

TABLE TENNIS BALL TRACKING

Ball Constants

- Size → Set By The User
- Perimeter
- Radius Range
- Roi
- ADAPTIVE Roi
- Blur kernel, closing kernel

Color of the Ball

* FIRST-PASS THRESHOLD

* SECOND-PASS "

HSV-MIN/MAX

* MAIN LOOP

while (video has frame)

- SKIP INITIAL FRAMES

- READ FRAMES

- IF FIRST-FRAME

- USER SELECTS Ball Initial Pos

- CROP ROI

- READ FRAME

• ELSE IF Ball Previous-found

- CROP ON THAT POS, Calc ROI

- ELSE

ENLARGE ROI RECT,

CROP ON LAST KNOWN POS.

- BLUR IMAGE

- FIRST PASS (COARSE THRESHOLD)

- IF OBJECTS FOUND

CROP A ROI ON EACH OBJECT POS

ELSE

CONTINUE USING frame AS ROI

- SECOND PASS (RELAXED THRESHOLD)

- FOR EACH CANDIDATE Ball

Recalc COORDINATES

Calc ~~next~~ frame time

- TWO STAGE Ball DETECTION

- CHOOSE SMALLEST error AS THE Ball

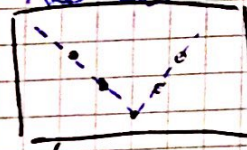
END

* Plot The Trajectory

- Calc Interval
- Use Spline Interpolation
- Plot in the picture

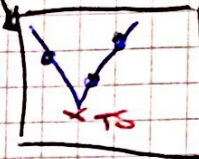
* Calc Time of Bounce

- IF By looking at 5 frames we have the y coord going down 3 times and going up 2 times
- set as a bounce point



intuition

- For each bounce point use spline interpolation to find to approx TS in between the frame that the bounce happened



* First Pass

while candidate not found AND not in max-iteration

- * Color threshold inc
- * Calc number of objects

IF number of objects == 1

candidate-found = true

set as

best-first-pass

best-threshold...

elseif num. objects > 1

IF num objects < best-first-pass (lowest num of objects)

set as

best-first-pass

best-threshold...

increase the threshold

else

decrease the threshold

check edges are ok

LEND

Restore best-threshold

iterate on threshold

* SECOND PASS

FOR EACH ROI

LOAD ROI

GET INITIAL THRESHOLDS

* while AREA-OK = false AND IT < MAX-IT

COLOR THRESHOLD IMG

CALC AREA OF OBJECTS

→ FOR EACH AREA FOUND

IF $(\text{BALL-AREA} - \text{AREA}) / \text{BALL-AREA} < 0.4$

AREA-OK = TRUE

SET AS BEST THRESHOLD

ELSE IF $\text{AREA} > \text{BALL-AREA}$

IF $\text{BEST-IT AREA} > \text{AREA} - \text{BALL-AREA}$
SET AS BEST-IT

INCREASE THE THRESHOLD

ELSE

IF $\text{AREA diff} < \text{BEST-IT AREA}$
SET AS BEST-IT

DECREASE THE THRESHOLD

HANDLE EDGES

ELSE (NO AREA FOUND)

- DECREASE THE THRESHOLD

- HANDLE EDGES

RESTORE BEST THRESHOLD

- SAVE BEST THRESHOLD FOR EACH ROI

- SAVE ROI BINARIZED

→ Fix the code!

→ CHANGE RESTORE LATER

* TWO STAGE Ball Detection

For each Boundary

Calc, Area, Centroid, Bounding Box

FIRST STAGE - Eliminary

- * Check ROUNDED UPPER Contour
 - Calc upper contour
 - fit IN A CIRCLE
 - Calc error
 - IF $error < THRESHOLD$
EUC = TRUE;
- * Fix CENTROID POS
- * IF enough POSITIONS
 - LINEAR extrapolation to predict Ball location
 - Compare
 - CANDIDATE Ball AND predicted Ball location
 - IF $smaller$ THAN THRESHOLD
T = TRUE
 - LAST Ball position AND CANDIDATE Pos.
IF THERE IS MOTION
MC = TRUE
 - Prediction ???
(THE proper way to check if the predictions IS OF A different pos.)
- * IF NOT enough
 - Give A free BOWT
 - CHECK movement

SECOND STAGE - Classfactory

* IF AT LEAST 2 OF THE ABOVE HOLD TRUE

Calc AREA, perimeter, ROUNDED,
maxHeight, maxWith

calc ERROR

compare with current Mv error,
IF smaller Select.