IDL Obsession (with examples)

Ronn Kling

October 16, 2008

Why Obsession?

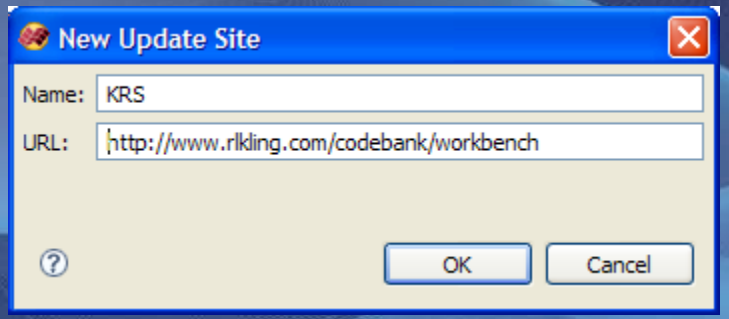
- First, I really do love IDL
  - When I see something my first thought is always “How would I do that in IDL?”
- Second, I like solving problems
- Third, I am a visual thinker
  - I don't think in equations but in visual concepts. Having the ability to see my data displayed is invaluable.
- Fourth, I am always partial to the underdog and IDL is not quite as famous as Matlab.
- Fifth, I have been using IDL for 15 years and I still am amazed at the things we can do.

