AIBOVision User Guide

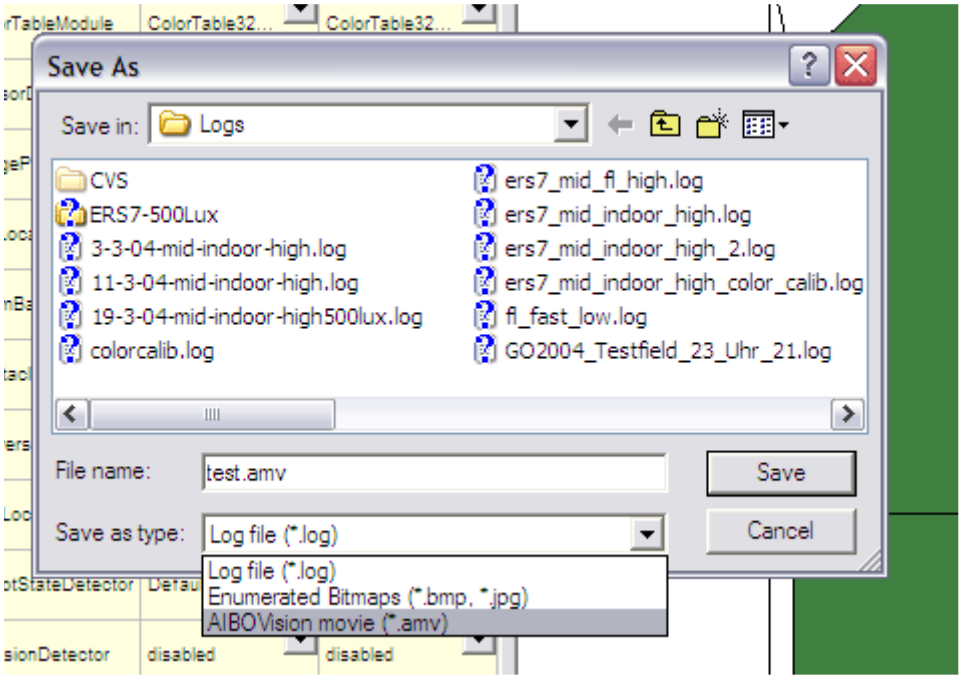
By Walter Nistico

Installation and running

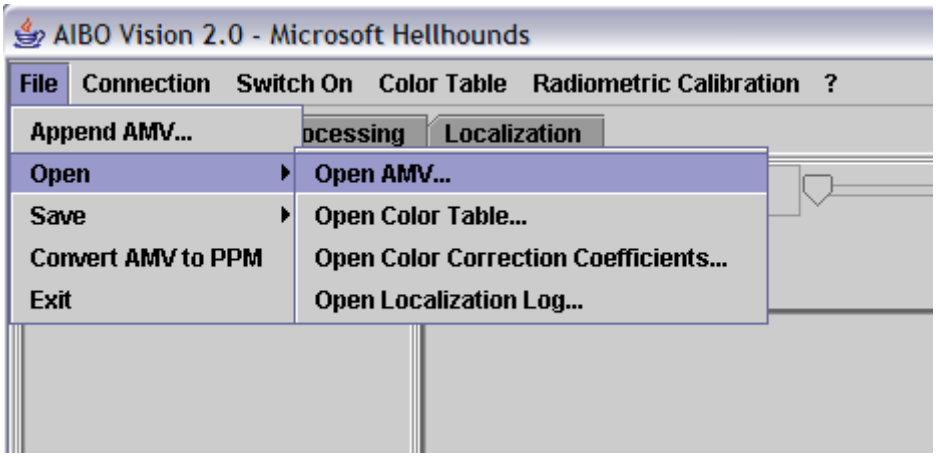
Install a recent version of a JDK (should run with j2sdk1.3 or higher); even if it's a java application, it won't run in OS other than windows, because makes use of the JNI to load *dll*s (they could be recompiled as lib<xxx>.so under linux though).
Place the files in a chosen folder, double click *install.bat*.
You can launch the application using *AIBOVision.bat*.

Importing a RobotControl log file

Open the chosen log file from robot control.



