Computational Graphics: Lecture 9

Alberto Paoluzzi

Thu, Mar 20, 2014

Outline

Examples of Pyplasm primitives

Use of MKPOL constructor

MKPOL: stands for MaKe POLyhedron

The definition of a single convex cell (with 5 vertices)

```
verts = [[0,0],[4,0],[4,4],[2,6],[0,4]]
cells = [[1,2,3,4,5]]
pols = None
muro = MKPOL([verts, cells, pols])
VIEW(muro)
VIEW(SKEL_1(muro))
```

Use of MKPOL constructor

MKPOL: stands for MaKe POLyhedron

The definition of a single convex cell (with 5 vertices)

```
verts = [[0,0],[4,0],[4,4],[2,6],[0,4]]
cells = [[1,2,3,4,5]]
pols = None
muro = MKPOL([verts, cells, pols])
VIEW(muro)
VIEW(SKEL_1(muro))
```

or (the ordering of points is immaterial)

```
verts = [[0,0],[4,0],[4,4],[2,6],[0,4]]
muro = JOIN(AA(MK)(verts))
VIEW(muro)
VIEW(SKELETON(1)(muro))
```

Use of MKPOL constructor

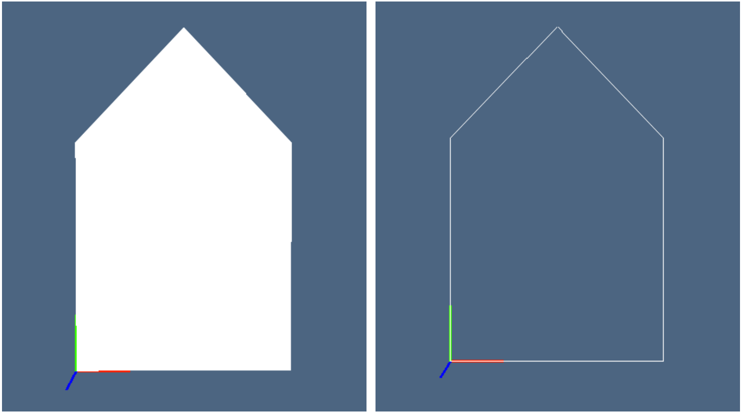


Figure : (a) `hpc` complex value constituted by a single convex cell; (b) its 1D skeleton

Use of primitive (translation) tensor

Translation: T(coords)(parameters)(object)

two primitive objects

```
door = CUBOID([1,3])
window = CUBOID([1,1.5])
```

one assembly

```
VIEW(STRUCT([muro, door, window]))
VIEW(SKELETON(1)(STRUCT([muro, door, window])))
VIEW(SKELETON(1)(STRUCT([muro, T(1)(1.5)(door),
    T([1,2])([2.75,1.5])(window)])))
```

Use of primitive (translation) tensor

\framesubtitle{STRUCT: from {**local**} **to** {**global**} **coordinates**

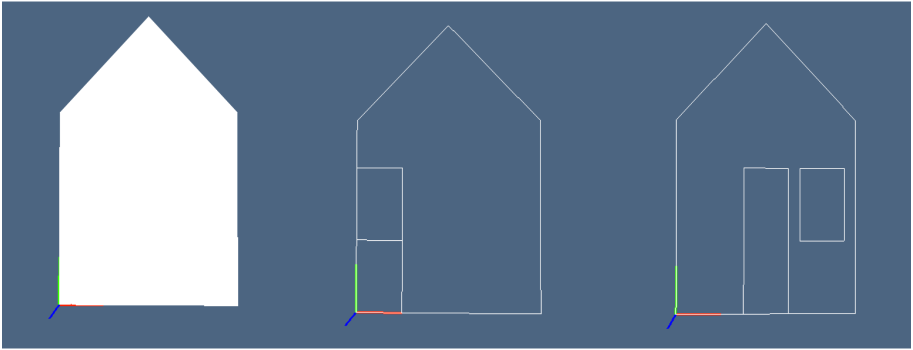


Figure : (a) three `hpc` values with a vertex on the origin; (b) their skeletons; (the translated skeletons)

STRUCT primitive

STRUCTture: used to assembly geometrical values

`COLOR(color)(obiect)` primitive

```
house = STRUCT([muro, COLOR(RED)(T(1)(1.5)(door)),
    COLOR(GREEN)(T([1,2])([2.75,1.5])(window))])
VIEW(house)
```

STRUCT primitive

STRUCTture: used to assembly geometrical values

COLOR(color)(obiect) primitive

```
house = STRUCT([muro, COLOR(RED)(T(1)(1.5)(door)),
    COLOR(GREEN)(T([1,2])([2.75,1.5])(window))])
VIEW(house)
```

