular é a chave para tal classificação: retas com coeficientes angulares distintos são concorrentes
- Caso duas retas tenham coeficientes angulares iguais, é necessário verificar também o coeficiente linear: se iguais, as retas são coincidentes
- Retas com coeficientes angulares iguais e coeficientes lineares distintos são paralelas
- A implementação destas verificações é trivial na representação baseada na equação reduzida, sendo necessário apenas o cuidado no trato do caso das retas verticais

Exemplo de implementação de classificação de retas em C++

```
1 // Definição da função equals()
2
3 template<typename T>
4 struct Line {
5     // Membros e construtores (equação reduzida)
6
7     bool operator==(const Line& r) const // Verdadeiro se coincidentes
8     {
9         if (vertical != r.vertical || !equals(m, r.m)) return false;
10
11         return equals(b, r.b);
12     }
13
14     bool parallel(const Line& r) const // Verdadeiro se paralelas
15     {
16         if (vertical && r.vertical) return b != r.b;
17         if (vertical || r.vertical) return false;
18
19         return equals(m, r.m) && !equals(b, r.b);
20     }
21 };
```

Exemplo de implementação de classificação de retas em C++

```
1 // Definição da função equals()
2
3 template<typename T>
4 struct Line {
5     // Membros e construtores (equação geral)
6
7     bool operator==(const Line& r) const
8     {
9         auto k = a ? a : b;
10        auto s = r.a ? r.a : r.b;
11
12        return equals(a*s, r.a*k) && equals(b*s, r.b*k)
13            && equals(c*s, r.c*k);
14    }
15
16    bool parallel(const Line& r) const
17    {
18        auto det = a*r.b - b*r.a;
19        return det == 0 and !(*this == r);
20    }
21 };
```

Retas perpendiculares

- Duas retas são perpendiculares se o produto de seus coeficientes angulares for igual a -1
- Outra maneira de checar se duas retas são perpendiculares é escolher dois pontos pertencentes a cada reta e montar dois vetores \vec{u} e \vec{v}
- Estes pontos podem ser escolhidos de forma eficiente, fazendo $x = 0$ e $y = 0$ (caso a reta não passe na origem)
- Se o produto interno dos dois vetores for igual a zero, as retas são perpendiculares
- Importante notar, porém, é que os coeficientes a e b da equação geral de uma reta formam um vetor $\vec{v} = (a, b)$ perpendicular à reta
- Tais vetores, denominados normais, podem ser utilizados na comparação descrita anteriormente

Exemplo de verificação de retas perpendiculares em C++

```
1 // Definição da função equals()
2
3 template<typename T>
4 struct Line
5 {
6     // Membros e construtores (equação reduzida)
7
8     bool orthogonal(const Line& r) const // Verdadeiro se perpendiculares
9     {
10         if (vertical && r.vertical)
11             return false;
12
13         if ((vertical && equals(r.m, 0)) || (equals(m, 0) && r.vertical))
14             return true;
15
16         if (vertical || r.vertical)
17             return false;
18
19         return equals(m * r.m, -1.0);
20     }
21 };
```


Exemplo de verificação de retas perpendiculares em C++

```
1 // Definição da função equals()
2
3 template<typename T>
4 struct Line
5 {
6     // Membros e construtores (equação geral)
7
8     bool orthogonal(const Line& r) const // Verdadeiro se perpendiculares
9     {
10         return equals(a * r.a + b * r.b, 0);
11     }
12 };
```

1. **HALIM**, Felix; **HALIM**, Steve. *Competitive Programming 3*, 2010.
2. **LAAKSONEN**, Antti. *Competitive Programmer's Handbook*, 2018.
3. **De BERG**, Mark; **CHEONG**, Otfried. *Computational Geometry: Algorithms and Applications*, 2008.
4. David E. Joyce. *Euclid's Elements*. Acesso em 15/02/2019¹
5. Wikipédia. *Geometria Euclidiana*. Acesso em 15/02/2019².

¹<https://mathcs.clarku.edu/~djoyce/elements/bookI/def11.html>

²https://pt.wikipedia.org/wiki/Geometria_euclidiana