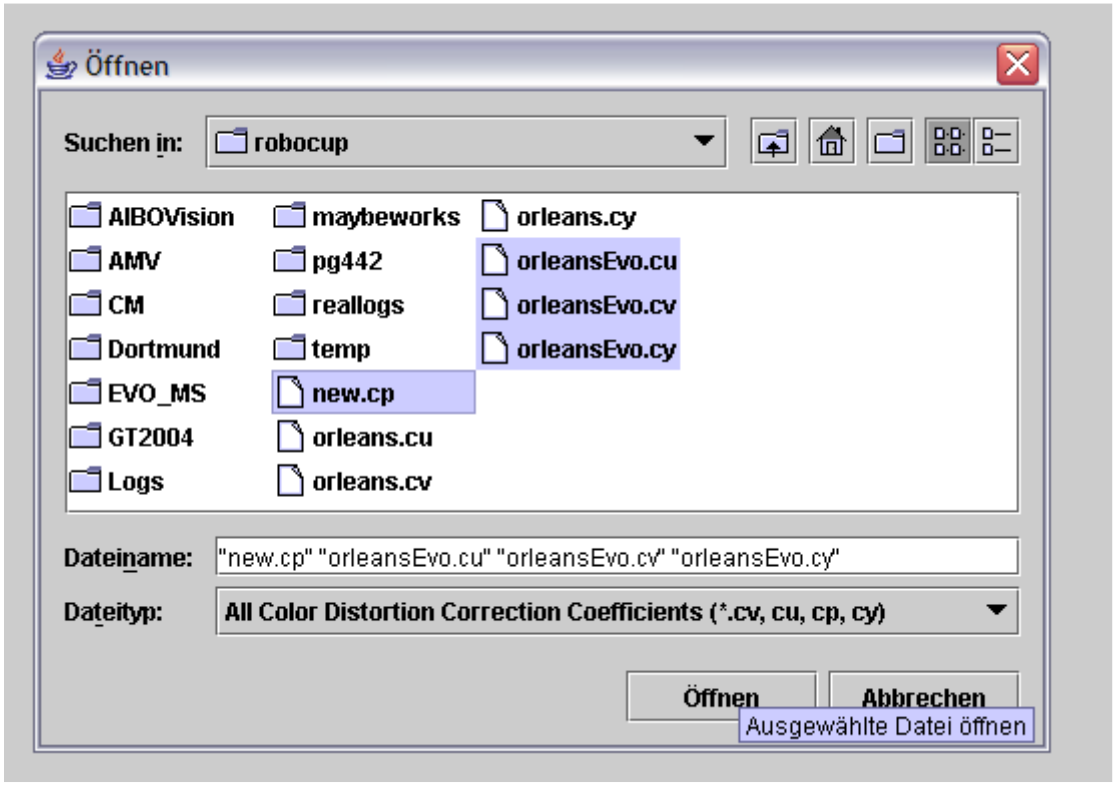
Save it with the log player as an AIBOVision movie (**NOTE**: especially if the original log file contained jpeg compressed pictures, the resulting *amv* file can be considerably bigger since it's uncompressed).
At the moment, ERS7 high resolution pictures (416x320) are unsupported in AIBOVision (could work, but not tested yet).



Now you can open the *amv* file with AIBOVision.

