τ+

Author: Matthew Fonken
Date: 7/24/17

Overview: Tau+ is a method for predicting the locations of a coupled object pair in a set of images. It involves two major steps: 1) image processing, and 2) location prediction.

Given:

- $X_{n+1}[1]$ = Initial X location prediction for beacon 1.
- $X_{n+1}[2]$ = Initial X location prediction for beacon 2.
- $Y_{n+1}[1]$ = Initial Y location prediction for beacon 1.
- $Y_{n+1}[2]$ = Initial Y location prediction for beacon 2.

Image Processing:

1. Map the X & Y densities of an image:

$$X_\rho[i] = \sum_{j=1}^{H_c} I(i,j)$$
$$Y_\rho[j] = \sum_{i=0}^{W_c} I(i,j)$$

2. Scan the densities maps for peaks and record index and height for each:

$$\frac{dh}{dx}X_\rho[i] = 0\ ,\ \frac{d^2h}{dx^2}X_\rho[i] < 0,\ X_p[m] = \{x[j], h^x[j]\}$$
$$\frac{dh}{dy}Y_\rho[j] = 0\ ,\ \frac{d^2h}{dy^2}Y_\rho[j] < 0,\ Y_p[n] = \{y[j], h^y[j]\}$$

3. Generate primary and secondary probabilities for X peaks using the equations:

$$L_n^x = 1 - \left| 1 - \frac{X_{n+1}}{x_p[m]} \right|$$
$$H_n^x = 1 - \left| 1 - \frac{H_{n-1}^x}{h_n^x[m]} \right|$$
$$X_n = aL_n^x + bH_n^x\text{ , where } a + b = 100\,\%$$
$$= L_n^x + b(H_n^x - L_n^x)$$

4. Select peaks with highest probability for new X's.
5. Repeat for Y's.
6. Output X and Y pairs into coordinate reconstruction.

Predict Next Locations:

1. Calculate rotation coupling:

$$X_c = \frac{\Delta X[2] + \Delta X[1]}{\Delta X[2] - \Delta X[1]}$$

2. Calculate linear velocities:

$$V_x^* = X_n - X_{n-1}$$

3. Calculate linear extrapolation:

$$\begin{aligned} X_{n+1}^L &= X_n + V_x^* \\ &= 2X_n - X_{n-1} \end{aligned}$$

4. Calculate radius between beacons:

$$r = \sqrt{(X[2] - X[1])^2 + (Y[2] - Y[1])^2}$$

5. Calculate angular extrapolation:

$$\frac{X_{n+1}}{r} = \cos\theta_{n+1}$$
$$\frac{X_n}{r} = \cos\theta_n$$
$$\frac{X_{n-1}}{r} = \cos\theta_{n-1}$$
$$\omega^* = \theta_n - \theta_{n-1}$$
$$\theta_{n+1} = \theta_n + \omega^* = 2\theta_n - \theta_{n-1}$$
$$X_{n+1} = r\cos\left(2\theta_n - \theta_{n-1}\right)$$
$$X_{n+1}^A = r\cos\left(2\cos^{-1}\left(\frac{X_n}{r}\right) - \cos^{-1}\left(\frac{X_{n-1}}{r}\right)\right)$$

6. Combine linear and angular extrapolation:

$$\begin{aligned} X_{n+1} &= X_{n+1}^L X_c + X_{n+1}^A\left(1 - X_c\right) \\ &= X_c\left(X_{n+1}^L - X_{n+1}^A\right) + X_{n+1}^A \end{aligned}$$

7. Repeat for Y.
8. Cycle back to **Image Processing** with predictions.

Note: velocity are period synchronized and thus time irrelevant

<u>Algorithm Angle Teardown:</u>

1. Solving for σ_r:

1.1. *Description of method*: σ_r represents the angle $\angle b_1 c_f b_2$. Due to b_1' and b_2' lying on the camera's line of site from c_f to b_1 and b_2, it also represents $\angle b_1' c_f b_2'$. Due to the characteristics of a linear lens on the camera, the coordinates of the visualized points b_1' and b_2' on the camera's plane correspond one-to-one and linearly to horizontal (ϕ) and vertical (θ) angles relative to the camera plane's normal $\overrightarrow{c_n}$. This normal is the vector (whose length is f) perpendicular to the camera's plane defined by the camera plane's center and the camera's focal point c_f.

