Porting the IRAF to x86_64 OS
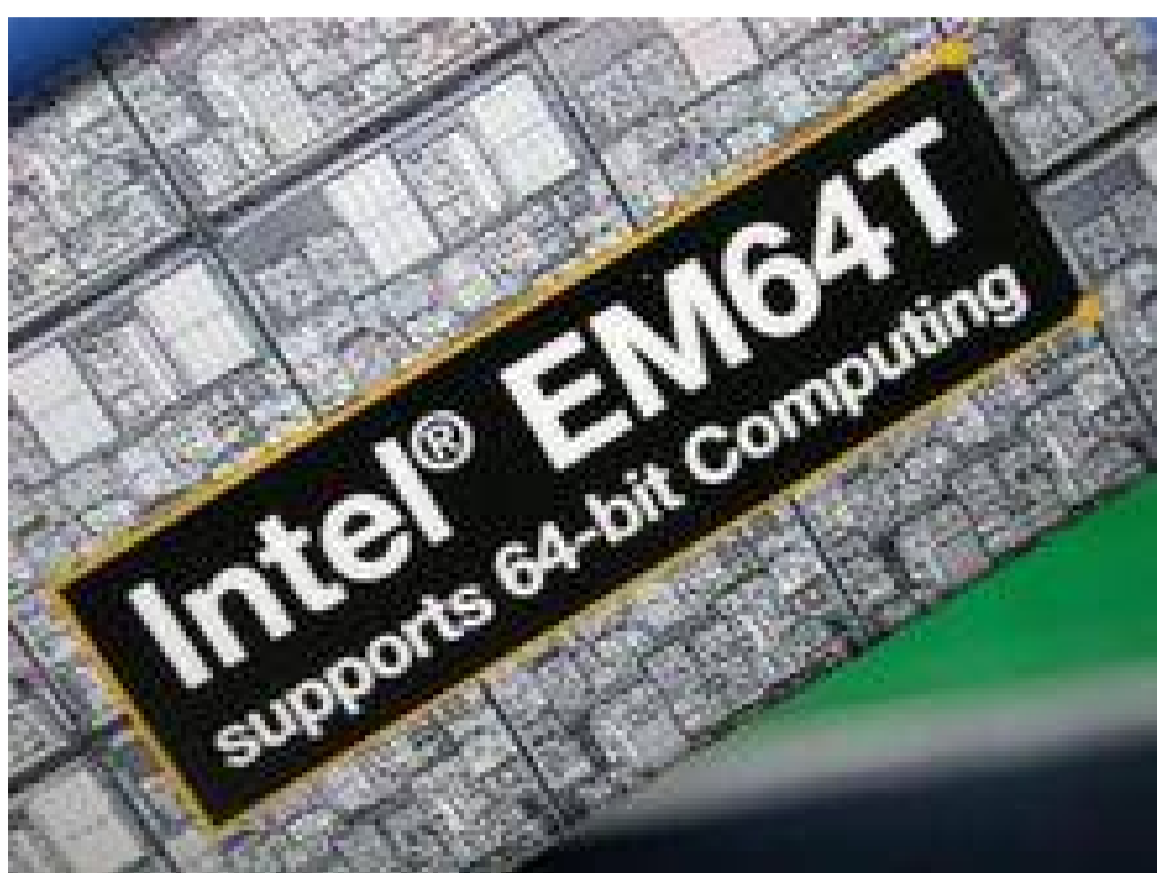
ADASS 2007 LONDON UK

Chisato Yamauchi

cyamauch@ir.isas.jaxa.jp

Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency

IRAF64 | Search

We have started the IRAF64 Project to port the IRAF to native x86_64, and modified necessary codes of the IRAF kernel. As a result, the IRAF software is compiled successfully on the x86_64 native environment (CentOS 4.3) and the xc compiler, mkpkg, cl and basic tasks (e.g. display, imstat, imcombine, phot, etc.) work properly.

Why difficult to port the IRAF to x86_64 OS?

The intrinsic problem is the size of 'long' in C on x86_64 OS (not 4bytes but 8bytes). The xc compiler replaces 'int' of SPP with 'long' of C. However, the IRAF's C code assumes that the size of 'int' in SPP is equal to that of 'int' in C (i.e. This is a potential bug). Therefore, huge modifications are required to C code.

Another problem is that the IRAF requires a small assembler code for each architecture.

