Localization in unknown environments using computer vision

Eduardo Perdices García

eperdices@gsyc.es

March 2013

Index

Introduction

- Computer vision
  - Cameras
  - Robots
  - Localization
- Kalman Filter

Computer Vision

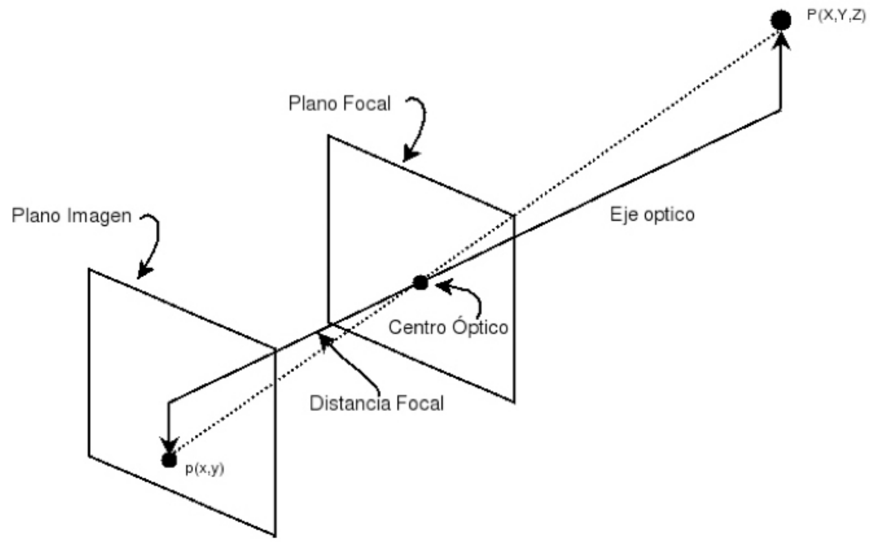
- Advantages
  - More information
  - 3D objects, colors, shapes
- Disadvantages
  - High execution time
  - Poor reliability
  - No depth

