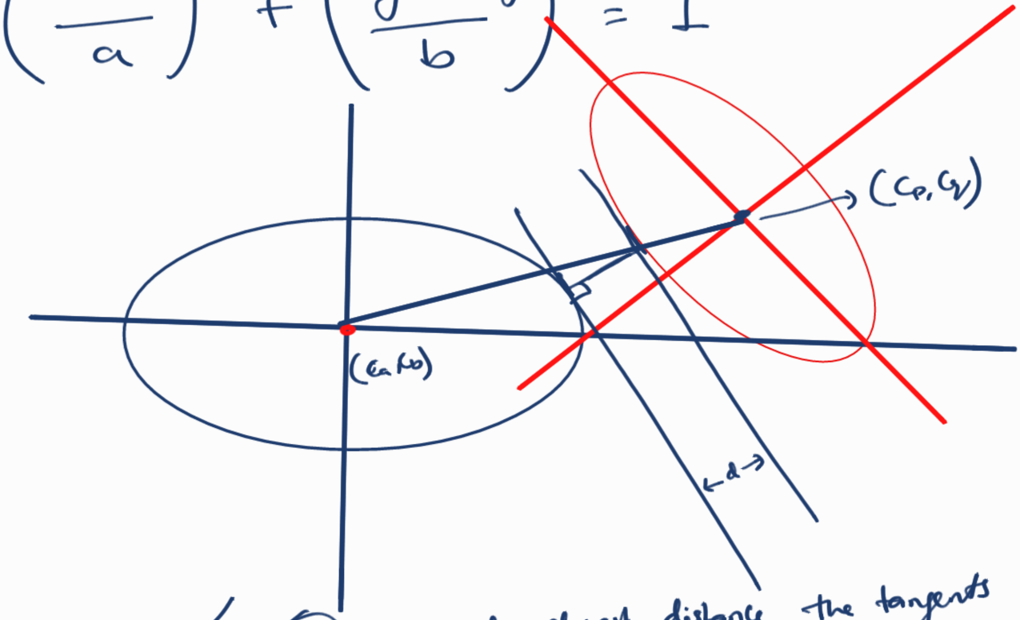
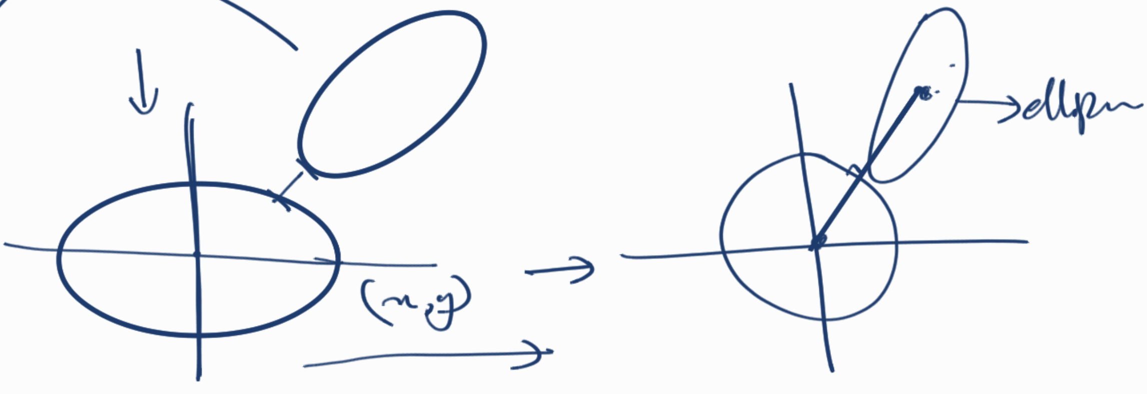
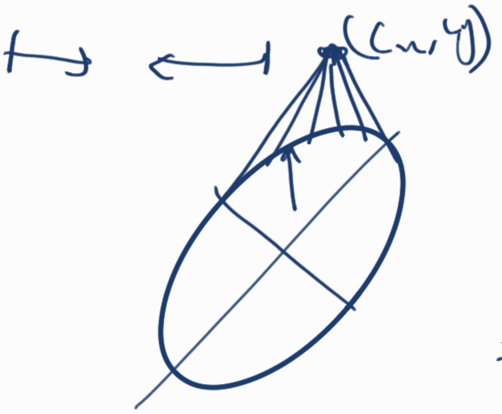
Collision Detection

$$\left(\frac{x-C_x}{a}\right)^2 + \left(\frac{y-C_y}{b}\right)^2 = 1$$



at closest distance, the tangents are parallel.