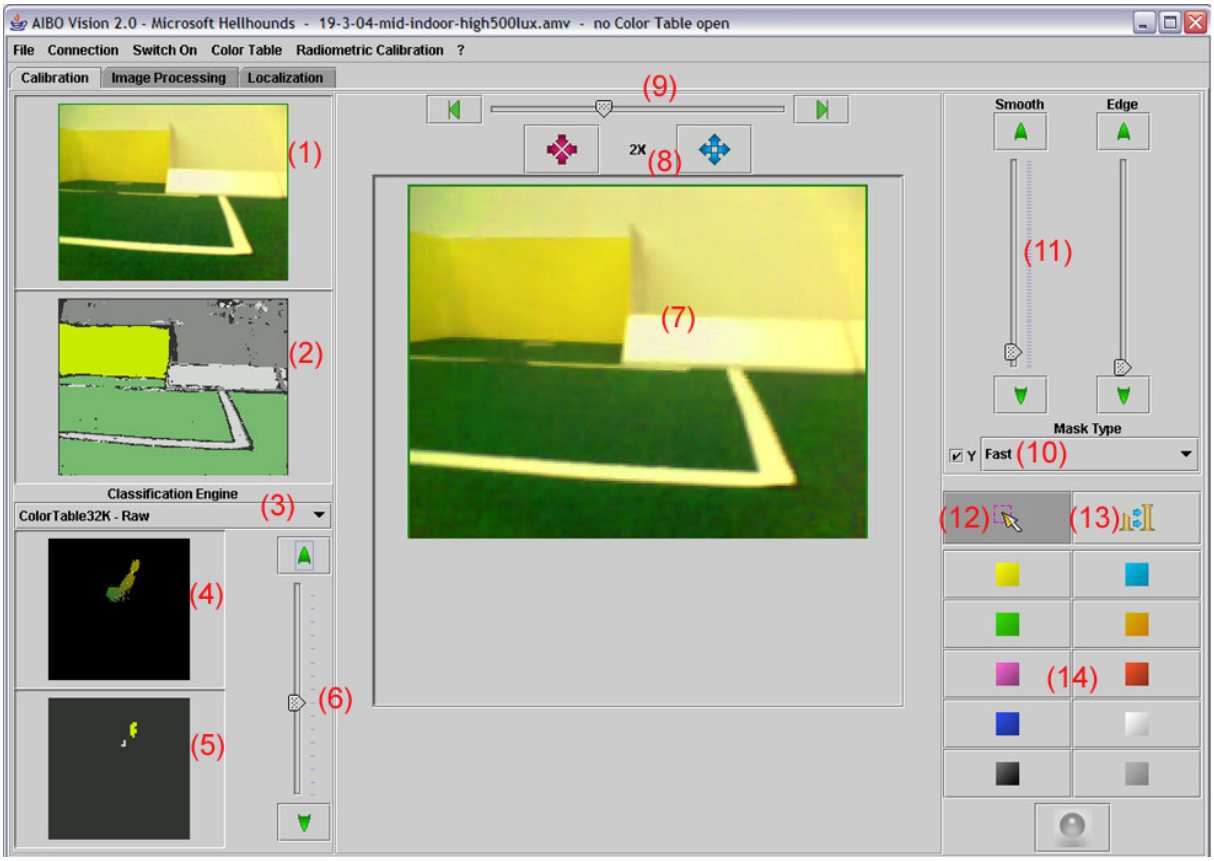
Before starting the creation of a new color table, it's advisable to switch on the color correction, otherwise it wouldn't be possible to correctly deal with objects in the corners of the image.

From the dialog reached through the menus *File -> Open -> Open Color Correction Coefficients* it's possible to load all the color correction coefficients which are to be used (.cy, .cu, .cv for the 3 chromatic components, additionally it's possible to load the center of distortion coordinates from a .cp file, currently unused in the robot code but coming soon).



Finally, the color correction is switched on by selected the checkbox located in the menu *Radiometric Calibration -> Correct Color*.

Color Table calibration



These are the functions of the main panel:

