西安交通大学

**计算机视觉与**

**模式识别**

计算机53班

龙思宇

2150500103

1. 填充“add the code here"部分缺失的内容，调试通程序

carv.m填充部分

% remove the horizontal seams

for i = 2 : nr + 1

%generate the energy map

e = genEngMap(TI{i - 1,1});

%dynamic programming matrix

[My,Tby] = cumMinEngHor(e);

[TI{i, 1}, E,~] = rmHorSeam(TI{i-1, 1}, My, Tby);

%accumulate the energy

T(i,1) = T(i - 1,1) + E;

end

% remove the vertical seams

for i = 2 : nc+1

e = genEngMap(TI{1, i-1});

[Mx,Tbx] = cumMinEngVer(e);

[TI{1, i}, E,~] = rmVerSeam(TI{1, i-1}, Mx, Tbx);

T(1, i) = T(1, i-1) + E;

end

% do the dynamic programming

for i = 2 : nr+1

for j = 2 : nc+1

e = genEngMap(TI{i-1, j});

[My, Tby] = cumMinEngHor(e);

[Iy, Ey,~] = rmHorSeam(TI{i-1, j}, My, Tby);

e = genEngMap(TI{i, j-1});

[Mx, Tbx] = cumMinEngVer(e);

[Ix, Ex,~] = rmVerSeam(TI{i, j-1}, Mx, Tbx);

if T(i, j-1) + Ex < T(i-1, j) + Ey

TI{i, j} = Ix;

T(i ,j) = T(i, j-1) + Ex;

% inherite from row direction

else

TI{i, j} = Iy;

T(i, j) = T(i-1, j) + Ey;

% inherite from col direction

end

% suppress the memory for recording intermediate results

TI{i-1,j} = [];

end

end

cumMinEngHor.m填充部分

for i = 2 : nx

for j = 1 : ny

if j == 1

[val,index] = min([My(j,i -1) My(j + 1,i - 1)]);

My(j,i)= e(j,i) + val;

index = index - 1;

Tby(j,i) = index;

elseif j == ny

[val,index] = min([My(j - 1,i - 1) My(j,i -1)]);

My(j,i)= e(j,i) + val;

index = index - 2;

Tby(j,i) = index;

else

[val,index] = min([My(j - 1,i - 1) My(j,i -1) My(j + 1,i - 1)]);

My(j,i)= e(j,i) + val;

index = index - 2;

Tby(j,i) = index;

end

end

end

rmHorSeam.m填充部分

[val,index] = min(My(:,end));

E = val;

for i = nx :-1 :2

Iy(1:index - 1,i,:) = I(1:index - 1,i,:);

Iy(index:end,i,:) = I(index + 1:end,i,:);

rmIdx(1,i) = index;

if Tby(index,i) == -1

index = index - 1;

elseif Tby(index,i) == 1

index = index + 1;

end

end

Iy(1:index - 1,1,:) = I(1:index - 1,1,:);

Iy(index:end,1,:) = I(index + 1:end,1,:);

rmIdx(1,1) = index;

cumMinEngVer.m填充部分

for j = 2 : ny

for i = 1 : nx

if i == 1

[val,index] = min([Mx(j - 1,i) Mx(j - 1,i + 1)]);

Mx(j,i)= e(j,i) + val;

index = index - 1;

Tbx(j,i) = index;

elseif i == nx

[val,index] = min([Mx(j - 1,i - 1) Mx(j - 1,i)]);

Mx(j,i)= e(j,i) + val;

index = index - 2;

Tbx(j,i) = index;

else

[val,index] = min([Mx(j - 1,i - 1) Mx(j - 1,i) Mx(j - 1,i + 1)]);

Mx(j,i)= e(j,i) + val;

index = index - 2;

Tbx(j,i) = index;

end

end

end

rmVerSeam.m填充部分

%% Add your code here

[val,index] = min(Mx(end,:));

E = val;

for i = ny :-1 :2

Ix(i,1:index - 1,:) = I(i,1:index - 1,:);

Ix(i,index:end,:) = I(i,index + 1:end,:);

rmIdx(i,1) = index;

if Tbx(i,index) == -1

index = index - 1;

elseif Tbx(i,index) == 1

index = index + 1;

end

end

Ix(1,1:index - 1,:) = I(1,1:index - 1,:);