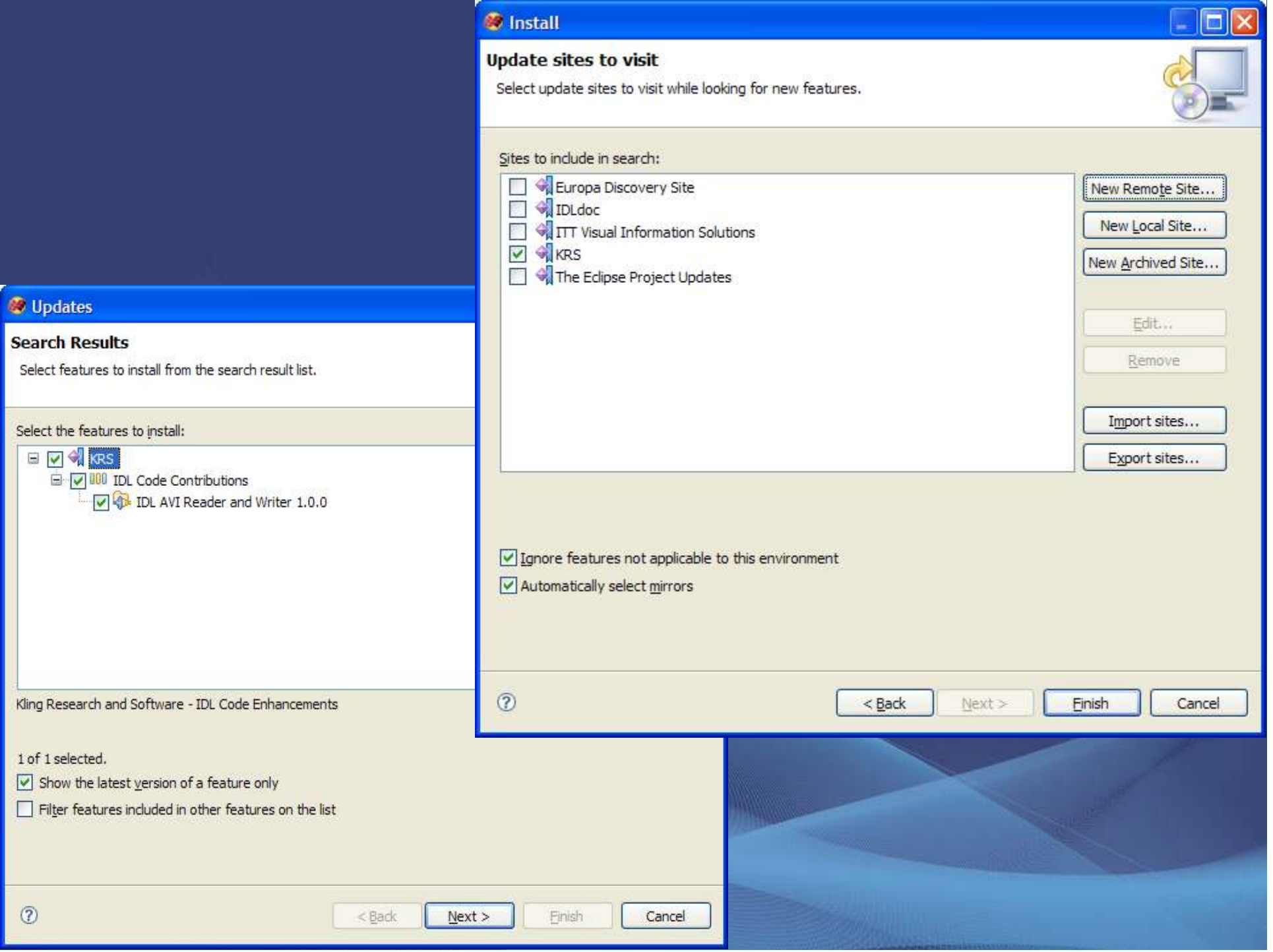
Outline

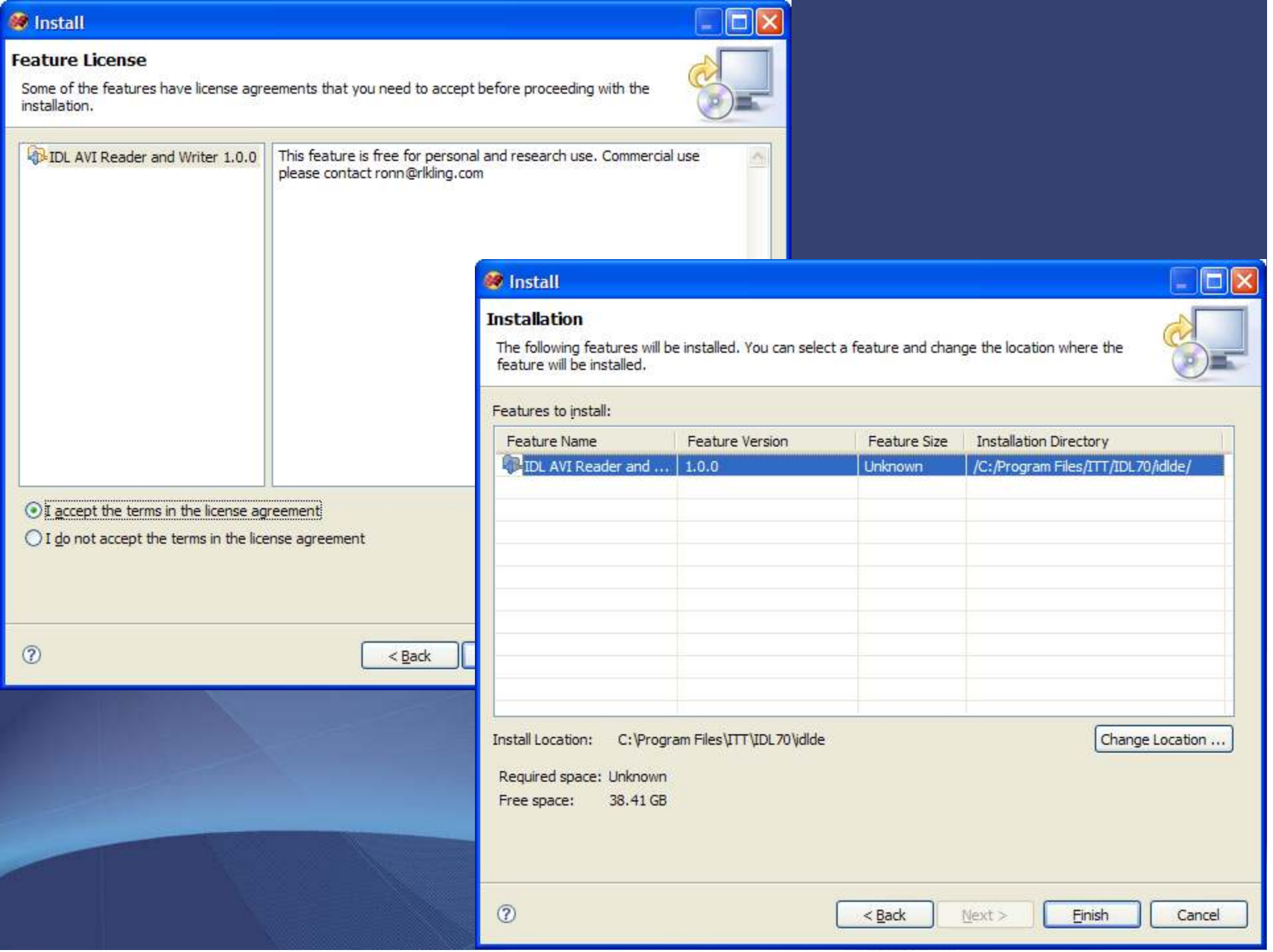
- AVI creation and display
  - New plugin for the IDL workbench
- Workbench
  - My own “special” workbench layout
- N-dimensional Ellipse fitting
  - Code for 2D and 3D with optional weighting
  - Easily extended to higher dimensions
- Point in a convex polygon/polyhedron
  - Completely new (?) way of solving the problem