Strategy and Current Status

1. Fix the build errors/warnings of gcc: DONE
2. Write an assembler code (zsvjmp.s): DONE
3. Add the x86_64 native support of packaging system: DONE
4. Improve a lot of security issues, and modify ancient style of C code: current work
5. Make the best use of GNU make, and replace the csh scripts with sh scripts. to be done
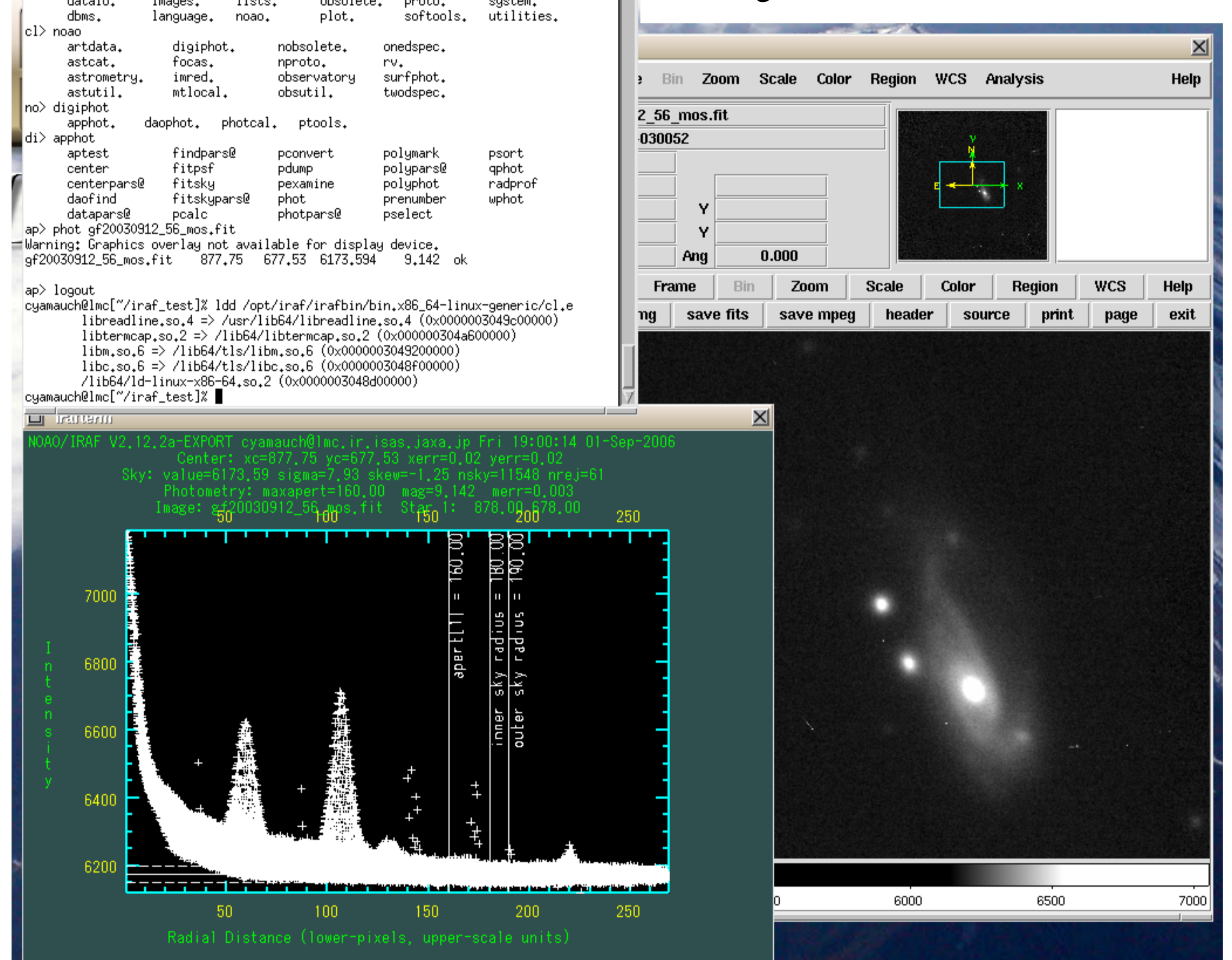6. Fix the SPP codes which cause errors. to be done

