



# Tough Image Enhancer

Гущин Елисей      10-1 класс  
Косуха Анатолий    9-1 класс  
Лаевский Игорь     9-3 класс  
Логачев Федор     9-3 класс  
Насимов Михаил    9-1 класс  
Розет Михаил        8-2 класс



Научный руководитель: Галинский Виталий Александрович,  
Преподаватель информатики и программирования ФМЛ № 30

\*

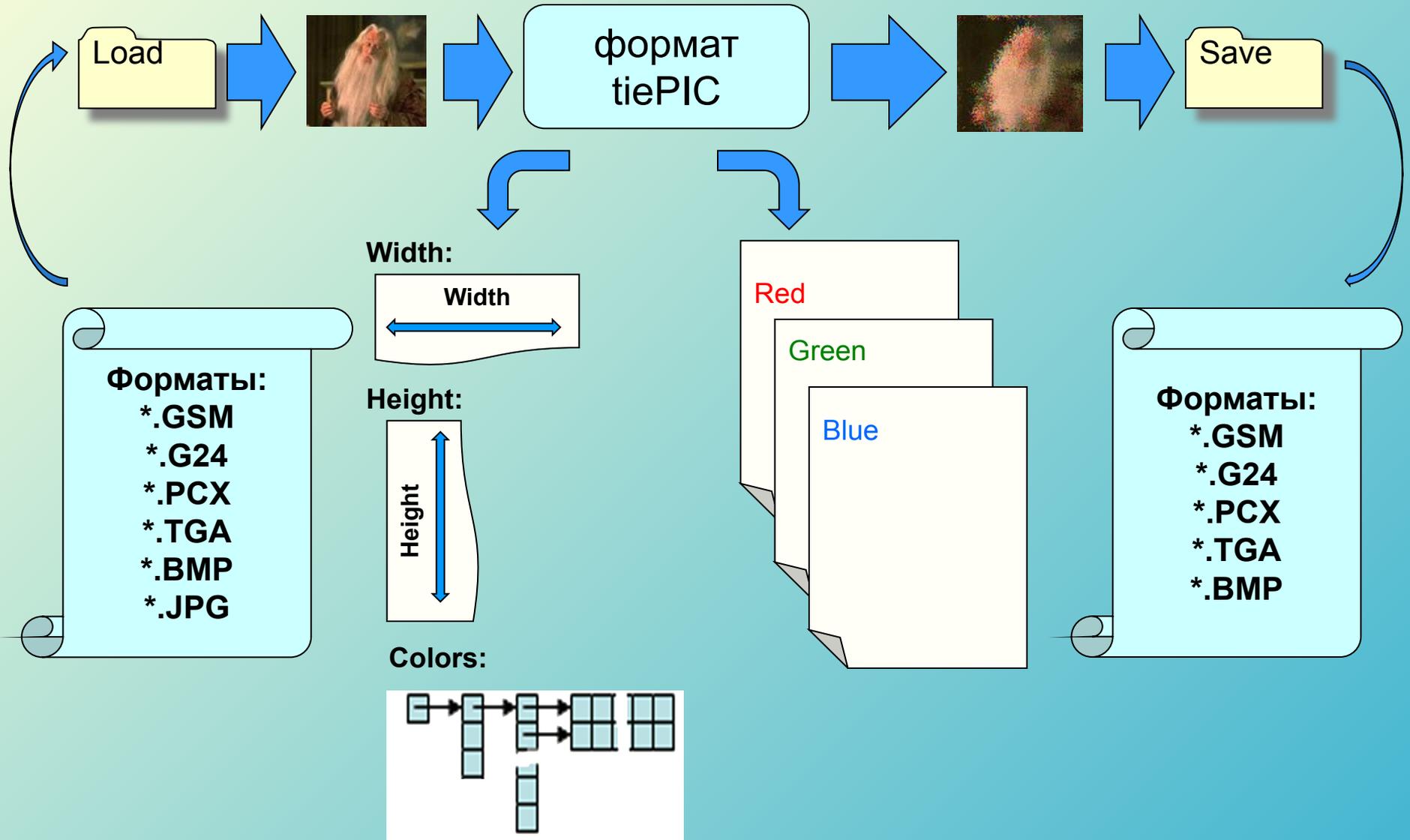


## Составные части проекта

- Система представления и визуализации данных
- Фильтры
- Таблицы подстановок
- Язык со стек-машиной



# Хранение и визуализация данных.





## Фильтры

### Линейная фильтрация

#### Линейная фильтрация:

Для локальной фильтрации изображений в нашей программе применяется матрица  $5 \times 5$ , устроенная следующим образом:

Каждому элементу этой матрицы, называемой также апертурой, сопоставляется некоторое число, обычно задаваемое некоторой функцией, называемой функцией апертуры. Это число называется весовым множителем. Фильтрация осуществляется перемещением апертуры фильтра по изображению. В каждом положении апертуры выполняются однотипные действия которые и определяют “отклик” фильтра.

При каждом положении апертуры соответствующий весовой коэффициент поэлементно умножается на значения соответствующих пикселей, произведения суммируются. Сумма делится на нормирующий коэффициент (  $freq$  ) к ней прибавляется сдвиг (  $bias$  ), результат присваивается результирующему пикселю, который находится в центре апертуры. Упомянутый выше делитель обычно берется равным сумме всех элементов апертуры.





