Geometric & Graphics Programming Lab: Lecture 10

Alberto Paoluzzi

November 4, 2016

1. Workshop N.4

2. Minimal git/github instructions

Workshop N.4

Parametric House Roofs

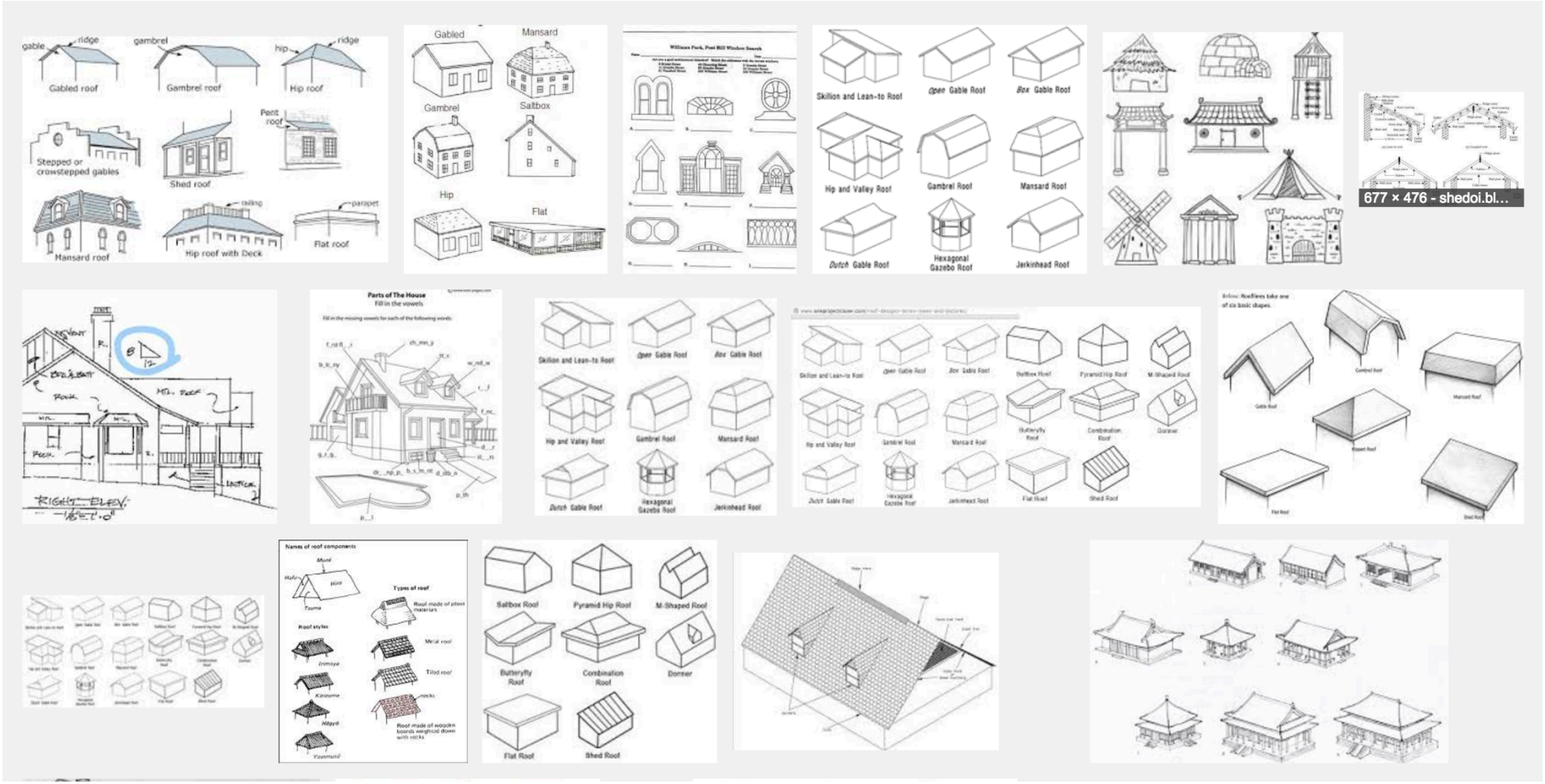
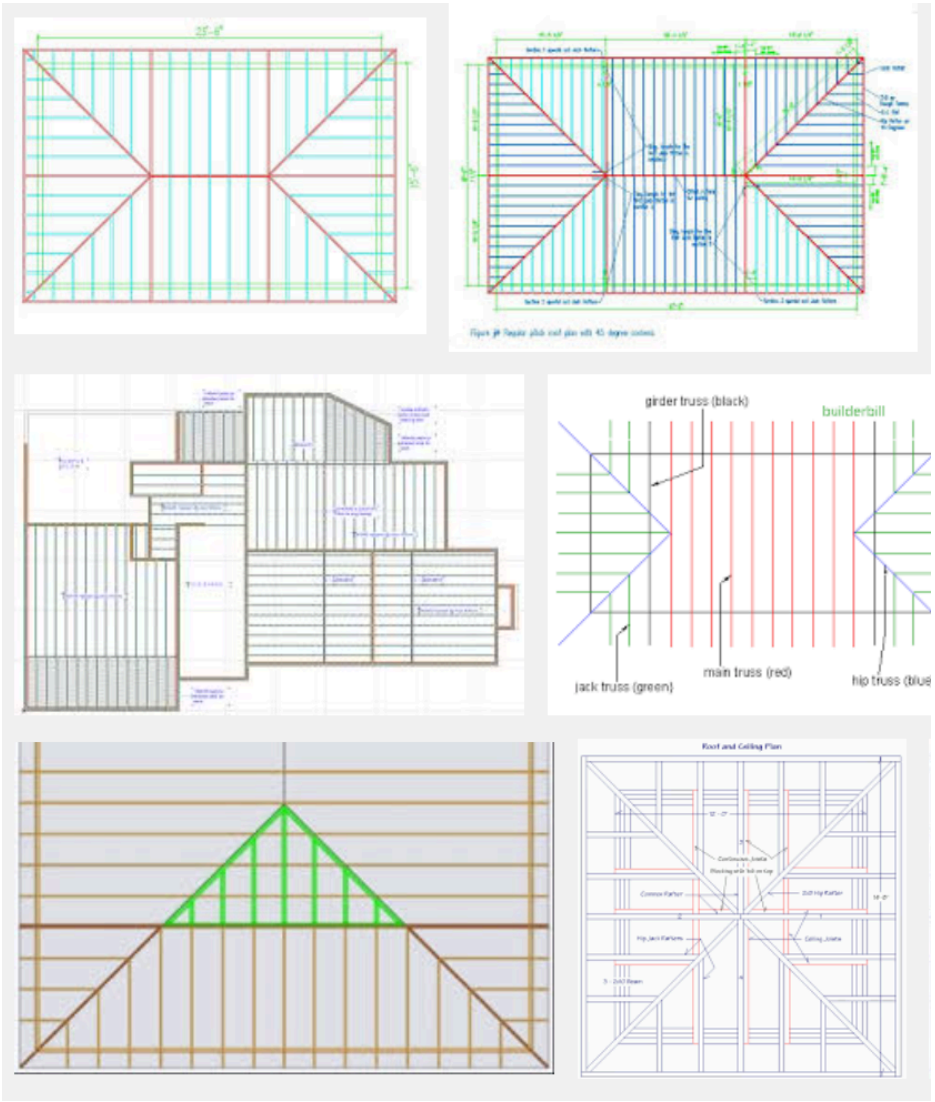