AVI plugin

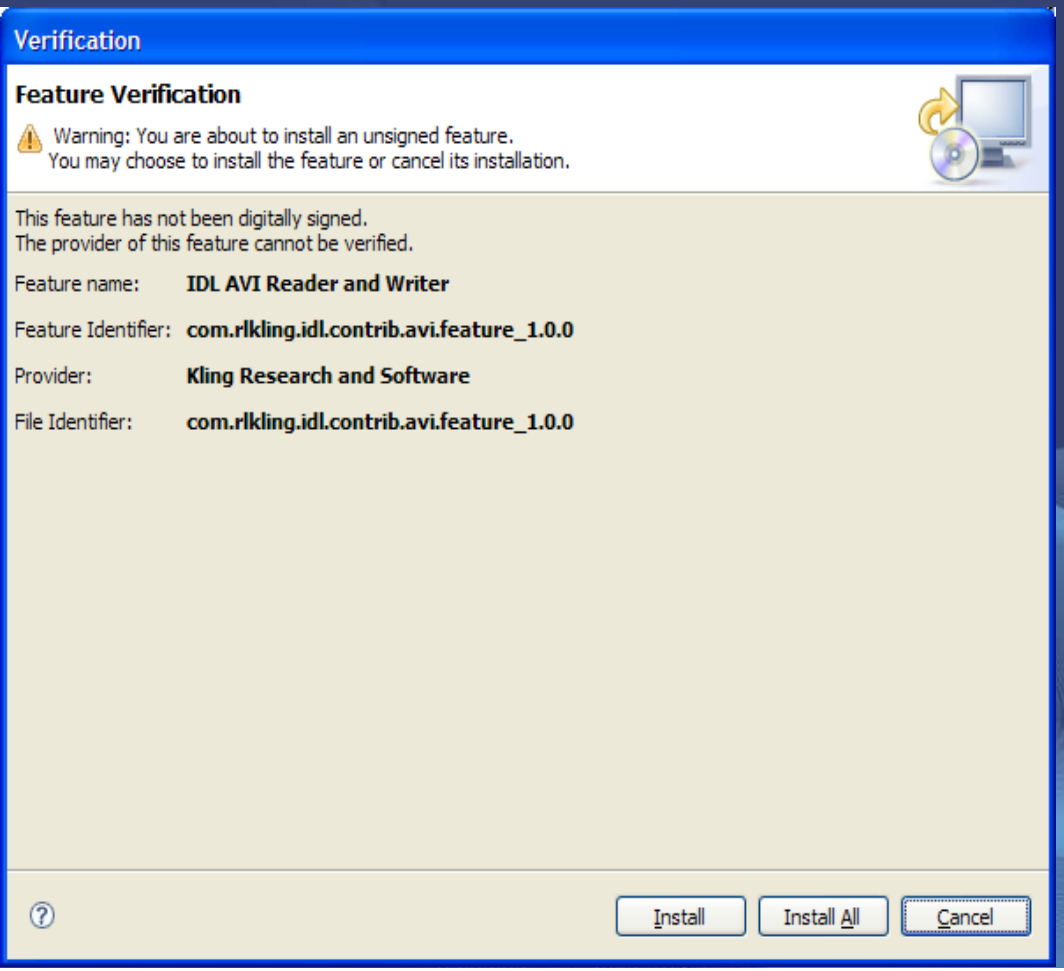
- The most popular DLL I have are the AVI (MPEG, Quicktime) reading/ writing routines
- With the new workbench I created a plugin that make it easy to install.
- Go to Help->Software Updates->find and Install. Then choose “find new updates”
- On the install panel choose New Remote Site and enter this information
- www.rlkling.com/codebank/workbench
-