1.2. Start with visualized beacons on camera's plane and identify horizontal θ and vertical ϕ angles:

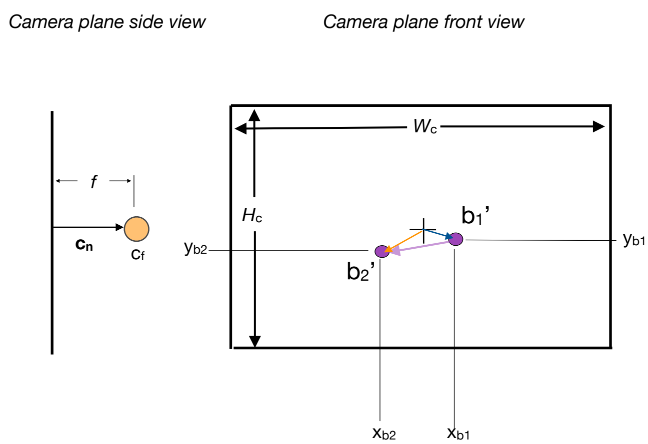

Figure 7

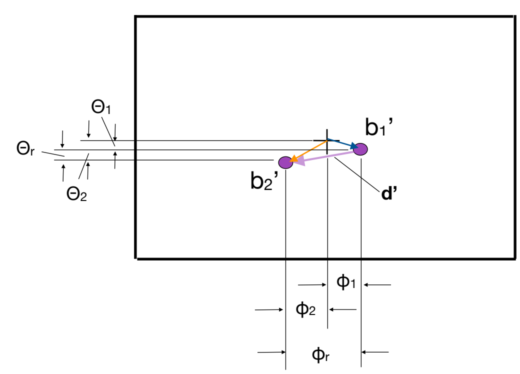

Figure 8

1.3.Calculate horizontal θ and vertical ϕ angles of both beacon points:

$$\theta_{1,2} = \alpha_H \left(\frac{x_{b_{1,2}}}{W_c} - \frac{1}{2} \right) \tag{1}$$

$$\phi_{1,2} = \alpha_V \left(\frac{y_{b_{1,2}}}{H_c} - \frac{1}{2} \right) \tag{2}$$

1.4. Calculate differences of angles and coordinates:

$$\theta_r = |\theta_2 - \theta_1| \tag{3}$$

$$\phi_r = |\phi_2 - \phi_1| \tag{4}$$

1.5. Calculate σ_r as the angle between detected beacon #1 and beacon #2:

$$\sigma_r = cos^{-1}(cos\theta_r cos\phi_r) \tag{5}$$

2. Solving for χ:

   2.1. *Description of method*: χ represents the angle $\angle b_1' b_2' c_f$ and equivalently $\angle b_1 b^* c_f$. It can be found using the law of sines using the recently found σ_r and the lengths of vectors $\overrightarrow{b_1' b_2'}$ (or more concisely $||\overrightarrow{d'}||$) and $\overrightarrow{b_1' c_f}$ (or mathematically $\frac{f}{cos\sigma_1}$).

   2.2. Calculate σ_1 as the angle between detected beacon 1 and camera normal:

$$\sigma_1 = cos^{-1}(cos\theta_1 cos\phi_1) \tag{6}$$

   2.3. Calculate distances between beacon coordinates:

$$\Delta x_b = x_{b_2} - x_{b_1} \tag{7}$$

$$\Delta y_b = y_{b_2} - y_{b_1} \tag{8}$$

   2.4. Calculate length of vector $\vec{d'}$ (**d'** in figure) between visualized beacons

$$||\vec{d'}|| = \sqrt{\Delta y_b^2 + \Delta x_b^2} \times \frac{units}{pixel} \tag{9}$$

   2.5. Calculate the inner angle χ between $\vec{d'}$ and $\vec{c_f b_2}$ (beacon #2 line-of-site):

$$\frac{\frac{f}{cos\sigma_1}}{sin(\chi)} = \frac{||\vec{d'}||}{sin(\sigma_r)} \tag{10}$$

$$\chi = sin^{-1}\left(\frac{f}{||\vec{d'}||}\frac{sin\sigma_r}{cos\sigma_1}\right) \tag{11}$$

3. Solving for μ:

   3.1. *Description of method*: μ represents the angle $\angle b^* b_1 b_2$. It can be extracted directly from the rotation quaternion H_r representing the camera's spatial orientation relative to the beacons.