Large Magellanic Cloud in 3,7 and 11 micrometers with the Near– and Mid– Infrared Camera (IRC) onbord AKARI satellite.
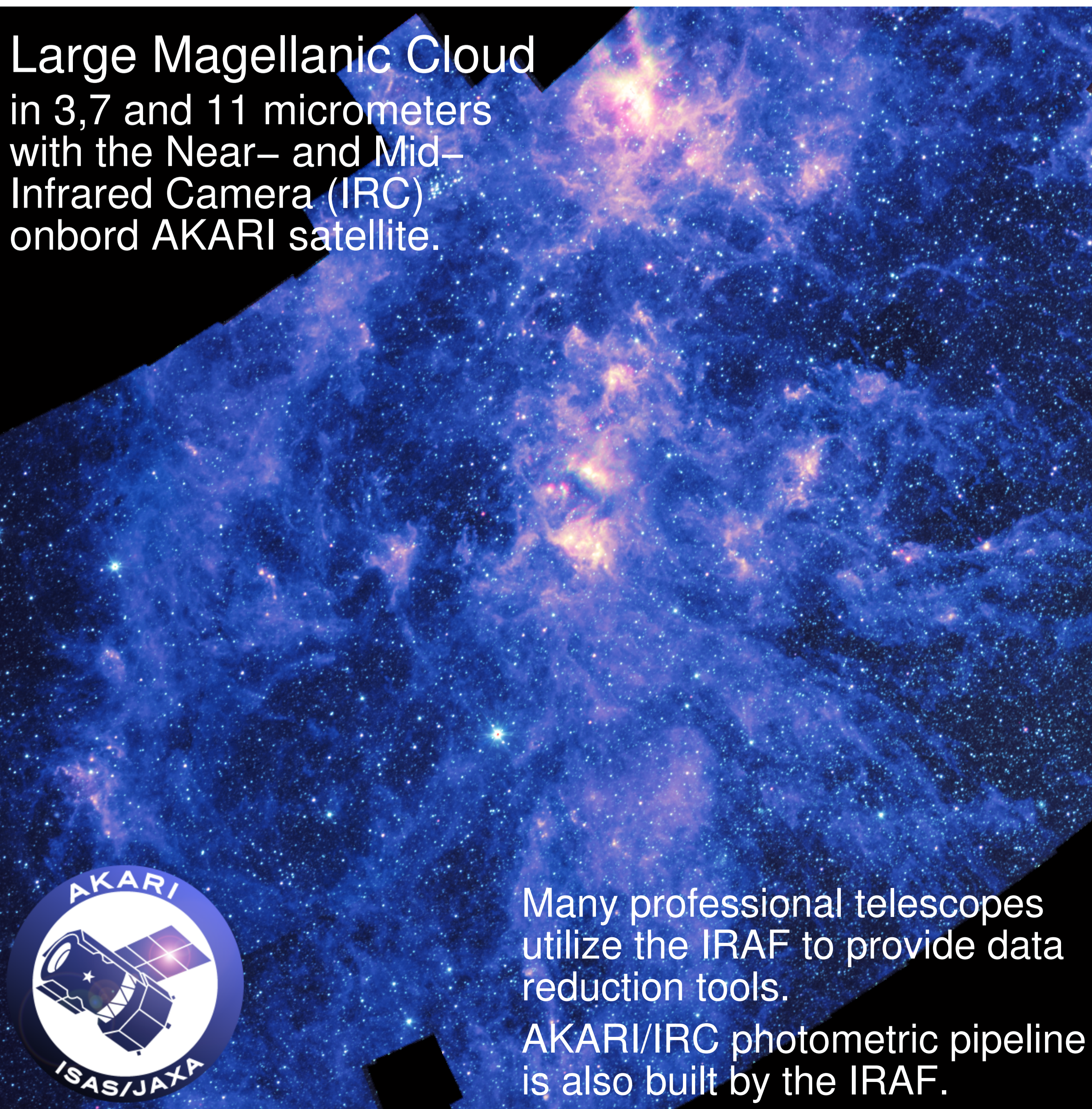
Many professional telescopes utilize the IRAF to provide data reduction tools.
AKARI/IRC photometric pipeline is also built by the IRAF.

The IRAF built for native x86_64 is running on CentOS 4.3.

Technical Details for C code:

– Append '`-Wall`' option to arguments of gcc. The 'incompatible pointer type' warning is indispensable to find '`int *`' (4bytes ptr) args which must be replaced with '`XINT *`' (8bytes ptr; 'int' in the SPP).

– Append complete prototype declarations to remove the potential problems and to meet the present standard.

– Fix the buffer overflow problem of fixed–length array. An argument is appended to public functions to specify the size of buffer (if missing). Make the most of safer APIs.

– Make the best use of '`const char *`' for function arguments to clarify whether input or output.

Technical Details for SPP code:

– `P2R()` macro is appended. This new macro is required to reference a member of REAL type in a struct.
    e.g. **`define PL_WIDTH Memr[P2R($1+2)]`**

Release and Future Plan

We aim to release the first test binary packages by the end of 2007. The source code will be maintained using the Subversion, and the repository will be public. We will concentrate on the problems of each SPP's package after first test release. Your report will be required to increase x86_64–supported packages.