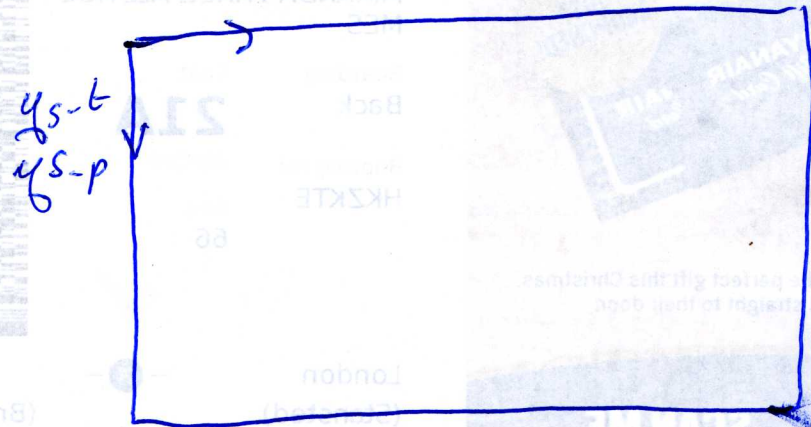
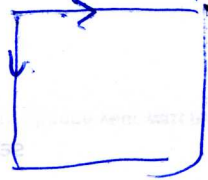


(float)  $ns-t$  (true)  
 (int)  $ns-p$  (pixel)

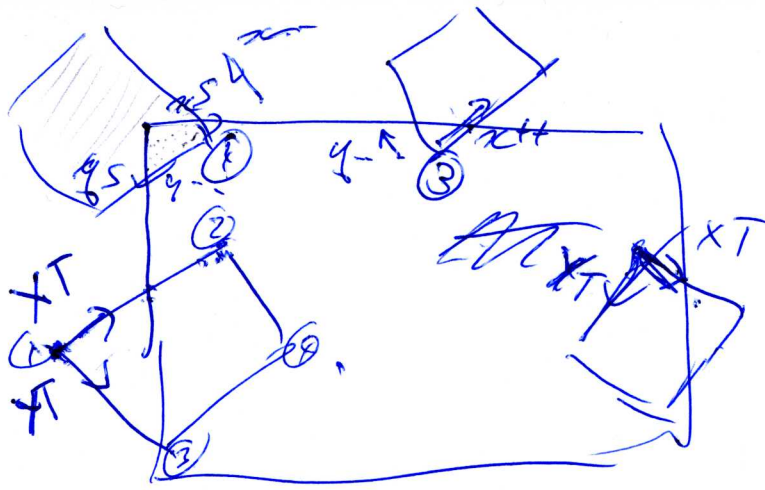
~~ns~~



double  $XT-T \leftarrow true$   
 int  $XT-P \leftarrow pixel$



$\} \rightarrow XT-st$   
 $XT-pt$



2

```
for (int YTsp = 0; YTsp < YT_hedge; YTsp++) {
```

```
    for (int XTsp = 0; XTsp < XT_midle; XTsp++) {
```

```
        if (XTsp > YTsp) { break; }
```

```
    }
```

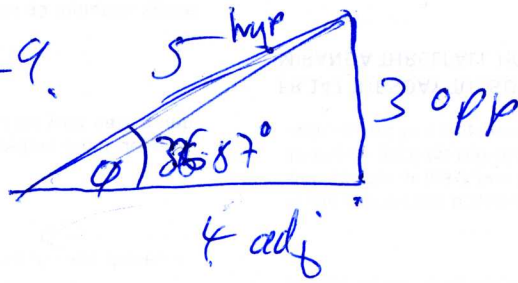
①  $if ((XT_{sp} < 0) || (YT_{sp} > YT_{hedge}))$

② (is ② inside)  
row loop end ← start pc--  
↑ ++

③.

③

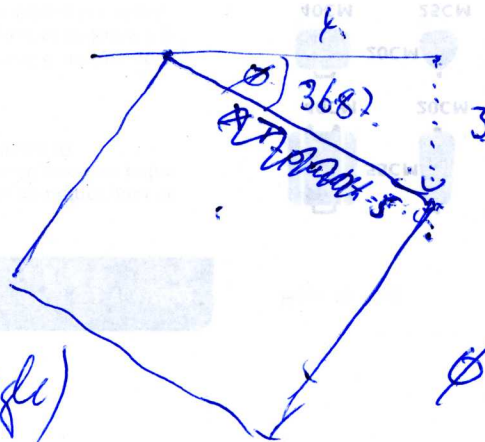
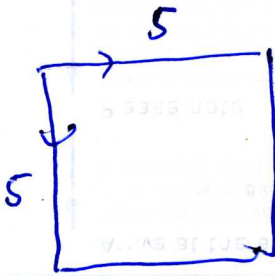
$$25 = 16 + 9$$



$$\sin = \frac{\text{opp}}{\text{hyp}}$$

$$\cos = \frac{\text{adj}}{\text{hyp}}$$

$$\tan = \frac{\text{opp}}{\text{adj}}$$



$$dY = [\text{hyp}] \cdot \sin(\text{angle})$$

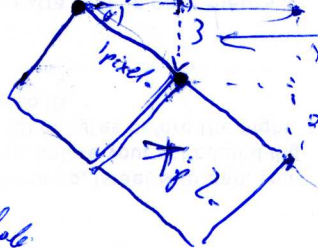
$$dX = [\text{hyp}] \cdot \cos(\text{angle})$$

$$\phi = \tan^{-1}\left(\frac{3}{4}\right)$$

④

$$\text{float } dX = \cos(\text{angle})$$

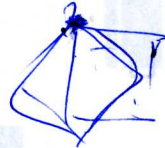
$$\text{float } dY = \sin(\text{angle})$$