$$\left\| (x(t), y(t)) - C_x, C_y \right\|_2$$

$$= 0$$

$$\left(\frac{x-C_{1x}}{a_1}\right)^2+\left(\frac{y-C_{1x}}{b_1}\right)^2=1,\quad \left(\frac{x-C_{2x}}{a_2}\right)^2+\left(\frac{y-C_{2y}}{b_2}\right)^2=1,$$

$$\boxed{(x,y)\rightarrow \qquad\qquad\qquad\qquad \Rightarrow `\alpha'}$$

ellipse → (C_x, C_y, α, a, b)

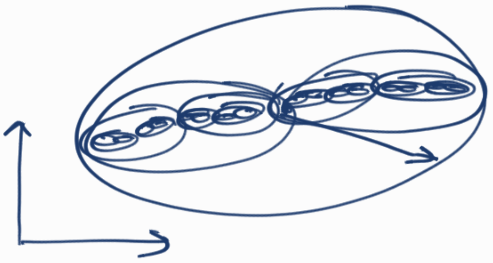
unrotate $\alpha = -\alpha$ $\quad T_r = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad$ unshift $P = C$ $\quad T_t = \begin{bmatrix} 1 & 0 & C_x \\ 0 & 1 & C_y \\ 0 & 0 & 1 \end{bmatrix}$

$R(-\alpha)$ $\qquad\qquad T(C_x, C_y)$

unscale $x \to xa$ $y \to y^b$ $\quad T_s = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad S(a,b)$

Final transformation = $\underline{T_s T_r T_t P}$ ⇒

↘ $x^2+y^2=1$



$T_s T_r T_t \times \text{ellipse2}$

$$(C_{2x}, C_{2y}, \alpha_2, a_2, b_2)$$

$$\Downarrow$$

$$\left(C_{2x}-C_{1x},\ C_{2y}-C_{1y},\ \alpha_2-\alpha_1,\ \frac{a_2}{a_1}\cdot\frac{b_2}{b_1}\right)$$

$\left(C_{2x}-C_{1x},\ C_{2y}-C_{1y},\ \alpha_2-\alpha_1,\ \frac{a_2}{a_1}\cdot\frac{b_2}{b_1}\right)$

$x^2+y^2-1=0$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$$

$x = \cos(t)$, $y = \sin(t)$

$\underline{(1^2)+(1^2)=1}$

$$\begin{matrix} x = \\ y = \\ \end{matrix} \begin{bmatrix} a\cos(t) \\ b\sin(t) \\ 1 \end{bmatrix}$$

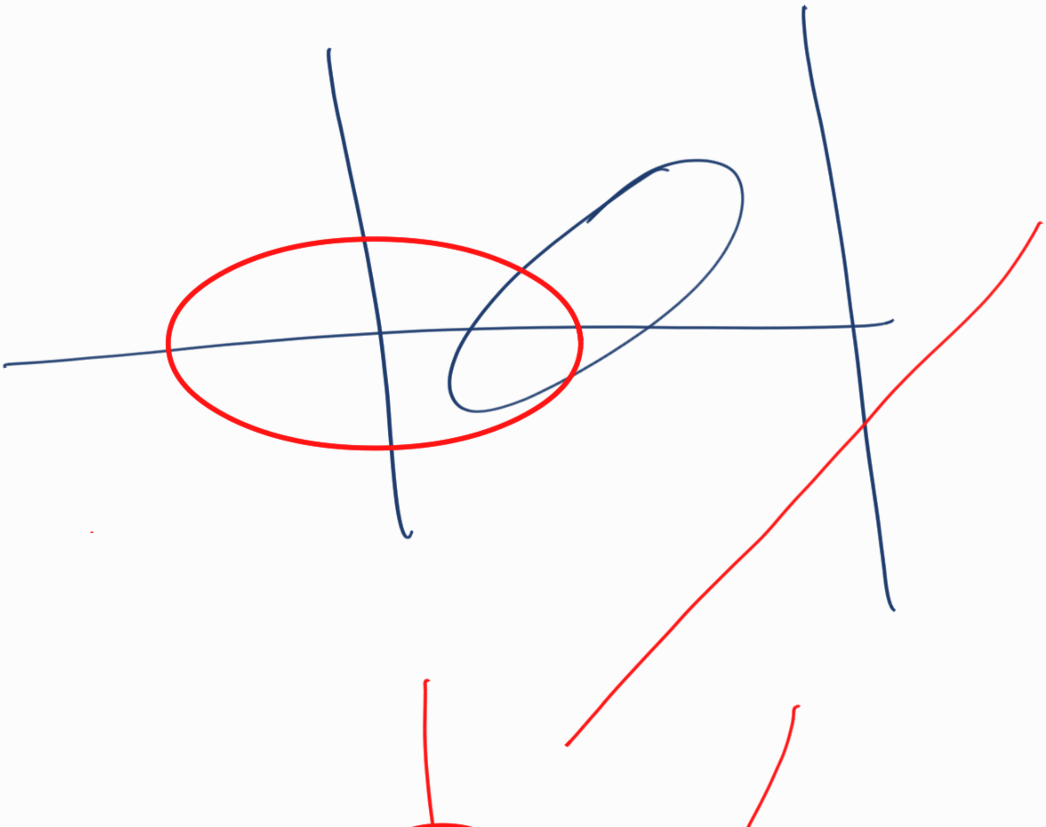
$$\begin{bmatrix} a\cos(t)\cos(\alpha) - b\sin(t)\sin(\alpha) \\ a\cos(t)\sin(\alpha) + b\sin(t)\cos(\alpha) \\ 1 \end{bmatrix}$$