   3.2. Generate rotation quaternion from beacon reference coordinate system to camera's using the first beacon b_1:

$$H_r = H(\phi_1, \theta_1, 0) \tag{12}$$

   3.3. Convert quaternion H_r rotation matrix R:

      3.3.1. Basic quaternion overview:

(13)

$$H_a = a + bi + cj + dk = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} q_{a_w} \\ q_{a_x} \\ q_{a_y} \\ q_{a_z} \end{bmatrix}$$

$$R = \begin{bmatrix} \angle\hat{z}_c\hat{z}_r & \angle\hat{y}_c\hat{z}_r & \angle\hat{x}_c\hat{z}_r \\ \angle\hat{z}_c\hat{y}_r & \angle\hat{y}_c\hat{y}_r & \angle\hat{x}_c\hat{y}_r \\ \angle\hat{z}_c\hat{x}_r & \angle\hat{y}_c\hat{x}_r & \angle\hat{x}_c\hat{x}_r \end{bmatrix} \tag{14}$$

$$= \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{bmatrix} \tag{15}$$

3.4. Extract single angle rotation from $\widehat{x_c}$ to $\widehat{x_r}$ from H_r as $R[\hat{x}_c\hat{x}_r]$:

$$\mu = R[\hat{x}_c\hat{x}_r] = a^2 - b^2 - c^2 + d^2 = q_{r_w}^2 - q_{r_x}^2 - q_{r_y}^2 + q_{r_z}^2 \tag{16}$$

4. Solving for γ & γ':

4.1. *Description of method*: γ represents the angle $\angle b^*b_2b_1$ and γ' is its supplement ($\gamma + \gamma' = 180°$). γ' Is required to define the fundamental triangle of Figure 2.

$$\gamma' = 180 - \gamma = 180° - \mu - \chi \tag{17}$$

$$\gamma = \mu + \chi \tag{18}$$

Algorithm Final Solution:

1. Solving for $||\vec{r}||$:

   1.1. *Description of method*: $\vec{r}$ represents the vector of length $||\vec{r}||$ from b_1 to b_1'. $||\vec{r}||$ can be solved first as the physical distance from the first beacon to the camera's focal point and then using the combined rotation quaternion of the full system comprising of four smaller rotation quaternions, $\vec{r}$ can be found. $\vec{r}$ directly represents the camera focal point's physical coordinates in space relative to the first beacon.

   1.2. With γ, σ_r, and $||\overrightarrow{d}||$ known $||\vec{r}||$ can be calculated using the law of sines.

$$\frac{||\vec{r}||}{\sin(\gamma)} = |\frac{|\overrightarrow{d}||}{\sin(\sigma_r)} \tag{19}$$

$$||\vec{r}|| = \frac{\sin(\gamma)}{\sin(\sigma_r)} \times ||\overrightarrow{d}|| \tag{20}$$

1. Solving for $\vec{r}$:

   1.1. *Description of method*: $\vec{r}$ is simply a unit vector $\hat{u}$ multiplied by the scalar $||\vec{r}||$ rotated by the combined rotation quaternions of the full system: H_d, H_c, H_r, and H_g (d - geomagnetic north to device, c - device to camera, r - camera to beacons, and g - beacons to geomagnetic north). H_d is actively calculated using the inertial measurement unit on the device. H_c is the pre-calculated rotation offset between the inertial measurement unit and the camera's plane. H_r is calculated previously as equation 12. H_g is the pre-calculated rotation offset of the stationary beacon and geomagnetic north.

   1.2. Combine all quaternions into H_a:

$$H_a = H_d H_c H_r H_g \tag{21}$$

   1.3. Rotate a unit vector $\overrightarrow{u}$ of length $||\vec{r}||$ by H_a and extract final coordinates:

$$\vec{r} = H_a(||\vec{r}||\widehat{u}) \tag{22}$$

$$(x, y, z)_d = [\hat{i}_{\vec{r}}, \hat{j}_{\vec{r}}, \hat{k}_{\vec{r}}] \tag{23}$$