


PICEL

PICture Editing Language

Project Manager	System Architect	System Architect	Language Guru	Testing Jedi
Chia-Hao Hsu	Chih-Sheng Wang	Ruijie Zhang	Rui Lu	Chang Liu

What's Cool?

Primitive type for picture

```
pic foo = load("bar.bmp");
```

R/W RGB values in a single statement

```
foo.r[1][1] = 255;
```

Shared/separated buffer pictures

```
pic baz = foo; /* shared buffer */

copy_pic(foo, baz) /* separated buffer */
```

Special syntax for convolution

```
mat kernel = {0, 1, 0, … 2, 0};

foo #kernel;
```

What about multiple convolution?

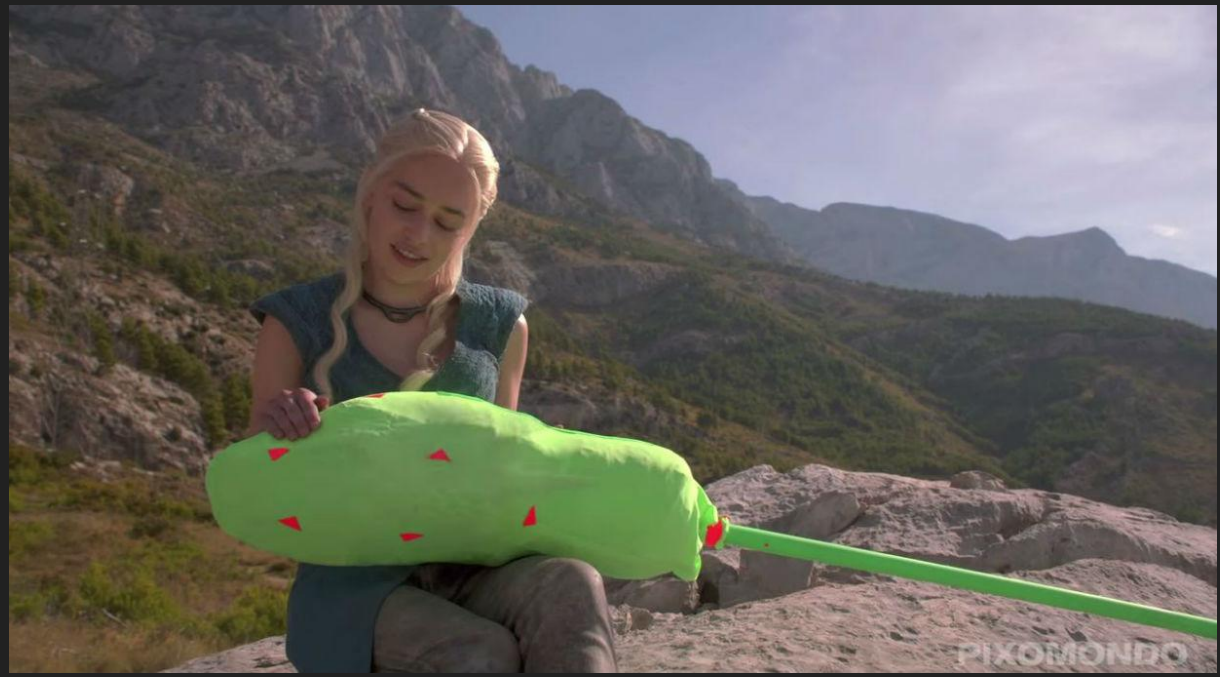
```
foo #kernel1 #kernel2 #kernel3...
```