Ix(1,index:end,:) = I(1,index + 1:end,:);

rmIdx(1,1) = index;

carvAdd.m

%% my code

%add the horizontal seams

for i = 2 : nr + 1

%generate the energy map

e = genEngMap(TI{i - 1,1});

%dynamic programming matrix

[My,Tby] = cumMinEngHor(e);

[TI{i, 1}, E] = addHorSeam(TI{i-1, 1}, My, Tby);

%accumulate the energy

T(i,1) = T(i - 1,1) + E;

end

%add the vertical seams

for i = 2 : nc+1

e = genEngMap(TI{1, i-1});

[Mx,Tbx] = cumMinEngVer(e);

[TI{1, i}, E] = addVerSeam(TI{1, i-1}, Mx, Tbx);

T(1, i) = T(1, i-1) + E;

end

for i = 2 : nr+1

for j = 2 : nc+1

e = genEngMap(TI{i-1, j});

[My, Tby] = cumMinEngHor(e);

[Iy, Ey] = addHorSeam(TI{i-1, j}, My, Tby);

e = genEngMap(TI{i, j-1});

[Mx, Tbx] = cumMinEngVer(e);

[Ix, Ex] = addVerSeam(TI{i, j-1}, Mx, Tbx);

if T(i, j-1) + Ex < T(i-1, j) + Ey

TI{i, j} = Ix;

T(i ,j) = T(i, j-1) + Ex;

% inherite from row direction

else

TI{i, j} = Iy;

T(i, j) = T(i-1, j) + Ey;

% inherite from col direction

end

% suppress the memory for recording intermediate results

TI{i-1,j} = [];

end

end

addHorSeam.m

function [Iy, E] = addHorSeam(I, My, Tby)

% I is the image. Note that I could be color or grayscale image.

% My is the cumulative minimum energy map along horizontal direction.

% Tby is the backtrack table along horizontal direction.

% Iy is the image removed one row.

% E is the cost of seam removal

[ny, nx, nz] = size(I);

Iy = uint8(zeros(ny + 1,nx,nz));

[val,index] = min(My(:,end));

E = val;

for i = nx :-1 :2

Iy(1:index,i,:) = I(1:index,i,:);

Iy(index + 1,i,:) = I(index,i,:);

Iy(index + 2:end,i,:) = I(index + 1:end,i,:);

if Tby(index,i) == -1

index = index - 1;

elseif Tby(index,i) == 1

index = index + 1;

end

end

Iy(1:index,1,:) = I(1:index,1,:);

Iy(index + 1,1,:) = I(index,1,:);

Iy(index + 2:end,1,:) = I(index + 1:end,1,:);

End

addHorSeam.m

function [Ix, E] = addVerSeam(I, Mx, Tbx)

% I is the image. Note that I could be color or grayscale image.

% Mx is the cumulative minimum energy map along vertical direction.

% Tbx is the backtrack table along vertical direction.

% Ix is the image removed one column.

% E is the cost of seam removal

[ny, nx, nz] = size(I);

Ix = uint8(zeros(ny, nx + 1, nz));

[val,index] = min(Mx(end,:));

E = val;

for i = ny :-1 :2

Ix(i,1:index,:) = I(i,1:index,:);

Ix(i,index + 1,:) = I(i,index,:);

Ix(i,index + 2:end,:) = I(i,index + 1:end,:);

if Tbx(i,index) == -1

index = index - 1;

elseif Tbx(i,index) == 1

index = index + 1;

end

end

Ix(1,1:index,:) = I(1,1:index,:);