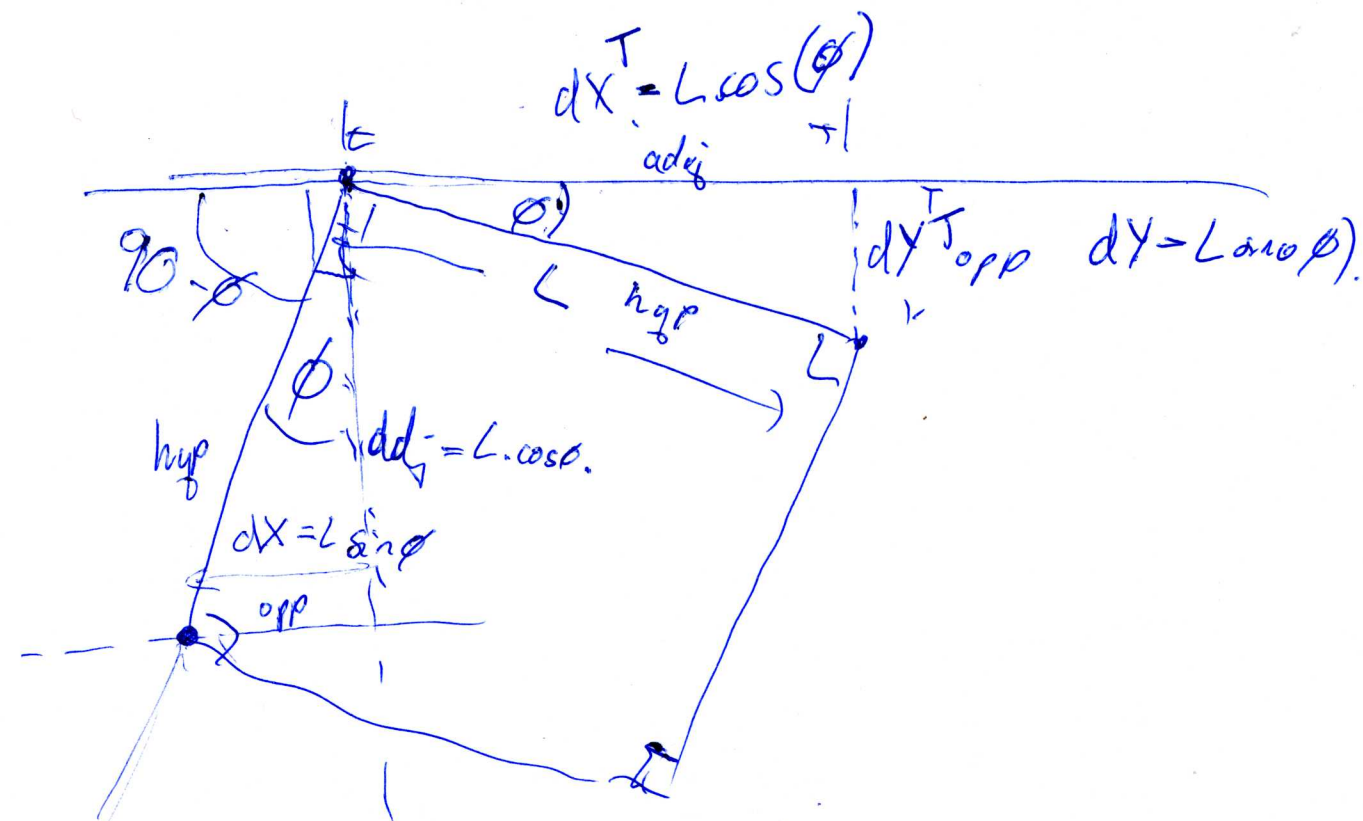
$$x_2 = 4$$

$$y_2 = 3$$

(double) (X.T.P.) (double) dx



5



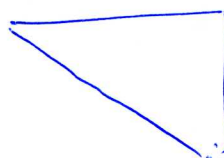
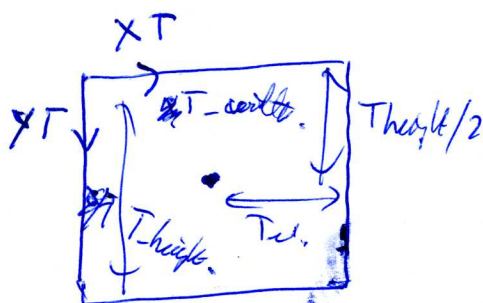
(int) Math. sound { double sum }

7



$y_0$   $x_1$  2. 3. 4 5. 6 7 8 9

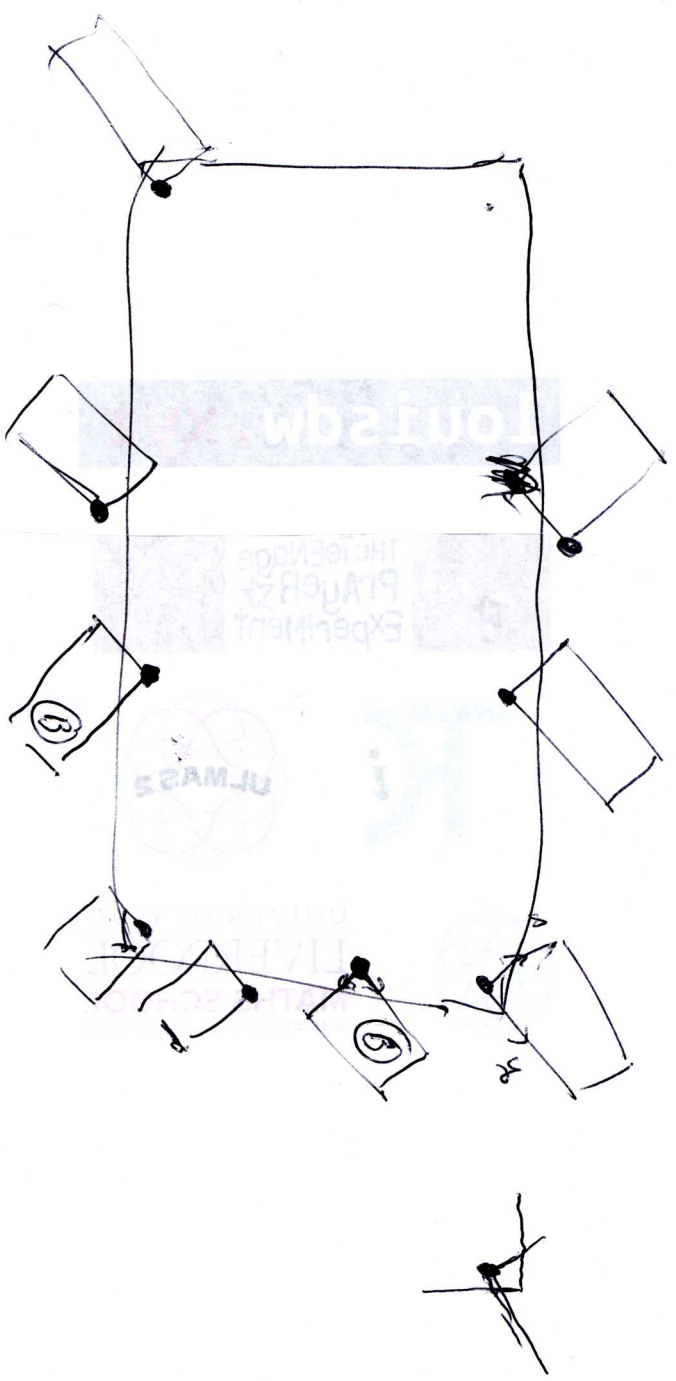
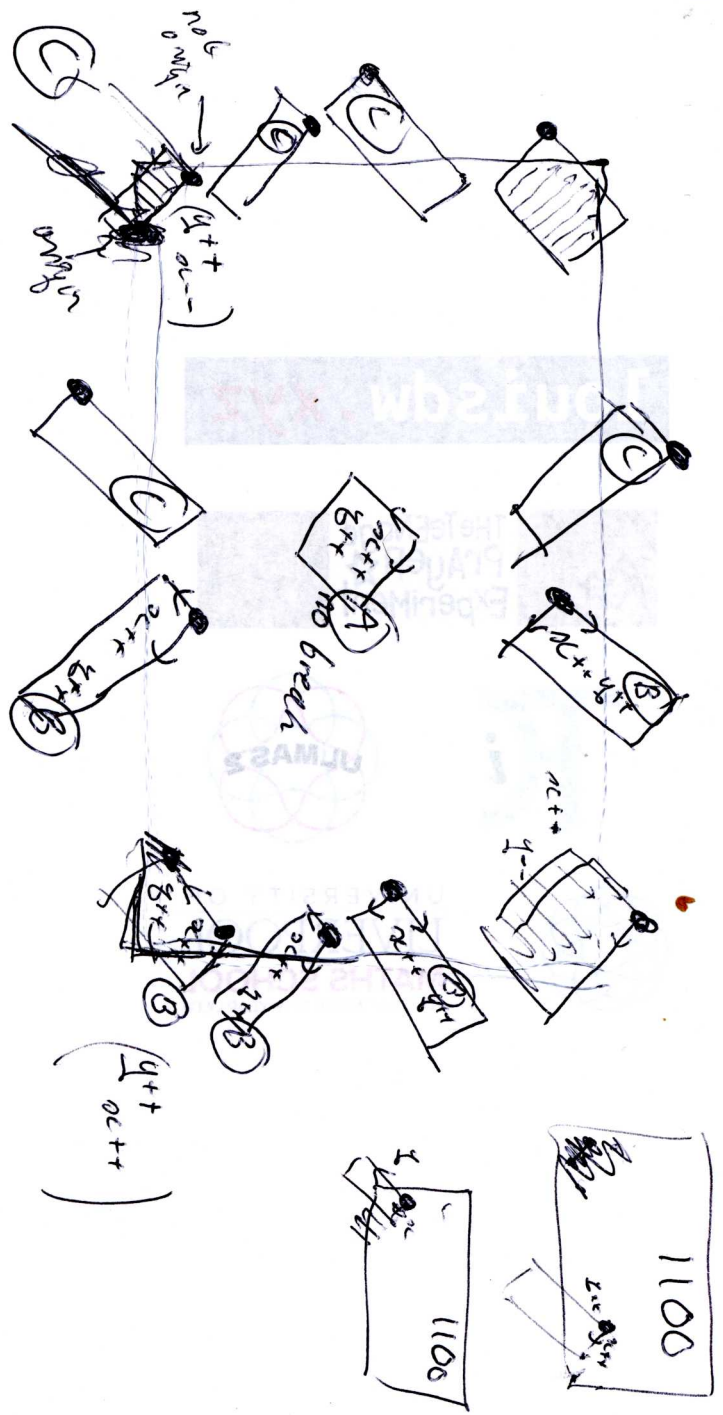
6