Intro - Picture editing on a dragon

C-like syntax

Customized syntax for picture editing

LLVM backend



AST, program structure

- Init_array
- S_init, F_init
- Getpic, Getmatrix, GetRGBXY
- Assignpic, Assignmatrix, AssignRGBXY

```
type expr = Literal of int
  | Id of string
  | StringLit of string
  | CharLit of char
  | BoolLit of bool
  | Binop of expr * op * expr
  | Unop of uop * expr
  | Assign of string * expr
  | Call of string * expr list
  | Getarr of string * expr
  | Assignarr of string * expr * expr
  | Getpic of string * string
  | GetRGBXY of string * string * expr * expr
  | Getmatrix of string * expr * expr
  | Assignpic of string * string * expr
  | AssignRGBXY of string * string * expr * expr * expr
  | Assignmatrix of string * expr * expr * expr
  | Convol of expr * expr
  | Noexpr
  | Init_array of string * expr list
```

```
type initialization = typ * string * expr

type vdecl =  Bind of bind

type for_init = F_init of initialization
  | F_expr of expr


type stmt = Block of stmt list
  | Expr of expr
  | If of expr * stmt * stmt
  | For of for_init * expr * expr * stmt
  | While of expr * stmt
  | Return of expr
  | S_bind of bind
  | S_init of initialization

type func_decl = {
  typ: typ;
  fname: string;
  formals: bind list;
  body: stmt list;
}

type decl = Vdecl of vdecl
  | Fdecl of func_decl

type  program = decl list
```

Parser & Scanner

```
typ:
    INT { Int }
  | BOOL { Bool }
  | CHAR { Char }
  | VOID { Void }
  | PIC {Pic}
```

```
| WHILE LPAREN expr RPAREN stmt { While($3, $5) }
| typ ID SEMI { S_bind($1, $2) }
| typ ID ASSIGN expr SEMI { S_init($1, $2, $4) }
| typ ID LBRACKET LITERAL RBRACKET SEMI {S_bind(Array($1, $4),$2)}
| MATRIX ID LBRACKET LITERAL RBRACKET LBRACKET LITERAL RBRACKET SEMI { S_bind(Matrix($4,$7),$2) }
```

```
| ID LBRACKET expr RBRACKET ASSIGN expr { Assignarr($1, $3, $6) }
| ID LBRACKET expr RBRACKET { Getarr($1, $3) }
| ID DOT ID          {Getpic($1, $3)}
| ID DOT ID LBRACKET expr RBRACKET LBRACKET expr RBRACKET {GetRGBXY($1, $3, $5, $8)}
| ID DOT ID ASSIGN expr {Assignpic($1, $3, $5)}
| ID DOT ID LBRACKET expr RBRACKET LBRACKET expr RBRACKET ASSIGN expr {AssignRGBXY($1, $3, $5, $8, $11)}
| ID LBRACKET expr RBRACKET LBRACKET expr RBRACKET { Getmatrix($1,$3,$6) }
| ID LBRACKET expr RBRACKET LBRACKET expr RBRACKET ASSIGN expr { Assignmatrix($1,$3,$6,$9) }
| expr CONV expr { Convol($1,$3) }
| ID PPLUS { Assign($1, Binop(Id($1), Add, Literal(1))) }
| ID MMINUS { Assign($1, Binop(Id($1), Sub, Literal(1))) }
| DELETE ID { Unop(Delete, Id($2)) }
| ID ASSIGN int_list { Init_array($1, $3) }
```

