SUMMER VISION PROJECT
INTERNAL MEMO No. 103

TO: Summer Vision Group
FROM: Gerald Jay Sussman and Adolfo Guzman
DATE: July 20, 1966

A Quick Look at Some of Our Programs

REGIONS1
Sussman

Input:
1. An array filled with numbers, which are intensities read from the vidisector.
2. A point (inside the array).
3. A predicate, which defines a region.

Output:
A list of two things:
1. Number of points on the boundary.
2. Unsorted list of boundary points. (A point is a list of two numbers, namely their bi-dimensional coordinates.)
3. Also, marks in the array the points belonging to the region found (with a - sign).

Purpose: Given a starting point, finds a region around it satisfying a given predicate. Marks this region, and returns its boundary.
Status: 1. Running.

2. Tested on made-up data.

Availability: In VISION LIBRARY tape in REGI SUSMAN (LISP s-expressions).

In VISION LIBRARY tape in REGI SUBRS (lap).

ARESET

Given a, b, and a given region, inside an array, ARESET resets the negative points inside the box generated by REGIONS1 and returns the number of points reset (to + again).

Status: Running.

Tested on dummy data.

Availability: On VISION LIBRARY tape in REGI SUSMAN.

On VISION LIBRARY tape in REGI SUBRS.
RANDOM UTILITY Programs for Arrays.

**FILL**  Fills an array with the elements of a list.

**PRINARRAY:** Outputs an array, from LISP, on teletype.

The format is:

(R1)
(R2)
(R3)

.
.
.

Availability: On VISION LIBRARY tape in REGI Sussman.

On VISION LIBRARY tape in REGI SUBRS.

**VDTAPE**

Reads and writes vidisector input on tape.

**DUMPVD**

Input:
1. Starting point on vidisector
2. Number of points on vidisector to be dumped in x and y directions (size of sample, i.e. m x n).
3. Density of sampling.

![Vidisector screen diagram]
Output: 1. Dumps area of vidisector specified on μ-tape.

Purpose: To make reference pictures of objects which can be used reproducably.

READPICT

Input: 1. Name of array to be defined.
2. Tape number to be read.
3. Name of file to be read.

Output: 1. List of specifications of picture when dumped by DUMPVD
(i.e. starting point; number of points in x, y; density).

Effect: Defines an array with name given in input and fills it with
picture dumped by DUMPVD.

Status: Running.
Tested on real data.

Availability: VISION SYSTEM as VDTAPE SUSMAN

VIDI DUMPS

A collection of pictures is beginning to be formed on microtape; this
tape contains several objects which were seen by the vidisector and
written with VDTAPE. They consist of a big collection of integers which
represent the light intensity at a given point.

Sussman tape #5 currently contains

WHITE CUBE -- a cube
CYLIND WOOD -- a wooden cylinder
CLOTH COUCH -- a piece of Seymour's couch.
SORTBOUND

Sussman

Experimental version of function for ordering boundary returned by

REGIONS1.

Input: 1. Array before reset by ARESET.

2. List of boundary points.

Output: 1. A list of segments.

Availability: On VISION LIBRARY tape as BDSORT SUSMAN.

SAL

Sussman

Experimental version of a scope assembly language, embedded in LISP.

Input: 1. List of macro-commands to scope, e.g.

   (PARAM (SCALE 2))

   (POINT (3 5) (4 6) (200 30))

   (VECTOR (25 60) (700 1005) (720 30)

   (CHAR (30 50) WHAT'S THIS HACK)

   (INCRE (100 250) (+ 0) (0 -) (- 0) (- -) (- -))

2. Name of display.

Output: 1. Array having name given as input with contents as scope

   words to be displayed.

Status: All implemented features working.

Main function is SAL.

Availability: On VISION LIBRARY tape as SAL SUSMAN.
CONVERT

This is a language for symbolic manipulation, imbedded in LISP.

Status: Running.

Tested (not exhaustively).

Availability: In VISION SYSTEM as CONVRT GUZMAN (dump of LISP + CONVRT).

DT

Input: 1. A scene, described by its regions, in symbolic form.

   Each region is an atom with special properties in its
   property list: shape, color, neighbors, etc.