Example - Faces

037	084	132	155	172	175	178	179	181	180	178	177	176	173	168	163	158	156	155	149	139	128	117	111	096	087	069	052	049	040	021	021
045	108	166	190	205	209	212	215	215	214	214	212	209	205	201	198	194	189	188	181	170	163	149	142	128	119	110	092	077	073	049	028
052	113	171	197	217	222	226	230	231	233	232	230	226	222	218	216	214	211	206	201	193	183	170	163	149	145	134	113	087	070	057	032
049	109	163	189	206	211	219	224	229	231	232	230	226	223	219	218	216	214	209	206	202	195	185	179	170	166	149	123	088	076	060	049
066	115	164	179	185	194	204	214	224	229	230	229	228	225	222	221	221	220	219	219	218	216	207	195	191	192	171	142	102	085	058	043
074	131	173	176	167	154	147	156	174	199	220	227	226	226	226	226	224	224	226	222	197	173	156	158	161	165	179	157	116	096	076	048
095	135	182	178	144	158	183	172	141	121	122	154	203	217	225	228	224	216	169	131	120	142	188	211	210	196	183	166	127	107	080	080
120	140	187	180	161	168	174	169	168	156	136	128	157	200	220	226	217	194	147	141	154	178	182	181	185	185	185	171	131	110	097	090
135	158	195	185	154	130	098	081	072	086	120	151	154	179	222	229	209	173	160	144	096	077	057	079	113	156	184	182	137	117	099	084
125	170	200	189	161	109	054	110	064	075	168	132	174	170	224	237	209	181	165	149	152	082	111	127	082	155	199	187	145	125	091	061
133	194	203	197	185	166	119	134	140	153	189	182	204	186	218	239	213	210	213	188	176	165	153	150	187	213	213	190	151	130	074	043
149	185	207	200	196	193	181	176	185	195	209	223	218	189	218	239	215	215	230	221	210	206	201	208	217	222	213	189	153	140	070	075
184	195	209	200	203	207	209	213	222	227	231	228	209	187	220	238	218	212	230	233	232	235	233	231	228	224	210	186	158	114	102	154
202	213	213	197	199	210	219	226	228	230	231	222	201	187	222	238	221	211	225	234	235	235	235	232	227	220	206	183	161	123	140	187
205	218	212	197	192	204	219	227	231	231	229	216	192	189	224	236	226	209	219	231	236	236	235	230	222	213	199	180	151	120	180	206
204	232	228	203	185	196	212	223	230	232	229	213	179	193	229	239	227	209	210	229	235	236	232	225	216	207	192	175	166	167	227	204
201	233	250	227	179	186	201	213	224	227	224	202	184	188	221	237	219	208	212	219	228	228	225	218	209	198	182	177	252	251	233	201
197	229	246	241	181	180	192	205	218	221	220	194	169	155	191	211	186	163	197	216	227	223	219	212	202	191	177	195	249	246	229	197
194	225	242	243	192	177	186	197	208	214	218	212	152	114	135	155	151	138	187	222	225	219	211	205	195	185	173	211	245	242	225	194
190	222	238	240	204	178	182	191	202	212	215	216	202	187	172	160	189	204	217	224	220	213	207	201	192	184	172	219	238	238	222	190
185	217	233	235	219	186	180	187	198	205	207	209	213	216	228	222	222	223	222	219	212	203	205	200	191	183	170	227	233	233	217	185
180	212	229	229	224	195	181	187	196	190	192	199	188	178	187	203	184	178	187	201	184	192	206	201	191	178	181	230	229	229	212	180
176	208	224	224	224	209	186	188	195	179	122	096	100	106	111	112	121	121	112	096	142	199	208	201	186	165	207	226	224	224	208	176
171	203	220	219	220	216	192	188	193	199	176	126	134	148	158	158	168	163	151	161	212	212	206	195	176	169	223	220	219	219	203	171
166	198	214	214	214	214	203	192	191	194	198	177	143	156	179	187	178	162	170	202	209	206	197	183	166	202	227	227	227	225	198	166
162	194	210	210	210	210	207	199	189	189	193	190	170	153	149	149	152	167	192	206	207	203	189	173	159	205	224	178	178	225	194	162
158	189	205	205	205	205	210	207	197	192	192	195	197	199	196	194	199	206	213	210	205	195	172	155	165	217	233	178	178	233	211	184
153	186	202	202	201	202	210	212	203	198	195	200	207	215	224	231	232	229	220	210	194	177	154	128	220	178	178	153	153	178	178	222
150	182	197	198	197	204	210	212	209	205	201	201	208	214	222	224	222	218	206	190	168	150	122	122	221	178	178	153	153	178	178	222
147	179	195	195	195	205	214	214	210	206	203	203	206	209	210	210	204	198	180	166	142	124	122	122	157	205	227	178	178	228	203	175
133	161	176	174	187	190	199	199	198	193	187	187	184	180	173	169	158	147	136	116	105	111	112	108	101	138	206	178	178	201	161	133
117	132	142	151	160	159	166	168	168	166	157	155	154	147	134	125	115	105	091	081	082	086	083	077	071	078	168	222	222	172	132	117

Pin-Hole Cameras

- Intrinsic parameters
  - Focal distance
  - Optical center
  - Skew
  - Others
- Extrinsic parameters
  - 3D position
  - Focus of attention
  - Roll



Quaternions

- Extension of complex numbers
- Represents a 3D rotation with 4 elements:
  - q_0: Real scalar, rotation angle
  - $q1, q2, q3$: Imaginary vector, rotation axis
- Easy translation to rotation matrices
- Rotation composition with Hamilton product

