Project 5 Code

**%--------------------------------------------------------------------------**

**% Jump Diffusion MCMC Algorithm**

**%--------------------------------------------------------------------------**

**% TuannD**

**% 04/04/13**

**% @see: Computer Vision slides**

**%--------------------------------------------------------------------------**

**clc;**

**clear all;**

**close all;**

**rng('shuffle');**

**fprintf('JUMP DIFFUSION MCMC ALGORITHM\n');**

**fprintf('Copyright-TuanND-04/2013\n');**

**%% PARAMETERS**

**lambda = 8;**

**Nsteps = 30; %Number of sampling steps**

**K\_MAX = 12; %Max (Number of objects)**

**init\_k = 9;**

**M\_BURN\_IN = 3;**

**STEP\_BURN\_IN = 2;**

**%% Step 0: Prepare data**

**% load image/target**

**fprintf('Loading image....\n');**

**imgpath = 'discs8.bmp';**

**[pathstr, name, ext] = fileparts(imgpath);**

**img=double(imread(imgpath))/255;**

**target=double(imread('target.bmp'))/255;**

**[rows cols] = size(img);**

**fprintf(['Loading image ', imgpath,' and target is done\n']);**

**%% Step 1: Initialization**

**%Initialize locations of k hypothesized objects and the maximum order Kmax.**

**allStartTime = tic;**

**num\_obj = zeros(Nsteps, 1);**

**num\_obj(1) = init\_k;**

**Oxy = cell(Nsteps, 1);%Objects position**

**Oxy{1} = [randi(rows, num\_obj(1) , 1) randi(cols, num\_obj(1), 1)];%k hypothesized object locations**

**showImg = drawcircle(img, (Oxy{1})', num\_obj(1));**

**figure(1); imshow(showImg);**

**obj\_fn = zeros(Nsteps, 1);**

**obj\_fn(1) = likelihood(img, target, Oxy{1}, num\_obj(1)) \* poisspdf(num\_obj(1), lambda); %aposterior for evaluation**

**fprintf('Iteration[%02u]--Discs:[%02u]--OBJ\_FN:[%d]--Duration[%5.5f]\n', 1, num\_obj(1), obj\_fn(1), toc(allStartTime));**

**Imframe(1:rows,1:cols,1)=showImg;**

**Imframe(1:rows,1:cols,2)=showImg;**

**Imframe(1:rows,1:cols,3)=showImg;**

**videoseg(1)=im2frame(Imframe); % make the first frame**

**%% Step 2: Iteration**

**%for i=1:N**

**for i = 2:Nsteps**

**iterStartTime = tic;**

**% Draw a sample a~U(0,1)**

**a = rand(1);**

**% If a<0.33 and k>1 (jump by -1)**

**fprintf('Iteration[%02u]--a:[%1.2f]',i, a);**

**if a < 0.33 && num\_obj(i-1) > 1**

**% k=k-1;**

**fprintf('--Jump-1');**

**num\_obj(i) = num\_obj(i-1) - 1;%**

**% else if a<0.66 and k<Kmax (jump by +1)**

**elseif a < 0.66 && num\_obj(i-1) < K\_MAX**

**% k=k+1;**

**fprintf('--Jump+1');**

**num\_obj(i) = num\_obj(i-1) + 1;**

**% else (no jump)**

**else**

**fprintf('--Jump+0');**

**num\_obj(i) = num\_obj(i-1);**

**% End**

**end**

**fprintf('--Discs:[%02u]',num\_obj(i));**

**% MCMC Gibbs sampling**

**[Oxy{i} VS] = gibss\_sampling(img, target, num\_obj(i));**

**% Accept or reject by Metropolis Sampling**

**obj\_fn(i) = likelihood(img, target, Oxy{i}, num\_obj(i)) \* poisspdf(num\_obj(i), lambda);**

**pa\_jump = min(obj\_fn(i)/obj\_fn(i-1), 1);%acceptance probability for jump/no jump (note that poisspdf is the same if no jump)**