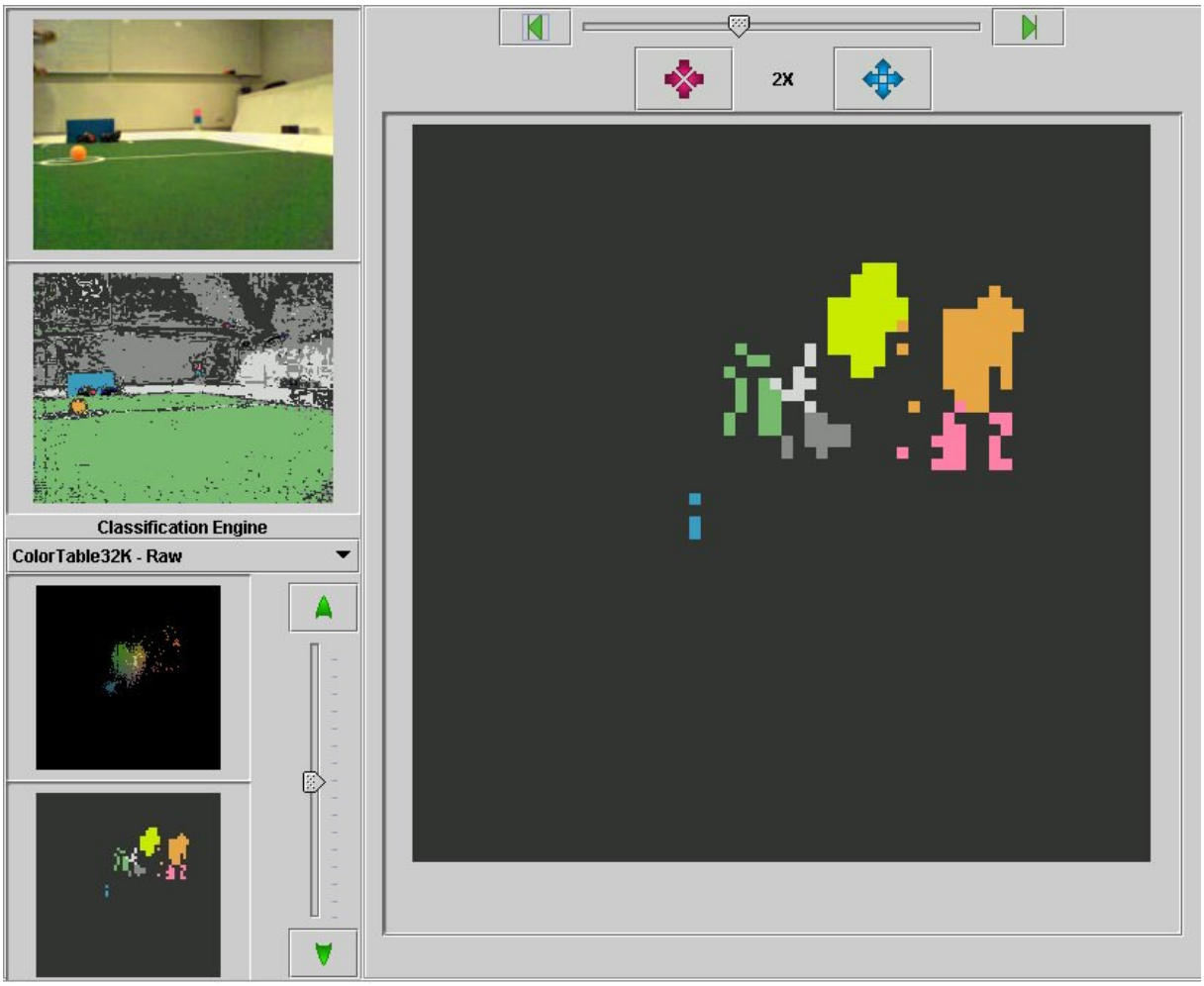
1. **Source image view**: right-clicking this panel offers the possibility to visualize the normal image (*Raw*), one of the color layers (*U* or *V*) or a transformed version where the brightness channel (*Y*) is replaced by the color saturation; this mode called *Color Enhanced* is useful to see the real color of objects, especially very dark ones such as the blue jersey of the robots (e.g. to determine if it can be assigned to the blue or black color class); all these visualization modes can be also *dynamically thresholded*, but this feature has no current application in the German Team code.
2. **Color classified image**: computed from the current color table.
3. **Classification engine selector**: it's possible to choose to buil a *ColorTable32K* (-Raw) or *ColorTable64*, the other modes are currently unused in GT code.
4. **Spectrum visualizer**: shows the UV spectrum of the current image, the brighter points represent the spectrum for the Y level selected in the **(6)** *brightness slider*; right-clicking this panel permits to visualized the dynamically thresholded spectrum, or the histograms of the *U* and *V* channel.
5. **Color table visualizer**: shows a *UV* layer of the current color table, corresponding to the *Y* value selected in the **(6)** *brightness slider*.
6. **Brightness slider**: selects the *Y* value for the visualization of **(4)** and **(5)** panels; the up and down arrows move the slider by fixed increments depending on the chosen color table quantization (16 levels for ColorTable32K, 4 levels for ColorTable64).
7. **Active area**: this zoomable and scrollable panel shows anything visualized in one of the fixed panels on the left, which can be selected by a left-click on the chosen one; when the *rectangular selection* button is on **(12)**, dragging the mouse over a region in this area selects the corresponding points to be assigned to a color class.

8. **Zoom buttons**: control the zoom factor of the active area, left (red) zoom in, right (blue) zoom out.
9. **Movie control**: selects the current frame to be processed, through the slider and the step back and forward buttons.
10. **Smoothing mask selector**: leave the *Fast* mode selected, the other modes can filter more noise but are way too slow to be used on the robot.
11. **Smoothing threshold slider**: controls the amount of blur VS noise-filtering balance, a good value for preserving all image features such as lines and edges, while rejecting most of the noise, is in the range 12-20 (clicking 3 to 5 times the up arrow, which steps up the slider by 4 level increments).
12. **Rectangular selection button**: when it's activated, it's possible to select rectangular regions in the *active area* by clicking and dragging the mouse.
13. **Undo button**: can undo up to 20 latest actions on the current color table.
14. **Color class buttons**: after a rectangular region in the *active area* has been selected, it can be assigned to a color class by clicking one of these buttons; the bottom button (*Remove*) assigns all selected pixels to the **null** color class; if the active area is currently representing a color table layer instead of a source camera image, is possible to directly edit it by dragging the mouse and assigning the corresponding region to a color class.