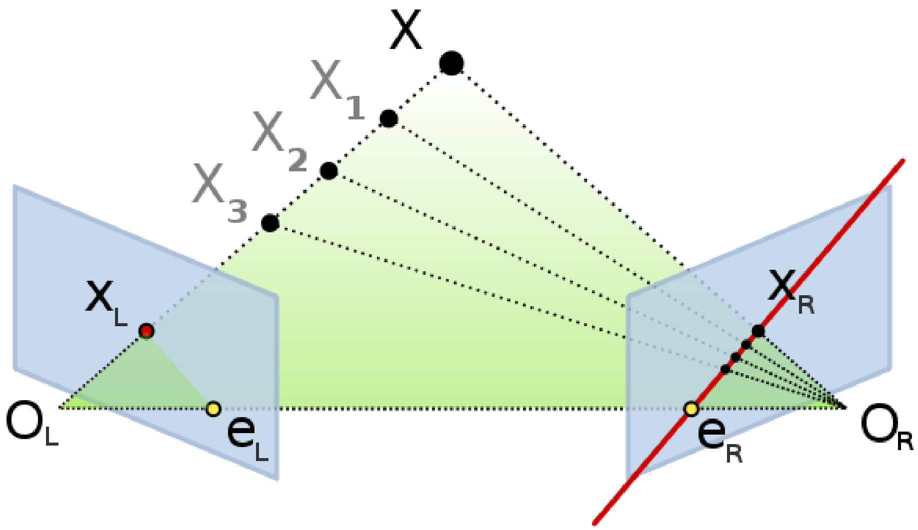
Project and unproject

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} -\frac{x}{z} \\ -\frac{y}{z} \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{u - u_0}{f_x} \\ \frac{v - v_0}{f_y} \\ 1{,}0 \end{bmatrix}$$

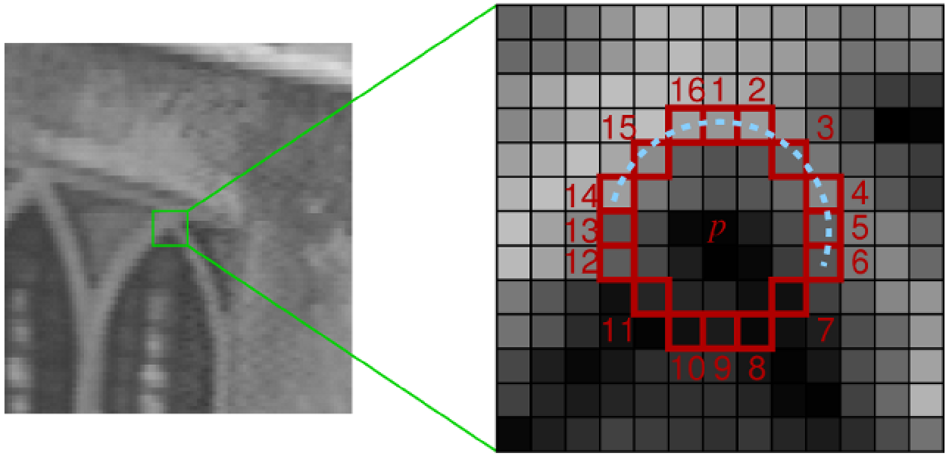
Getting 3D

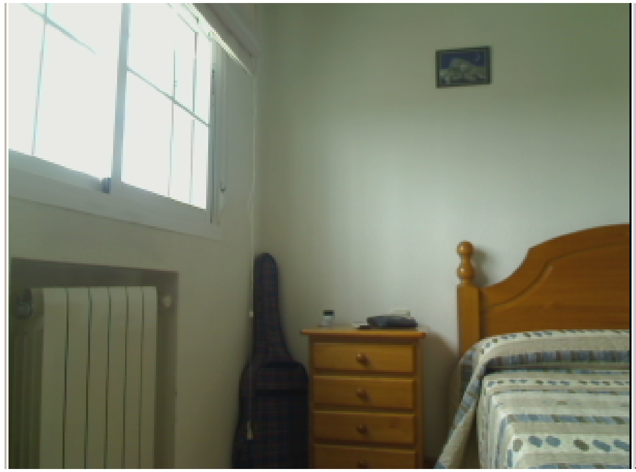
- Epipolar geometry:
  - Two cameras at the same time
  - One camera at different times (and movement)

