Tuesday, August 29, 2006
3:33 PM

Paper review and Point

Tuesday, August 29, 2006
3:33 PM

Paper Review

Notation: $\vec{x_i} \in \mathbb{R}^3$ means $(x, y, z)_i$

there's n of them

$$\|\vec{x}\| = \text{magnitude of } \vec{x} = \sqrt{\vec{x}\cdot\vec{x}} = \sqrt{x^2 + y^2 + z^2}$$

$\downarrow$

L^3-norm (?)

$X = \underbrace{\{\vec{x}_1, \ldots, \vec{x}_n\}}_{\text{need a } \underline{\underline{\text{list}}} \text{ of points}}$: our sampled data, our input

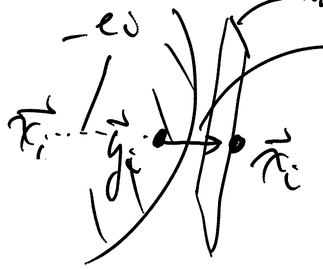
Tuesday, August 29, 2006
3:33 PM

M: unknown surface — this is the real surface

— we just assume some small measurement error

notion of ρ-density:

what we're after: a small piece of surface at $\vec{x}_i$



$e_i \Rightarrow \vec{x}_i = \vec{y}_i + \vec{e}_i$

(really this should be $y_i \pm e_i$)

$\vec{y}_i$ is on the surface

data is ρ-dense if $\|e_i\| \leq \delta \; \forall i$

Step 1 Compute oriented tangent plane for each point $\vec{x}_i$

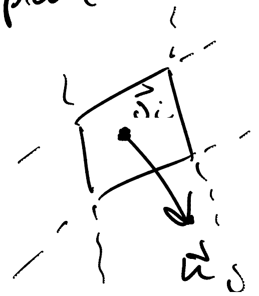
$$Tp(\vec{x}_i) = \{\vec{o}_i, \vec{n}_i\}$$

$\underbrace{\text{local "origin"}}_{\text{center of the plane}}$ normal (orientation of plane)

$A\vec{o}_{i_x} + B\vec{o}_{i_y} + C\vec{o}_{i_z} + D = 0$

$D = -(A\vec{o}_{i_x} + B\vec{o}_{i_y} + C\vec{o}_{i_z})$

$= -(\vec{n}_i \cdot \vec{o}_i)$



plane of ∞ extent

$Ax + By + Cz + D = 0$

where $(A, B, C) = \vec{n}$

$D = -(\vec{n}_i \cdot \vec{o}_i)$

Tuesday, August 29, 2006
3:33 PM

Signed distance of arbitrary point $\vec{p}_i \in \mathbb{R}^3$ to $Tp(\vec{x}_i)$

$$dist_i(\vec{p}) = \underbrace{(\vec{p} - \vec{o}_i)} \cdot \vec{n}_i$$

$$= \left\{ \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \underset{\uparrow}{-} \begin{pmatrix} o_{ix} \\ o_{iy} \\ o_{iz} \end{pmatrix} \right\} \cdot \begin{pmatrix} n_{ix} \\ n_{iy} \\ n_{iz} \end{pmatrix}$$

in C++, we'd like to use this

almost verbatim : operator-(· ·)

Tuesday, August 29, 2006
3:33 PM

How to calculate $\vec{o}_i$ & $\vec{n}_i$?

find k-nearest neighbors of $\vec{x}_0$, $\boxed{\mathrm{Nbhd}(\vec{x}_0)}$

e.g. $\mathrm{Nbhd}(\vec{x}_i) = \{ \vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4, \vec{x}_5 \}$, $K=5$ say

(what do they actually use for k?)