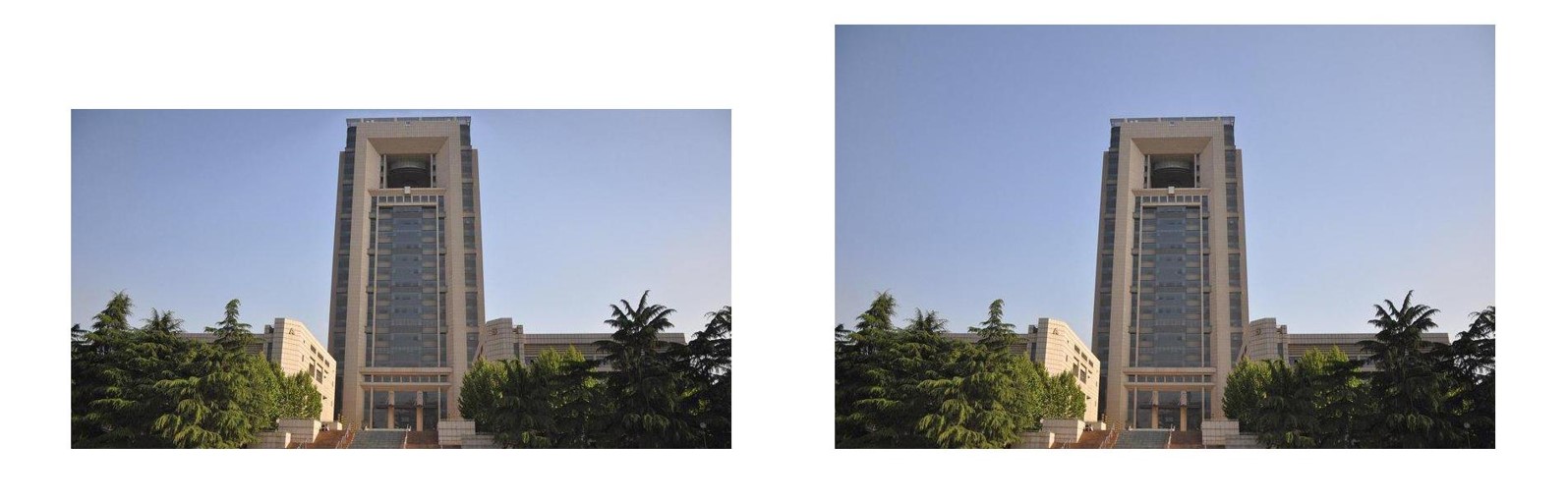
Ix(1,index + 1,:) = I(1,index,:);

Ix(1,index + 2:end,:) = I(1,index + 1:end,:);