8

XIX 1



1 1 1 1 1

0 1 1 1 1

1 0 1 1 1

1 1 1 0 1

1 1 1 1 0

0 0 1 1 1

0 1 1 0 1

0 1 1 1 0

0 1 0 0 1

0 0 0 1 0

0 0 0 0 0

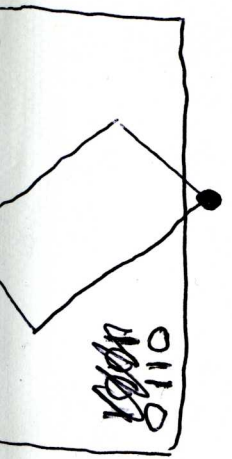
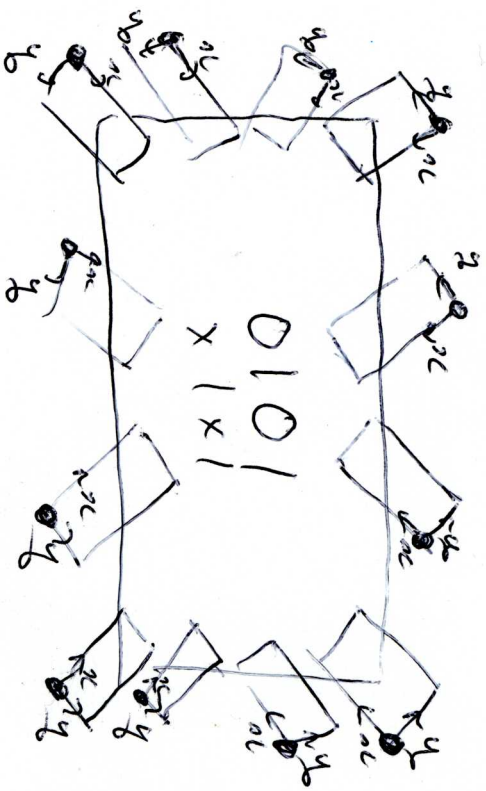
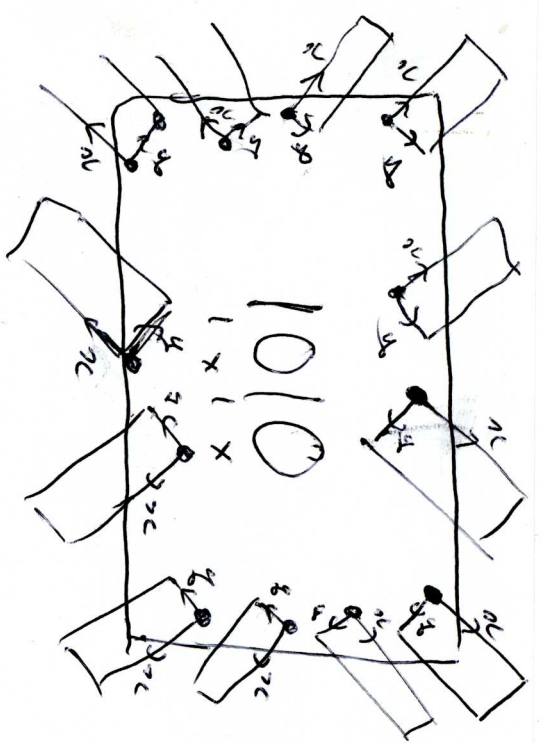
8

1 0 0 1

1 0 1 0

1 0 0 0

1 0 0 0



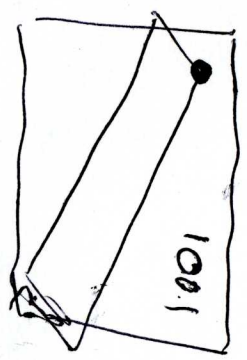
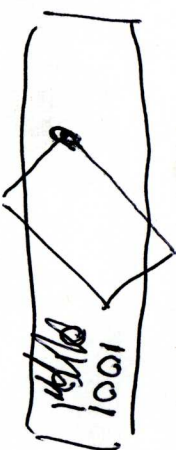
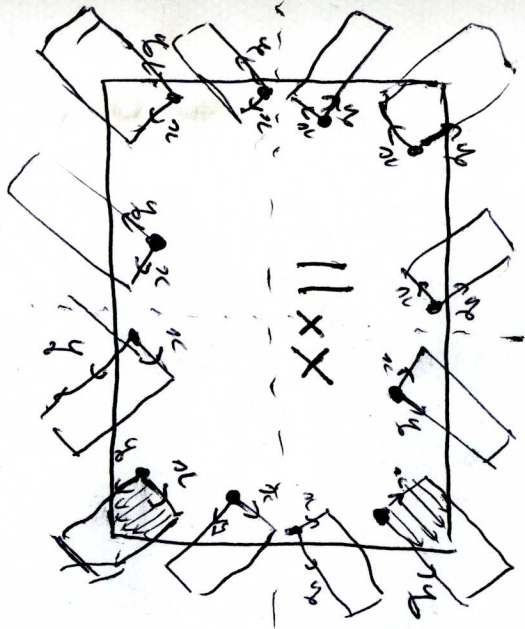
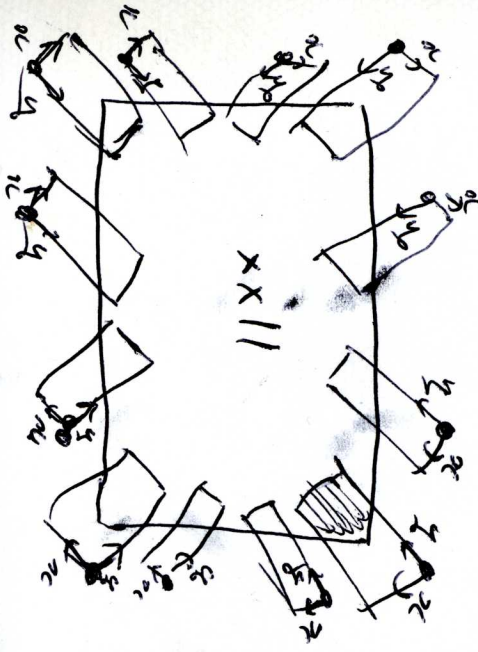
both  
if left corner are in image,  
can always scan  $y^{++}$   
regardless of the rotation or centre.

