Automatic Site Construction, Extension, Detection and Refinement

The ASCENDER System v1.0 Users Guide

RADIUS Working Group

Computer Vision Research Laboratory
University of Massachusetts, Amherst

April 14, 1995

1 Installing ASCENDER

The ASCENDER system is delivered as a compressed archive file named **ASCENDER.tar.Z**. This archive contains the 'C' executable code, lisp files, and data files necessary for running the ASCENDER system within RCDE. The tar file should be installed in the $CME/code/CME-5d directory. The $CME variable determines the root directory of the CME system and may be different for each site. If $CME is undefined on your system, consult the "Installation Guide" document that is delivered along with RCDE.

Copy the **ASCENDER.tar.Z** file into $CME/code/CME-5d directory and uncompress the file:

```
tsh> uncompress ASCENDER.tar.Z
tsh>
```

The directory should now contain, **ASCENDER.tar**. When this file is unarchived, it will create all of the necessary subdirectories and files for the ASCENDER system. In order to execute this command, the user must have write privileges in the $CME/code/CME-5d directory. Type:

```
tsh> tar xvf ASCENDER.tar
```

When complete a subdirectory **UMass** will have been created that contains the ASCENDER system files. If this is not the case, there may be a permission problem within the directory. If the installation fails consult the Help section in this document.

Once the code has been installed on the local system the directory that ASCENDER will use for temporary disk storage needs to be setup. Change the global lisp variable **user::*temp-directory*** to be the local directory that will be available for temporary storage. This variable can be found in the system startup file:

$CME/code/CME-5d/UMass/umass-system.lisp

2 Preparing a Site

This document assumes that the user is familiar with the RCDE system and that RCDE is running on the user's machine. The user should also have a working familiarity with LISP and have access to the RCDE-LISP window.

In order to load the ASCENDER system into RCDE, the user should load the system startup file into LISP. Within the LISP window type:

```
> (load "$CME/code/CME-5d/UMass/umass-system.lisp")
```

This initalizes all LISP and 'C' code and creates the *UMass* menu in RCDE. In order to automatically include ASCENDER as part of RCDE, "umass-system.lisp" can be loaded from the file, `$CME/RCDE_2.0/local/cme-site-init.lisp`. Thereafter, when RCDE is loaded, it will include the ASCENDER system.

Once the ASCENDER system has been loaded, the user can load a site model and use RCDE as normal. However, in order to access the ASCENDER functions, feature sets must be loaded and the site must be initalized.

The required steps in preparing a site for use with ASCENDER are shown below. The final step "Initalize site model" can be performed by simply executing *Initalize Site Model* from the *UMass* menu. Loading "site specific feature sets" is a bit more involved but details about saving and loading site features can be found in section 3.

ASCENDER Startup Procedure

- Load the RCDE system.
- Execute (load "$CME/code/CME-5d/UMass/umass-system.lisp") .
- Load a site model.
- Load site specific feature sets.
- Initalize site model.

3 Site Features

Automated site construction makes use of special features that may have been previously extracted from an image (see Line Segment Extraction). Before extracting building rooftops, the proper line features must be associated with each image in the site model by loading them from file. Likewise, after new features have been detected (buildings, for example) they can be saved to a feature set file for permanency.

3.1 Line Segment Extraction

To run boldt lines on an image

Put the following environment variable in your initialization files

```
setenv BOLDTDIR $CME/code/CME-5d/UMass/Boldt
```

To run boldt lines on an image, lets call it “image.g0”, do the following steps:

1. convert rcde g0 image to llvs plane file:
   `$BOLDTDIR/g0_to_plane image.g0 image.plane`
2. run boldt lines:
   `$BOLDTDIR/bigboldt image.plane label minlength mincontrast partitionwidth`
   where image.plane is the name of the plane file label is a label that will be used to generate filenames minlength is the minimum length of lines that will be produced (generally not less than 5 pixels) mincontrast is the minimum contrast of lines that will be produced (generally in the range 5-10). and partitionwidth tells how the image will be broken up into pieces

