
match3d Documentation

Release 0.0.1

Ryan Henderson

May 19, 2016

1	Contents	3
1.1	Installation	3
1.2	Basic usage	4
1.3	About Image Match Generator	5
1.4	Generator of Images for Humans	7
1.5	Documentation	8
2	Indices and tables	9

`match3d` is an extension for our image matching library. It allows you to store and search 3D models (STL files) for similar designs.

Contents

1.1 Installation

We hope to streamline the setup process soon. In the meantime, you follow the *Manual setup*, or use *Using Docker*.

1.1.1 Manual setup

image-match

You will need `image-match` installed. Follow the install directions in the link. Along the way you will get elasticsearch and the scientific python libraries set up.

blender

3D-match uses `blender` for rendering. The blender packages are out-of-date on Ubuntu 15.10, so to be safe install blender as explained [here](#):

```
$ sudo add-apt-repository ppa:thomas-schiex/blender
$ sudo apt-get update
$ sudo apt-get install blender
```

match3d

To install `match3d` along with its python dependencies:

```
$ pip install -e .
```

1.1.2 Using Docker

Run elasticsearch:

```
$ docker-compose up -d es
```

Then to start an ipython session:

```
$ docker-compose run --rm m3d ipython
```

1.2 Basic usage

match3d is essentially a collection of blender scripts for generating views of 3D models, and Python scripts for turning those into something searchable by image-match.

1.2.1 API operations

The most important functionality is provided in `api_operations.py`.

```
from match3d.api_operations import APIOperations
api = APIOperations(index_name='3d_test')
```

`index_name` is the name of the elasticsearch index to use. Defaults to `match3d` if none is specified.

ADD

You can add a model from URL or local file. You must specify some kind of label either way. Label uniqueness isn't enforced (yet), but duplicate labels will be ignored in searches:

```
api.add('human', stl_url='http://people.sc.fsu.edu/~jburkardt/data/stla/humanoid_tri.stl')
api.add('human_other', stl_url='http://people.sc.fsu.edu/~jburkardt/data/stla/humanoid.stl')
```

Some more example **STL** files. For example, download and unzip the Porsche, then:

```
api.add('porsche', stl_file='/home/ryan/Downloads/porsche.stl')
```

SEARCH

Search the renderings of different objects and the score of the single best view. Lower numbers are better matches.

```
api.search(stl_file='/home/ryan/Downloads/porsche.stl')
```

Gives the result:

```
{ '/home/ryan/Downloads/porsche.stl':
  {
    u'human_other': 0.44157460914354879,
    u'porsche': 0.0
  }
}
```

Or:

```
api.search('http://people.sc.fsu.edu/~jburkardt/data/stla/humanoid.stl')
```

gives:

```
{ 'http://people.sc.fsu.edu/~jburkardt/data/stla/humanoid.stl':
  {
    u'human': 0.13292872986279264,
    u'human_other': 0.0,
    u'porsche': 0.24738691224575407
  }
}
```


LIST

```
api.list_designs()
```

returns

```
[u'porsche', u'human_other', u'human']
```

1.3 About Image Match Generator

The purpose of Image Match Generator is to make a set of “standard” images of a 3D object. Other software converts those images to image signatures. Image signatures can be stored in a database. Given a new image signature, one can find similar image signatures in the database.

1.3.1 Prerequisites

To use Image Match Generator, you need to install Blender 2.74 or later.

`image_match_generator.py` is a Python script but you don’t need to install Python to use it. That’s because Blender comes bundled with a version of Python (currently Python 3.4) and it uses that.

1.3.2 Assumptions

All 3D objects are assumed to be modeled using STL files. Those STL files are assumed to be in subdirectories of a directory.

`image_match_generator.py` has a command-line interface.

1.3.3 Example 1

The simplest use of `image_match_generator.py` has the form:

```
$ blender -b -P image_match_generator.py -- -d ~/stl_set_a/ -o test_output/
```

- `blender -b` launches Blender in the background (rather than launching the Blender GUI).
- `-P image_match_generator.py` tells Blender to run the Python script named `image_match_generator.py`.
- The double dash (`--`) indicates the beginning of the arguments for the Python script (i.e. not for Blender).
- `-d ~/stl_set_a/` tells the Python script that the directory with subdirectories containing STL files is `~/stl_set_a/`.
- `-o test_output/` tells the Python script that the generated images should be written to the directory `test_output`. If that directory doesn’t exist, it will be created.

In this case, 48 images get generated, plus a report file named `image_match_generator_report.csv`. Each generated image file will have a filename similar to:

```
cf4a7d5060943dd196b1e34fb6cfbf74.2.3.back.1.png
```

Here’s what that filename means:

- `cf4a7d5060943dd196b1e34fb6cfbf74` is a hash of the name of the *directory* containing the STL file (not the name of the STL file).