Using a Boolean operator

```
house = DIFFERENCE([muro, COLOR(RED)(T(1)(1.5)(door)),
    COLOR(GREEN)(T([1,2])([2.75,1.5])(window))])
VIEW(house)
```

Use of STRUCT primitives

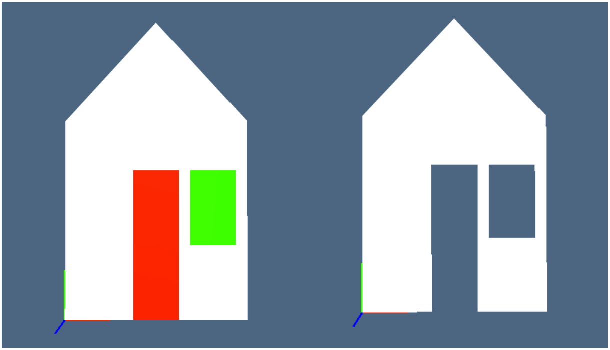


Figure : (a) substituting `DIFFERENCE` for `STRUCT`; (b) using the `COLOR` primitive.

Use of PROD primitive

PROD: used to make the Cartesian product of geometrical objects (pointsets)

Cartesian product times an interval of size 4

```
house3D = PROD([house, Q(4)])   # properties (color) are lost
VIEW(house3D)
```

Use of PROD primitive

PROD: used to make the Cartesian product of geometrical objects (pointsets)

Cartesian product times an interval of size 4

```
house3D = PROD([house, Q(4)])    # properties (color) are lost
VIEW(house3D)
```

New assembly

```
muro = PROD([muro, Q(4)])
door = T(1)(1.5)(PROD([door, Q(4)]))
window = T([1,2])([2.75,1.5])(PROD([window, Q(4)]))
house = STRUCT([muro, COLOR(RED)(door), COLOR(GREEN)(window)])
VIEW(house)
```

Use of PROD primitive

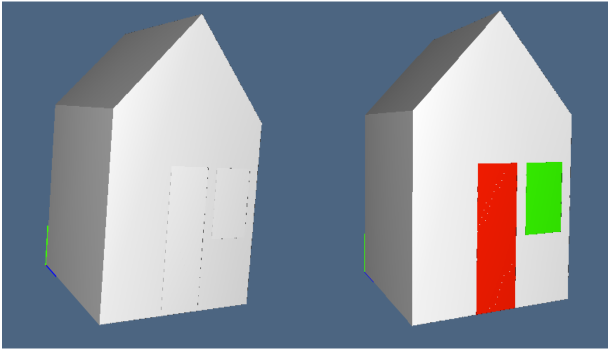


Figure : some solid operations loose the property values of assemblies

Advanced use of affine tensors within an assembly

PROD: used to make the Cartesian product of geometrical objects (pointsets)

$$\text{STRUCT}([Q, hpc_1, Q, hpc_2, \ldots, Q, hpc_n) \equiv$$
$$\text{STRUCT}([Q(hpc_1), Q^2(hpc_2), \ldots, Q^n(hpc_n))$$

```
pair_x = [T(1)(4), house]
houseRow = STRUCT(NN(10)(pair_x))
VIEW(houseRow)
```

assembly of assemblies

```
pair_z = [T(3)(14), houseRow]
neighbourhood = STRUCT(NN(10)(pair_z))
VIEW(neighbourhood)
```