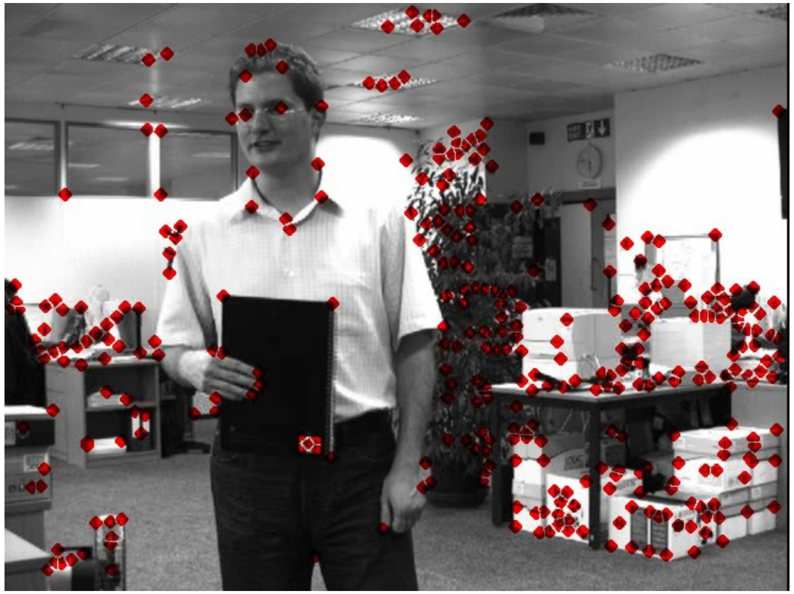


Features detection

- Feature types:
  - Points
  - Lines
  - Complex objects
- Techniques:
  - Color filters
  - Edge filters (Canny, Sobel)
  - Morphology
  - Hough transform
  - Shi-Tomasi
  - FAST Corner Detection

