

Polygon Mesh Processing Library

동아대학교 컴퓨터시공학부

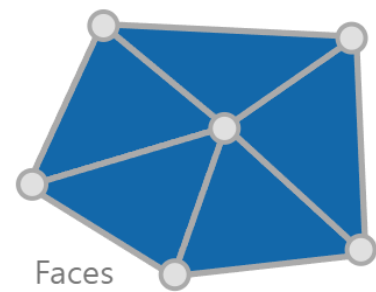
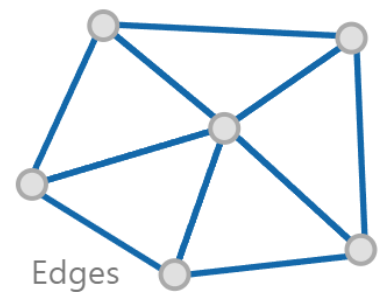
박영진

- 다각형 표면 메시를 처리하고 시각화하기 위한 C++ 오픈 소스 라이브러리
 - An efficient and easy-to-use mesh data structure
 - Standard algorithms such as decimation, remeshing, subdivision, or smoothing
 - Ready-to-use visualization tools
 - Seamless cross-compilation to JavaScript
 - <https://www.pmp-library.org/demos/mpview.html>

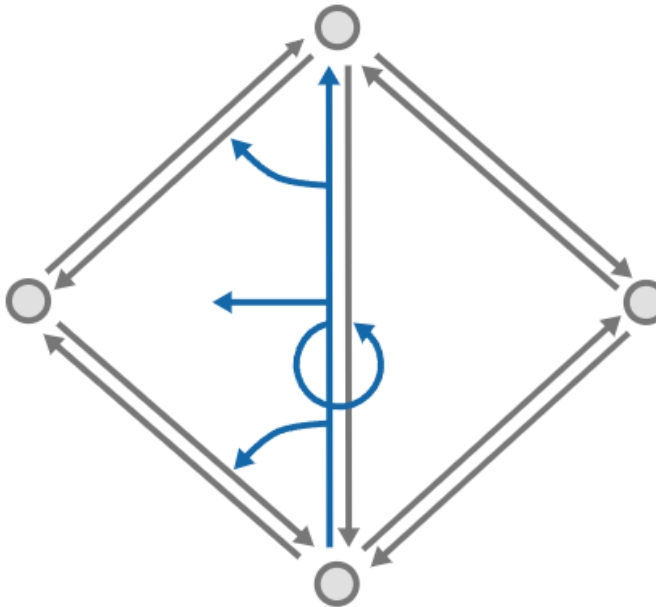
```
#include <pmp/surface_mesh.h>
int main()
{
    pmp::SurfaceMesh mesh;
    pmp::read(mesh,"input.obj");
    // .. do awesome things with your mesh
    pmp::write(mesh,"output.obj");
}
```

Polygonal Mesh

- 다각형 메쉬(Polygonal Mesh)는 컴퓨터 그래픽스와 3D 모델링에서 3D 표면을 표현하기 위한 기본 데이터 구조
- 메쉬는 꼭짓점, 가장자리 및 면으로 구성
- **Vertex:** 개체의 모양을 정의하는 개별적인 점
 - 각 꼭짓점에는 일반적으로 3D 좌표(x,y,z)로 표현
 - 색 또는 질감(texture)과 같은 추가적인 속성은 각 꼭짓점에 대해 저장
- **Edge:** 두 꼭짓점을 연결하는 선분 각 면의 경계를 정의
 - 모든 모서리는 정확히 두 면에 연결
- **Face:** 물체의 표면을 구성하는 다각형
 - 각 면에는 세 개 이상의 정점이 존재하며, 일반적으로 삼각형과 사각형(꼭짓점 4개)은 가장 자주 사용



- 모든 3D 데이터를 저장하고 조작하기 위한 핵심 클래스
- Halfedge 구조를 기반으로 연결 정보를 저장
 - 각 vertex은 out-going halfedge를 가지고 있음
 - 각 face은 인접한(보유하고 있는) halfedge를 가지고 있음
 - 각 halfedge는 인접한 면, target vertex, 이전 halfedge와 다음 halfedge를 가지고 있음



Halfedge connectivity.

```
// add 4 vertices
const auto v0 = mesh.add_vertex(pmp::Point(0, 0, 0));
const auto v1 = mesh.add_vertex(pmp::Point(1, 0, 0));
const auto v2 = mesh.add_vertex(pmp::Point(0, 1, 0));
const auto v3 = mesh.add_vertex(pmp::Point(0, 0, 1));
// add 4 triangular faces
mesh.add_triangle(v0, v1, v3);
mesh.add_triangle(v1, v2, v3);
mesh.add_triangle(v2, v0, v3);
mesh.add_triangle(v0, v2, v1);
std::cout << "vertices: " << mesh.n_vertices() << std::endl;
std::cout << "edges: " << mesh.n_edges() << std::endl;
std::cout << "faces: " << mesh.n_faces() << std::endl;
```

```
auto n = mesh.get_vertex_property<pmp::Normal>("v:normal");

for (auto f : mesh.faces()) {
    glBegin(GL_TRIANGLES);
    for (auto v : mesh.vertices(f)){
        auto p = mesh.position(v);
        glNormal3dv(n[v].data());
        glVertex3dv(p.data());
    }
    glEnd();
}
```

[과제] 정육면체 만들기

- InitMyMesh() 활용하여 크기가 2인 정육면체 만들어보기
- `pmp::write(mesh, "output.obj");` 활용하여 obj 파일로 출력한 후 meshlab 등 다양한 외부 3D 모델링 프로그램에서 열어보기
- 마감 : 5/10(금) 23:59
- 제출물 : PDF 보고서

Any Questions?