Semantic Checker

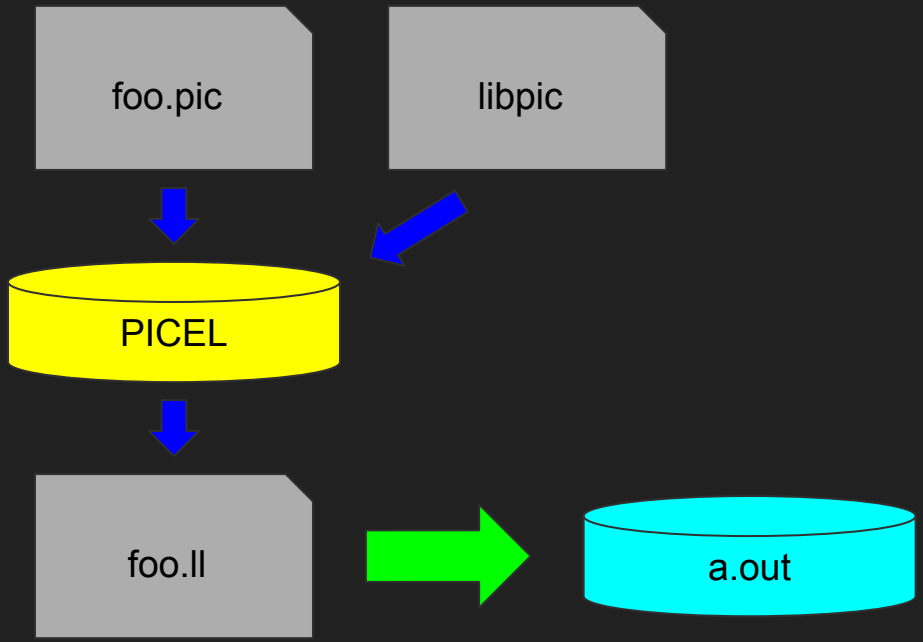
- For_init var => local var => block local hash list => global var
- Pic, RGBXY, Matrix type checking
- Convolution checking

```
let type_of_identifier local_hash_list s =
  try Hashtbl.find for_init_symbols s
  with Not_found ->
     try Hashtbl.find local_symbols s
     with Not_found ->
        try search_var_in_locals s local_hash_list
        with Not_found ->
           try Hashtbl.find global_symbols s
           with Not_found -> raise (Failure ("undeclared identifier " ^ s))
```

```
| GetRGBXY(s1, s2, e1, e2) -> ignore(type_of_identifier local_hash_list s1);
                              rgb_attr_checker s2;
                              ignore(expr local_hash_list e1);
                              expr local_hash_list e2
| AssignRGBXY(s1, s2, e1, e2, e3) ->  ignore(type_of_identifier local_hash_list s1);
                                      rgb_attr_checker s2;
                                      ignore(expr local_hash_list e1);
                                      ignore(expr local_hash_list e2);
                                      check_int_assign e3 (expr local_hash_list e3)
| Getpic(s1, s2) -> ignore(type_of_identifier local_hash_list s1);
                    ignore(pic_attr_checker s2);
                    StringMap.find s2 pic_attrs
| Assignpic(s1, s2, e) -> ignore(type_of_identifier local_hash_list s1);
                          ignore(pic_attr_checker s2);
                          ignore(expr local_hash_list e);
                          check_int_assign e (expr local_hash_list e)
```

```
| Convol(e1, e2) -> ignore(expr local_hash_list e1);
                    expr local_hash_list e2
| Getarr(s, e) -> ignore(type_of_identifier local_hash_list s);
                  expr local_hash_list e
| Assignarr(s, e1, e2) -> ignore(type_of_identifier local_hash_list s);
                          ignore(expr local_hash_list e1);
                          (* expr local_hash_list e2 *)
                          let st = type_of_identifier local_hash_list s
                          and e2t = expr local_hash_list e2
                          in
                          check_arr_assign st e2 e2t
| Init_array(s, el) -> List.iter (fun e -> ignore(expr local_hash_list e)) el;
                       type_of_identifier local_hash_list s
| Getmatrix(s, e1, e2) -> ignore(type_of_identifier local_hash_list s);
                          ignore(expr local_hash_list e1);
                          expr local_hash_list e2
| Assignmatrix(s, e1, e2, e3) -> ignore(type_of_identifier local_hash_list s);
                                 ignore(expr local_hash_list e1);
                                 ignore(expr local_hash_list e2);
                                 check_int_assign e3 (expr local_hash_list e3)
```

Code Gen




```
struct pic {
    unsigned int width;
    unsigned int height;
    unsigned int bytes_per_pixel;
    unsigned char* data;
};
```

Code Gen - Tricks

- Using hash maps:
  - Nested variable table:
    - variable name -> variable address (llvalue)
  - Type table:
    - llvalue -> variable type (used for type extend/casting and matrix dimension)

- 3rd-party library for bitmap read/write

Bitmap Image Reader Writer Library
Author: Arash Partow - 2002
URL: http://partow.net/programming/bitmap/index.html

