Computer Graphics
Prof. Dr.-Ing. Hendrik P.A. Lensch
Jieen Chen

Eberhard Karls
Universität
Tübingen

Displays - WS 2017/18

Exercise 2

Submission deadline: Thursday, 23.11.17

- Please upload your source code and result images to ILIAS before the next exercise meeting
- C++ and OpenCV are mandatory

2.1 Cascaded Displays - Simulation of high resolution image formation (20 points)

During the lecture, we've learnd one spatiotemporal resolution enhancement method called Cascaded Displays which subject to the lateral displacement of layers of low resolution displays to synthesize higher resolution display. The simpliest setup of this technique contains only two shifted layers of grid panels, where optimized low resolution images are shown. The subpixel fragments are shown in Figure 2.1, where intensity of each high resolution image pixel is a multiplication of the transmittance of two corresponding pixels of the two layers. When two static images are shown ($k = 1$ in Equation 1 of the paper, see Hints), the intensity follows Equation 1, where a non-negative factorization algorithm will estimate the optimized images.

$$s_{i,j} = w_{i,j}(a_i b_j) \tag{1}$$

In Equation 1, $w_{i,j}$ is a weighting factor denotes the overlap of pixel a_i and b_j, where i and j are the pixel indices. The dual-layer image formation is expressed as

$$\mathbf{S} = \mathbf{W} \circ (\mathbf{a}\mathbf{b}^T) \tag{2}$$

where **a** and **b** are column vector representation of two low resolution images.
假设 We assume a dual-layer static display scenario without rotation of layers throughout this exercse. In this case, $w_{i,j}$ is a binary number to denote the overlap.
标示 重叠 方案

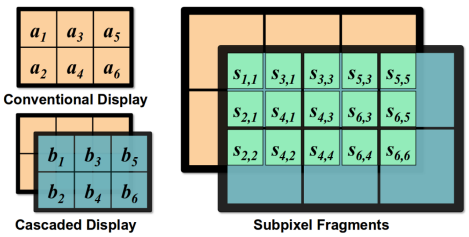
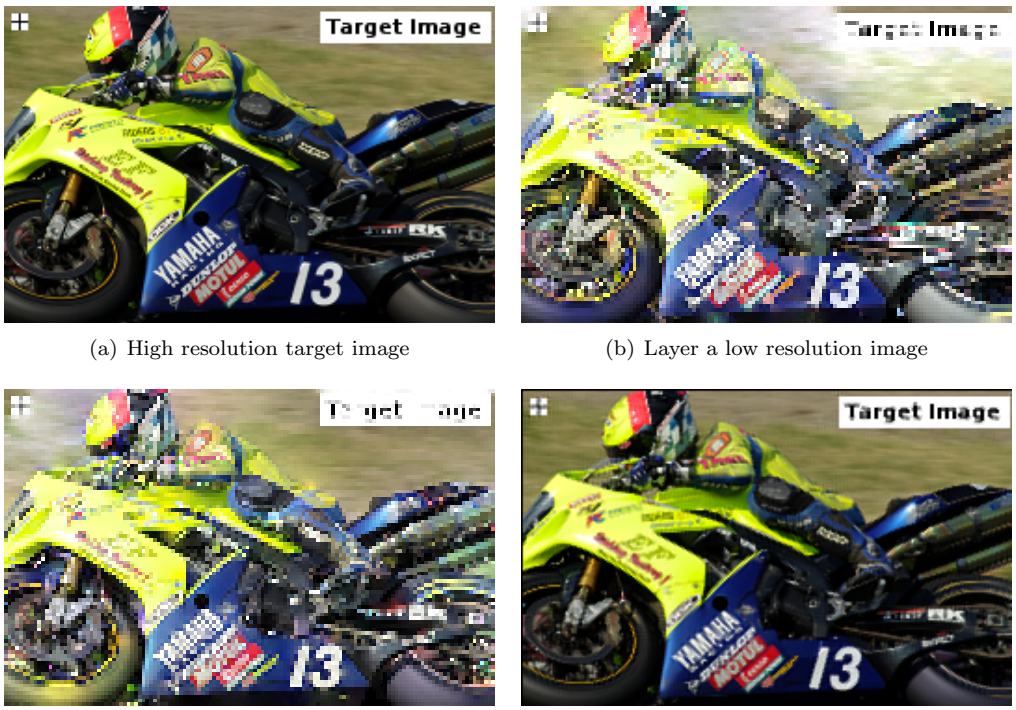


Abbildung 1: Subpixel fragments by Cascaded Displays.

