

Crystal Match - System User Guide

Software version: V0.1.1

Last Updated: 30/11/2016

Contributions:

- Chris Martin - Contractor (Tessella) - Sept 2016 - Current
- Kris Ward - Contractor (Tessella) - Jan 2016 - Sept 2016

Overview

CrystalMatch was developed for the VMXi beamline at the Diamond Light Source to map points on an image of a sample to their current location using Feature Matching techniques. This is a two phase process which accounts for both the movement of the plate (the Alignment phase) and the movement of crystals in the sample (the Crystal Matching phase).

Context

VMXi is a macromolecular crystallography (MX) beamline which uses X-Ray diffraction to infer the 3-dimensional structure of protein molecules. It is necessary for the sample to be crystalline as the molecules are arranged in a regular lattice structure sharing the same orientation.

The current approach to MX involves growing crystal samples in solution before isolating them manually and fixing them in place using oil, glass or a loop of nylon/plastic. VMXi aims to be the World's first high throughput MX beamline which targets crystal samples in solution without the need to isolate the crystals beforehand.

Plates of protein samples sent to Diamond are stored in Formulatrix cooling units near the beamline - periodically the samples are photographed by the storage units. Scientists can log into a remote system to view the samples and mark areas of interest to be sampled on the beamline.

Once the sample is transported to the beamline a second image is taken. CrystalMatch is then used to analyse the image and account for differences in the alignment of the plate on the beamline and the movement of crystals in the sample caused by either storage or the transport process.

Solution Description

Crystal Match is implemented in Python and uses different Feature Matching techniques found in the OpenCV toolbox. The process is split into two phases:

The Alignment phase uses a single Feature Matcher (ORB at time of writing) to compare the Formulatrix and Beamline images calculating a translation between the two. Note that this does not take into account scaling between the images which must be set manually. If this process fails the sample cannot be aligned and the second phase is abandoned.

The Crystal Matching phase uses the average result of a series of localised Feature Matchers to map a Point of Interest (POI) marked on the Formulatrix Image to a location in the Beamline image. It does this by taking a Region of Interest (ROI) around the POI and comparing it to a larger ROI in the Beamline image to allow for movement of the crystal. If this process fails the co-ordinates reported will have the transform from the Alignment phase applied accounting for the plate alignment but not the possible movement of the crystal.

Assumptions

The following assumptions have been made during the development of this software:

- The images will be of the same resolution or the scale will be defined in the command line.
- Focus stacking will have been carried out on both the Formulatrix Image and Beamline Image.
- The aspect of the crystal facing the camera in both images will be similar - Feature Matching aims to be a viewpoint-invariant process but crystals which rotate entirely cannot be matched using this method.

Quick Guide

CrystalMatch can be run from the command line using the following basic syntax:

```
CrystalMatch Formulatrix_image beamline_image [x,y [x,y ...]]
```

The app will attempt to locate a configuration directory in the current working directory; if one is not found a `config` directory will be created at the current location. The location of the configuration directory can be set using a command line flag - see *Configuration and Log Files*.

Co-ordinate System

Co-ordinates for selected points should be given from the top-left of the image and will be output in the same format. Transforms will also be expressed from the top-left pixel.

CrystalMatch provides all output in pixels for accuracy and is unit-agnostic.

NOTE: In some cases debug information and configuration options are expressed in micrometers (um) - this is a hangover from the original implementation part of the testing GUI - these values should be disregarded or treated as unit-less values.

Scale

The Feature Matching does not account for scaling and assumes that both the Formulatrix and beamline image have the same resolution.

The scale can be specified using the `--scale s1:s2` flag where `s1` and `s2` are the pixels per [unit] (resolution) of the Formulatrix and beamline images respectively. Note that the actual unit used does not matter but the values must be relative to each other.

The default values for scale are set in the Alignment phase configuration file (`align.ini`) and can be modified - these will still be overridden by the command-line flag.

Configuration and Log Files

By default the app will attempt to locate a configuration directory at the current working path; a `config` directory with default values will be created if one is not found. This behaviour can be overridden using the `--config path` flag which sets the path of the configuration directory - again, if one is not found at this path a `config` directory will be created with default values.

A `log` directory will also be created at the same file level as the configuration directory by default. This can be overridden by setting a permanent path in `config/settings.ini`.

The configuration files are designed to allow limited changes to be made to the app and algorithm parameters without needing to recompile from source. To this end they are self-documented and can be edited using any text editor. A summary of the settings file contents is as follows:

- `settings.ini` - App level settings - primarily log file options.
- `align.ini` - Settings for the Alignment phase including acceptance metrics, Feature Matcher used and default scaling options.
- `crystal.ini` - Settings for the Crystal Matching phase such as the size of ROI and the transform method.
- `gui.ini` - The GUI is currently for testing purposes only and cannot be accessed in the compiled app - this file should be ignored.
- `licensing.ini` - Activate/Deactivate SIFT and SURF proprietary algorithms in the OpenCV toolbox. These are not currently free for commercial use.
- `det_*.ini` - Where `*` is the name of a feature detector. Settings specific to that detector.

Output

CrystalMatch outputs results in a human-readable format by default - when being called by a service it is recommended that the `--to_json` flag is used to return JSON formatted results instead:

JSON schema

- `input_image` - Absolute path to Formulatrix Image.
- `output_image` - Absolute path to Beamline Image.
- `exit_code` - Describes the exit status of the application, used to flag abnormal runs while returning some results if possible.
 - `code` - Exit status - 0 for success or 1 for failure - **note:** not to be confused with status code (see later).
 - `msg` - Human-readable error message.
- `alignment` - Alignment phase results.
 - `status` - Alignment status.
 - `code` - 1 for success or 0 for failure.
 - `msg` - Human readable error message.

- `scale` - Scale between Formulatrix and Beamline images - this is calculated from the input scale ratio.
- `translation` (x and y values) - The translation required to map a point from the Formulatrix image to the Beamline image.
- `mean-error` - Mean error value after image alignment.
- `poi` (optional, array) - An array of Crystal Matching phase results - will not be present if input points were not specified or Alignment phase failed.
 - `status` - Results status
 - `code` - 1 for success or 0 for failure.
 - `msg` - Human readable error message.
 - `location` (x and y values) - Calculated location of the POI in the co-ordinate space of the Beamline image. Note that if the match fails this will be the original point with the Alignment transform applied.
 - `translation` (x and y values) - The translation required to account for Crystal movement (excludes Alignment transform).
 - `mean_error` - Mean error value for this POI match.

Command Line Options

Command	Action
<code>-h, --help</code>	Shows help information
<code>--config path</code>	Sets the configuration directory.
<code>--scale scale</code>	Sets the scale between the Formulatrix and Beamline images given as two floats separated by a colon. Note this is relative (1:2 is the same as 2:4) and a value must be specified for each image using the format '[input_image_resolution]:[output_image_resolution]'.
<code>--version</code>	Shows the program's version number.
<code>-v, --verbose</code>	Increase output verbosity on the console.
<code>-d, --debug</code>	Output debug information to the console.
<code>-j job_id,</code> <code>--job job_id</code>	Specify a <code>job_id</code> - this will be reported in the output to help identify this run.
<code>--to_json</code>	Output results as a JSON object.
<code>--log path</code>	Write log files to the directory specified by path.