Q9

if both right corner are in image  
can always scan  $y^{++}$   
regardless of the rotation or centre

isn't possible unless the texture is wide  
compared to screen height.





or  
might high  
1 empow to  
screen width.

97

37-2007



Magnetic Sheet Self Adhesive A4 X 0.75mm

Quantity 1

www.rapidonline.com

Tel: 01206 751166 Fax: 01206 751188

52015

F022C

Rapid

PLEASE  
DO NOT  
BEND

But this is star-  
no more!

4+

if both bottom corners are in image

ac++

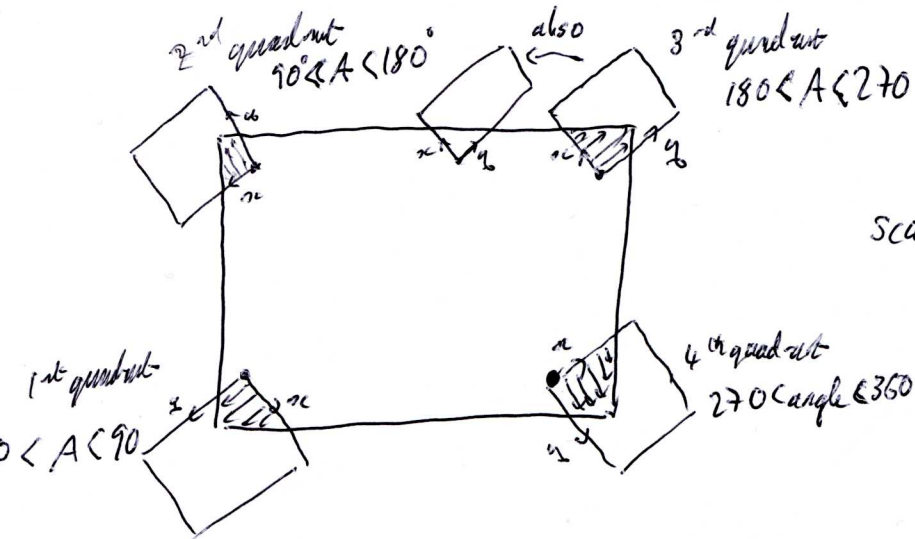
g--

If there is only one corner in the image.

(10)

There are two possibilities in each corner, depending on the orientation of the texture.

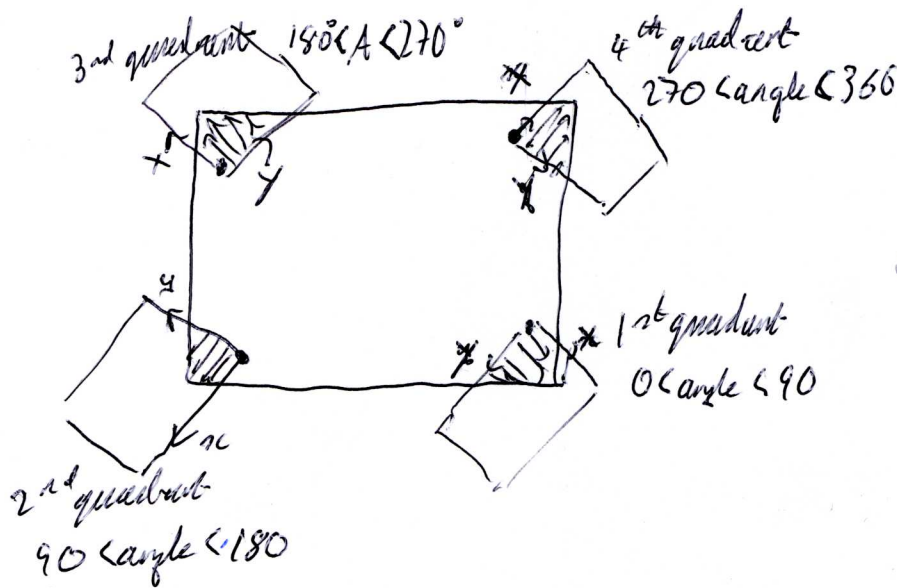
Example: only Top left corner is in image.



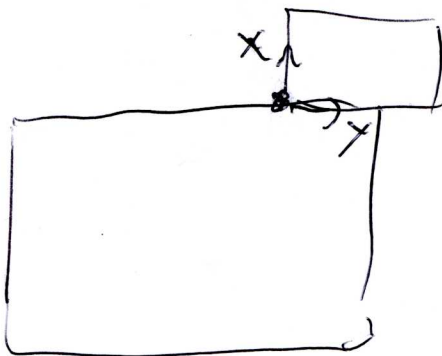
same as  $[1 \times 1]$

Scan from Top Left corner

$y++$  in  $x++$  outer loop



the simple obvious scan as per  $[1 \times 1]$   
scan from top left  $x++$  in  $y++$  outer loop