then, calculate centroid of $\mathrm{Nbhd}(\vec{x}_i)$ to get $\vec{o}_i$

$$\vec{o}_i = \frac{1}{k} \sum_{i=1}^{k} \vec{x}_i = \frac{1}{k} \begin{pmatrix} x_{1x} + x_{2x} + x_{3x} + x_{4x} + x_{5x} \\ x_{1y} + x_{2y} + x_{3y} + x_{4y} + x_{5y} \\ x_{1z} + x_{2z} + x_{3z} + x_{4z} + x_{5z} \end{pmatrix}$$

have to write this

for scalability: need a kd-tree

Tuesday, August 29, 2006
3:33 PM

- here's what the future code may look like:

```
                Point  o(0.Ø, Ø.Ø, Ø.Ø);
                for(k=Ø; k < Nbhd.size(); k++) {
replace    {            o[Ø] += (Nbhd[k])[Ø];
with loop  {            o[1] += (Nbhd[k])[1];
           {            o[2] += (Nbhd[k])[2];
                }
                o[Ø] = o[Ø]/(float)k;
                o[1] = o[1]/(float)k;
                o[2] = o[2]/(float)k;
```

Tuesday, August 29, 2006
3:33 PM

what's Nbhd ?

vector< Point* > Nbhd;

↳ C++ Standard Template Library

STL

vector

#include <vector>

a list, basically.

Here, in my example it's a list of pointers to Point

So what's the point? Let's start with a 2D point. (x,y)

old C-way:

```
typedef struct {
        double x, y;   -OR-   double *val   -OR-
                              double val[2]
} Point;
```

---

whatever you choose, it'd be nice to use Point like this:

```
Point p;  Point q;
p[0] = 42.0;
q = p+p;
```

operator[] (mutator; not accessor)

Tuesday, August 29, 2006
3:33 PM

```
class Point {
    private:
    vector<double>   point;   // internal data member
    public:
    double& operator[](int k) { return (point[k]); }
}
```

when you have

```
Point p;
p[0] = 42.0
```

what happens:

```
p.operator[](0) = 42.0

p.point[k]& = 42.0
```

pass by reference

Tuesday, August 29, 2006
3:33 PM

what else do we want Point to do?

C++'s "Big Three":

constructor
destructor
copy assignment (operator =)

```
Point(double x=0.0, double y = 0.0)
{
      point.push_back(x);
      point.push_back(y);
}
```

Tuesday, August 29, 2006
3:33 PM

Point operator = (const Point& rhs); } point.h

how does this get called?

```
        Point a, b;
        a = b
        ____________
        a.operator=(b)   : assumption is that a exists
```

---

in ← cpp file

```
    Point Point::operator= (const Point& rhs)
    {
      if (this != &rhs) {   // aliasing test; testing for a = a
            point.clear();
            for (int i = 0; i < rhs.size(); ++i)
                 point.push_back(rhs.point[i]);
      }
      return *this;
    }
```

point = rhs.point;

Tuesday, August 29, 2006
3:33 PM

what else? I/O

I'd like to say this:

```
Point p, q;
std::cout << p;    // print p to stdout
std::cin >> q;     // read in data to q
```

read as:
operator<<(cout, p)
global scope → not a member of Point

in point.h: #include <iostream>

```
friend ostream& operator<<(ostream& s, const Point& rhs);
friend istream& operator>>(istream& s; Point& rhs);
```

```
std::cout << "hello";  ⟶  operator<<(cout, char*)
std::cout << "p = " << p) << std::endl;
            cout.operator<<(cout, p)
```

Tuesday, August 29, 2006
3:33 PM

in point.cpp:

```
ostream& operator<<(ostream& s, const Point& rhs)
{
    s << rhs.point[0] << "␣";
   s<< rhs.point[1] << "␣";
    s<< std::endl;
    return s;
}
```

add whatever
int you
Petty pr
want

<2 3>

[2 3)

Tuesday, August 29, 2006
3:33 PM

```
istream &   operator>> (istream & s, Point& rhs)
 {                double f;  int i = 0;  char c, n;

        do {
            s>>f;  rhs[i] = f ;  i++;

        } while (s.get(c) && (c != '\n') &&
                                ((n = s.peek(1) != '\n'));
      return s;

  }
```