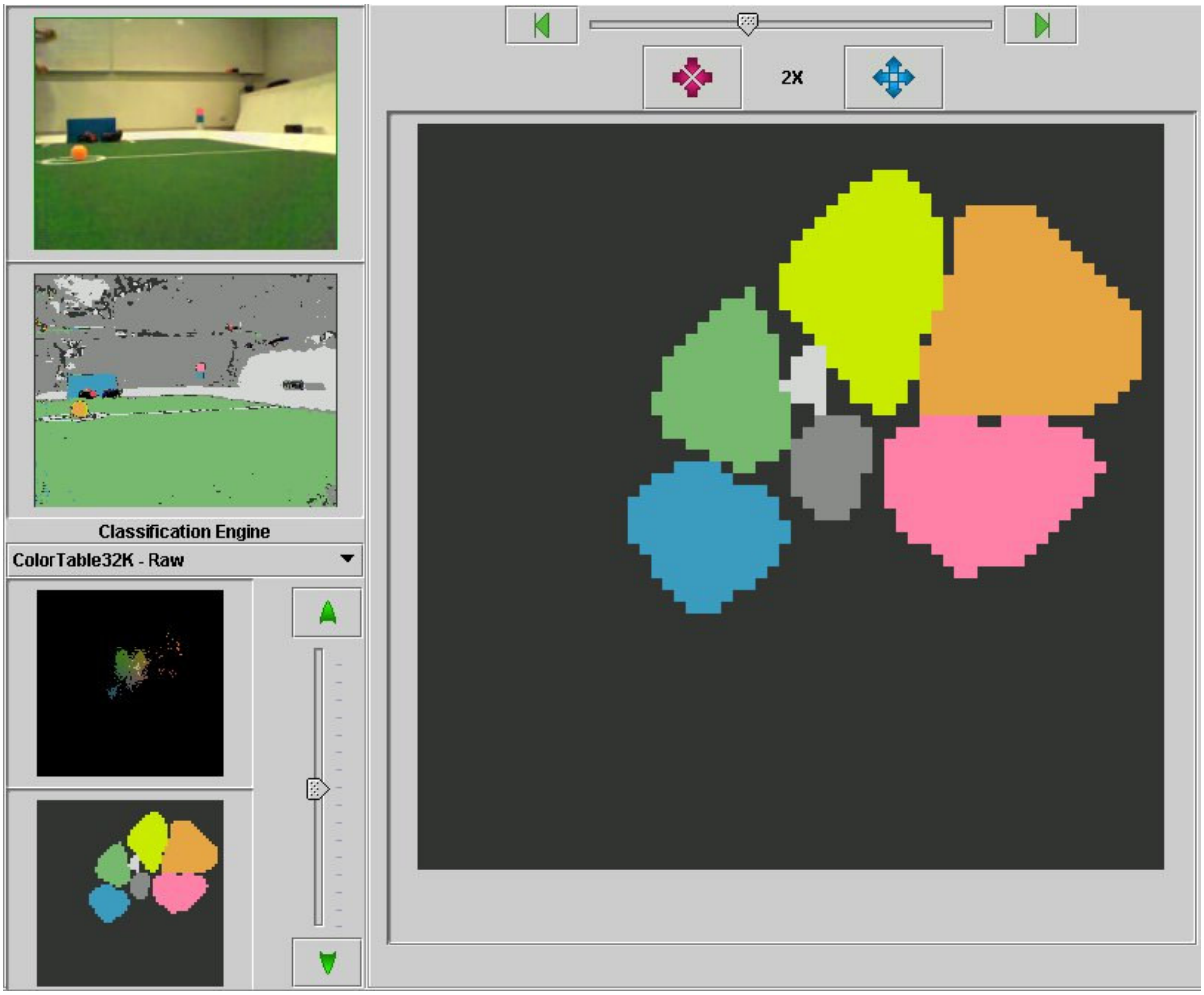
While the color table can be built interactively in a similar way as in the ColorTable64 dialog, in general is not advisable to insist on similar images until they all look good, but instead taking few samples of most lighting situations to be found in the log file and resolve crucial conflicts like yellow-orange.
Afterwards, selecting *Color Table -> Global Exponential Generalization* is possible to choose among 3 types of optimizations:

- **Increase Stability**: can be applied on an already optimized color table, increases the distance among neighbouring color classification regions;
- **Normal**: generalizes the color table to lighting conditions not present in the sample set and resolves some conflicts and calibration errors;
- **High Generalization**: same as Normal, but with a higher generalization factor.