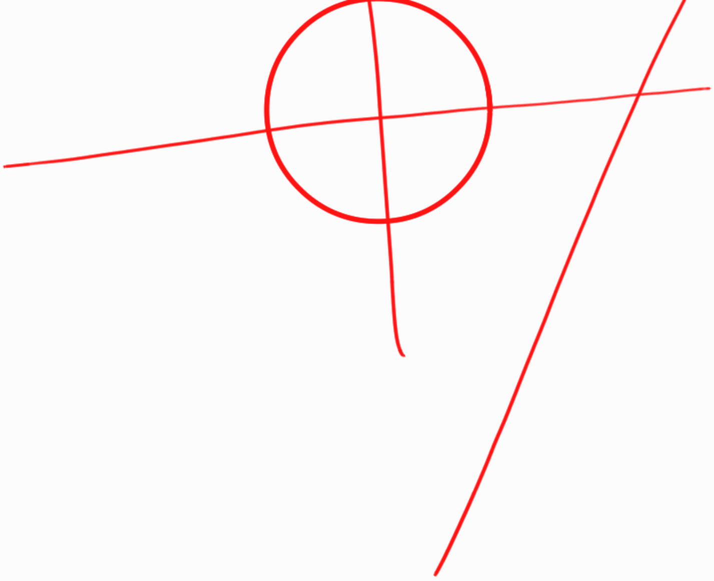
<u>ellipse parametric equation</u>

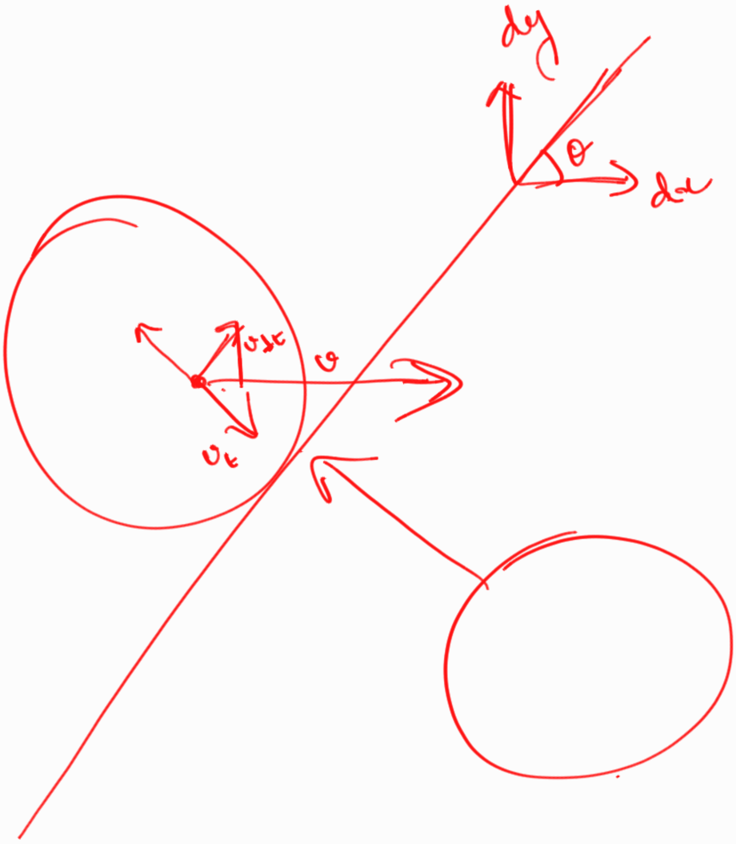
$$\begin{bmatrix} a\cos(t)\cos(\alpha) + b\sin(t)\sin(\alpha) + t_x \\ -a\cos(t)\sin(\alpha) + b\sin(t)\cos(\alpha) + t_y \\ 1 \end{bmatrix}$$