- The `.2` following that means the camera was at one end of eigenvector 2. Eigenvector 0, 1, and 2 are the three eigenvectors of the moment of inertia matrix of the object (treating it as a hollow shell). That is, they are the three principal axes.
- The `.3` following that means the object was rotated by $3/4$ of a full rotation (270 degrees). That slot can have the values 0, 1, 2, and 3.
- The `.back` means the camera was at the back end of the eigenvector, not the front end.
- The `.1` means the the object was reflected. If it had been `.0` then that would mean it wasn't reflected.

1.3.4 Command Line Arguments

Image Resolution

The default image resolution is 1024px by 1024px, but that can be changed using the `--resolution RESOLUTION` argument. For example, to render images that are 200px by 200px, use something like:

```
$ blender -b -P image_match_generator.py -- -d ~/stl_set_a/ -o test_output/ --resolution 200
```

Turning off Rotations, Front+Back, and/or Reflections

By default, for each of the three eigenvectors, both the front and back view will be generated, and for each of those four rotations will be generated, and for each of those two reflections will be generated. That makes $3 \times 2 \times 4 \times 2 = 48$ images total. You can turn off any or all of front+back, rotations, or reflections using the following flags:

```
--no-rotations
--only-front-view
--no-reflections
```

You can use one, two, or all three of those flags together. For example, here's how you'd use all three, so only three images would be generated (the front view of three eigenvectors):

```
$ blender -b -P image_match_generator.py -- -d ~/stl_set_a/ -o test_output/ --no-rotations --only-front-view --no-reflections
```

Turning off Image Transforms

By default, Blender isn't used to render every image, because rendering is computationally expensive. Instead, whenever possible, image rotations or image reflections are used instead. Collectively, those are known as image transforms. You can turn off image transforms using the `--no-image-transforms` flag. If you do, every image will be generated by reflecting or rotating the *object*, rendering that object, and then unrotating and unreflecting the object as necessary.

1.3.5 Generating Images for the Database

When generating images for the database, you want all 48 images, so just call `image_match_generator.py` as in the simplest use case.

1.3.6 Generating Images for Search

When generating images for search, you only need a small number of images, say three or six.

1.3.7 Getting Help from the Command Line

To get help with Image Match Generator from the command line, use

```
$ blender -b -P image_match_generator.py -- -h
```

Here's the current output:

```
usage: blender [-h] [--resolution RESOLUTION] [--no-rotations]
              [--only-front-view] [--no-reflections] [--no-image-transforms]
              d o

Generate oriented images for image matching

positional arguments:
  d                      directory containing STL files
  o                      directory where to put images

optional arguments:
  -h, --help            show this help message and exit
  --resolution RESOLUTION
                        resolution of renderings (n x n)
  --no-rotations        do not generate rotations
  --only-front-view     only generate front views
  --no-reflections      do not generate reflections
  --no-image-transforms
                        generate all images by rendering
```

1.4 Generator of Images for Humans

1.4.1 Overview

This Python Blender script produces images, one image per STL file, showing nine renderings of the 3D object pasted into a single image as a 3x3 collage. The idea is to have one image that can give a human a good sense of what the 3D object is. The camera positions are chosen to be fairly evenly distributed on a sphere (not randomly).

1.4.2 Usage

Example usage:

```
$ blender -b -P generate_images_for_humans.py -- -d ~/Documents/ascribe/cad_files/testset1/ -o test_o
```

The `-d` argument is the directory containing subdirectories which contain STL files. All of those subdirectories will be scanned and the STL file they contain will be rendered.

The `-o` argument is the output directory where all the final images will be saved. If that directory doesn't exist yet, it will be created. Each image is named after the parent directory of the associated STL file.

Note that we're assuming that each subdirectory contains only one STL file.

1.5 Documentation

This section contains instructions to build and view the documentation locally, using the `docker-compose.yml` file of the `image-match` repository: <https://github.com/ascribe/3d-match>.

If you do not have a clone of the repo, you need to get one.

1.5.1 Building the documentation

To build the docs, simply run

```
$ docker-compose up bdocs
```

Or if you prefer, start a bash session,

```
$ docker-compose run --rm bdocs bash
```

and build the docs:

```
root@a651959a1f2d:/usr/src/app/docs# make html
```

1.5.2 Viewing the documentation

You can start a little web server to view the docs at <http://localhost:50080/>

```
$ docker-compose up -d vdocs
```

Note: If you are using `docker-machine` you need to replace `localhost` with the `ip` of the machine (e.g.: `docker-machine ip tm` if your machine is named `tm`).

1.5.3 Making changes

The necessary source code is mounted, which allows you to make modifications, and view the changes by simply re-building the docs, and refreshing the browser.

Indices and tables

- `genindex`
- `modindex`
- `search`