Figure 1: Images from Google

Look at some examples

- roof design types
- hip roof design plans
- hip roof design plans
- complex hip roof design
- house roof models
- house roof design
- house roof structure design
- house roof styles



Roof terminology

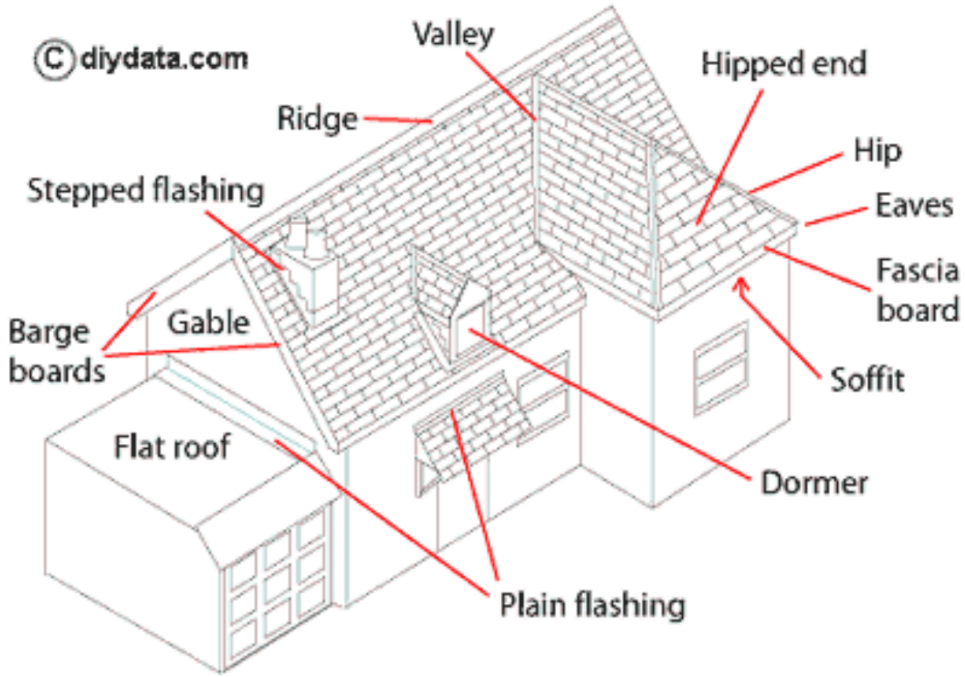


Figure 3: Glossary of roofing terms

- Wikipedia: Illustrations of common and simple roof shapes

Roof shape input

The volumetric shape of the roof will be defined using the 'MKPOL' primitive PLaSM operator

- draft the plan view of the roof on a paper sheet;
- identify and number (starting from 1) the vertices of the drawn graph
- decompose the graph faces into convex cells, without introduce new vertices;
- build 3D convex cells by adding new vertices where necessary
- number (starting again from 1) the cells;
- set an origin point and a Cartesian frame on the drawing;
- provide 3 coordinates for each vertex (write close to it);
- check the consistency of coordinates;
- look at your model: `VIEW(MKPOL([verts,cells,None]))`
- take any necessary feedback action

Requirements

- Write a single notebook, named `workshop_04.ipynb`
- Choose a notebook Title, for example `<my_roof_builder>`
- Start the notebook with a web reference and one/more image/s of your type of roof (i.e. your chosen kind of roof model)
- List the variables used in your code, with a textual definition
- Provide a short description of the geometric method you are going to implement
- Include the coding of a single Python function named `ggpl_<my_roof_builder>`
- Provide only 2 formal parameters, of type `list`, named `verts,cells`, respectively
- Provide the images generated by at least two executions with different actual parameters.
- Use measures in meters (m)

Hints to solution

1. for each vertex compute the incident faces (`SKEL_2`,`UKPOL`, identification of coincident points)
2. verify the planarity of faces (. . . ;-)
3. compute the 1-skeleton of the roof (`SKEL_1`)
4. make the `OFFSET` of the 1-skeleton with a small cuboid to get a solid roof frame
5. select the boundary 2-faces oriented upwards, and make thick (. . . ;-)
6. return a single `hpc` value including the `structure` and the roof `rising faces`

IMPORTANT ADVICE

- Start with the simple points, and produce an output
- Only later refine your solution with the more difficult points . . .

Style specs (1/2)

- produce a notebook file, of type `.ipynb` (The ipynb file extension is associated with the IPython notebook and/or Jupiter, a rich architecture for interactive computing written in Python and available for various platforms.)

Style specs (2/2)

- output: a single HPC value
- use meaningfull identificators (variables and parameters)
- use camelCase ids
- add Python `docstrings` (google for it)
- produce a single notebook file, named workshop_04.ipynb
- file path: `your_repo/2016-11-04/workshop_04.ipynb`

Minimal git/github instructions

Minimal git/github instructions (1/2)

create your local repository

```
$ mkdir 2016-11-04
$ cd 2016-11-04
$ touch workshop_04.ipynb
```

Minimal git/github instructions (2/2)

commit your work

```
$ git add -A .
$ git commit -m "add a short note to commit"
$ git push origin master
```