Install
Update sites to visit
Select update sites to visit while looking for new features.
Sites to include in search:
Europa Discovery Site
IDLdoc
ITT Visual Information Solutions
KRS
The Eclipse Project Updates
New Remote Site...
New Local Site...
New Archived Site...
Edit...
Remove
Import sites...
Export sites...
Ignore features not applicable to this environment
Automatically select mirrors
< Back
Next >
Finish
Cancel
Updates
Search Results
Select features to install from the search result list.
Select the features to install:
KRS
IDL Code Contributions
IDL AVI Reader and Writer 1.0.0
Kling Research and Software - IDL Code Enhancements
1 of 1 selected.
Show the latest version of a feature only
Filter features included in other features on the list
< Back
Next >
Finish
Cancel


Install
Feature License
Some of the features have license agreements that you need to accept before proceeding with the installation.
IDL AVI Reader and Writer 1.0.0
This feature is free for personal and research use. Commercial use please contact ronn@rlkling.com
I accept the terms in the license agreement
I do not accept the terms in the license agreement
< Back
Install
Installation
The following features will be installed. You can select a feature and change the location where the feature will be installed.
Features to install:
Feature Name
Feature Version
Feature Size
Installation Directory
IDL AVI Reader and ...
1.0.0
Unknown
/C:/Program Files/ITT/IDL70/idlde/
Install Location: C:\Program Files\ITT\IDL70\idlde
Change Location ...
Required space: Unknown
Free space: 38.41 GB
< Back
Next >
Finish
Cancel

- Look in C:\Program Files\ITT\IDL70\idlde\plugins\com.rlkling.idl.contrib.avi.windows_1.0.0 for the final installation instructions



Creating an AVI movie

- The syntax for creating an AVI writer is
  - oAVIwriterObj = obj_new('krsGRaviWriter',filename, width, height, codec=codec, framerate=framerate, quality=quality, nBits=nBits, red=red, green=green, blue=blue)
    - Filename must end in an AVI or an error will be generated
    - Codec is the fourcc abbreviation for the compressor/decompressor (codec)
    - If codec is blank then a windows dialog appears with all the codecs available on that machine displayed
- Adding a frame is simple
  - status = oAVIwriterObj->addFrame(image)
    - If the AVI object was created with nBits=8 then image must be a 2D array. If nBits=24 then image must be a 3xMxN array
- The object must be destroyed in order to close the file
  - obj_destroy, oAVIwriterObj

Reading, AVI, MPEG, Quicktime

- The syntax for creating the AVI reader is
  - oAVIreaderObj = obj_new('KRSgrAVIreader', file, width=width, height=height, numberOfFrames=numberOfFrames, nBits=nBits, red=red, green=green, blue=blue)
- If file has either an .mpg or .mov (quicktime) extension then the AVI reader object will attempt to use the avisynth frame server to read the file.
  - Avisynth is a free third party scripting tool for windows and can be downloaded at www.avisynth.org.
  - The user need only install this program and the AVI reader object takes care of all the calls to avisynth.
  - This process is completely transparent to the user.
- To get a frame
  - image = oAVIreaderObj->getFrame(frameNumber)
- The object must be destroyed in order to close the file
  - obj_destroy, oAVIreaderObj

Hybrid Workbench

- Right after 7.0 came out there was a lot of discussion on the newsgroup about changing the layout of the workbench
  - David Fanning has a picture of his layout with this quote
  - “Don't ever use anything but the *Debug Perspective* in the Workbench.”
- Getting the layout right can be very frustrating
- I have copied all of the relevant files (for windows) and have them on my website, **www.kilvarock.com/plugins**
  - This will reset your workbench back to the default state so you will lose any project settings!
  - It will not delete any IDL files, but you will need to recreate the projects again.
  - Once you get the .plugin directory from the site just replace yours in the IDLWorkspace\.metadata directory