Advanced use of affine tensors within an assembly

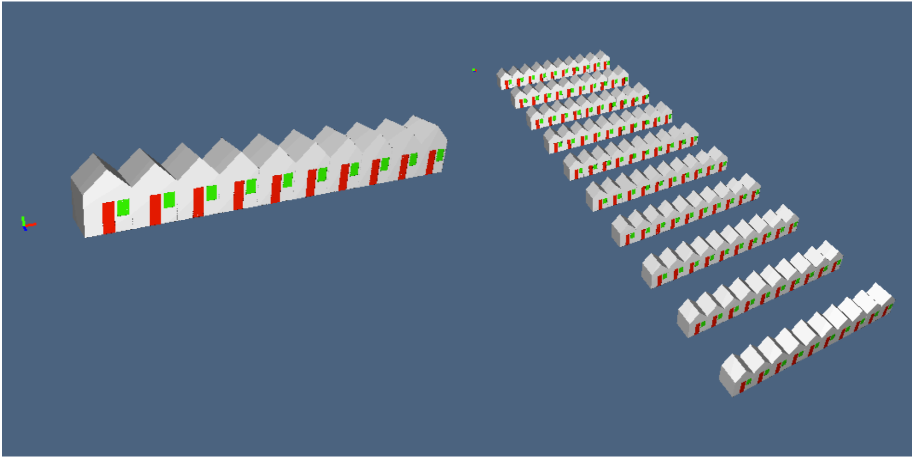
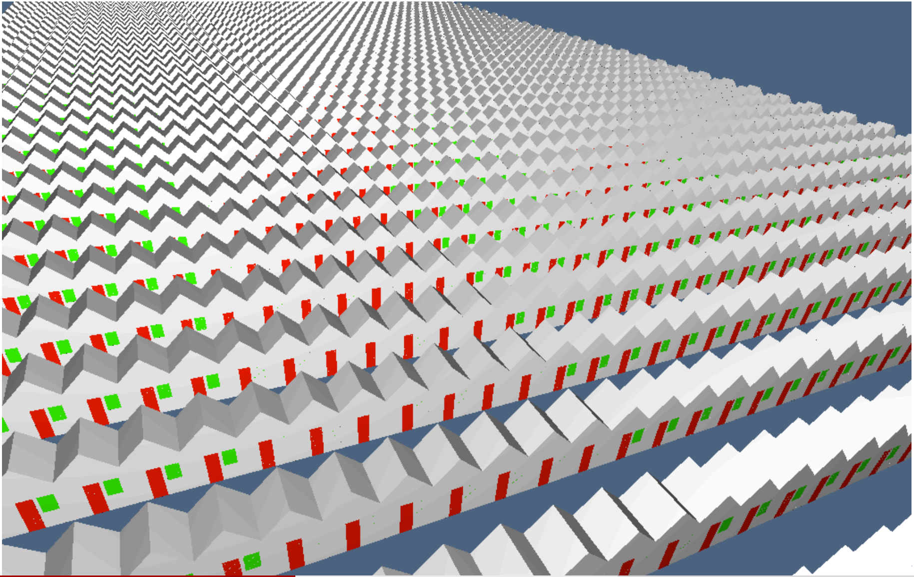


Figure : automatic composition of affine tensors within an assembly

Example (1/2)

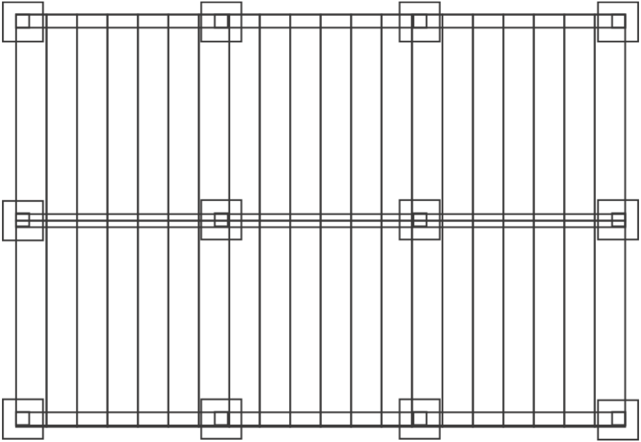
```
from pyplasm import *

verts = [[0,0],[4,0],[4,4],[2,6],[0,4]]
muro = JOIN(AA(MK)(verts))
door = CUBOID([1,3])
window = CUBOID([1,1.5])

bianco = PROD([muro,Q(4)])
rosso = COLOR(RED)(T(1)(1.5)(PROD([door,Q(4)])))
verde = COLOR(GREEN)(T([1,2])([2.75,1.5])(PROD([window,Q(4)])))

house3D = STRUCT([bianco, rosso, verde])
VIEW(house3D)
fila = STRUCT([house3D, T(1)(4)] * 50)
VIEW(fila)
spiaggia = STRUCT([fila, T(3)(10)] * 50)
VIEW(spiaggia)
```

Example (2/2)



Assignment: 3D model of building framework

Using only QUOTE,Q,PROD,INSR,STRUCT operators



A solution (1/2)

```
from pyplasm import *

x_plinti = QUOTE([1.2,-4.8] * 8)
y_plinti = QUOTE([1.2,-4.8] * 6)
plinti = INSR(PROD)([x_plinti,y_plinti,Q(0.6)])
VIEW(plinti)

x_pilastri = QUOTE([-0.4,0.4,-0.4,-4.8] * 8)
y_pilastri = QUOTE([-0.4,0.4,-0.4,-4.8] * 6)
pilastri = INSR(PROD)([x_pilastri,y_pilastri,QUOTE([-0.6,3.6] * 12)])
VIEW(STRUCT([plinti,pilastri]))

x_travi = QUOTE([-0.4,6*7+.4])
y_travi = QUOTE([-0.4,6*5+.4] )
travi = INSR(PROD)([x_travi,y_travi,QUOTE([0.6,-3,0.6] * 12)])
VIEW(STRUCT([plinti,pilastri,travi]))
```

A solution (2/2)