end

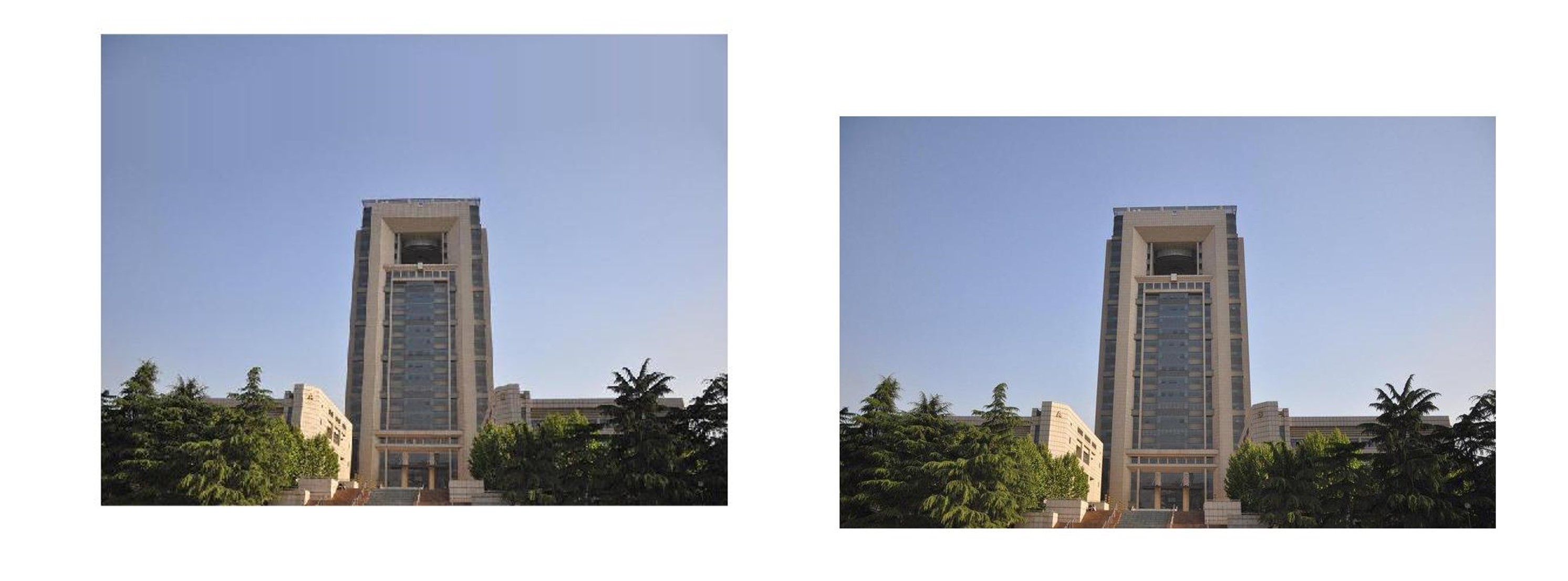
1. 找到自己拍摄的两幅照片，实现图像的缩放和图像的膨胀

图像的缩放



左边是缩放后的图片，右边是原图。

图像的膨胀



左边是膨胀后的图片，右边是原图。

1. 找到自己拍摄的两幅照片，设置需要抹掉的照片部分，通过carving实现物体的擦除

Carv\_with\_mask源码

function [Ic, T] = carv\_with\_mask(I, nr, nc,mask,r\_s)

% I is the image being resized

% [nr, nc] is the numbers of rows and columns to remove.

% Ic is the resized image

% T is the transport map

% mask is where you want to remove or save

T = zeros(nr+1, nc+1);

TI = cell(nr+1, nc+1);

TI{1,1} = I;

Masks = cell(nr + 1,nc + 1);

Masks{1,1} = mask;

if ~exist('r\_s','var')

r\_s = 1;

end

%% Add your code here

% remove the horizontal seams

for i = 2 : nr + 1

%generate the energy map

e = genEngMap(TI{i - 1,1});

if r\_s == 1

e = e - e .\* Masks{i - 1,1};

elseif r\_s == 0

e = e - e .\* Masks{i - 1,1} + 255 \* Masks{i - 1,1};

end

%dynamic programming matrix

[My,Tby] = cumMinEngHor(e);

[TI{i, 1}, E,~,Masks{i,1}] = rmHorSeam\_with\_mask(TI{i-1, 1}, My, Tby,Masks{i - 1,1});

%accumulate the energy

T(i,1) = T(i - 1,1) + E;

end

% remove the vertical seams

for i = 2 : nc+1

e = genEngMap(TI{1, i-1});

if r\_s == 1

e = e - e .\* Masks{1,i - 1};

elseif r\_s == 0

e = e - e .\* Masks{1,i - 1} + 255 \* Masks{1,i - 1};

end

[Mx,Tbx] = cumMinEngVer(e);

[TI{1, i}, E,~,Masks{1,i}] = rmVerSeam\_with\_mask(TI{1, i-1}, Mx, Tbx,Masks{1,i - 1});

T(1, i) = T(1, i-1) + E;

end

% do the dynamic programming

for i = 2 : nr+1

for j = 2 : nc+1

e = genEngMap(TI{i-1, j});

if r\_s == 1

e = e - e .\* Masks{i - 1,j};

elseif r\_s == 0

e = e - e .\* Masks{i - 1,j} + 255 \* Masks{i - 1,j};

end

[My, Tby] = cumMinEngHor(e);

[Iy, Ey,~,mask\_y] = rmHorSeam\_with\_mask(TI{i-1, j}, My, Tby,Masks{i - 1,j});

e = genEngMap(TI{i, j-1});

if r\_s == 1

e = e - e .\* Masks{i,j - 1};

elseif r\_s == 0

e = e - e .\* Masks{i,j - 1} + 255 \* Masks{i,j - 1};

end

[Mx, Tbx] = cumMinEngVer(e);

[Ix, Ex,~,mask\_x] = rmVerSeam\_with\_mask(TI{i, j-1}, Mx, Tbx,Masks{i,j - 1});

if T(i, j-1) + Ex < T(i-1, j) + Ey

TI{i, j} = Ix;

T(i ,j) = T(i, j-1) + Ex;

Masks{i,j} = mask\_x;

% inherite from row direction

else

TI{i, j} = Iy;

T(i, j) = T(i-1, j) + Ey;

Masks{i,j} = mask\_y;

% inherite from col direction

end

% suppress the memory for recording intermediate results

TI{i-1,j} = [];

Masks{i - 1,j} = [];

end

end

Ic = TI{nr+1,nc+1};

end



左边是擦除的图像，右边是原图像。

1. 找到自己拍摄的两幅照片，设置需要保留的照片部分，通过carving实现物体的保留



左边是处理后的图片，右边是原图。