

```

close all;

I = imread('bull.jpg');

I = im2double(I); %读入图片并转换类型

hsv = rgb2hsv(I); %转成 hsv

M = size(hsv,1);

N = size(hsv,2);

figure(1), imshow(I);

% choose point of interest

h1 = impoint; %在图像上选一个点（取在红布上）

pos1 = wait(h1);

pos1 = round(pos1); %取整，得到点的位置

a = [hsv(pos1(2),pos1(1),1) hsv(pos1(2),pos1(1),2) hsv(pos1(2),pos1(1),3)]; %把点的 hsv 得到，放在 1*3 的向量里

R = 90 / 255; % radius

hue = abs(hsv(:,1) - a(1)); %得到色调的差

hue(hue > 0.5) = 1 - hue(hue > 0.5); %因为在圆周上是循环的，所以色调差不超过 0.5

D = hue.^2 + (hsv(:,2) - a(2)).^2 + (hsv(:,3) - a(3)).^2;

mask = D <= R * R; %设计出一个掩模，通过 mask 可以找到与选定点 hsv 差比较小的点

% 选择感兴趣区域，下面用的是让用户选择一个矩形

figure(2), imshow(mask);

h2 = imrect; %让用户选一个矩形

pos2 = wait(h2);

pos2 = round(pos2);

roi = false(size(mask)); %初始化感兴趣区域，让其全 0

```

```

roi(pos2(2) : pos2(2) + pos2(4),pos2(1) : pos2(1) + pos2(3)) = 1;           %使矩形域里值为 1

mask = mask & roi;                                                         %与运算，得到矩形域里在与
选定点 hsv 值接近的点，只有这些点在新的模板上被赋值为 1

imwrite(mask,'mask0.jpg');

J0 = hsv;

J0(:,1) = zeros(M,N);

J0(:,2) = zeros(M,N);                                                     %把 hsv 图的 hue 和 saturation 都
变成 0，使整个图变成灰度图

idx = find(mask);                                                         %找出矩形域内的“红点”

J0(idx) = hsv(idx);

J0(idx + M * N) = hsv(idx + M * N);                                       %对 hsv 前 2 个分量分别还原，把
上述点还原为彩色

J = hsv2rgb(J0);                                                         %转回 rgb 图像

figure(3),imshow(J, []);

imwrite(J,'bull_final0.jpg');

```

Published with MATLAB® R2015b