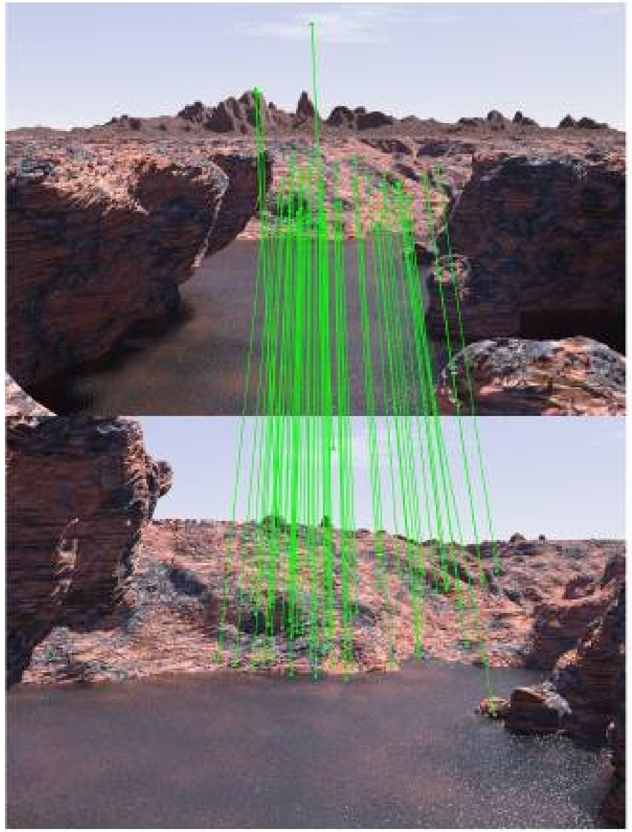






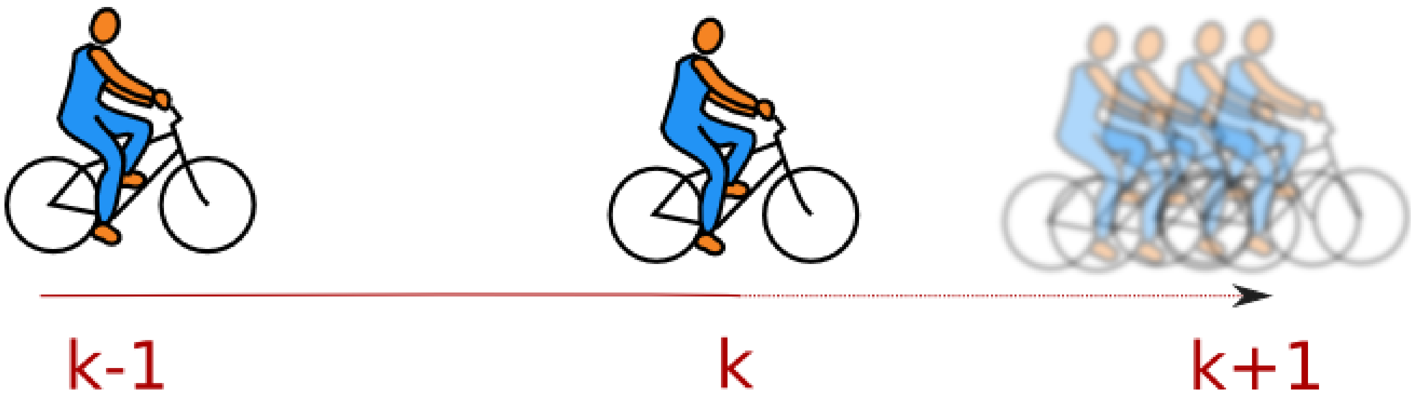
Feature matching

- With descriptors:
  - SURF
  - SIFT
  - ORB
- Patch to Patch:
  - Difference
  - Normalized squared difference
  - Avg color, color counting, ...



Kalman Filter

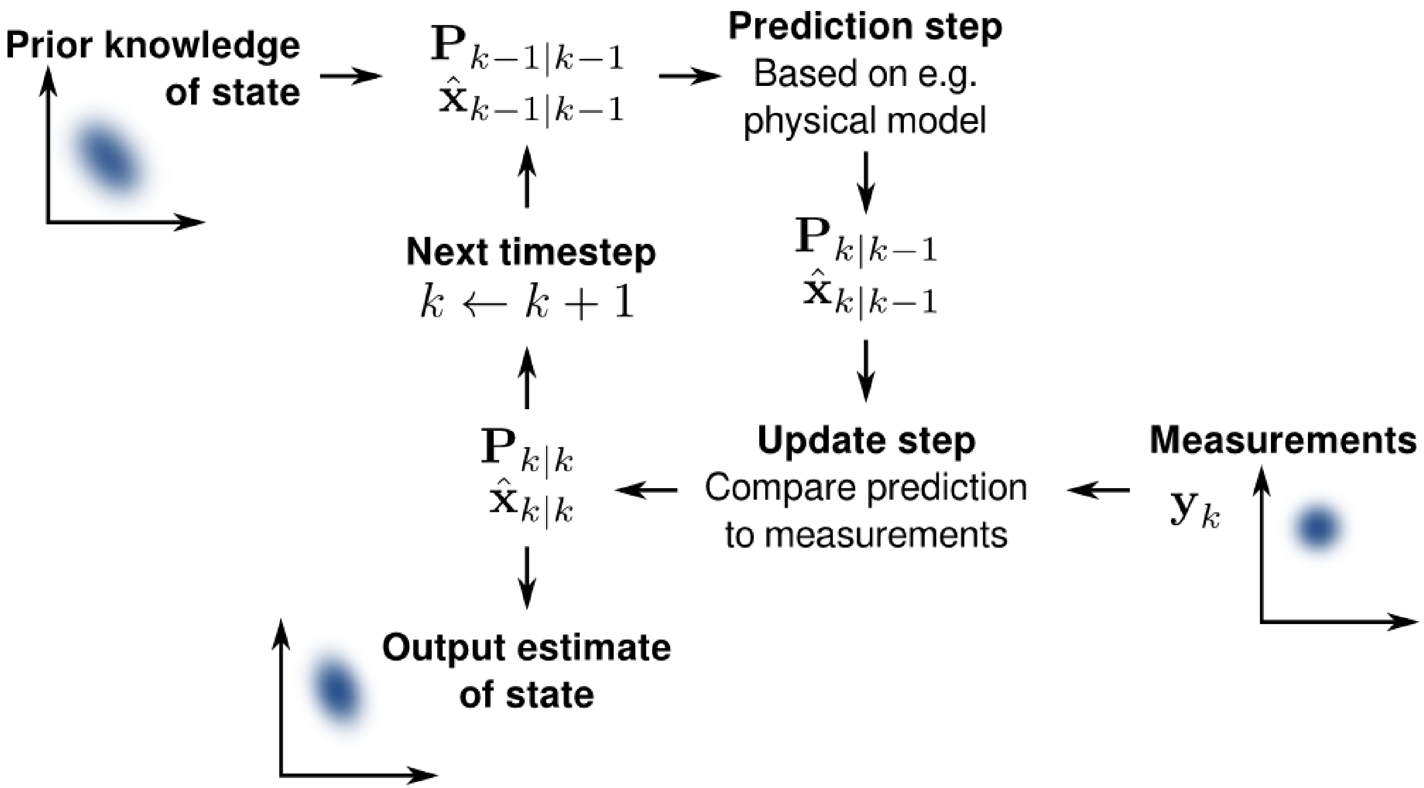
- Linear systems and Gaussian distributions
- Estimates state based on observations



