

# Reference frame conversions

Laurens Valk

January 14, 2018

## Contents

<b>1</b>	<b>Preliminaries</b>	<b>1</b>
1.1	Points and Vectors . . . . .	1
1.2	Transformation . . . . .	2
1.3	Inverse Transformation . . . . .	2
<b>2</b>	<b>From Pixels to Meters</b>	<b>2</b>
2.1	Transformation Matrices . . . . .	2
2.2	Scaling Factor . . . . .	4
<b>3</b>	<b>Local Robot Frames</b>	<b>5</b>
3.1	From Label Markers to Label Frame . . . . .	5
3.2	From Label Frame to Robot Frame . . . . .	6
<b>4</b>	<b>Relative Agent Positions and Orientations</b>	<b>6</b>

## 1 Preliminaries

### 1.1 Points and Vectors

$$r_{b/a}^c = p_b^c - p_a^c \tag{1}$$

add picture

### 1.2 Transformation

$$p^i = \alpha \left( R_j^i p^j + r_{j/i}^i \right) \tag{2}$$

$$\begin{bmatrix} p^i \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} R_j^i & r_{j/i}^i \\ 0 & 1 \end{bmatrix}}_{H_j^i} \begin{bmatrix} p^j \\ 1 \end{bmatrix} \quad (3)$$

### 1.3 Inverse Transformation

$$\begin{bmatrix} p^j \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} (R_j^i)^T & -(R_j^i)^T r_{j/i}^i \\ 0 & 1 \end{bmatrix}}_{H_j^i} \begin{bmatrix} p^i \\ 1 \end{bmatrix} \quad (4)$$

## 2 From Pixels to Meters

### 2.1 Transformation Matrices

The symbol  $\tilde{p}$  expresses points in units of pixels. Any other point  $p$  has position units of meters.

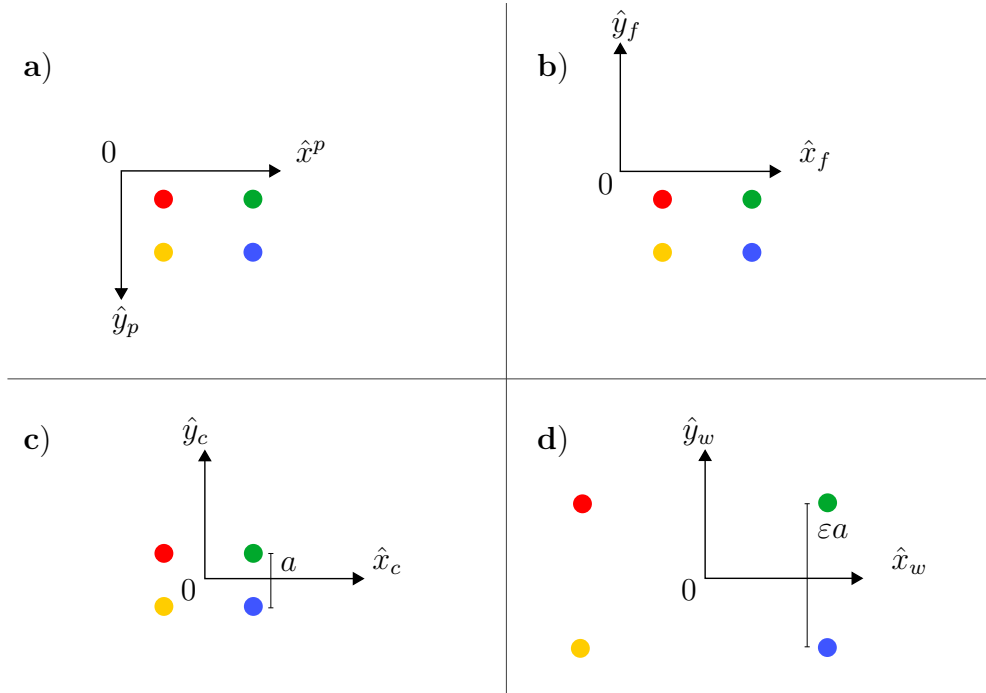


Figure 1: Four coordinates: a) in the camera picture, b) with flipped y axis, c) with the origin at the midbase of the picture, d) scaled to physical distances.

$$\begin{bmatrix} \tilde{p}^f \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{p}^p \\ 1 \end{bmatrix} = H_p^f \begin{bmatrix} \tilde{p}^p \\ 1 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \tilde{p}^c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{1}{2}\tilde{d}_x \\ 0 & 1 & \frac{1}{2}\tilde{d}_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{p}^f \\ 1 \end{bmatrix} = H_f^c \begin{bmatrix} \tilde{p}^f \\ 1 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} p^w \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{p}^c \\ 1 \end{bmatrix} = H_c^w \begin{bmatrix} \tilde{p}^c \\ 1 \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} p^w \\ 1 \end{bmatrix} = H_c^w H_f^c H_p^f \begin{bmatrix} \tilde{p}^p \\ 1 \end{bmatrix} \quad (8)$$

