The 2005 LANL/RIPS project statement

The project we are proposing involves the use of low resolution, off-the-shelf camera hardware to build a super-resolution imaging system. Students will be given a pair of computer controlled cameras and a computer controlled platform for changing their position. They will then build software to take the images from both cameras and construct super-resolution images based on the raw image data and information taken from the second motion-correction camera.

The main idea is that what makes or breaks super-resolution is the estimate of motion. Finding a low-cost way to generate a high-quality estimate of motion should permit a significant improvement in the super-resolution results.

Open source software implementing the solutions found by the students will be a main deliverable of the project for the summer.

Outline of tasks and deliverables:

1. Purchase or obtain the following items and software to perform experiments and analysis with:

   - One windows XP pro workstation / laptop.
   - Two Logitech USB web cams. The "Quickcam Pro 4000" seems reasonable.
   - Lego Mindstorms robotics set
   - Extra Lego pieces, super glue and misc. hardware for constructing a way to mount the cameras on the Lego robot.
   - The programming environment for the project will be
     - Matlab 7.0 for algorithm programming and data processing,
     - Java 1.4.2 software development kit (free from http://java.sun.com/), and
     - Java Media Framework 2.0 (also free: http://java.sun.com/products/java-media/jmf/).

     The Java Media Framework provides a simple programming interface to the simple webcams we will use for capturing the data. An example/ tutorial on how to do this is found at :

http://java.sun.com/dev/evangcentral/totallytech/jmf.html

http://ltu164.ltu.edu/itseng/version.htm

Matlab allows external routines to be called from inside Matlab that are written in Java. It should be straightforward once code is written in Java using the JMF to capture frames from the camera, to then take this code and make it callable from Matlab to gather image data.

2. Initial Tasks: The following tasks using the Lego robot, one camera, the Java and Matlab software and some freely available super-resolution software package should help get the project started:

   - Find methods and algorithms already used to do super-resolution from a series of images and try them out (see notes below). Some of these will be proprietary and impossible to get complete details on.
   - Test the methods on data collected from a series of experiments conducted using the camera and camera motion system. Find the limit as the motions between images become large and non-affine, where the super-resolution methods break down.

3. Innovate: a method for using a very low spatial resolution, but high temporal resolution video input to generate a high fidelity estimate of the motion. Use this estimate to produce an improved super-resolution of a sequence of frames with higher spatial, but lower temporal resolution.

   - The challenge here is a lowcost, high-fidelity estimate for motion to improve the super-resolution results. Therefore even though developing your own super-resolution method from scratch is permissible and may be desirable at the latter stages, using what free software you can find to insert your better estimate of motion into will possibly be good enough.
   - Suggestion: that the robot be navigated around a lab environment that is fixed so that the only thing that varies is the two cameras horizontal position and rotation around the vertical access.
   - Note:The motion will be limited by lengths of cables to the cameras, but not that much motion is really necessary given the fact that the scene items can be placed quite close to the robot's path.

- Suggestion: use one camera at the full frame rate of 30 frames a second to collect the very low resolution (say 5x5 to 10x10) "image" and use this to estimate the motion in the sequence of images taken with the other camera at 1 to a few frames per second.

4. Extra Challenge: Concoct your own super-resolution method to work with your motion estimate from the task above.

5. Deliver Code : Open source implementations of existing methods as well as the new methods the projects generates.

6. Deliver Reports: 1) users manual for the code as well as 2) report documenting the project.

Notes:

Though we can't get a license for his software, you should find the paper from Milanfar's group at UC Santa Cruz useful: it is titled "Fast and Robust Multiframe Super Resolution" by Farsiu et al on Milanfar's webpage:

http://www.cse.ucsc.edu/~milanfar/

Some free software for doing super-resolution can be found here:

http://und.edu/instruct/young/toast/Super-Resolution.html

http://auricle.dyndns.org/ALE/

Your LANL points of contact for the project will be Kevin R. Vixie (vixie@lanl.gov), Matt Sottile (matt@lanl.gov) and Brendt Wohlberg (brendt@lanl.gov). We will be available for consultation and visits to UCLA. We will also be at IPAM for much of July during the IPAM graduate summer school.

All of the above research mathematicians and scientists are on the "Data Driven Modeling and Analysis" (DDMA) team at LANL. Our webpage can be found at:

http://ddma.lanl.gov