Graphic Library

- Load
- Save
- newpic
- Convolution
- Copy_pic
- To_BW

Testing

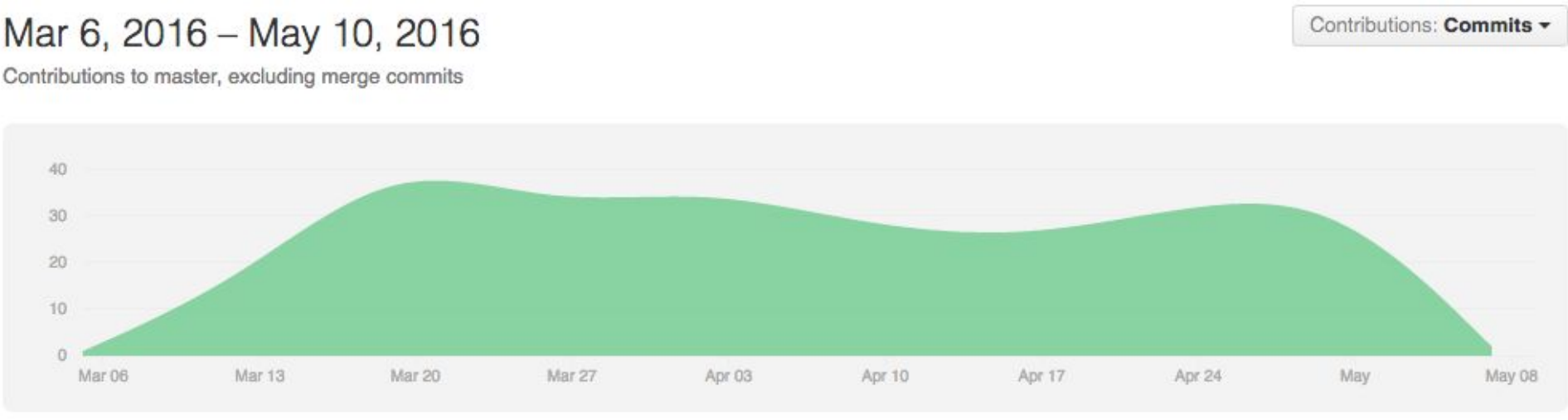
```
fail-assign4.pic    fail-assign4.err    fail-for1.pic       fail-for1.err
fail-for6.pic       fail-for6.err       test-array1.pic     test-array1.out
test-array2.pic     test-array2.out     test-array3.pic     test-array3.out
test-array4.pic     test-array4.out     test-assign1.pic    test-assign1.out
test-block1.pic     test-block1.out     test-for1.pic       test-for1.out
test-for2.pic       test-for2.out       test-for4.pic       test-for4.out
test-for5.pic       test-for5.out       test-global3.pic    test-global3.out
test-hello1.pic     test-hello1.out     test-keyword.pic    test-keyword.out
test-local2.pic     test-local2.out     test-local3.pic     test-local3.out
test-mat1.pic       test-mat1.out       test-minus.pic      test-minus.out
test-ops3.pic       test-ops3.out       test-var2.pic       test-var2.out
```

```
conv.pic            demo_color.pic      hello.pic           source.pic
struct_test.pic     test-conv1.pic      test-conv2.pic      test-conv3.pic
test-conv4.pic
```

Planning & Processes

- Team Distribution: 2 * Codegen, 2 * Semantic, 1 * Testing
- Always Physical Meeting (Most Amazing!)
- Internal Deadline
- Version Control

Project Timeline



Development Challenges

- Codegen:
  - Type extend/casting problem
  - Alloca in loop causes stack overflow
    - LLVM's mem2reg pass helps?
  - Memory corruption debugging
    - struct/array indexing
    - Bitmap buffer init/free
  - Difference between LLVM versions
- Variable Scoping (semantic/codegen):
  - Nested local variable scope
- Testing
  - Coming up with tricky test cases

demo!