VPHOT Technical Documentation

VPHOT was developed using Visual Studio, utilizing ASP.NET, C++ and javascript, along with MySQL5.1 for data storage.

The solution consists of five projects:

- VPHOT: web application
- BL: shared application logic
- PhotLib: the C++ photometry library
- PhmServer: the server application doing plate solving etc.
- Watchdog: a simple app monitoring PhmServer

When zipping the files to disk you should have the latter four placed in My documents/Visual Studio xxxx/Projects/VPHOT_proj, while the web application should be placed at My documents/Visual Studio xxxx/websites/VPHOT. Open the solution by selecting My documents/Visual Studio xxxx/Projects/VPHOT_proj/VPhot.sln.

FITS files are stored in a folder on the server, the path to this folder is given in the CurrentFITSFolder key in web.config, located in the root of the web application.

Information about FITS files, user etc. is stored in a MySQL database named Photometrica.

VPHOT - WEB APPLICATION

This is an ASP.Net web application currently running under .Net 4.

Third-party components

The following components are used by VPHOT:

- MySql.Data.dll. Used to access the MySQL database. Freeware.
- ICSharpCode.SharpZipLib.dll. Used to zip and unzip files during upload / download. Freeware
- cfitsio.dll . CFITSIO library from HEASARC, NASA.
- Dart.FileUpload.dll. Used to facilitate advanced uploading via the browser. We have bought a license for this one.

These three ddl's are located in the bin folder.

Main file structure

All web pages use the phm.master file for common layout elements, such as menu and footer. Common styles are located in prof.css.

Users log in via the AAVSO web pages. From there they are re-directed to Login.aspx, where the username is decrypted and checked against the database. If this is a new user a new record is inserted in the USERS table.

All other web pages are located in the Restriced folder to prevent unauthorized access. After logging in the user is redirected to Restricted/files5.aspx, which lists images available to the user.

Application settings, such as folder paths, are found in Web.config, located in the web application root directory.

Help files are located in the Help folder.

Images are found in the img folder below root. Both static images used for the web pages are found here, as well as dynamically generated images resulting from image analysis. These are stored in temporary folders named after the user's session ID. These folders contain jpgs for displaying FITS files in the browser, as well as jpegs from time series analysis.

UploaderTemp is used to temporary store files uploaded via the web user interface as they are streamed to the server, before they are moved to the incoming directory of the PhmServer application.

Main features overview

The following list the main features in relation to file structure:

- Image list: files5.aspx.
- Time Series analysis: parameters are set in TimeSeries.aspx, then redirected to TimeseriesProgress.aspx, and results are shown in TimeseriesResults.aspx. Details shown in TSDetails.aspx, and there is export in TSExport.aspx and AAVSO export in AAVSOReport_TS.aspx.
- Stacking: Stack.aspx.
- Sharing: Share.aspx.
- Renaming/change of filter or telescope: Rename.aspx.
- Uploading of images: Upload wizard in Uploading/UploadWizStep1.aspx through Uploading/UploadWizStep5.aspx. Quick upload in Uploading/Upload2.aspx. View server processing queue in Uploading/UploadProgress.aspx. View processed images in Uploading/ProcessedImages.aspx.
- Analysis log: log.aspx.
- Administration of sequences: Sequences.aspx.
- Administration of telescopes: Telescopes.aspx and details in Telescope.aspx.
- Administration of accounts: AccountTransfer.aspx.
- Display and analysis of image: SelectStars.aspx.
- Photometry report: SingleImageReport.aspx.

More Advanced Features

The code in many of the files above is pretty self-explanatory. The more advanced features are outlined a bit more in this section.

Single image analysis

SelectStars.aspx. The majority of the work is done in java scripts, located in Analysis.js. The Ajax-based PageMethods of Visual Studio is used to communicate between the server (ASP.Net code in SelectStars.aspx) and the client (javascript code in Analysis.js). The FITS image, sequence information,

selected stars etc. is stored in session variables on the server. They are also used to create the report in SingleImageReport.aspx.

To display the image a set of jpg tiles are created and stored in img/xxx, where xxx is the current session ID.

Time series

When the parameters of the time series analysis are set, the user is redirected to TimeseriesProgress.aspx, where the progress of the analysis is shown. The analysis is run asynchronically, code located in App_Code\TimeseriesAsyncRun.vb. Progress info is written in a file that the web page periodically reads and use to display the progress bar.

