

Relatório Detalhado do Projeto Mosaico

Desenvolvido em C com OpenCV

1 Introdução

O presente relatório detalha o funcionamento dos scripts que compõem o projeto Mosaico. Este projeto, desenvolvido em C utilizando a biblioteca OpenCV, tem como objetivo transformar uma imagem original em um mosaico. Para isso, duas etapas principais são realizadas: a preparação de um banco de dados de tiles (imagens menores) e a criação do mosaico propriamente dito, onde cada região da imagem original é substituída pelo tile cuja média de cor melhor se aproxima da média da região.

2 Objetivo do Projeto

O projeto Mosaico visa:

- Processar um conjunto de imagens (tiles) para calcular suas médias de cor e armazená-las no arquivo **mosaic.txt**.
- Utilizar essas informações para transformar uma imagem original em um mosaico, substituindo cada região da imagem pelo tile com a média de cor mais próxima.

3 Arquivos e Componentes

O projeto é composto pelos seguintes arquivos:

- **fazmosaico.c**: Implementa a criação do mosaico. Este programa divide a imagem original em regiões, calcula a média de cor de cada região, busca no banco de dados (arquivo **mosaic.txt**) o tile com o perfil de cor mais próximo e substitui a região pelo tile redimensionado.
- **mosaic.c**: Processa os tiles, calculando a média dos valores de cor (RGB) de cada imagem e gravando os resultados no arquivo **mosaic.txt**. Cada linha deste arquivo possui o formato:

```
nome_imagem.jpg:red:green:blue
```

- **mosaic.txt**: Banco de dados gerado pelo **mosaic.c**, contendo as médias de cor dos tiles.

4 Funcionamento dos Scripts

4.1 Preparação do Banco de Dados (**mosaic.c**)

O programa **mosaic.c** realiza as seguintes operações:

1. Abre imagens nomeadas sequencialmente (por exemplo, `1.jpg`, `2.jpg`, `3.jpg`, etc.).
2. Para cada imagem, a função `encontrar_car` é chamada para calcular a média dos valores de cor (vermelho, verde e azul).
3. Os valores calculados são gravados no arquivo **mosaic.txt** no formato:

```
nome_imagem.jpg:red:green:blue
```

4.2 Criação do Mosaico (**fazmosaico.c**)

O programa **fazmosaico.c** executa o seguinte processo:

1. A imagem original é carregada e dividida em uma grade de células. O tamanho da grade é definido por um dos parâmetros numéricos passados via linha de comando.
2. Para cada região da imagem, o programa calcula a média dos valores de cor.
3. Utilizando o arquivo **mosaic.txt**, a função `dados` é chamada para identificar o tile cuja média de cor seja a mais próxima da média da região.
4. **ci067** é um parâmetro auxiliar, indicando a pasta onde estão armazenados os tiles.
5. A função `subimg` é utilizada para redimensionar o tile selecionado e substituir a região da imagem original pelo tile.
6. Ao final do processamento, a imagem resultante é salva como **nova.jpg**.

5 Detalhamento das Funções Principais

5.1 Função `subimg`

Esta função tem como objetivo substituir uma região da imagem original pela imagem de um tile. Ela realiza os seguintes passos:

- Calcula as dimensões da região com base na divisão da imagem original.
- Carrega o tile (usando o nome passado como parâmetro) e o redimensiona para se ajustar à região.
- Copia os pixels do tile redimensionado para a posição correspondente na imagem original.

Trecho de código relevante:

```

void subimg(IplImage *img, char *nome, int k, int l, int tamm) {
    IplImage *imagem;
    int a, b, i, j, x = ZERO, y = ZERO;
    int altura = img->height;
    int largura = img->width;
    int step = img->widthStep;
    int channels = img->nChannels;
    uchar *data = (uchar *)img->imageData;

    a = (altura / tamm);
    b = (largura / tamm);

    imagem = cvLoadImage(nome, 1);
    printf("%s", nome);

    IplImage* imgnew = cvCreateImage(cvSize(b, a), img->depth, 3);
    cvResize(imagem, imgnew, CV_INTER_LINEAR);
    int step2 = imgnew->widthStep;
    uchar *data2 = (uchar *)imgnew->imageData;

    for(i = k * a, x = 0; i < (k+1) * a; i++, x++)
        for(j = l * b, y = 0; j < (l+1) * b; j++, y++) {
            data[i*step + j*channels + 0] = data2[x*step2 + y*channels + 0];
            data[i*step + j*channels + 1] = data2[x*step2 + y*channels + 1];
            data[i*step + j*channels + 2] = data2[x*step2 + y*channels + 2];
        }
    cvReleaseImage(&imgnew);
}

```

5.2 Função dados

Esta função abre o arquivo **mosaic.txt** e lê cada linha para encontrar o tile cuja média de cor seja a mais próxima da média de cor calculada para uma região. A função compara a soma das diferenças absolutas entre os canais de cor e retorna o nome do tile com a menor diferença.