for example: `$BOLDTDIR/bigboldt image.plane image 5 10 200`

says to run boldt lines on the file image.plane, to call the output line file image.asc (because image is the label that is specified), filter output lines on length ≥ 5, contrast ≥ 10, and to use a partition width of 200 pixels. The Boldt algorithm cannot run directly on very large images. Instead, the "bigboldt" algorithm chops the image up into smaller pieces, runs the original boldt algorithm on each piece, then merges the resulting lines from all pieces back together. The partitionwidth parameter tells how big each piece should be (in our example, a 200x200 chunk of pixels). The optimal size of partitionwidth depends on your system's resources. If it is too big, the boldt algorithm can fail and you will be missing a chunk of lines from your image. If this happens (it is very noticeable), try rerunning bigboldt with a smaller value for partitionwidth.

When bigboldt is done, you will see two new files in your directory, which in our example will be image.asc and image.log. These are an ascii file of line segments produced, and a log of the boldt batch job. The lines can be examined in RCDE using the *Install Image Lines* option of the *UMass* pulldown menu. If they are acceptable, the image.plane file (created by g0_to_plane) is no longer needed and can be deleted.

The format of the ascii line file *.asc is the following:

```
3D: NO      - header information
DATA:       - header information
x1 y1 x2 y2 contrast    - line endpoints and contrast value
.
.
.
```

If you would like to run the UMass building detection system using some other set of line segments (produced by another algorithm) you can do it by formatting the lines in the above ascii file format, then doing *Install Image Lines* using that file.

3.2 Installing Line Features

The line segments for a single image can be loaded by selecting *Install Lines* from the *UMass* menu. The user will then be prompted to select an image and to type the full pathname of the feature set file.

Once the user clicks on the "Do it" button the system will load features from the file and associate them with the clicked image. The token set will be displayed on the image after it has been loaded. Display of the lines can be toggled as though it is any other feature set. The *View* menu controls the display of feature sets within the current site.

3.3 Saving line feature sets

After installing line features for several images, they can be stored as a block for fast loading later on (see next section). This is invoked using the *Save Line Feature Sets* option on the UMass menu.

3.4 Loading line feature sets

Line feature sets for an entire site can be loaded by selecting *Load Feature Sets* from the *UMass* menu. The interface is the same as when loading the line features into a single image. Select an image that displays the site and type the complete path name of the file that contains the feature set information. When the "Do it" button is clicked, all feature sets will be loaded into the associated images and displayed. Because the size of a feature set for a single image is often large, this process can take some time.

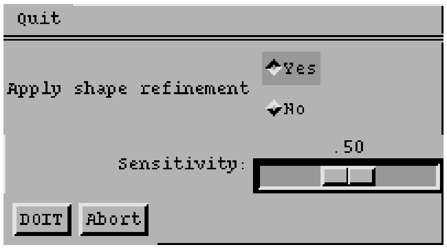
4 Detecting Building Rooftop Hypotheses

Building detection can either be performed on a particular image region or over an entire image. The detector uses the current line feature set and attempts to detect polygons that aid in the construction of the geometric site model.

The building detector requires the user to select a particular image of the site in which to search for building polygons. If the associated line feature set contains three dimensional information, then the detector will produce three dimensional models of building hypotheses. These models will be stored in a three dimensional feature set named "UMass 3d Buildings".

If the line features contain only two dimensional information, the detector will produce a set of two dimensional polygons that denote rooftop boundaries. The two dimensional polygons for a single image are stored in a two dimensional feature set "UMass 2d Buildings" associated with that image.

In order to perform building detection over an image region, select *Detect Building Rooftops* from the UMass menu. A window will popup that allows the user to set the building detector sensitvity and determine if shape refinement will be used. The popup menu is shown in figure 4



Apply shape refinement ◆Yes ◇No

Sensitivity: .50

The sensitivity can be set from 0.0 to 1.0. Default sensitivity is 0.5 and is probably a good setting for many images. At maximum sensitivity (1.0) the building finder will produce polygons that are supported by weak image evidence. Higher sensitivity is usefull for darker regions of the image. If the output of the building detector will be verified in three dimensions, it is usefull to set a higher sensitivity in order to avoid false negatives. Likewise, a low sensitivity will eliminate problems with false positives.