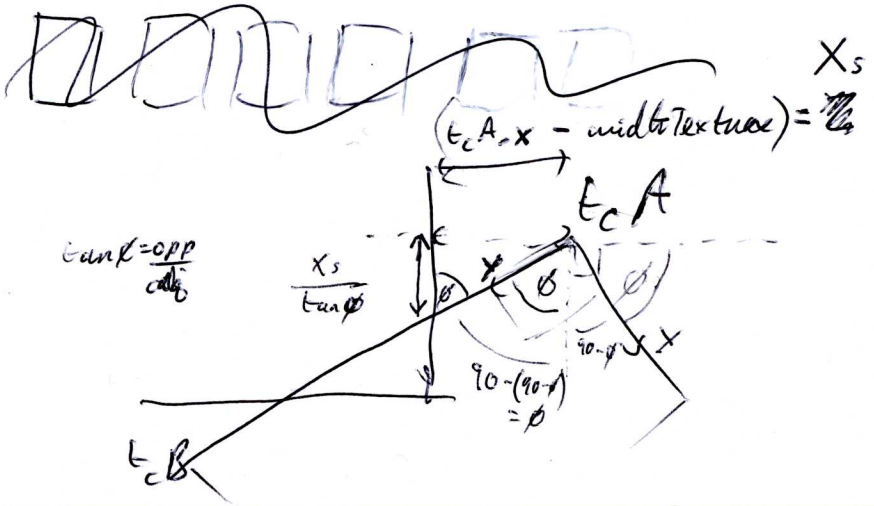
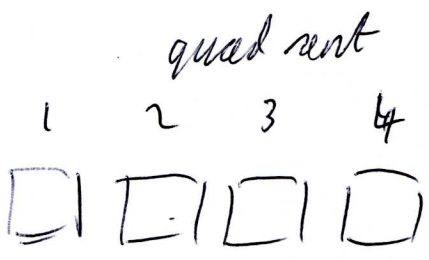
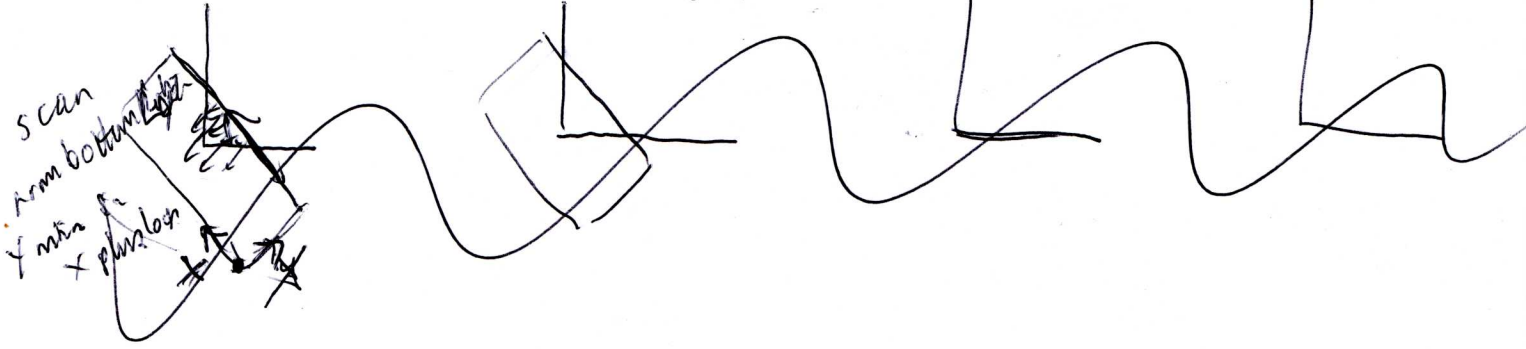
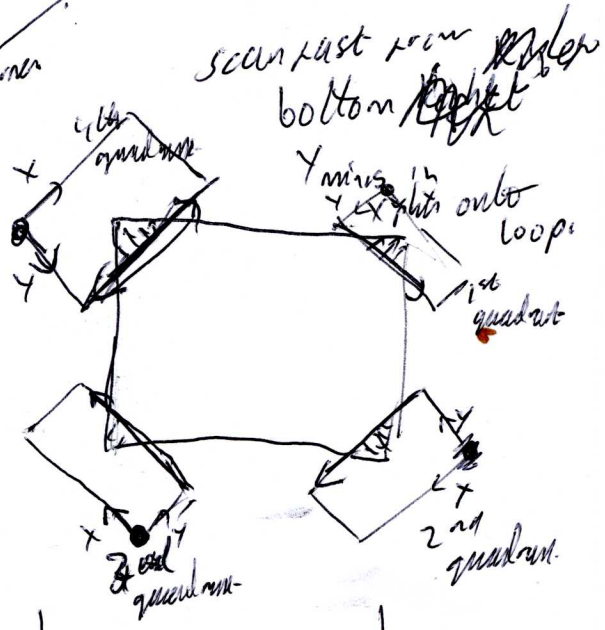
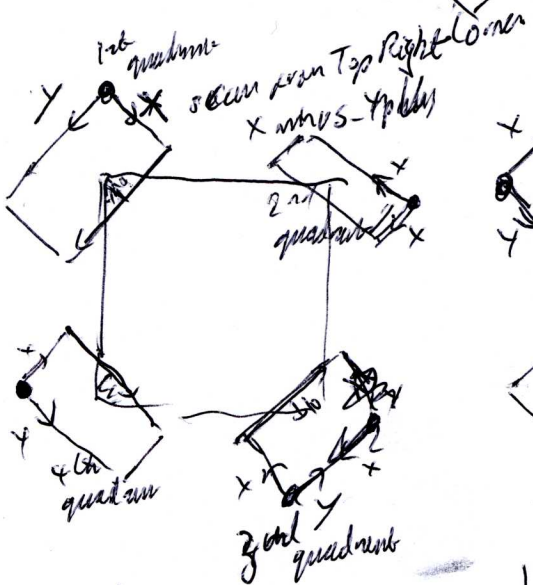
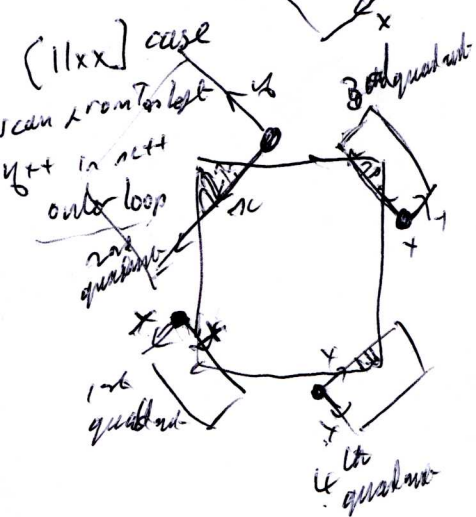
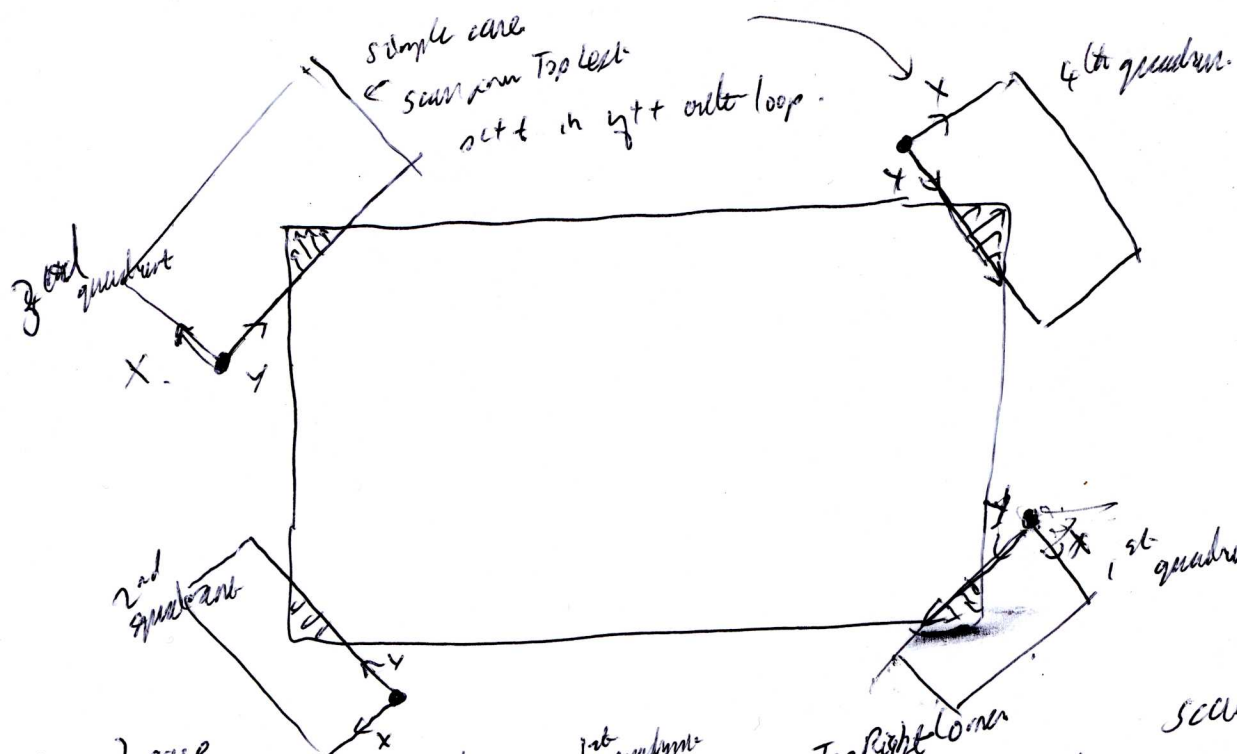