2. A model of an object, also in symbolic form with
   properties, etc.

Output: 1. Prints all the objects in the scene which match the
   model.

2. Returns as value the remainder of the scene.

The analysis which DT does at the present time is relatively unsophisticated; it does not recognize (yet) partially occulted objects.

Status: Running.

Tested in a couple of simple (simulated) scenes.

Availability: In VISION LIBRARY tape as DT GUZMAN.
EXEC1

Input:
1. A list of predicates.
2. A number of boxes.
3. An array.

Output:
1. Divides the array (the world, the scene) into boxes.
2. Applies REGIONS1, with each of the predicates, to the central point of each box.
3. Finds all the regions.
4. Returns a list of three elements.
   The first is a list of lists of regions for each box.
   The second is the number of regions found.
   The third is a list of regions found and the predicates that each satisfies.
5. This is the driver for REGIONS1.

Status: Running.
Tested on dummy data.
No problems in its interface with REGIONS1.

Availability: On VISION LIBRARY tape as EXEC1 LAMPET.

DISP

Displays the regions found by EXEC1.

Functions: INITDISPLAY sets up display arrays.
            REGDISPLAY displays region defined by region number.
            PREDISP displays region defined by predicate.
Status: Running.

Availability: In VISION LIBRARY as DISP LAMFRIT 9.

QFUNSL

Quality functions.

Measures:
1. Grooviness (GROOVES).
2. Average intensity (AVGINT).

GROOVES

Input:
1. Name of array (SPACE).
2. Point about which grooviness is to be measured (x, y).
3. Portion of array to be examined (XMIN, YMIN, XMAX, YMAX).

Output:
1. A list of the density of the grooves about the given point in the examined portion of the array, and their orientation (sin θ · cos θ).
   (density = 1/period of grooves)

Status: Running.

Tested on real data.

AVGINT

Input:
1. Name of intensity array (SPACE).
2. Portion of array to be examined (XMIN, YMIN, XMAX, YMAX).

Output:
1. A number: the average intensity of the examined portion of the array.

Status: Running.

Tested on real data.
ANGSEG

Input:
1. An ordered list of points.
2. The angles between them.

This list is a boundary found by EXEC1.

Output:
1. An analysis of the region, specifying the concave and convex parts, the direction in which the given region was traversed, and its area. (Additional information may be easily added.)

Each one of the parts is simply specified by a list of its points; terminal points may or may not be repeated.

Example:

```
1 2 3 4 7 8 10 11
5 6 9
13 12
```

will be described as `[[ 10 11 12 13 1 2 3 ] [ 3 4 ] [ 4 5 6 7 ] [ 7 8 ] [ 8 9 10 ]]` where we see that convex and concave parts alternate.

Status: Coded.
CS-TR Scanning Project
Document Control Form

Report # A1m-103

Each of the following should be identified by a checkmark:
Originating Department:
☒ Artificial Intelligence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:
☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other: _________________

Document Information

Number of pages: 9 (12-Images)
Not to include DOD forms, printer instructions, etc... original pages only.

Intended to be printed as:
☒ Single-sided or
☐ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☐ Laser Print
☒ InkJet Printer ☐ Unknown ☒ Other: MEMOGRAPH

Check each if included with document:
☐ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _________________

Page Data:

Blank Pages (by page number): _________________

Photographs/Tonal Material (by page number): _________________

Other (note description/page number):
Description: ___________________________
Page Number: _______________________

Images Map: (1-9) Unthe End Title Page, 2-9
(9-13) Scancontrol, TRGTS 2

Scanning Agent Signoff:
Date Received: 12/14/95 Date Scanned: 1/23/96 Date Returned: 1/25/96

Scanning Agent Signature: Michael W. Cook

Rev 8/94 D5/LCS Document Control Form castform.vsd
Scanning Agent Identification Target

Scanning of this document was supported in part by the Corporation for National Research Initiatives, using funds from the Advanced Research Projects Agency of the United states Government under Grant: MDA972-92-J1029.

The scanning agent for this project was the Document Services department of the M.I.T Libraries. Technical support for this project was also provided by the M.I.T. Laboratory for Computer Sciences.

Scanned
Date: 1/23/1996
M.I.T. Libraries
Document Services