

# MVC Instant Image Cloning

## 实验报告

赵欣昊

计 65

2016011351

2018 年 6 月 30 日

# 1 实验原理

## 1.1 Mean-Value Coordinates(MVC)

对于一个多边形中任意一点，它的某个关于坐标的函数可以表示为顶点坐标的函数的加权和，公式如下：

设多边形的边界  $\partial P = (p_0, p_1, \dots, p_m = p_0)$ ,  $p_i \in R^2$ , 则有

$$\lambda_i(x) = \frac{w_i}{\sum_{j=0}^{m-1} w_j}, i = 0, 1, \dots, m-1$$

$$w_i = \frac{\tan\left(\frac{\alpha_{i-1}}{2}\right) + \tan\left(\frac{\alpha_i}{2}\right)}{\|p_i - x\|}$$

其中  $\alpha_i$  为  $p_i$  与  $p_{i+1}$  的夹角。

$$f(x) = \sum_{i=0}^{m-1} \lambda_i(x) f(p_i)$$

## 1.2 MVC Seamless Cloning

---

### Algorithm 1 MVC Seamless Cloning

---

```

1: Preprocessing stage
2: for each pixel  $x \in \mathcal{R}$  do
3:   {Compute the mean-value coordinates of  $x$  w.r.t.  $\partial P_s$ }
4:    $\lambda_0(x), \dots, \lambda_{m-1}(x) = MVC(x, y, \partial P_s)$ 
5: end for
6: for each new  $P_t$  do
7:   {Compute the differences along the boundary}
8:   for each vertex  $p_i$  of  $\partial P_t$  do
9:      $diff_i = f^*(p_i) - g(p_i)$ 
10:  end for
11:  for each pixel  $x \in P_t$  do
12:    {Evaluate the mean-value interpolant at  $x$ }
13:     $r(x) = \sum_{i=0}^{m-1} \lambda_i(x) \cdot diff_i$ 
14:     $f(x) = g(x) + r(x)$ 
15:  end for
16: end for

```

---

计算出边界上的颜色的差，对所有源图片的像素计算边界顶点加权和，并将其颜色按照加权和进行修正。

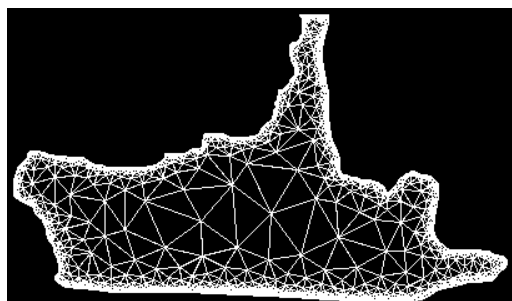
## 1.3 Adaptive Mesh

将源图片进行三角剖分，将每个三角的顶点表示为边界顶点的加权和，将其它点表示为所在三角形三个顶点的加权和，从而加速算法。

## 2 实验结果

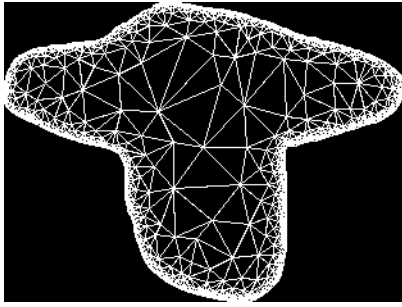
测试方法：执行 demo 文件夹下的 exe 文件

### 实验 1

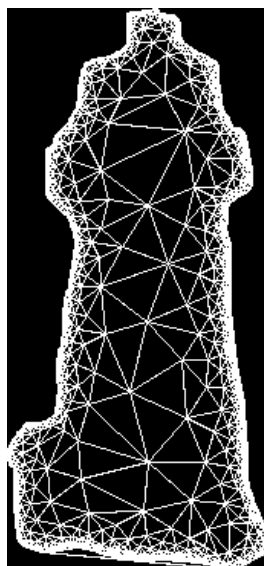


### 实验 2





### 实验 3



注：原图来自 <https://github.com/Polar1s/MVCCloning/tree/master/demo>

### 3 参考文献

Farbman Z, Hoffer G, Lipman Y, et al. Coordinates for instant image cloning[J]. Acm Transactions on Graphics, 2009, 28(3):1-9.