0, 90, 180, 360 parallel to edge of texture lies along boundary (or close to it).

Doesn't make much difference which method you choose.

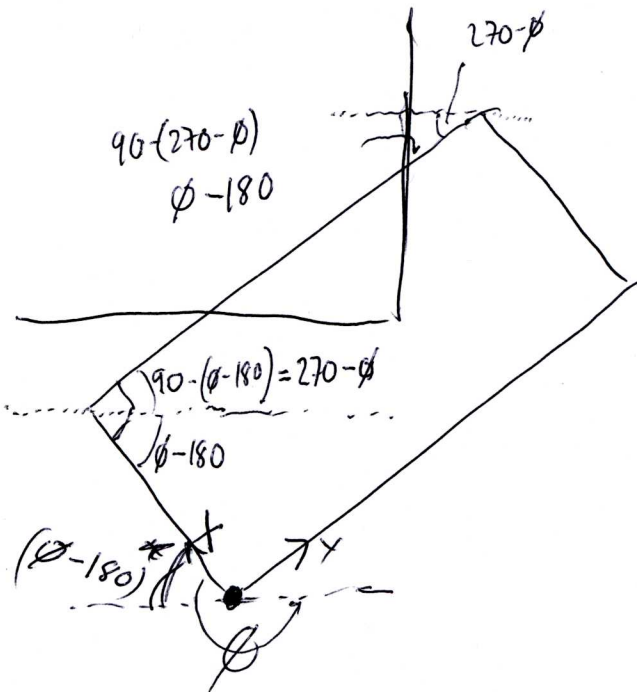
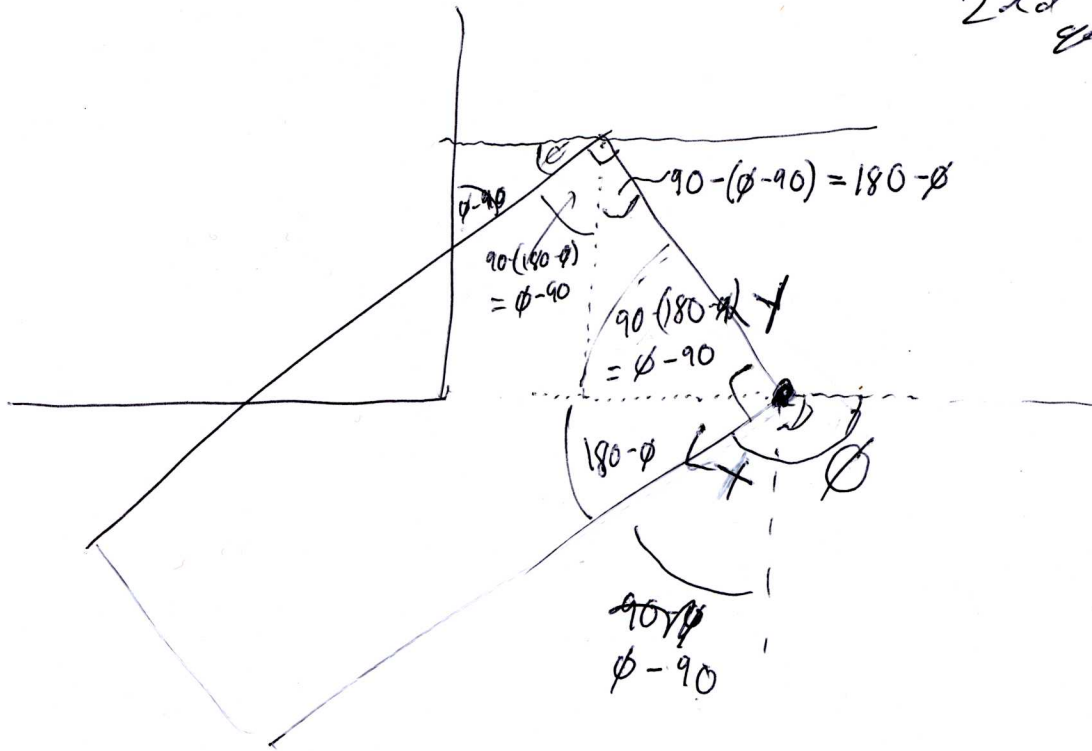


