Manga Colourisation

Team name: IFHTP
TA Mentor: Haripraveen Subramanian

Team members:
- Rutvij Menavlikar(2019111032, CSD)
- Tejas Chaudhari(2019111013 ,CSD)
- Ansh Khandelwal (2019102008, ECE)
- CYK sagar (2019101076, CSE)

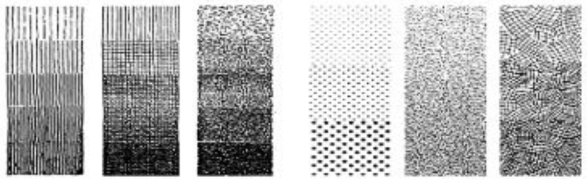
Repository: https://github.com/Digital-Image-Processing-IIITH/dip-project-ifhtp

Need for a new method

- Existing colorization methods relied on a rough continuity of grey levels to bring colorization.
- However, there are no gray level continuities in mangas. Instead they have rough continuities of black and white patterns.
- Implementing the algorithm which colorises on basis of pattern-continuity and intensity-continuity
.

Hashing and Screening

- In mangas, hatching and screening patterns are used in mangas for shading , textures and background.



(a) Hatching (b) Screening

- Only two colors are used in hatching and screening (black and white). Since there is no continuity of intensity, we take advantage of the continuity of the patterns for color propagation.

Working of our application

The command to run the application is: python main.py <path_to_input_image>. This pops up three windows with the input window where the user scribbles, output window where the final output is shown and key inputs are taken from, and the trackbar where you can set the color of the scribble being drawn.

User begins by scribbling colors on regions of our interest in the gui. The user is prompted to choose a mode for segmentation between pattern continuous and intensity continuous. After the segments are obtained, the user is then prompted to choose either Color replacement, Stroke preserving, or Pattern to shading colorization option.

The user can set the values of different parameters using the config.json file as desired

Level Set Method

The colorization of both pattern-continuous regions and intensity continuous regions can be formulated under same mathematical framework of level set method.

The main idea is to raise the modeling of boundaries from a 2D planar curve into a three-dimensional curved surface, by taking the curves as the zero level set of a higher dimensional surface.

So as the user inputs his color preferences with a scribble on top of a part in the image, we propagate level set to reach all boundaries of the part to be coloured.

This method actuals helps to get us a parameter-free representation and also gives the capability in dealing with local deformation.

Level Set: Algorithm

- Level set propagation starts by initializing Φ. In our application, Φ is the distance from the user scribble
- The evolving curve Γ is obtained using $\Phi = 0$
- A narrow band with a specified width is constructed around Γ
- For each pixel inside the band, Φ is updated using $(\delta \Phi)/(\delta t) = h \times (F_A + F_B)|\nabla \Phi|$
- Using the updated Φ, we obtain a new Γ
- Repeat until convergence

Pattern and Intensity continuity

Pattern continuous regions

- To evolve the boundary over pattern continuous regions we need to measure change of pattern instead of the change of intensity.
- We measure the changes in pattern at the level set boundary and at the scribble region to estimate when changes in pattern have taken place.
- We define the filter (h) in the PDE as: $$h_P(x,y) = \frac{1}{1 + |D(T_{\text{user}}, T_{\text{front}}(x,y))|}.$$
- T_{user}, T_{front} are pattern features.

Pattern continuous regions

- Pattern features are generated by applying gabor wavelets on windowed parts of the image.

$$W_{m,n}(u,v) = \int_{\Omega} I(x,y) g^{*}_{m,n}(u-x, v-y)\,dxdy,$$

$$\begin{aligned} \mu_{m,n} &= \int\int |W_{m,n}(x,y)|\,dxdy, \\ \sigma_{m,n} &= \sqrt{\int\int (|W_{m,n}(x,y)| - \mu_{m,n})^2\,dxdy}. \end{aligned}$$

$$T = [\mu_{0,0}\ \sigma_{0,0}\ \mu_{0,1}\ \cdots\ \mu_{3,5}\ \sigma_{3,5}].$$

Intensity continuous regions

The color propagation over intensity continuous regions are treated under LSE.

