

Relatório Aula CG 2018.2 IFCE  
Prática: Ajuste de Alpha - 13/08/2018  
Professor: Lucas Sousa.  
Alunos: Gabriela Bezerra, Ryllari Santana.

## - Introdução

Implementamos o ajuste de Alpha utilizando três imagens: background.jpg, foreground.png, mask.png, todas com as mesmas dimensões.

Utilizamos a biblioteca OpenCV e C++ para ler os arquivos e fazer as operações necessárias à sobreposição de imagens utilizando Alpha.

Código:

```
#include <stdio.h>
#include <opencv2/opencv.hpp>

using namespace std;
using namespace cv;

Mat changeAlpha(Mat mask, Mat fore, Mat back, float alpha) {
    Mat result = Mat::zeros(fore.size(), fore.type());

    fore.convertTo(fore, CV_32FC3);
    back.convertTo(back, CV_32FC3);

    // Normalizando a mascara
    mask.convertTo(mask, CV_32FC3, alpha/255);

    // Multiplicando a mascara pelo foreground e atribuindo ao foreground
    multiply(mask, fore, fore);

    // Multiplicando o background por ( 1 - mascara ) e atribuindo ao background
    multiply(Scalar::all(1.0)-mask, back, back);

    // Adicionando background e foreground após terem sido aplicadas as
    // mascaras...
    add(fore, back, result);

    return result/255;
}

int main() {

    Mat mask = imread("imgs/alpha_channel.png");
    Mat fore = imread("imgs/foreground.png");
    Mat back = imread("imgs/background.jpg");

    float alpha;
    cout<<"Digite o alpha"<<endl;
    cin>>alpha;

    Mat r = changeAlpha(mask, fore, back, alpha);

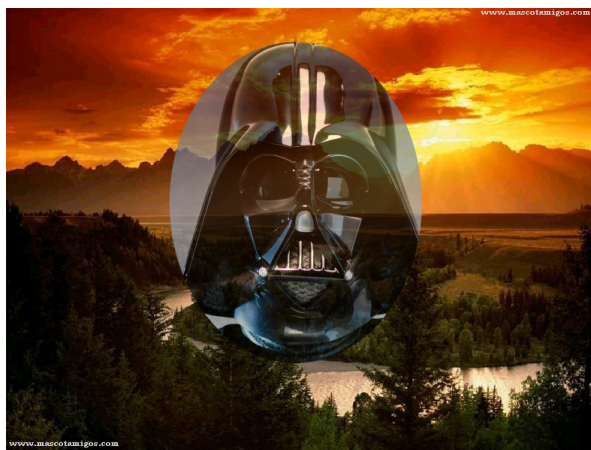
    cv::imwrite("Result.png", r*255);
    cv::imshow("Result", r);
    cv::waitKey(0);

    return 0;
}
```

## - Resultados



Alpha 0.0



Alpha 0.7



Alpha 1.0



Alpha 2.0

#### - Conclusão

O algoritmo consiste em uma multiplicação escalar das matrizes das imagens envolvidas aplicando a fórmula:

$$r = \text{fore} * (\text{mask} * \alpha) + \text{back} * (1 - (\text{mask} * \alpha))$$

Primeiro, foi feita uma tentativa manual de multiplicação das matrizes em loops aninhados, mas ela não funcionou muito bem, então, estudando a documentação do opencv, foi tomada a decisão de usar a função `convertTo` para normalizar o data type de todas as matrizes, e a função `multiply` para realizar a multiplicação escalar. O resultado foi como esperado: a multiplicação de duas matrizes de acordo com o alpha inserido, com a subtração da área de valor 0 da máscara.