NO  
CORNERS  
ON  
SCREEN

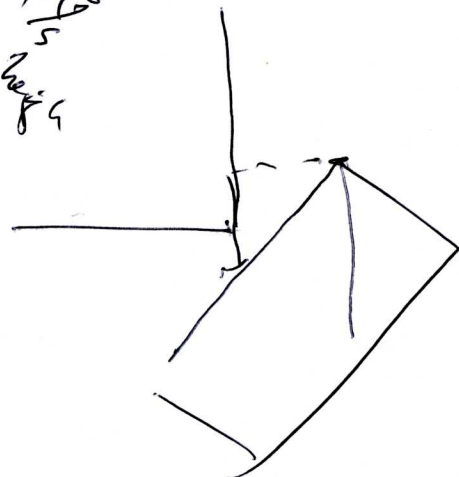
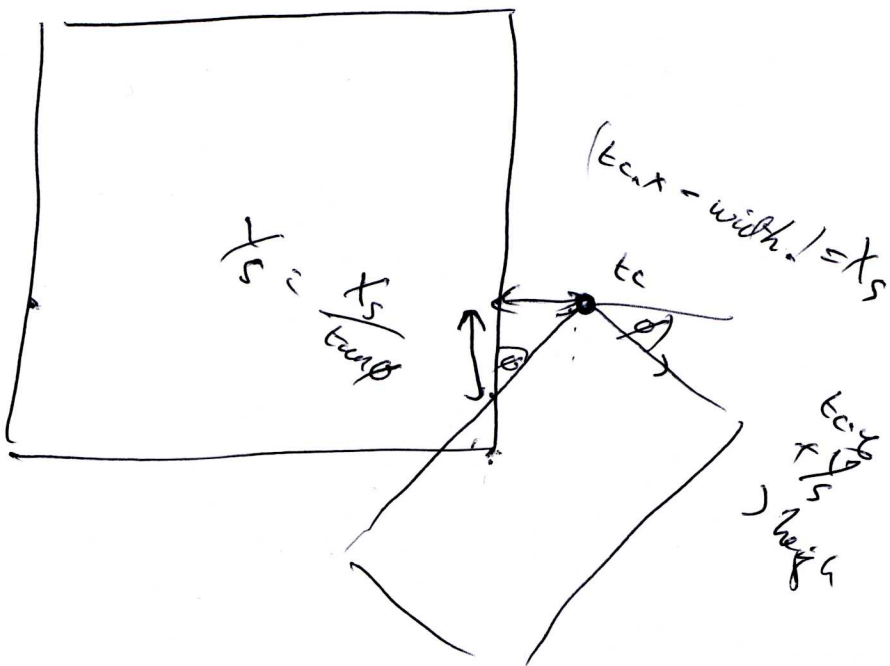
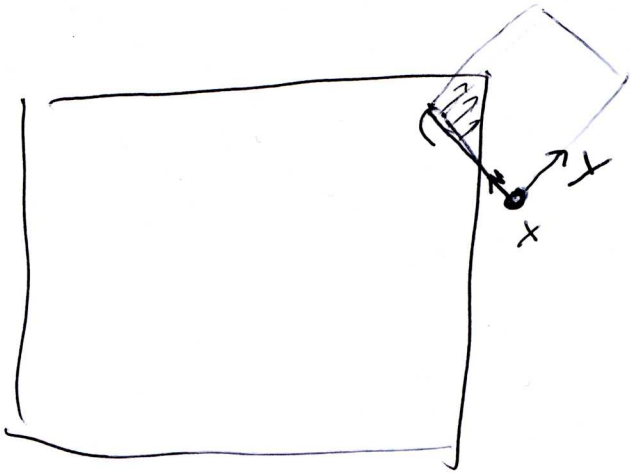


2nd quad

12

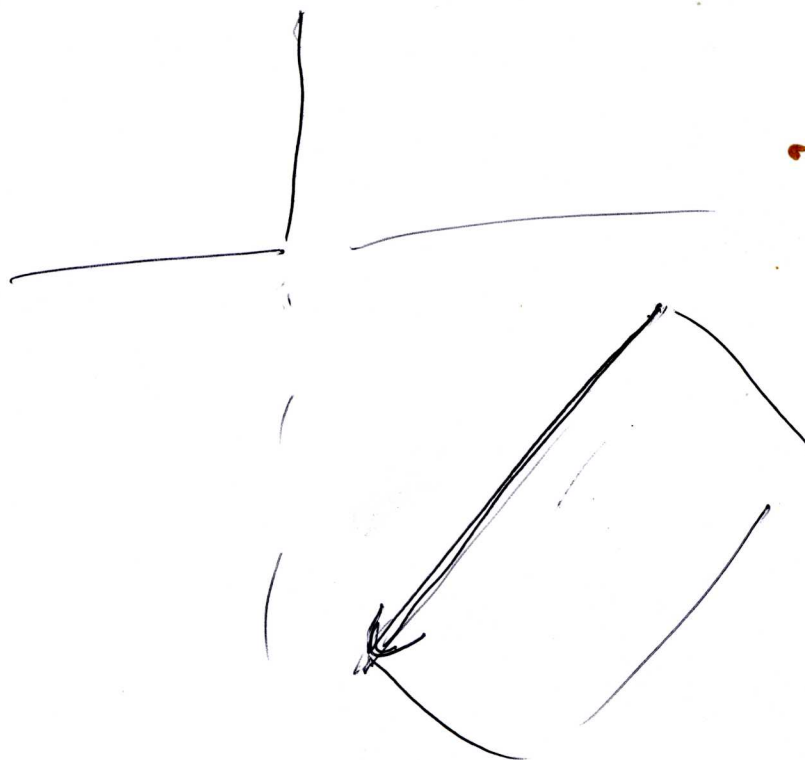
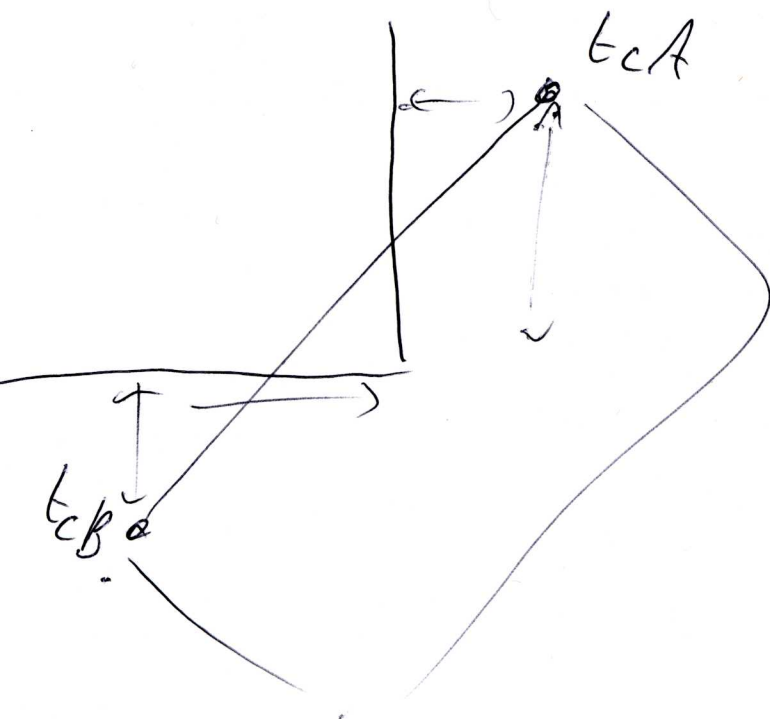


(13)