In this first exercise, we provide two optimized images (also shown in Figure 2), `layer_a.png`, `layer_b.png` and the target high resolution image `motor.png`. `layer_a.png` and `layer_b.png` represent the low resolution images as resulted from the non-negative factorization algorithm for Cascaded Displays. Your task is to simulate one high resolution image by multiplying `layer_a` elementwise with `layer_b` following Equation 2. The latter has to be shifted by half a pixel horizontally and vertically.

Hints:

- Use nearest neighbour method when upscaling images, you can use function `cv::resize`
- The coordinate origin is at upper-left corner
- Useful functions: Element-wise multiplication `cv::Mat::mul`, Rectangle class `cv::Rect`,
- Cascaded Displays paper link



(a) High resolution target image

(b) Layer a low resolution image

(c) Layer b low resolution image

(d) Simulated high resolution image

Abbildung 2: Simulation of Cascaded Display: The high resolution target image, the low resolution images that can be seen using a single display and the image displayed on Cascaded Displays.

2.2 Cascaded Displays - Optimization with non-negative factorization algorithm WRRI (80 points)

因子分解

In this exericse, your task is to implement the non-negative factorization algorithm WRRI (**Algorithm 1** of the paper) to generate two low resolution images from one high resolution image. After that, these two images will be printed on two sheets of transparencies (maximum A4 size) for you to align and replicate the Cascaded Displays method.

As it is described in the previous exercise, we assume static display scenario, that is to say $k = 1$ and $\mathbf{R} = \mathbf{T}$. Note that the number of elements of the weighting matrix and target matrix with high resolution image stored, $\mathbf{W}$ and $\mathbf{T}$, are rather large because the two low resolution images are represented as column vectors $\mathbf{a}$ and $\mathbf{b}$ with N and M elements (in our case, $N = M$), which are the number of pixels. However, both of the matrices are sparse. Please refer to the lecture slides about weighting matrix for more details. You may want to have a look at OpenCV's SparseMat class. You can also solve this exercise without using sparse matrices at all but with matching of pixel indices.

Hints:

生产量 发出

- The maximal light throughput of a emitting pixel can never be greater than 1.0 (= white, full throughput of backlight) or smaller than 0.0. Consider this when assigning the updated pixel values in the iterations.
- You can initialize the two images with random values using `cv::RNG::RNG`
- You can use a high iteration number as the stopping condition

Algorithm 1 Weighted Rank-1 Residue Iterations (WRRI)

1: Initialize $\mathbf{A}$ and $\mathbf{B}$
2: **repeat**
3: $k = 1$
4: $\mathbf{R}_k = \mathbf{T}$ ▷ Evaluate rank-1 residue.
5: $\mathbf{a}_k \leftarrow \left[\frac{[(\mathbf{W}\circ\mathbf{R}_k)\mathbf{b}_k]_+}{\mathbf{W}(\mathbf{b}_k\circ\mathbf{b}_k)}\right]_+$ ▷ Update column k of $\mathbf{A}$.
6: $\mathbf{b}_k \leftarrow \left[\frac{[(\mathbf{W}\circ\mathbf{R}_k)^T\mathbf{a}_k]_+}{\mathbf{W}^T(\mathbf{a}_k\circ\mathbf{a}_k)}\right]_+$ ▷ Update column k of $\mathbf{B}$.
7: **end for**
8: **until** Stopping condition

Abbildung 3: Algorithm 1: WRRI non-negative factorization

- Errata: $\mathbf{R} = \mathbf{T}$ in lecture slide 12
- Besides printed transparencies, multiple 7-inch 1280×800 LCD panels are offered in the student lab (C410). Please feel free to use them to perform further testing.