Kalman Properties

- Main Matrices:
  - $\hat{x}_k$: Estimated status vector
  - P_k: Estimated covariance matrix
  - y_k: Observation vector
- Models:
  - F_k: Prediction model
  - H_k: Observation model
- Steps:
  - Prediction step
  - Update step

Basic algorithm



Complete KF algorithm

Predict

Status prediction	$\hat{x}_{k\|k-1} = F_k \hat{x}_{k-1}$
Covariance prediction	$P_{k\|k-1} = F_k P_{k-1} F_k^T + Q_k$

Update

Residual	$\tilde{y}_k = z_k - H_k \hat{x}_{k\|k-1}$
Residual covariance	$S_k = H_k P_{k\|k-1} H_k^T + R_k$
Kalman Gain	$K_k = P_{k\|k-1} H_k^T S_k^{-1}$
Status stimation	$\hat{x}_k = \hat{x}_{k\|k-1} + K_k + \tilde{y}_k$
Covariance stimation	$P_k = (I - K_t H_k) P_{k\|k-1}$

Extended Kalman Filter

- Use Kalman with non linear systems
- More dynamic systems but less optimal
- Linearize models:
  - $F_k \rightarrow J_{F_k}$ and $f()$
  - $H_k \rightarrow J_{H_k}$ and $h()$

Complete EKF algorithm

Predict	
Status prediction	$\hat{x}_{k\|k-1} = f(\hat{x}_{k-1})$
Covariance prediction	$P_{k\|k-1} = J_{F_k} P_{k-1} J_{F_k}^T + Q_k$
Update	
Residual	$\tilde{y}_k = z_k - h(\hat{x}_{k\|k-1})$
Residual covariance	$S_k = J_{H_k} P_{k\|k-1} J_{H_k}^T + R_k$
Kalman Gain	$K_k = P_{k\|k-1} J_{H_k}^T S_k^{-1}$
Status stimation	$\hat{x}_k = \hat{x}_{k\|k-1} + K_k + \tilde{y}_k$
Covariance stimation	$P_k = (I - K_t J_{H_k}) P_{k\|k-1}$

Discrete bicycle example

$$x_k = \begin{bmatrix} x \\ v \end{bmatrix} \qquad x_k = Fx_{k-1} + Ga_k$$

$$z_k = Hx_{k-1} + u_k$$

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$x_k = Fx_{k-1} + w_k$$

$$R = E\begin{bmatrix} u_k & u_k^T \end{bmatrix} = \begin{bmatrix} \sigma_z^2 \end{bmatrix}$$

$$Q = GG^T\sigma_a^2 = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \sigma_a^2$$

Continuous bicycle example (Kalman-Bucy filter)

$$X = \begin{bmatrix} x \\ v \end{bmatrix}$$

$$\frac{dx}{dt} = v \qquad \frac{dv}{dt} = a + w$$

$$z = x + u$$

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a + \begin{bmatrix} 0 \\ w \end{bmatrix}$$

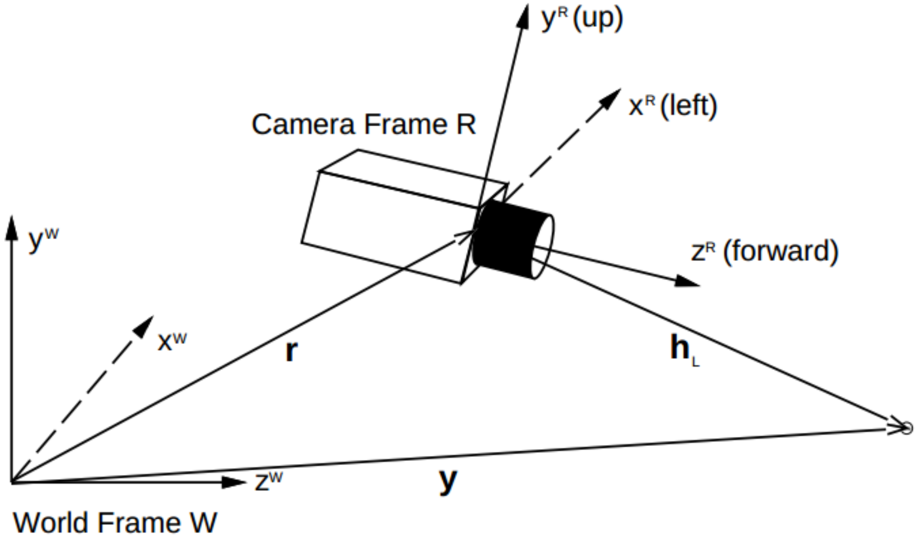
$$z = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + u$$