N-dimensional Ellipse Fitting with Weighting

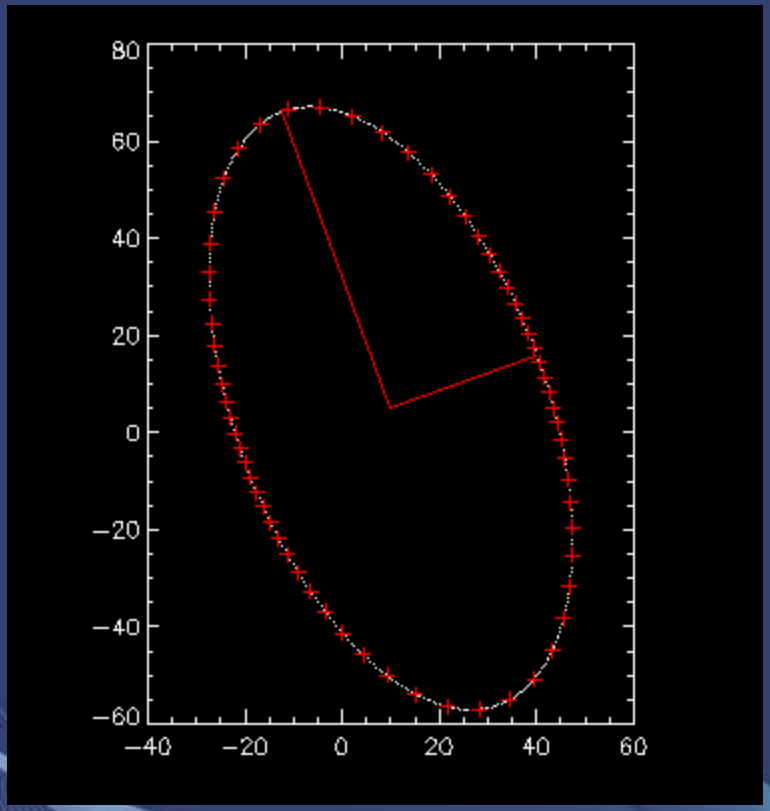
- Very similar to David Fanning's fit_ellipse routine that he first wrote in 2002 and updated in August 2008
- Uses the method of moments and the quadratic form of the ellipse/ellipsoid equation
  - Let a_i = amplitude at i^{th} point
  - $\underline{X}_i$ = vector to the i^{th} point
  - $\underline{X}_c$ = vector to the (weighted) center
  - Then $\underline{X}_c = \sum a_i \underline{X}_i / \sum a_i$ (First weighted moment)
  - Second moment is $P = \sum a_i \underline{dx}_i^T \underline{dx}_i / \sum a_i$ where $\underline{dx}_i = \underline{X}_i - \underline{X}_c$
- As you might expect if we find the eigenvectors and eigenvalues of P we can get the ellipse axes and rotation angle
- But take it one step further first

Ellipse fitting (cont)

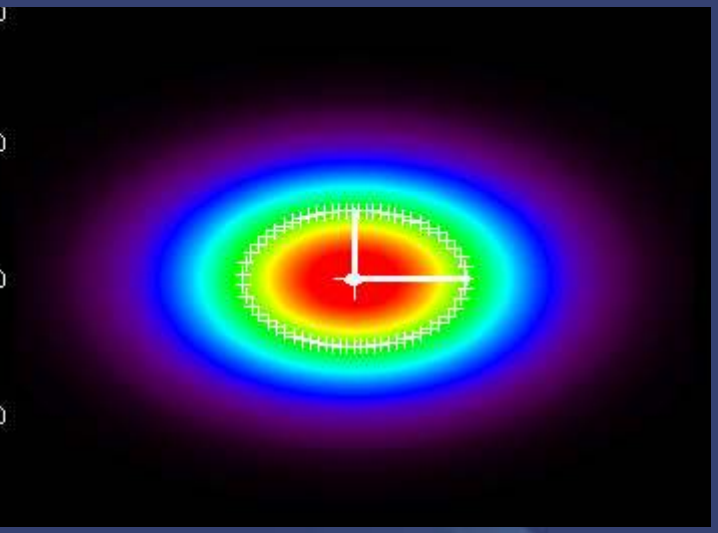
- Quadratic form of the ellipse equation is
  - $|\underline{X} - \underline{X}_c|^T P^{-1} | \underline{X} - \underline{X}_c | = k^2$
    - Independent of coordinate system
    - True for all dimensions
- If the data being fitted is gaussian then k is the sigma factor
  - i.e k=1 is 1 sigma, k=2 is 2 sigma etc.
- If the data is not gaussian then k is still useful
  - Calculate all of the k values from the equation above
  - Using the max ensures that you will encompass all of the points. Picking a value of k that is 90% of the max will create an ellipse that encompasses 90% of the points and so on

