

COLUMBIA UNIVERSITY
The Fu-Foundation School of Engineering and Applied Science

YAIL

Yet Another Image-processing Language

Andrew Kisch, Aniket Phatak, Pranay Prabhakar, Uday Chandrasen
(aik2113) (avp2110) (ppp2113) (uc2124)

Prof. Stephen A Edwards

COMS W4115 - Fall 2010

Proposal for YAIL: Yet Another Image-processing Language.

Introduction and Motivation:

Image processing has been an integral part of computer science ever since signal processing and digitizing analog data became possible. Digitized images lend themselves perfectly to classical applications of Mathematical Transformations like Fourier, Laplace et al.

The applications of this fascinating field include Medical Imaging, Computer Vision, Biometrics, Computer Animation, Digital Cameras, Remote Sensing, Entertainment etc. Also, the fields of Artificial Intelligence like pattern matching etc. have found a new application with images. Moreover, we now live in a world where new art seems to be judged less and less on content and increasingly on groundbreaking originality. The ability to easily customize photo filters would appear to be an extremely powerful tool.

The entire process of Image Processing involves many stages, right from image acquisition, sampling, storing to processing and rendering. We are motivated to work on that part of Image Processing which involves the application of various transforms and computations in order to have the desired effect on a digital image.

Objectives:

The main goal of our language will be to facilitate ease of access to already digitized images and provide simple interfaces to manipulate the digital data. More specifically, the language would enable users to do routine tasks like fetch, crop and resize an image.

YAIL will also aim at providing an ability to write custom kernels/filters in order to process the images as per the user requirements. Writing kernels for operations like gray-scaling, blurring, sharpening etc. should be an easy process with this language.

YAIL will provide support for basic primitive data types like integers, floats, doubles and strings. Apart from the primitives the language will be able to define objects to hold images, filters and other user defined structures.

Example Syntax:

1. Fetch an image from an existing directory by instantiating object.

```
Image img = new Image(Path to the image file);
```

2. Declare primitive types.

Type Identifier VariableName

```
int foo;  
double bar = value;  
char a = character_literal;  
char [ ] a = "string";  
pixel pix_a = [x,y]; //(x,y) are the x-y co-ordinates  
color color_a = [x1,y1,z1]; //(x1,y1,z1) are the RGB values
```

Additional Functionality:

1. User defined data types like arrays and structures.
2. We also intend to provide in-built functions like to zoom and resize.
3. All expressions and statements will be terminated in C style, i.e with a semicolon.
4. Iterative control structures like WHILE, FOR and DO WHILE will be available.
5. Conditional control structures like IF and ELSE will be available.
6. All arithmetic operators would be available for primitive types with the general precedence and associativity of C.