The overall transformation from pixels to the real world representation is

$$\begin{bmatrix} p^w \\ 1 \end{bmatrix} = H_p^w \begin{bmatrix} \tilde{p}^p \\ 1 \end{bmatrix} \Rightarrow \mathbf{p}^w = H_p^w \tilde{\mathbf{p}}^p \quad (9)$$

fix scaling outside  $H$

## 2.2 Scaling Factor

consider working with centimeters

The distance  $d_{\text{apex/midbase}}$  in Figure 3 is known, constant, and equivalent for each agent:

$$d_{\text{apex/midbase}} \equiv \|p_{\text{apex},i} - p_{\text{midbase},i}\| \quad \forall i \quad (10)$$

Per agent, we can estimate the scaling factor  $\epsilon_i$  as:

$$\epsilon_i = \frac{d_{\text{apex/midbase}}}{\|\tilde{p}_{\text{apex},i} - \tilde{p}_{\text{midbase},i}\|} \quad \text{m/pixel} \quad (11)$$

A more accurate estimate for the overall scaling factor is:

$$\varepsilon = \frac{1}{N} \sum_{i=1}^N \epsilon_i \quad (12)$$

However, because  $\varepsilon$  is constant it may be more desirable to calculate it once and use it as a constant parameter. If so, we may divide the arena width by the picture with in pixels:

$$\varepsilon = \frac{d_{\text{field}}}{\tilde{d}_x} \quad \text{m/pixel} \quad (13)$$

### 3 Local Robot Frames

cross check p's and then define vector as difference between two points;  
and then vectors cannot be transformed; just rotated

#### 3.1 From Label Markers to Label Frame

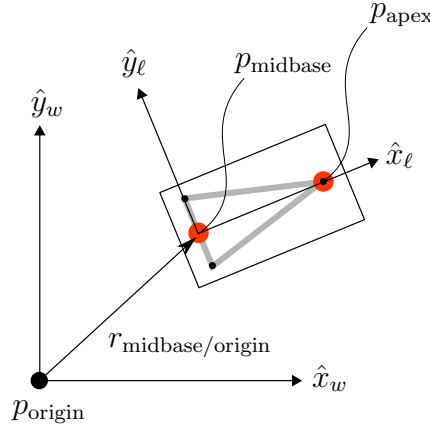


Figure 2: Local vehicle frame definitions

First, transform pixel locations to world coordinates:

$$\mathbf{p}_{\text{midbase}}^w = H_p^w \tilde{\mathbf{p}}_{\text{midbase}}^p, \quad \mathbf{p}_{\text{apex}}^w = H_p^w \tilde{\mathbf{p}}_{\text{apex}}^p \quad (14)$$

Obtain the label x and y axes in world coordinates:

$$\hat{x}_\ell^w = \frac{p_{\text{apex}}^w - p_{\text{midbase}}^w}{\|p_{\text{apex}}^w - p_{\text{midbase}}^w\|}, \quad \hat{y}_\ell^w = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \hat{x}_\ell^w \quad (15)$$

The transformation matrix becomes:

$$\mathbf{p}^w = H_\ell^w \mathbf{p}^\ell, \quad H_\ell^w = \begin{bmatrix} \hat{x}_\ell^w & \hat{y}_\ell^w & r_{\text{midbase/origin}}^w \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Agree on a better name than center. This is confusing

replace terminology: top = apex, center = midbase

FIX: Inside H matrix are always VECTORS. not p's.

### 3.2 From Label Frame to Robot Frame

distinguish between points and vectors: Then vectors suddenly make sense: points in a frame are always defined with respect to the origin of that frame

$$\mathbf{p}^b = \begin{bmatrix} I_2 & r_{\text{midbase/robot}}^b \\ 0 & 1 \end{bmatrix} \mathbf{p}^\ell \quad (17)$$

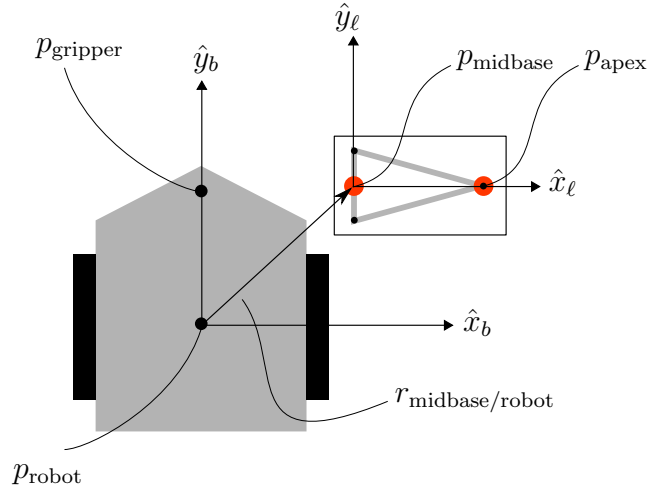


Figure 3: Local vehicle frame definitions

## 4 Relative Agent Positions and Orientations