Trecho de código:

```

char * dados(int red, int green, int blue, char *arq) {
    FILE *car;
    auxiliar aux;
    int a, b = 10000;
    char *nome;

    car = fopen(arq, "r");
    nome = (char*)malloc(sizeof(char)*10);

    while (!feof(car)) {
        fscanf(car, "%[^:]:%d:%d:%d\n", aux.n, &aux.r, &aux.g, &aux.b);
        a = (abs(aux.r - red) + abs(aux.g - green) + abs(aux.b - blue));
        if (a < b) {
            b = a;
            sprintf(nome, "%s", aux.n);
        }
    }
}

```

```

    }
    fclose(car);
    return nome;
}

```

5.3 Função mosaico

Esta função gerencia o processo de criação do mosaico. Ela:

- Divide a imagem original em uma grade com número de células definido pelo parâmetro `tammatriz`.
- Para cada célula, calcula a média dos valores de cor.
- Utiliza a função `dados` para buscar o tile cujo perfil de cor seja o mais próximo.
- Chama a função `subimg` para substituir a região correspondente na imagem original pelo tile selecionado.

Trecho de código:

```

void mosaico(IplImage *img, char *inf[]) {
    int i, j, a, b, c, k, l;
    int red, green, blue;
    int altura, largura, step, channels;
    uchar *data;
    char *nome;
    altura = img->height;
    largura = img->width;
    step = img->widthStep;
    channels = img->nChannels;
    data = (uchar *)img->imageData;

    c = atoi(inf[5]); // tammatriz
    a = (altura / c);
    b = (largura / c);

    for (k = 0; k < c; k++) {
        for (l = 0; l < c; l++) {
            red = 0; green = 0; blue = 0;
            for (i = k * a; i < (k + 1) * a; i++)
                for (j = l * b; j < (l + 1) * b; j++) {
                    red += data[i * step + j * channels + 0];
                    green += data[i * step + j * channels + 1];
                    blue += data[i * step + j * channels + 2];
                }
            red /= (a * b);
            green /= (a * b);
            blue /= (a * b);
            printf("ok1_␣%d_␣%d\n", k, l);
            // Busca no banco de dados a caracterstica mais proxima
            nome = dados(red, green, blue, inf[3]);
            // Substitui o pedao da imagem pelo tile selecionado
            subimg(img, nome, k, l, c);
        }
    }
}

```

```

        printf("ok3_█%d_█%d\n", k, l);
    }
}

```

5.4 Função encontrar_car (em mosaic.c)

Esta função é utilizada para calcular a média dos valores de cor (RGB) de um tile inteiro. Ela percorre todos os pixels da imagem e acumula os valores para cada canal, dividindo pelo número total de pixels para obter a média.

Trecho de código:

```

void encontrar_car(IplImage *img, int *red, int *green, int *blue) {
    int altura, largura, step, channels;
    uchar *data;
    int i, j;
    altura = img->height;
    largura = img->width;
    step = img->widthStep;
    channels = img->nChannels;
    data = (uchar *)img->imageData;

    *red = *green = *blue = 0;
    for(i = 0; i < altura; i++)
        for(j = 0; j < largura; j++){
            *red += data[i*step + j*channels + 0];
            *green += data[i*step + j*channels + 1];
            *blue += data[i*step + j*channels + 2];
        }
    *red /= (altura * largura);
    *green /= (altura * largura);
    *blue /= (altura * largura);
}

```

6 Compilação e Execução

6.1 Geração do Banco de Dados

Para compilar e executar o programa que gera o arquivo **mosaic.txt**, use os seguintes comandos:

```

gcc 'pkg-config --cflags --libs opencv' -o mosaic mosaic.c
./mosaic mosaic.txt

```

6.2 Criação do Mosaico

Para compilar e executar o programa que cria o mosaico, use:

```

gcc 'pkg-config --cflags --libs opencv' -o fazmosaico fazmosaico.c
./fazmosaico 14.jpg ci067 mosaic.txt 1 10

```

Nesta execução:

- `14.jpg` é a imagem de entrada.
- `ci067` é um parâmetro auxiliar, indicando a pasta onde estão armazenados os tiles.
- `mosaic.txt` é o arquivo com os dados dos tiles.
- `1` e `10` são parâmetros numéricos; o último (10) define a quantidade de células na grade (por exemplo, uma divisão 10x10).

A imagem resultante será salva como **nova.jpg**.

7 Conclusão

O projeto Mosaico integra processamento de imagens e técnicas de substituição de regiões para criar composições visuais artísticas. Utilizando OpenCV em C, o sistema:

- Processa um conjunto de tiles para gerar um banco de dados de médias de cor.
- Divide a imagem original em regiões e, para cada uma, seleciona o tile cujo perfil de cor se aproxima mais da média calculada.
- Substitui cada região pelo tile correspondente, resultando em uma nova imagem (**nova.jpg**) que preserva, de forma aproximada, a tonalidade e a estrutura da imagem original.

Essa abordagem pode ser aplicada em áreas como reconhecimento de padrões, criação artística e experimentação visual.