Invariant GHT code

Source: http://sci.tech−archive.net/Archive/sci.image.processing/2006−08/msg00331.html

- *From*: "Ashwin Nanjappa" <ashwin.n@xxxxxxxxx>
- *Date*: 15 Aug 2006 02:19:52 −0700

Hi all,

I'm studying the Generalized Hough Transform (GHT) from the book _Feature Extraction And Image Processing_ [1]. The book has Matlab code for the Invariant GHT. I'm using that, but there seems to be a bug in the book's code. I'm not seeing any significant peaks in the 2D accumulator array produced by this GHT. I've rechecked the code many times but am not able to find any error.

Has anyone faced this problem? The code from the book is pasted below this post.

[1] <http://www.ecs.soton.ac.uk/~msn/book/>

TIA,
~ash

```
% ========================================
% Invariant RTable
% Inputs are the desired number of entries in the R−table and the
template
% image. Output is the R−table
function t = rtableInv(entries, img)

% Image size
[rows, cols] = size(img);

% Edges
[mag, ang] = Edges(img);
mag = MaxSupr(mag, ang);

alfa = pi / 4;
d = pi / entries;
s = 0; % Number of entries in the table
t = [];
f = zeros(entries, 1); % Number of entries in the row

% Compute reference point
xr = 0; yr = 0; p = 0;
```

```
for x = 1 : cols
for y = 1 : rows
if (mag(y, x) ~= 0)
xr = xr + x;
yr = yr + y;
p = p + 1;
end
end
end

xr = round(xr / p);
yr = round(yr / p);

% For each edge point
for x = 1:cols
for y = 1:rows
if (mag(y, x) ~= 0)
% Search for the second point
x1 = −1;
y1 = −1;
phi = ang(y, x);
m = tan(phi − alfa);

if (m > −1 & m < 1)
for i = 3:cols
c = x + i;
j = round(m * (c − x) + y);
if (j > 0 & j < rows & c > 0 & c < cols & mag(j, c)
~= 0)
x1 = c;
y1 = j;
i = cols;
end

if (i ~= cols)
c = x − i;
j = round(m * (c − x) + y);
if (j > 0 & j < rows & c > 0 & c < cols &
mag(j, c) ~= 0)
x1 = c;
y1 = j;
i = cols;
end
end
end
else
for j = 3:rows
c = y + j;
i = round(x + (c − y) / m);
if (c > 0 & c < rows & i > 0 & i < cols & mag(c, i)
~= 0)
```

```
x1 = i;
y1 = c;
i = rows;
end

if (i ~= rows)
c = y − j;
i = round(x + (c − y) / m);
if (c > 0 & c < rows & i > 0 & i < cols &
mag(c, i) ~= 0)
x1 = i;
y1 = c;
i = rows;
end
end
end
end

if (x1 ~= −1)
% Compute beta
phi = tan(ang(y, x));
phj = tan(ang(y1, x1));
if ((1 + phi * phj) ~= 0)
beta = atan((phi − phj) / (1 + phi * phj));
else
beta = 1.57;
end

% Compute k
if ((x − xr) ~= 0)
ph = atan((y − yr) / (x − xr));
else
ph = 1.57;
end
k = ph − ang(y, x);

% Insert in the table
i = round((beta + (pi / 2)) / d);
if (i == 0)
i = 1;
end
v = f(i) + 1;

if (v > s)
s = s + 1;
t(:, s) = zeros(entries, 1);
end

t(i, v) = k;
f(i) = f(i) + 1;
end
```

```
end
end
end

% ========================================

% Invariant Generalized Hough Transform
% Input is the image and the R−table (produced by the function above)
% Output is the 2D accumulator array "acc"
function ghtInv(img, Rtable)

% Image size
[rows, cols] = size(img);

% Table size
[rowsT, colsT] = size(Rtable);
d = pi / rowsT;

% Edges
[mag, ang] = Edges(img);
mag = MaxSupr(mag, ang);

alfa = pi / 4;

% Accumulator
acc = zeros(rows, cols);

% For each edge point
for x = 1:cols
for y = 1:rows
if (mag(y, x) ~= 0)
% Search for the second point
x1 = −1;
y1 = −1;
phi = ang(y, x);
m = tan(phi − alfa);

if (m > −1 & m < 1)
for i = 3:cols
c = x + i;
j = round(m * (c − x) + y);
if (j > 0 & j < rows & c > 0 & c < cols & mag(j, c)
~= 0)
x1 = c;
y1 = j;
i = cols;
end

if (i ~= cols)
c = x − i;
j = round(m * (c − x) + y);
```

```
if (j > 0 & j < rows & c > 0 & c < cols &
mag(j, c) ~= 0)
x1 = c;
y1 = j;
i = cols;
end
end
end
else
for j = 3:rows
c = y + j;
i = round(x + (c − y) / m);
if (c > 0 & c < rows & i > 0 & i < cols & mag(c, i)
~= 0)
x1 = i;
y1 = c;
i = rows;
end

if (i ~= rows)
c = y − j;
i = round(x + (c − y) / m);
if (c > 0 & c < rows & i > 0 & i < cols &
mag(c, i) ~= 0)
x1 = i;
y1 = c;
i = rows;
end
end
end
end

if (x1 ~= −1)
% Compute beta
phi = tan(ang(y, x));
phj = tan(ang(y1, x1));
if ((1 + phi * phj) ~= 0)
beta = atan((phi − phj) / (1 + phi * phj));
else
beta = 1.57;
end

i = round((beta + (pi / 2)) / d);
if (i == 0)
i = 1;
end

% Search for k
for j = 1:colsT
if (Rtable(i, j) == 0)
j = colsT; % No more entries
```

```
else
k = Rtable(i, j);
% Lines of votes
m = tan(k + ang(y, x));
if (m > −1 & m < 1)
for x0 = 1:cols
y0 = round(y + m * (x0 − x));
if (y0 > 0 & y0 < rows)
acc(y0, x0) = acc(y0, x0) + 1;
end
end
else
for y0 = 1:rows
x0 = round(x + (y0 − y) / m);
if (x0 > 0 & x0 < cols)
acc(y0, x0) = acc(y0, x0) + 1;
end
end
end
end
end
end
end
end
end

% =========================================

.
```