$$z = HX + u$$

$$\frac{d}{dt}X = FX + Ba + \begin{bmatrix} 0 \\ w \end{bmatrix}$$

MonoSLAM

- Monocular Simultaneous Localization and Mapping
- Calculates camera position and orientation (6 degrees of freedom)
- Creates a map of features automatically
- Uses an EKF to estimate its status



EKF with MonoSLAM

$$\hat{x} = \begin{bmatrix} x_v \\ f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{bmatrix} \qquad P = \begin{bmatrix} P_{x_v x_v} & P_{x_v f_1} & P_{x_v f_2} & \dots & P_{x_v f_N} \\ P_{f_1 x_v} & P_{f_1 f_1} & P_{f_1 f_2} & \dots & P_{f_1 f_N} \\ P_{f_2 x_v} & P_{f_2 f_1} & P_{f_2 f_2} & \dots & P_{f_2 f_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{f_N x_v} & P_{f_N f_1} & P_{f_N f_2} & \dots & P_{f_N f_N} \end{bmatrix}$$

Status Model

$$x_v = \begin{bmatrix} r^W \\ v^W \\ q^{WR} \\ \omega^R \\ p^W_{f_1} \\ \dots \\ p^W_{f_N} \end{bmatrix} \qquad x_{v|k} = f(x_{v|k-1}) = \begin{bmatrix} r^W_{|k-1} + v^W_{|k-1}\Delta k \\ v^W_{|k-1} \\ q_g(\omega^R_{|k-1}\Delta k) \times q^{WR}_{|k-1} \\ \omega^R_{|k-1} \\ p^W_{f_1|k-1} \\ \dots \\ p^W_{f_N|k-1} \end{bmatrix}$$

Observation Model

- Project each point 3D to camera pixels

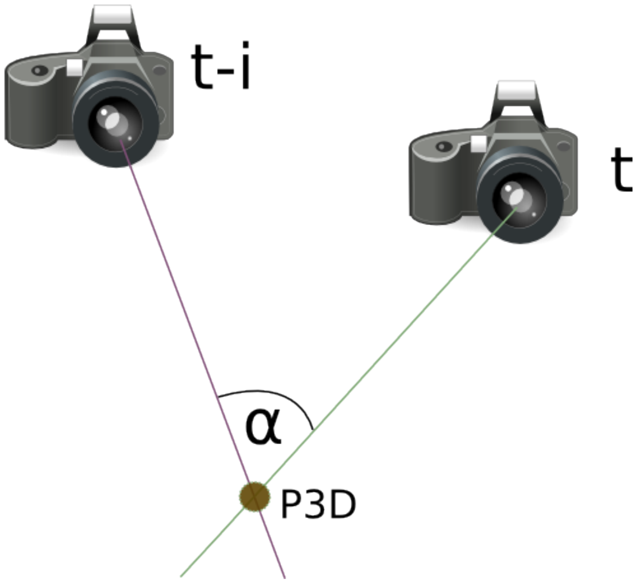
$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = h \begin{bmatrix} x_v \\ f_i^W \end{bmatrix} = K\left(R\left(T\begin{bmatrix} x_v \\ f_i^W \end{bmatrix}\right)\right)$$

Feature Matching

- Features (points) detected with FAST
- Matching patch to patch:
  - Search inside uncertainty regions
  - Normalized squared difference
- Initial features known

Creating new Features

- Initialize on the infinite
- Calculate 3D Matrix:
  - $M = p1 * p2' - p2 * p1'$
- Concatenate matrices:
  - $TM = [TM; M]$
- Get 3D point (autovector) with SVD:



Localization in unknown environments using computer vision

Eduardo Perdices García

eperdices@gsyc.es

March 2013