The photometry is stored in data tables (System.Data.DataTable), which in turn are kept in session variables. These are documented at the top of TimeSeries.aspx.vb.

When the analysis is done the report is created in TimeSeriesResults.aspx. At the same time the results are stored to the analysis log.

Analysis log

Analysis results from a single image are kept in a session variable structure on the server, see BL.ImageAnalysisHelper.vb. This structure is serialized to disk when the user opts to save the analysis, and a reference to this file is created in the ANALYSIS database table. Time series results are stored in a similar way when the time series completes (in SaveTS in TimeseriesProgress.aspx.vb).

BUSINESS LOGIC (BL)

The BL project is a dll project that contains functionality common to both the web application and PhmServer. Files are:

- Analysis.vb: functions for creating reports from an image and a sequence, + saving to the analysis log
- DB.vb: A wrapper for communicating with the database. The conStr constant defines the user name and password to the database.
- FITSWrapper.vb: A wrapper for reading and writing to FITS files. Uses the CFITSIO library (http://heasarc.gsfc.nasa.gov/fitsio/). The cfitsio.dll must be in the bin folder of the web application.
- ImageAnalysisHelper.vb: a helper class for storing single image analysis results. The class is serializable, so that it can be stored directly to disk.
- Measurement.vb: Represents a measurement, holding the position of the centroid, and a reference to the image (via the FITSWrapper class). The UpdateStatistics method uses this reference to calculate statistics from the C++ PhotLib library, and the measurement.vb class then holds those statistics in memory.
- PhotLib.vb: A wrapper class for the PhotLib library to expose its methods to the VB part of the project.
- StarImportvb: Methods for working with sequences. Import and parse data from AAVSO, VSX and GCVS. Resulting stars are measured in AddStarFromSequence2. This method finds the centroid of each star in the sequence for the given image, create a new measurement class

(Measurement.vb), and add it to a global session variable - `SESSION_ACTIVE_MEASUREMENTS`. The data is stored in memomory during the session, or until a new sequence in created.

- TimeAndCoord.vb: Contains various helper functions.
- Util.vb: Contains various helper functions.

PhotLib – PHOTOMETRY ALGORITHMS

Numerical calculations are done in a C++ dll project called PhotLib. It exposes its methods through Exports.def and PhotLib.cpp. Other files are:

- Measurement.cpp: calculates signal, FWHM, skyglow etc. for a given centroid
- CCDImage.cpp: calculates image statistics, and estimates centroid around initial coordinates.
- CListAnalysis.cpp: measures a list of stars in one go (optimization). Used by PhmServer when creating photometry lists for each incoming frame.

The main steps involved in the aperture photometry analysis process are:

- Centroid determination
- Sky fitting
- Aperture integration
- Final magnitude estimate
- Error estimation

Algorithms for these tasks are briefly discussed in the following sections.

Centroid determination

The centroiding algorithm is based on the DAOPHOT FIND algorithm in IRAF (1987PASP...99..191S). In short, it goes something like this:

- When the image is clicked a region centered on the click is scanned for star centroids
- The size of this region is determined from the aperture size
- A unit height Gaussian is fitted to every pixel in the region, based on that pixel's surrounding pixels. The fit tend to be good for pixels that happen to be in a star center, bad otherwise.
- A sharpness criteria is calculated to separate stars from hot pixels and cosmic rays.
- The pixel with the best fit is selected as the final centroid.

Finally the exact star center is calculated using center of mass calculations on the marginals, see the IRAF specification at http://iraf.noao.edu/docs/photom.html.

Sky fitting

VPHOT uses the following outlier rejection algorithm to remove high valued pixels in the sky annulus (from stars, hot pixels etc.):

- Calculate the mean and standard deviation of the ADUs in the annulus.
- Remove all pixels with ADU greater than 3 * std.
- Repeat until no more pixels are rejected

Then the sky glow is estimated using the mode:

mode = 3 * median - 2 * mean

This is the value subtracted from each pixel in the aperture.

Aperture integration

The star signal (instrumental magnitude) is estimated as

-2.5 * LOG(SUM(ADU - Sky) / exptime)

where the sum runs over all pixels in the aperture. That is, for each pixel that is fully contained inside the aperture, the software sums the pixels ADU, subtracted the estimated sky background (from step 2). The sum is divided by exposure time to get total intensity pr second before it is converted to the magnitude scale.