14



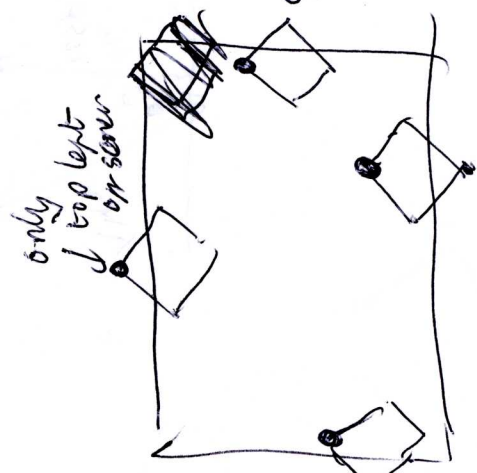
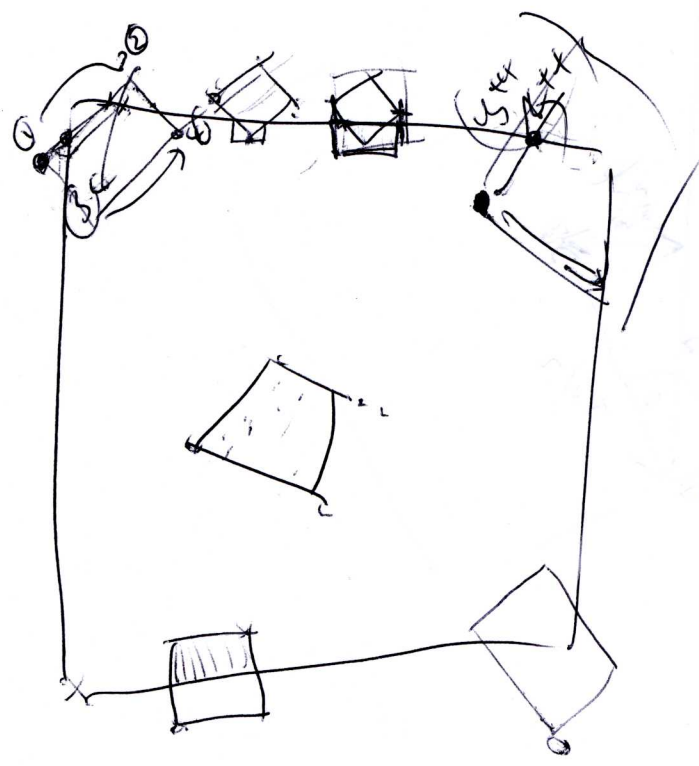
Top half

(S)

S x

All over (=)

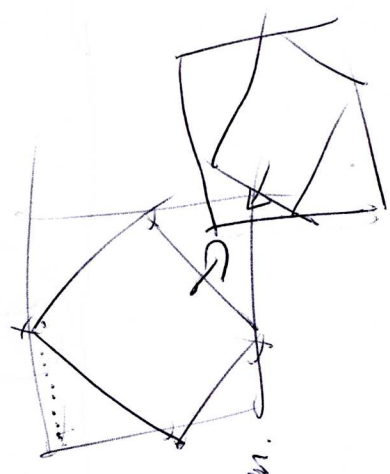
Port. x, y (?)

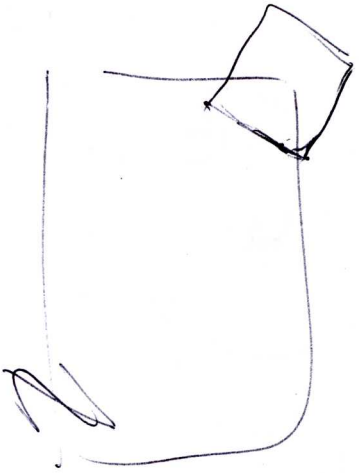
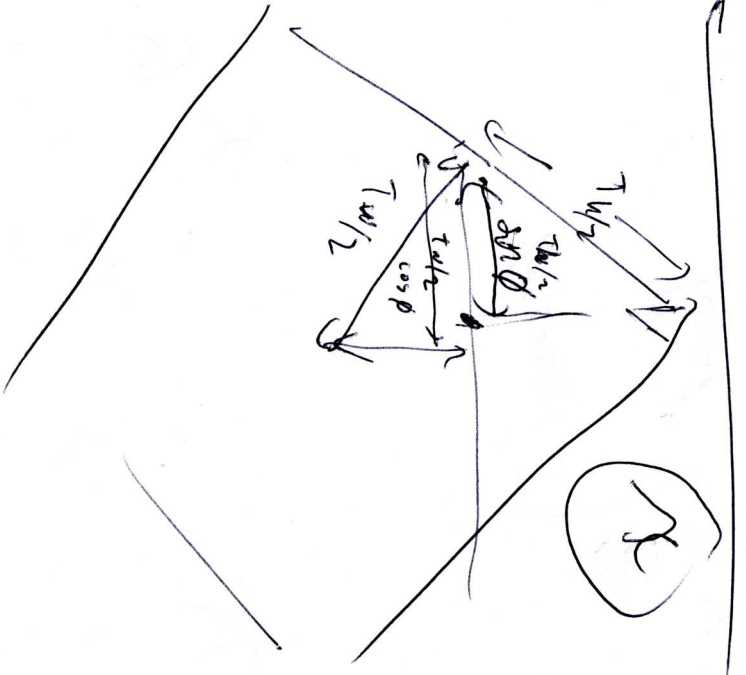
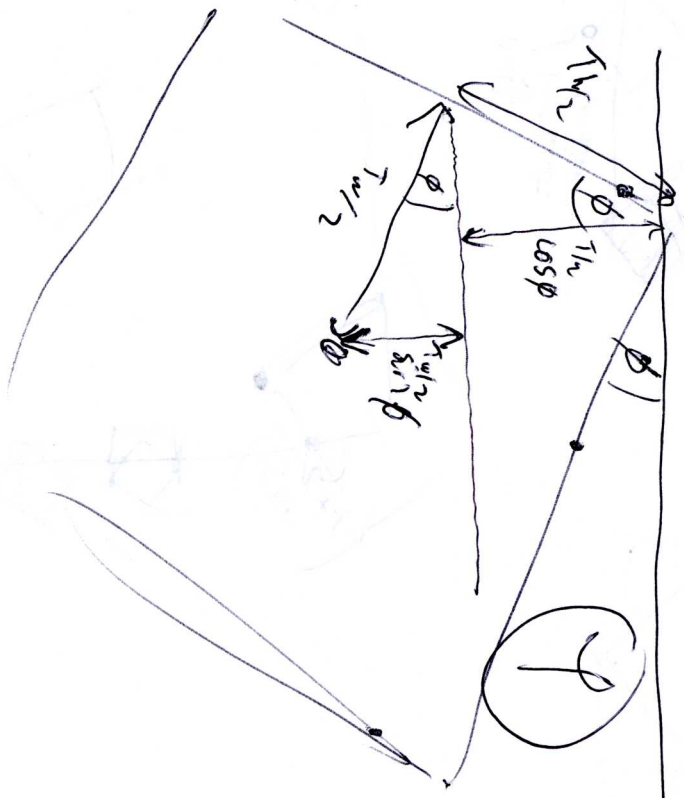


only  
top left  
on screen

only  
top right  
on screen

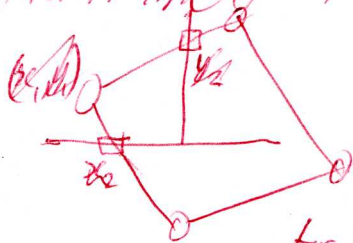
only  
bottom  
left  
on screen





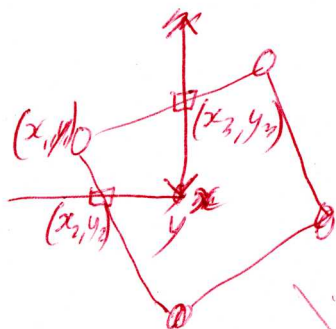


Is corner of screen inside image?  
 ↳ if not cut through start line (ignore this sample row for now)



check each  
 line between a  
 vertices in the box  
 and one outside the box  
 for the intercept with exterior

~~$x_2 < x$~~   
 ~~$x_2 < x$~~



~~$(x_1, y_1)$~~   
 ~~$(x_2, y_2)$~~

$$y = \frac{(y_1 - y_2)}{(x_1 - x_2)} x + y_2$$

~~relative position for  $(x_2, y_2)$~~