In general, depending on the time spent in the manual classification phase and the generalization factor that is desired to be achieved, Normal or High Generalization are the best choices, while Increase Stability is suitable to clean up an already generalized color table which has been further manually edited.
It's not advisable to perform 2 times Normal or High Generalization on the color table, and it's a good habit to save the "raw" color table before applying any generalization phase; if the result hasn't been satisfactory it's generally best to get back to the raw state and resolve there the conflicts, before applying the generalization again; if the raw table hasn't been saved the *undo* button can be used to get back as well.
The typical running time for the exponential generalization of a ColorTable32K is about 20sec - 1min on a 2.6GHz processor, for a ColorTable64 this has to be multiplied by a factor of 16 (the cost of the algorithm is $O(n^2)$, where n is the number of elements in the color table).



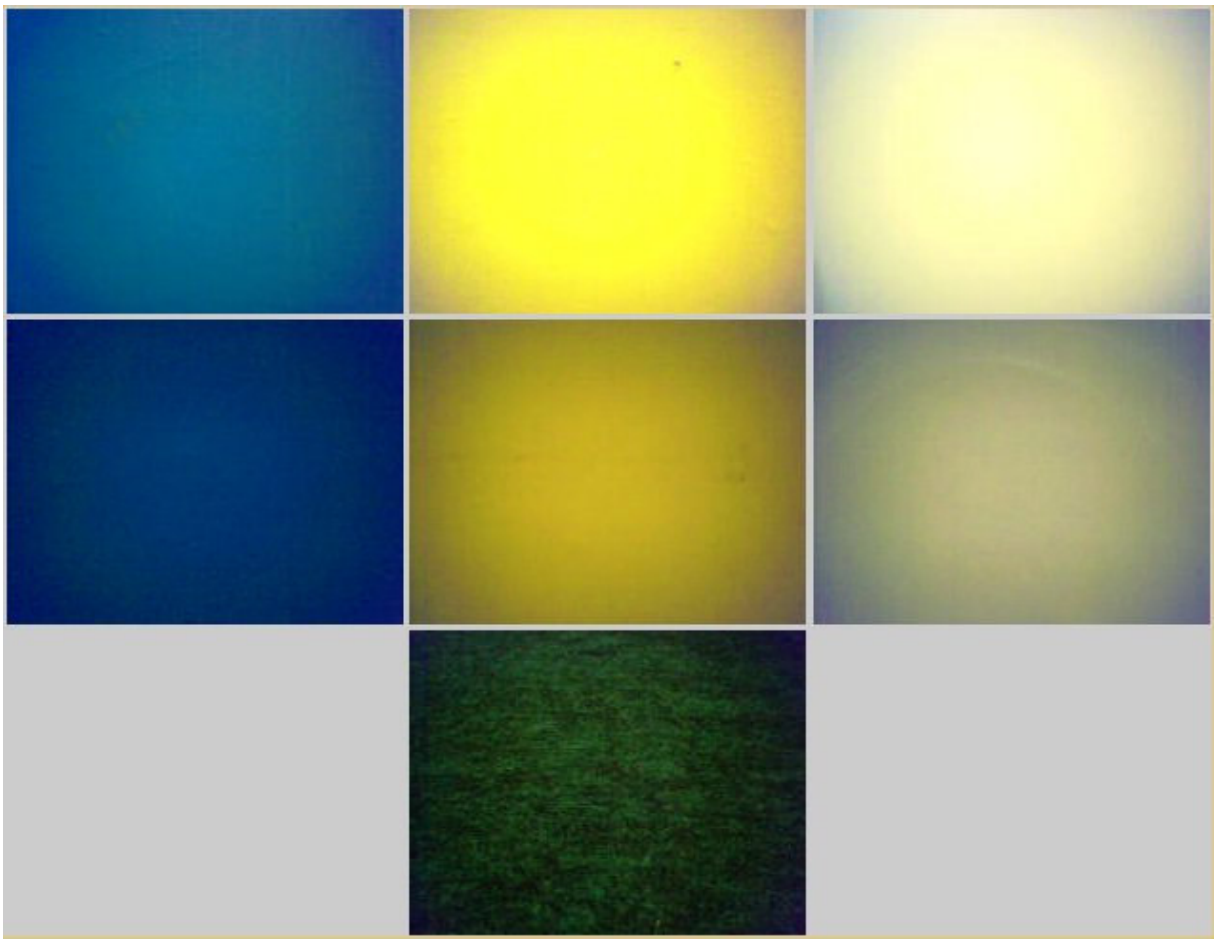
This is the color table before the exponential generalization, in the active area a layer of the color table is represented magnified by a factor of 2.