Along the rim of the aperture there are pixels that are only partially inside the aperture. If the distance from a given pixel to the centroid is less than the aperture radius - 0.5, it is included in the sum as described above. If the distance is greater than the aperture radius + 0.5 the pixel is excluded. If the distance is in between, a fraction of (ADU - Sky) is included, proportional to the amount of the pixel inside the aperture.

This is an approximate algorithm, but works fairly well. It is similar to the aperture integration algorithm in PHOT IRAF module, see http://iraf.noao.edu/docs/photom.html.

Final Magnitude Estimate: Calculating the instrumental magnitude of a target, It, and a comp star Ic, as described in the sections above, the magnitude estimate of the target is given as

V = It - Ic + C

where C is the known magnitude of the comp star. If we use more than one comp star, we get instrumental magnitudes I1, I2, ..., In. And hence n estimates of the targets magnitude, V1, V2, ..., Vn. VPHOT calculates the final magnitude as the average of these n estimates.

Error Estimation

In an ensemble solution with more than two comp stars, the magnitude is estimated as the average of the individual comp stars estimate, and the error is taken as the standard deviation of this sample. This error estimate will cover all error sources.

If one or two comp stars are used, the error estimate is based on the SNR of each measurement (the target measurement and the comp stars measurements). The standard error of a measurement is defined as

2.5 * LOG(1 + 1 / SNR)

where LOG is the 10 based logarithm, and SNR is defined as

S / Sqrt(S / (G + Ns * Std^2 * (1 + 1 / Nr)))

S: Total ADU in aperture
Ns: Number of pixels in aperture
Nr: Number of pixels is sky annulus

Std: ADU standard deviation in sky annulus
G: Gain of the CCD detector

For more information see the AAVSO CCD Observing Manual and 'Handbook of CCD Astronomy' by Steve B. Howell, 2000.

Finally, the standard error of each measurement is squared and summed, and the error estimation is the square root of this number.

PhmServer – IMAGE PROCESSING

As images are uploaded to the server, either via FTP from robotic observatories such as AAVSOnet or iTelescope, or via the web interface by users, they end up in a specific folder on the server – currently this is D:\Incoming. PhmServer is an application that checks that folder to see if there are any new images. If so, the image is opened and the FITS header is read. It looks for the key USERNAME as this specifies the owner of the image. It also looks up the TELESCOP field and then retrieves image scale etc. from the database for that telescope.

If the image is not plate solved it tries to solve it using PinPoint (for which we have a free license). It will try UCAC-3 first, and if it fails it will try UCAC-2, then USNOA, then GSC_ACT and finally a bright star version of GSC_ACT.

Plate solved or not, image information is stored in the database, in the IMAGES table (see database section). The image is then moved to the current image folder.

Finally, if the image is plate solved, a photometry list of all stars is produced and stored in a file, with the same name as the image but 'phot' as extension. These photometry files are used to enable variable star searching.

PhmServer will also do automated disk cleaning, removing temporary image (jpeg) files etc., and will also do a backup of the database each night.

Also, images uploaded more than 4 months ago are automatically deleted.

DATABASE

The database can be accessed via MySQL Administrator and MySQL QueryBrowser, both installed on the server and available via shortcuts on the desktop. Log in using

- Stored Connection: Local
- Server Host: localhost
- Port: 3306
- Username: <removed>
- Password: <removed>

The database (catalog) is called Photometrica, containing the following tables:

- analysis: results of both single image, time series and multi-color image analysis. One row pr analysis.
- imagesinanalysis: image list pr analysis
- bandtransforms: transformation coefficients pr telescope and filter band
- colortransforms: transformation coefficients pr telescope and color
- filtermappings: mapping between custom filter name and standard filter name pr telescope
- filter: list of filter names with ID
- images: information about images. One image pr row. Contains link to file on disk, RA/Decl of center, image scale (calculated), target name, reference to owner (user), telescope, date-time of observation and of upload etc.
- sequences: one row pr sequence, with reference to owner. The actual sequence is stored as a text string in the FileContent field.
- telescopes: telescope information. A user can have several telescopes.
- users: user registry. A user will automatically be registered first time he uses VPHOT. This is because the user is already registered at the AAVSO web site. The username is sent over to VPHOT as part of the link the user clicks on the AAVSO web site.