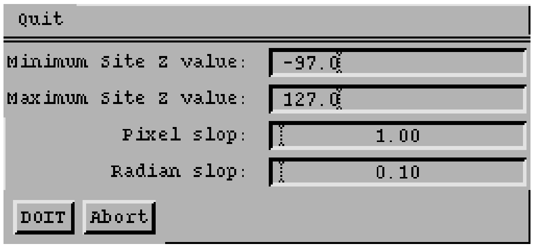
The building detector is able to use the known camera position for a particular image and the expected angles within a building to refine the final shape of the polygons through a least squares method. To activate this feature, click "Yes" next to the "Apply shape refinement" field on the popup window.

After setting the detection parameters, click on the "Do it" button. The system will ask the user to click the lower left and upper right corners of a bounding box. Detection takes place within the image that has been clicked and is limited to the defined bounding box.

5 Epipolar Rooftop Matching and Triangulation

After detecting a building rooftop in one image, a facility is provided for automatically gathering corroborating evidence from other views of the site. The epipolar matching facility is invoked by choosing *Match/Triangulate Rooftops* from the UMass pulldown menu. The results from epipolar matching are a set of corresponding rooftop edges in multiple images, an estimate of the height of the rooftop in the scene, and an initial estimate of the 3D building wireframe structure, which is entered as an object in the "UMass 3D Buildings" feature set.

After matching, the corresponding rooftop edges and initial building wireframe are send to a precise, multi-image triangulation routine that determines a more accurate 3D wireframe structure. This final estimate is also placed in "UMass 3D Buildings" for comparison with the initial estimate.

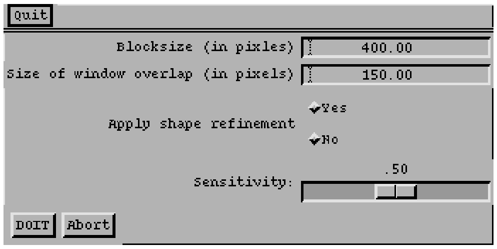


Minimum Site Z value: -97.0
Maximum Site Z value: 127.0
Pixel slop: 1.00
Radian slop: 0.10

6 Automatic Site Acquistion

Rather than search for building rooftops over a subregion of the image, ASCENDER is able to automatically search for all possible building rooftops in an entire image.

A set of two dimensional rooftop hypotheses for a single image are created by selecting *Acquire Site Model* from the *UMass* menu. The user is then required to supply values for regionsize and overlap. Given these values, the building detector is iterativly run over all the subregion in the image of size **`regionsize`**, with an overlap size of **`overlap`** pixels. As in the case of a single region, the sensitivity value and shape refinement parameters must be set. The popup window for *Acquire Site Model* is shown in figure 6.



Blocksize (in pixles) 400.00
Size of window overlap (in pixels) 150.00
Apply shape refinement ◆Yes ◆No
Sensitivity: .50

In order to generate a three dimensional site model, the user can use the rooftop hypotheses generated by *Acquire Site Model* as input to the *Match/Triangulate Rooftops* function. Currently, then, acquisition of an entire model must be performed in two steps. Future revisions of ASCENDER will perform the detection, matching, and triangulation of rooftops within *Acquire Model* in order to produce a three dimensional site model in a single step.

7 Help

The UMass RADIUS Working Group is able provide assistance if there is trouble installing ASCENDER or if there are problems executing the code. Contact information for the members of the UMass RADIUS Working Group is given below.

Allen Hanson	(413)-545-2746	Principal Investigator
Robert Collins	(413)-545-3482 / (413)-545-2746	Technical Director
Christopher Jaynes	(413)-545-0528 / (413)-545-2746	Research Assistant
Frank Stolle	(413)-545-0528	Research Assistant
Yong-Qing Cheng	(413)-545-0528 / (413)-545-2746	Research Assistant
Xiaoguang Wang	(413)-545-0528	Research Assistant

The ASCENDER system is part of an active research area for the RADIUS working group at the UMass Computer Vision Laboratory. Any bugs, suggestions, and requests can be sent to:

ascend@cicero.cs.umass.edu

Any feedback will be greatly appreciated and considered for the next release of ASCENDER.