Ellipse fitting (cont)

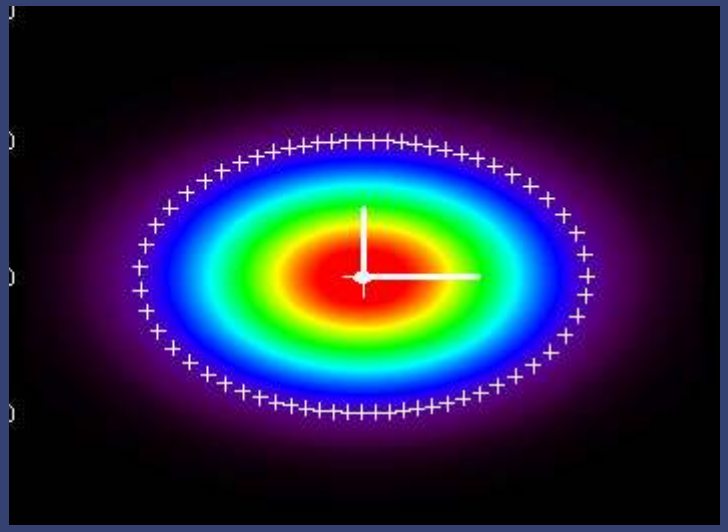
- Once you have chosen a k you can draw the ellipse
- In 2D we let $\underline{X} = r\,\underline{u}$ where $\underline{u} = [\cos(\theta), \sin(\theta)]$ then
  - $r^2 |\,\underline{u}\,|^T P^{-1} |\,\underline{u}\,| = k^2$
  - Or $r = k / (|\,\underline{u}\,|^T P^{-1} |\,\underline{u}\,|)^{1/2}$
  - So $X(\theta) = r\cos(\theta) + \underline{X}_c$ and $Y(\theta) = r\sin(\theta) + \underline{Y}_c$
- In 3D $\underline{u} = [\cos(\theta)*\cos(\varphi), \sin(\theta)*\cos(\varphi), \sin(\varphi)]$



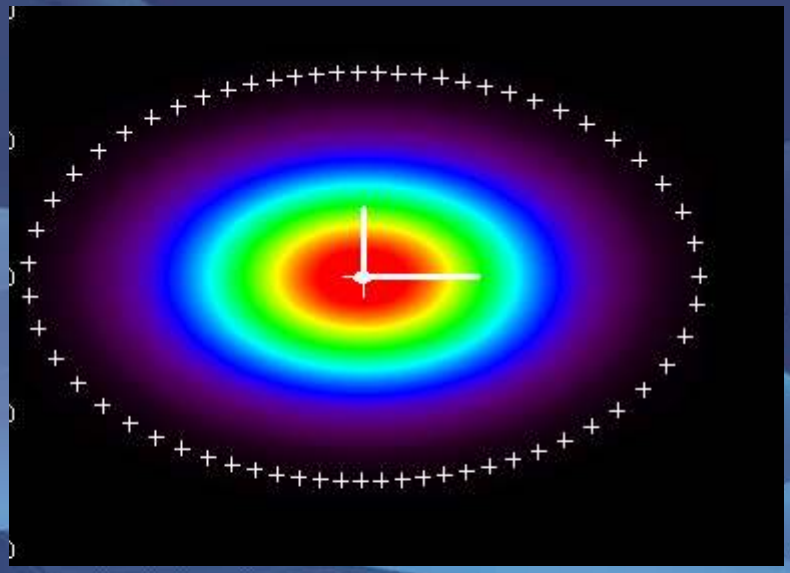
- No Weighting
- White dots are the original data
- Red +'s are the fitted
- Semi-major and semi-minor are from the eigenvectors
  - evals = eigenql(P, eigenvectors = evec)
  - semiMajor = sqrt(evals[0])
  - semiMinor = sqrt(evals[1])
  - angle = atan(evec[1,0],evec[0,0])