After applying *High Generalization*, this is the result.

Radiometric calibration (color distortion correction)

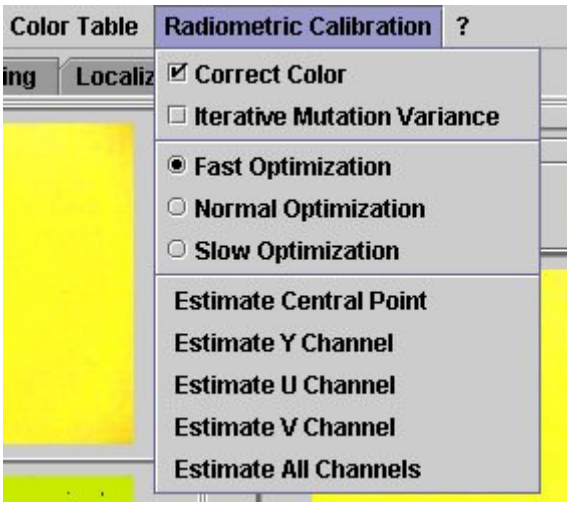
To generate the coefficients for the color correction algorithm, it's necessary a log file which contains only pictures of uniform color: to obtain a high lighting independence, it's advisable to take at least a yellow, a white and a skyblue picture, possibly at different brightness levels (such as 500 and 1000 lux); yellow and skyblue exhibit extreme levels for the 2 color components in the *YUV* color space.



Snapshot of a possible log file used for radiometric calibration.

To help to create such a log file, it's possible to use the *Append AMV* function from the *File* menu: anytime you open a new file this way, it is appended into a temporary file stored on disk; once you have finished this process, just select *Save AMV* from the *File* menu and the temporary file will be copied into a permanent one.
After opening a calibration log file, go through the *Radiometric Calibration* menu and activate one of the 3 radio buttons to select the total optimization time:

- *Fast Optimization* takes about 3 minutes (on a 2.6GHz processor) to calculate the coefficients for the 3 image spectra (*Y*, *U*, *V*), but it's not suitable for an accurate calibration starting from scratch; it's useful for a quick extension of previously generated coefficients, especially if applied iteratively a couple of times;
- *Normal Optimization* takes roughly 40 minutes, and can generate reasonable coefficients in a single run;
- *Slow Optimization* takes more than 8 hours, but calculates completely new accurate coefficients and is suitable for a run overnight.



Chosen the desired optimization time, the final step is activating the estimation of a single channel, all, or the calculation of the central point of distortion, which appears to be slightly dependent on the particular robot used; the estimation of the central point can be performed only after at least one of the color channels coefficients has already been calculated or loaded. During the calculation phase, it's possible to see on the console the current "*energy*" level minimization (which represents the sum of the variances of all pixels compared to the reference values for each color) and the "*temperature*" the annealing process is running at, once it reaches zero the algorithm ends; please notice if the final residual energy is lower than the starting one, if this is not true then the newly computed coefficients are worse than the starting ones (this can happen in case of *Fast Optimization*, starting from highly optimized coefficients).
The *Iterative Mutation Variance* checkbox selection triggers the use of previously computed coefficients as a base for the variance of the random mutation of the new ones; in some cases this can lead to a faster convergence to lower energy levels using iteratively *Fast Optimization*, but it can get stuck around local minima and in general is not recommended for *Slow Optimization*.
The coefficients can be saved individually or all at once, the latter is done by selecting in the *File -> Save -> Save Color Correction Coefficient* dialog as a **file type**: *All Color Distortion Correction Coefficients* and providing a single name.