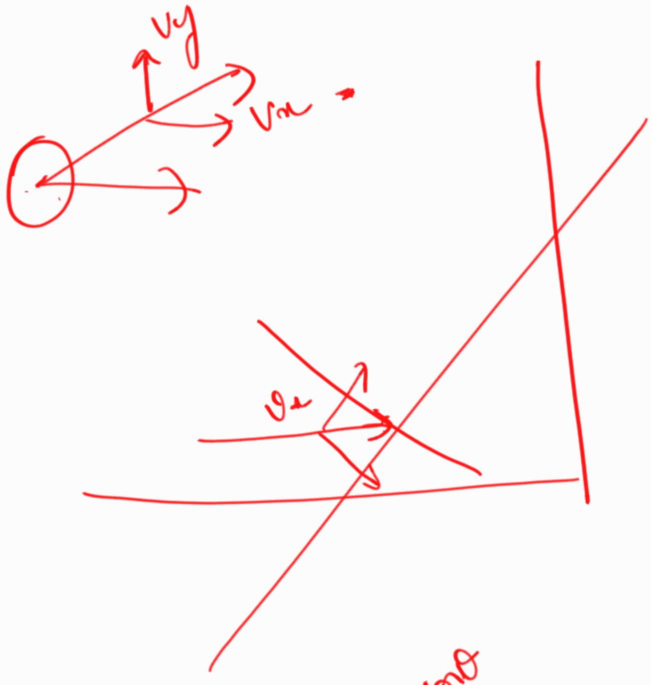
$$= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} ax \\ by \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos t \\ \sin t \\ 1 \end{bmatrix}$$









$v_x \cos\theta$

MATLAB CODE USED TO DERIVE THE COLLISION EXPRESSIONS

```
syms t real
x = cos(t);
y = sin(t);
% syms x y real

XY = [x y 1]';
E = [1 0 0; 0 1 0; 0 0 -1];
ellipse = XY'*E*XY;
ellipse

%%%% ellipse 1
syms cx1 cy1 alpha1 a1 b1 real
ellipse1XY = get_ellipse(cx1, cy1, alpha1, a1, b1)*XY
% ellipse1 = ellipse1XY'*E*ellipse1XY;
% ellipse1

%%%% ellipse 2
syms cx2 cy2 alpha2 a2 b2 real
ellipse2XY = get_ellipse(cx2, cy2, alpha2, a2, b2)*XY
% ellipse2 = ellipse2XY'*E*ellipse2XY;
% ellipse2

%%%%
% syms p q l a b real
% ellipse2XY = simplify(get_ellipse(p, q, l, a, b)*XY)
% ellipse2 = ellipse2XY'*E*ellipse2XY;
% ellipse2
%%%%
```

```
%%% unwarp test %%%%
un_warp = warp_ellipse2circle(cx1, cy1, alpha1, a1, b1);
ellipse1XY = un_warp*get_ellipse(cx1, cy1, alpha1, a1, b1)*XY;
ellipse1XY = simplify(ellipse1XY);
% disp(ellipse1XY')
% ellipse1 = ellipse1XY'*E*ellipse1XY;
% ellipse1 = simplify(ellipse1);
% ellipse1

%%% unwarp one ellipse based on other
un_warp_e1 = warp_ellipse2circle(cx1, cy1, alpha1, a1, b1);
ellipse2 = get_ellipse(cx2, cy2, alpha2, a2, b2);
ellipse2WarpedXY = un_warp_e1 * ellipse2 * XY
dxdyd1 = diff(ellipse2 * XY, t)
test=dxdyd1(1)/dxdyd1(2)

distance_fn = simplify(ellipse2WarpedXY'*ellipse2WarpedXY)
% diff_dist_fn = diff(distance_fn, t)
% solve(diff_dist_fn, t)
% latex(diff(distance_fn, t))

% ellipse2Warped = ellipse2WarpedXY'*E*ellipse2WarpedXY;
% ellipse2Warped=simplify(ellipse2Warped);
% ellipse2Warped




function op = warp_ellipse2circle(cx, cy, alpha, a, b)
   op = get_scale(1/a,1/b)*get_rotn(-alpha)*get_trans(-cx,
-cy);
end
```

```
function ell = get_ellipse(cx, cy, alpha, a, b)
    ell = get_trans(cx,cy)*get_rotn(alpha)*get_scale(a,b);
end
function t_mat = get_trans(tx, ty)
     t_mat = [1 0 tx; 0 1 ty; 0 0 1];
end
function r_mat = get_rotn(alpha)
    r_mat = [cos(alpha) -sin(alpha) 0; sin(alpha) cos(alpha) 0; 0
0 1];
end
function s_mat = get_scale(a, b)
    s_mat = [a 0 0; 0 b 0; 0 0 1];
end
```