K = 1



K = 2

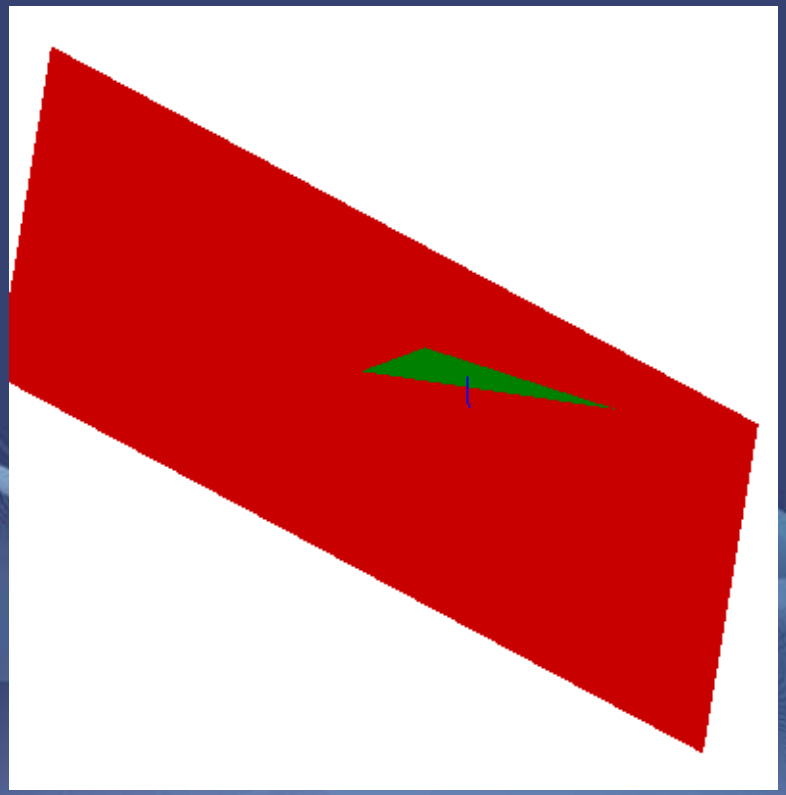


K = 3

Point in a convex polygon

- Many routines and techniques exist to see if a single point is inside a polygon
  - Algorithm is repeated for every testing point and every polygon vertex
- I know of no methods that are optimized to see if multiple points are inside the polygon
- Credit where credit is due. Original idea as applied to triangles was developed by a friend of mine, Jerry Lefever
- Demonstration

The Key



- Turn the 2D problem into a 3D one!
- Every edge intersects a plane with a z value of 1

The Steps for a Single Triangle

- Calculate the distance along each edge segment
- Find the plane that passes through this line with a Z gradient of 1
  - dist1 = sqrt((x2 - x1)^2. + (y2 - y1)^2.)
  - a1 = (y2 - y1)/dist1
  - b1 = -(x2 - x1)/dist1
  - c1 = -a1*x1 - y1*b1
- Find the center of the triangle
  - xc = (x1 + x2 + x3)/3 & yc = (y1 + y2 + y3)/3
- Calculate the distance from the center to the plane, if it is negative then flip the normal and find the new plane
  - Do this so that later any positive distance values will have a certain meaning
- Repeat for other two edges
- When done we return a 3x3 array where each row has the coefficients for the equation of a plane

Steps (cont)

- For every point calculate the distance to each of the three planes. If all three distances are positive then point is inside the pyramid formed by the planes. If not, then it is outside
  - If any distance is negative then you can stop testing immediately
- Inside the triangle this distance is the distance to the closest edge
- Outside it is the perpendicular distance from the plane of the triangle to the closest plane
- Extending this method to any convex polygon is simple
- This also extends to a point in a convex polyhedron
  - Simple example
- Lesson: adding a dimension to a problem may actually make it easier to solve

Summary

- I am an IDL geek and proud of it
- My routines for ellipse fitting and points in a polygon/polyhedra will be available on my website and the user contrib site