**fprintf('\n\t\tOBJ\_FN:[%d]--AcceptRate:[%2.2f]',obj\_fn(i), pa\_jump);**

**u0 = rand(1);**

**if pa\_jump > u0 %Accept jump with prob. of p\_jump**

**%keep the jump**

**fprintf('--Accept');**

**num\_frame = length(VS);**

**cur\_num = length(videoseg);**

**videoseg(cur\_num + 1:cur\_num + num\_frame) = VS;**

**else %Reject jump**

**fprintf('--Reject');**

**%duplicate previous step**

**num\_obj(i) = num\_obj(i-1);**

**Oxy{i} = Oxy{i-1};**

**obj\_fn(i) = obj\_fn(i-1);**

**end**

**fprintf('--Duration:[%3.3f]\n', toc(iterStartTime));**

**end**

**%% Step 3**

**% Select samples after M iterations (burn-in);**

**% Obtain a set of samples with certain step size.**

**fprintf('Burning to get Experimental Result...\n');**

**BI\_AOxy = Oxy(M\_BURN\_IN + 1:STEP\_BURN\_IN:Nsteps);**

**bi\_num\_obj = num\_obj(M\_BURN\_IN + 1:STEP\_BURN\_IN:Nsteps);**

**%% Step 4**

**% Compute the mean estimate of the object number k\***

**final\_num\_obj = round(mean(bi\_num\_obj));**

**fprintf('Number of objects: %02u\n', final\_num\_obj);**

**% For samples with k\*, re-order all object locations and compute the mean location for each object.**

**K\_BI\_AOxy = BI\_AOxy(bi\_num\_obj == final\_num\_obj);**

**num\_sp = length(K\_BI\_AOxy);**

**S = zeros(final\_num\_obj, 2, num\_sp);**

**for i = 1:num\_sp**

**S(:,:,i) = K\_BI\_AOxy{i};**

**end**

**OS = reorder\_samples(S);**

**final\_Oxy = round(mean(OS,3));**

**fprintf('Object locations:\n');**

**reshape(final\_Oxy, 2, [])'**

**%% Show the result**

**fprintf('See figure(2) for final result');**

**final\_img = drawcircle(img, final\_Oxy, final\_num\_obj);**

**figure(2);imshow(final\_img);**

**cur\_num\_frame = length(videoseg);**

**movie2avi(videoseg(1:(cur\_num\_frame)),['JD\_MCMC\_',name,'.avi'],'FPS',10,'COMPRESSION','None');**

**%% Plot**

**figure; plot(num\_obj);title('Number of objects vs. iteration');**

**figure; plot(obj\_fn); title('Object function vs. iteration');**

**function [Mxy videoseg] = gibss\_sampling(img, tg, M)**

**% GIBSS\_SAMPLING do gibs sampling**

**% @see Computer Vision slides**

**% TuanND**

**% 04/06**

**% Input:**

**% img: image on which have objects need to be dectected**

**% target: target knowledge**

**% k: number of objects**

**fprintf('\n\t\tGibss:[Step/Discs-100/25]');**

**%PARAMETER**

**T = 100;%Number of MCMC steps**

**M\_BURN\_IN = 50;**

**N\_BURN\_IN = 2;**

**K\_MAX = 300;**

**[rows cols] = size(img);**

**%1. Initialize {zi: i = 1, ..., M}**

**AOxy = zeros(M, 2, T);**

**AOxy(:,:,1) = [randi(rows, M, 1) randi(cols, M, 1)];%All object position**

**Cur\_Oxy = AOxy(:,:,1);**

**showImg = drawcircle(img, Cur\_Oxy, M);**

**figure(1); imshow(showImg);**

**L1 = likelihood(img, tg, Cur\_Oxy, M);**

**Imframe(1:rows,1:cols,1)=showImg;**

**Imframe(1:rows,1:cols,2)=showImg;**

**Imframe(1:rows,1:cols,3)=showImg;**

**videoseg(1) = im2frame(Imframe); % make the first frame**

**for t = 2:T**

**for i = 1:M**

**fprintf('\b\b\b\b\b\b\b');**

**fprintf('%03u/%02u]', t, i);**

**Oxy = Cur\_Oxy(2\*i-1:2\*i);%init position of ith object**

**for j = 1:K\_MAX**

**%Sampling ith variable**

**Dxy = Oxy + round(randn(1,2)\*20);**

**Dxy=clip(Dxy,1,rows);% make sure the position in the image**

**New\_Cur\_Oxy = Cur\_Oxy;**

**New\_Cur\_Oxy(2\*i-1:2\*i) = Dxy;**

**L2=likelihood(img,tg,New\_Cur\_Oxy,M);% evaluate the likelihood**

**v=min(1,L2/L1); % compute the acceptance ratio**

**u=rand; % draw a sample uniformly in [0 1]**

**if v>u**

**Oxy = Dxy;% accept the move**

**Cur\_Oxy = New\_Cur\_Oxy;**

**L1 = L2;**

**% showImg = drawcircle(img, Cur\_Oxy, M);**

**% figure(1); imshow(showImg);**

**else**

**end**

**end**

**AOxy(:,:, t) = Cur\_Oxy;**

**showImg = drawcircle(img, Cur\_Oxy, M);**

**figure(1); imshow(showImg);**

**Imframe(1:rows,1:cols,1)=showImg;**

**Imframe(1:rows,1:cols,2)=showImg;**

**Imframe(1:rows,1:cols,3)=showImg;**

**videoseg((t-2)\*M + i + 1) = im2frame(Imframe);**

**end**

**end**

**fprintf('...Burning...');**

**%Burn-in**

**% S = AOxy(:,:, M\_BURN\_IN+1:N\_BURN\_IN:T);%do burn-in, drop fist M samples, keep N-steps samples**

**% OS = reorder\_samples(S);**

**% Mxy = round(mean(OS, 3));**

**Mxy = Cur\_Oxy;**

**showImg = drawcircle(img, Mxy, M);figure(1);imshow(showImg);**

**Imframe(1:rows,1:cols,1)=showImg;**

**Imframe(1:rows,1:cols,2)=showImg;**

**Imframe(1:rows,1:cols,3)=showImg;**

**videoseg((T-1) \* M + 2) = im2frame(Imframe);**

**fprintf('Done Gibss');**

**end**

**function OS = reorder\_samples(S)**

**num\_sp = size(S, 3);%number of samples**

**OS = S;**

**for i = 1:num\_sp**

**Si = S(:,:,i);**

**Si = reshape(Si, 2, []);**

**D = Si(1, :).^2 + Si(2,:).^2;**

**[B IX] = sort(D,'ascend');**

**Si = Si(:, IX);**

**Si = reshape(Si, [], 2);**

**OS(:,:,i) = Si;**

**end**

**end**