Voorbereiding Programmeerwedstrijden

najaar 2021

https://liacs.leidenuniv.nl/~vlietrvan1/vbpw/

Rudy van Vliet
kamer 140 Snellius, tel. 071-527 2876
rvvliet(at)liacs(dot)nl

college 6, 18 oktober 2021

Geometry

Computational Geometry

13.4. Faster Than a Speeding Bullet

- given:
  - obstacles do not overlap
  - start and target positions lie outside of obstacles
- algorithm. . .

Faster Than a Speeding Bullet

$$\text{travel} = \text{distance}(s, t) + \sum_{\substack{\text{intersecting} \\ \text{circles}}} (\text{arclength} - \text{linesegmentlength})$$

Representation Point

```
typedef double point[2];
const int X = 0;
const int Y = 1;

int main ()
{ point p;

  p[X] = ...;
  p[Y] = ...;
}
```

Representation Circle

```
typedef struct
{ point center;    // center of circle
  double r;        // radius of circle
} circle;

int main ()
{ circle c;

  c.center[X] = ...;
  c.center[Y] = ...;
  c.r = ...;
}
```

Representation Line

Representation Line

```
typedef struct
{ double a,   // a, b and c are coefficients
         b    // in equation ax + by + c = 0,
         c;   // describing the line.
} line;

int main ()
{ line l;

  l.a = ...;
  l.b = ...;
  l.c = ...;
}
```

b (or a) is normalized to 1

Closest Point On Line

```
void closest_point (point p_in, line l, point p_c)
{ line perp;      // perpendicular to line l through point p

  if (fabs(l.b) <= EPSILON)  // vertical line
  { p_c[X] = -l.c;
    p_c[Y] = p_in[Y];
    return;
  }

  if (fabs(l.a) <= EPSILON)  // horizontal line
    ...   // analogous



  // Otherwise ...

}  // closest_point
```

Closest Point On Line

- find perpendicular line (how?)
- find intersection point (how?)

Perpendicular Line

$y = mx + k1$
is perpendicular to
$y = -(1/m)x + k2$

```
void point_and_slope_to_line (point p, double m, line *l)
{ ...
}
```

Point and Slope To Line

```
void point_and_slope_to_line (point p, double m, line *l)
{
  l->a = -m;
  l->b = 1;
  l->c = - ((l->a)*p[X] + (l->b)*p[Y]);

}
```

Closest Point On Line

```
void closest_point (point p_in, line l, point p_c)
{ line perp;      // perpendicular to line l through point p

  if (fabs(l.b) <= EPSILON)  // vertical line
  { p_c[X] = -l.c;
    p_c[Y] = p_in[Y];
    return;
  }

  if (fabs(l.a) <= EPSILON)  // horizontal line
    ...   // analogous

  point_and_slope_to_line (p_in, 1/(l.a), &perp);
  intersection_point (l, perp, p_c);

}  // closest_point
```

Intersection Point

(general case)

$$
\begin{aligned}
a_1 x + b_1 y + c_1 &= 0 \\
a_2 x + b_2 y + c_2 &= 0
\end{aligned}
$$

Intersection Point

(general case)

$$
\begin{aligned}
a_1 b_2 x + b_1 b_2 y + c_1 b_2 &= 0 \\
b_1 a_2 x + b_1 b_2 y + b_1 c_2 &= 0
\end{aligned}
$$

Intersection Point

(general case)

$$
\begin{aligned}
a_1 b_2 x + b_1 b_2 y + c_1 b_2 &= 0 \\
b_1 a_2 x + b_1 b_2 y + b_1 c_2 &= 0
\end{aligned}
$$

$$(a_1 b_2 - b_1 a_2)x + c_1 b_2 - b_1 c_2 = 0$$

$$x = \frac{b_1 c_2 - c_1 b_2}{a_1 b_2 - b_1 a_2}$$

Intersection Point

(general case)

$$
\begin{aligned}
a_1 b_2 x + b_1 b_2 y + c_1 b_2 &= 0 \\
b_1 a_2 x + b_1 b_2 y + b_1 c_2 &= 0
\end{aligned}
$$

$$
(a_1 b_2 - b_1 a_2) x + c_1 b_2 - b_1 c_2 = 0
$$

$$
x = \frac{b_1 c_2 - c_1 b_2}{a_1 b_2 - b_1 a_2}
$$

$$
y = -\frac{a_1 c_2 - c_1 a_2}{a_1 b_2 - b_1 a_2}
$$

unless lines are parallel / equal...

Point In Box

```
bool point_in_box (point p, point b1, point b2)
{
  return ( (p[X] >= min (b1[X], b2[X]) - EPSILON) &&
           (p[X] <= max (b1[X], b2[X]) + EPSILON) &&
           (p[Y] >= min (b1[Y], b2[Y]) - EPSILON) &&
           (p[Y] <= max (b1[Y], b2[Y]) + EPSILON) );
}
```

Triangles and Trigonometry

- $\sin(\alpha)$, $\cos(\alpha)$, $\tan(\alpha)$
- $(\sin(\alpha))^2 + (\cos(\alpha))^2 = 1$
- degrees vs. radians: 360 degrees $\approx 2\pi$
- $\cos(\alpha) = \sin(\alpha + (\pi/2))$
- $\arcsin(x)$, $\arccos(x)$, $\arctan(x)$

13.2.3. Solving Triangles

- Pythagoras: $a^2 = b^2 + c^2$

- In general:

$$a^2 = b^2 + c^2 - 2bc\,\cos(\alpha)$$

- In general:

$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$$

13.2.3. Solving Triangles

- In general:

$$a^2 = b^2 + c^2 - 2bc\cos(\alpha)$$

- In general:

$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$$

- given two angles and one side...

- given two sides and one angle...

13.2.3. Solving Triangles

area $A(T)$ of triangle $T\ldots$

13.2.3. Solving Triangles

area $A(T)$ of triangle T

- $A(T) = (1/2)ab$, for altitude and base
-
$$2A(T) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} = a_x b_y - a_y b_x + a_y c_x - a_x c_y + b_x c_y - c_x b_y$$

  (in absolute value)...

  Here, $a = (a_x, a_y)$, $b = (b_x, b_y)$, $c = (c_x, c_y)$ are vertices (not lengths of edges)

To Which Side of a Line

point c is to right of $a \rightarrow b$, if

$$a_x b_y - a_y b_x + a_y c_x - a_x c_y + b_x c_y - c_x b_y < 0$$

13.3. Circles

- line tangent to circle

- intersection points of two circles

Area of Convex Polygon

Area of Convex Polygon

triangulation from arbitrary vertex

Van Gogh's Algorithm

for general polygon

13.6.3. The Knights of the Round Table?

13.6.7. Is This Integration?

13.6.8. How Big Is It?