We define a halting term h_I which measures change of intensity gradient, which will be used as a filter in the LSE PDE.

$$h_I(x,y) = \frac{1}{1 + |\nabla(G_\sigma \otimes I(x,y))|}$$

$G_\sigma \otimes I(x,y)$ denotes convolution of Image with gaussian filter G_σ.

Colorisation

The three methods implemented:

- Color replacement

  In this method irrespective of the pixel being black or white, within the boundary, is replaced by the scribble color.

- Stroke preserving
- Pattern to shading

Stroke preservation

As discussed earlier, in some cases hatching and screening is used to express material textures, reflectances and sometimes shapes. So instead of replacing all pixel values with scribble color, we retain the strokes and fill in the color in between them.

The user color is multiplied with halting term h_I in the YUV space.

$$Y_{new}(x,y) = Y_{user} \otimes |1 - h_I(x,y)|^2,$$
$$(U,V)_{new} = (U,V)_{user}$$

Y represents the luminance channel.

$(U,V)_{new}$, $(U,V)_{user}$ denotes color of output and user input respectively.

Pattern shading

Sometimes the hatching and screening are used to recreate shading effects. Here the pattern strokes are replaced by smooth transition of color shading.

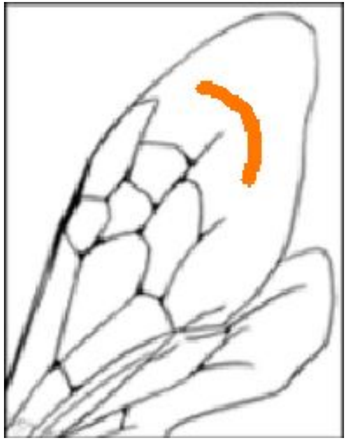
Local intensity within pixel neighbourhood is calculated as $s = f \otimes Y_{\text{image}}$.
Where Y_{image} is the pixel gray value in the input image, and f is a box filter.

Then the Y channel is linearly mapped as:

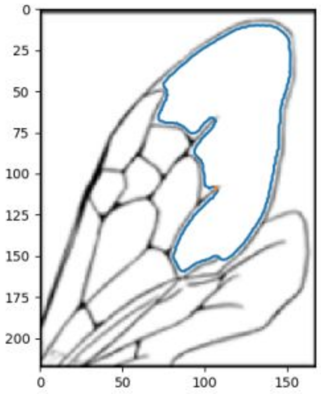
$$Y_{\text{new}} = sY_{\text{user}},$$
$$(U, V)_{\text{new}} = (U, V)_{\text{user}}$$

Results

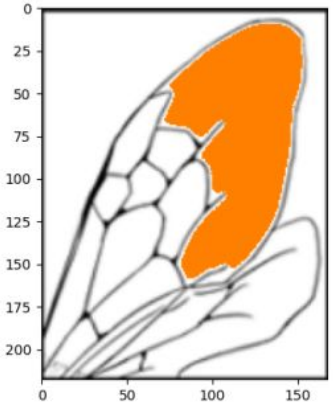
Input image



Stroke



Border



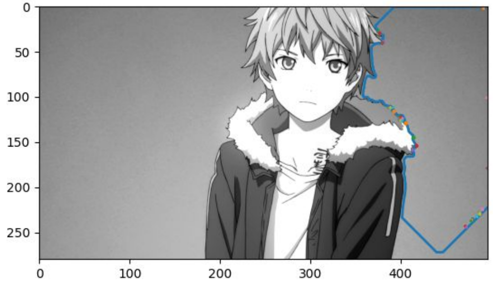
Coloured Image

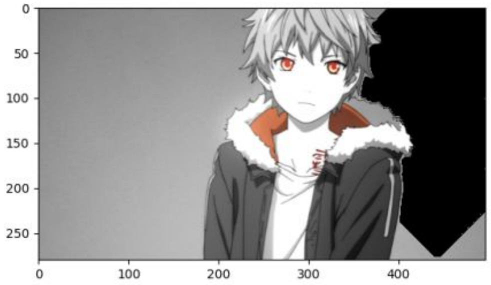
Results

Input image

Stroke

Border





Coloured Image

Division of work

- Rutvij: pattern continuous, color replacement
- Tejas: intensity continuous, pattern to shading,
- Ansh: gradient, stroke preserving gui, presentation
- Sagar: convolution, get band, gui, presentation