## Фильтры

### Процедурные фильтры

#### Фильтр Робертса

Цвет результирующей точки вычисляется по формуле:

$$\sqrt{X^2 + Y^2}$$

Где

$F(X, Y)$  – цвет точки  $(X, Y)$

$X = F(i, j) - F(i + 1, j + 1)$

$Y = F(i, j + 1) - F(i + 1, j)$

#### Медианная фильтрация

Выделяется апертура  $N \times N$ , она передвигается по изображению аналогично линейной фильтрации.

Она сортируется, элемент, находящийся в центре отсортированного массива присваивается результирующему пикселю (центральному в апертуре).

#### Процентильный фильтр:

Выделяется апертура  $N \times N$ , она передвигается по изображению аналогично линейной фильтрации.

В ней выбирается центральный элемент, все элементы меньшие(большие) его обращаются в ноль, другие делаются равным ему.

#### Фильтр Кувахары

Выделяется апертура  $N \times N$ , она передвигается по изображению аналогично линейной фильтрации.

В ней выделяются квадраты  $N/2 + 1 \times N/2 + 1$ . В каждом из них ищется минимальная разница. В квадрате с минимальной разницей ищется среднее значение, оно и присваивается результирующему пикселю (центральному в апертуре).

#### Фильтр Собела

Цвет результирующей точки высчитывается по формуле:

$$\sqrt{X^2 + Y^2}$$

Где

$F(X, Y)$  – цвет точки  $(X, Y)$

$X = [F(i - 1, j + 1) + 2F(i, j + 1) + F(i + 1, j + 1)] - [F(i - 1, j - 1) + 2F(i, j - 1) + F(i + 1, j - 1)]$

$Y = [F(i - 1, j + 1) + 2F(i - 1, j) + F(i - 1, j + 1)] - [F(i + 1, j - 1) + 2F(i + 1, j) + F(i + 1, j + 1)]$





# Таблицы подстановок

Look Up Tables( таблицы подстановки ) - таблица поточечной обработки.

Принцип действия состоит в создании самой таблицы и замены в изображении всех цветов, значения которых совпадают с индексом одного из элементов таблицы, на значение данного элемента.



```
for (y = 0; y < Pic->H; y++)
```

```
for (x = 0; x < Pic->W; x++)
```

Номер плоскости

Y -координаты точки

X – координаты точки.

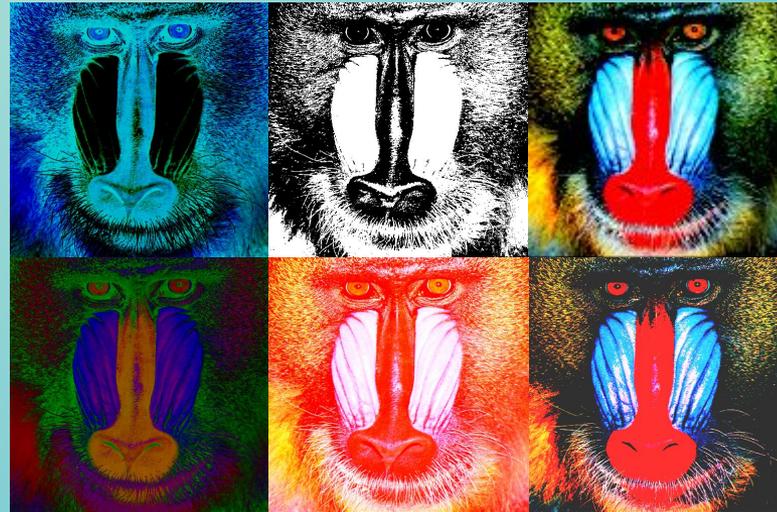
```
Pic[0][y][x] = LUT[Pic[0][y][x]];
```

```
Pic[1][y][x] = LUT[Pic[1][y][x]];
```

```
Pic[2][y][x] = LUT[Pic[2][y][x]];
```

```
}
```

Массив из 256 элементов





# Примеры языка

```
load X:\PICS\PCX\TR.PCX  
push  
load  
X:\PICS\PCX\mandrill512.bmp  
push  
load X:\PICS\PCX\ROSE.PCX  
combine_image 0.8  
save E:\cgsg22\tr1.bmp bmp
```





# Примеры языка

```
; Loading picture  
load X:\pics\pcx\tr.pcx  
Pop  
; Make contrast  
lut contrast 100 200  
; Mediana filtration  
filter mediana 11  
; Push result  
Push  
; Roberts filtration  
filter roberts 2  
; Combine roberts and  
; mediana results  
combine 0.4 stretch
```





# Примеры языка

```
load X:\pics\m.g24
pop
lut contrast 100 200
filter kuwahara 10
push
filter sobel 1
combine 0.4 stretch
```





# Примеры языка

```
load X:\pics\f.bmp ; load to top of the stack
                    ; main window
pop                ; get picture from
                    ; current picture
load X:\pics\stone.g24 ; load
                    ; background picture
load X:\pics\pcx\rose.pcx ; load mask file
add 0.5
```



