Convergence domain of image-based visual servoing with a line-scan camera

Vincent Drevelle

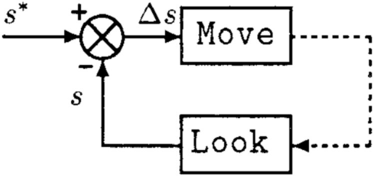
Lagadic, Université de Rennes 1, IRISA/Inria Rennes Bretagne Atlantique (France)

SWIM 2016, Lyon

Visual Servoing

Definition

Control the motion of a robot by using feedback information from a vision sensor



- Used a lot in industrial applications (assembly, alignment...)
- Large convergence domain in practice, but not always guaranteed

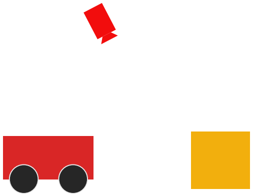
Outline

Visual Servoing

Visual Servoing: robot / camera configuration

“Eye-in-hand”

“Hand-to-eye”



Visual Servoing types

Different kinds of features can be used

Image-based (IBVS)

Use features directly available in the image data :

- points
- lines
- histogram
- ...

Position-based (PBVS)

Feature is object / camera pose, extracted from image by a pose estimation method

Hybrid

2D 1/2...

Image-based Visual Servoing (IBVS)

- Principle
  - Minimize the error between the current feature vector $\mathbf{s}(t)$ and desired value s^*

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*$$

- Interaction matrix $\mathbf{L_s}$
  - Relates the features time derivative $\dot{\mathbf{s}}$ to the camera velocity $\mathbf{v}_c$

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}_c$$

- Control scheme
  - For a fixed target, $\dot{\mathbf{e}} = \dot{\mathbf{s}}$. We have

$$\dot{\mathbf{e}} = \mathbf{L_e}\mathbf{v}_c \text{ with } \mathbf{L_e} = \mathbf{L_s}$$

  - Exponential decoupled decrease of the error is obtained with

$$\mathbf{v}_c = -\lambda \mathbf{L_e}^{+}\mathbf{e}$$

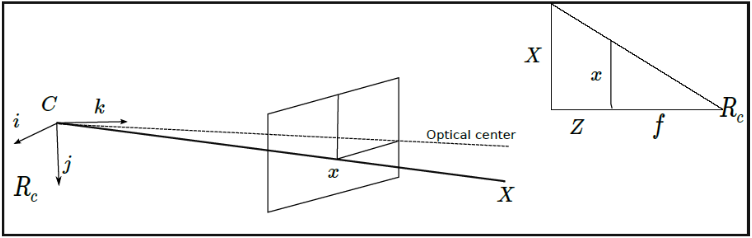
Image-based Visual Servoing (IBVS)

- In practice, neither the interaction matrix $\mathbf{L_e}$ nor $\mathbf{L_e^+}$ are known perfectly
- The control law uses an approximate value $\widehat{\mathbf{L_e^+}}$

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L_e^+}} \mathbf{e}$$

Line-scan camera IBVS with point features

- Line-scan camera: 1-D sensor
- Evolution in a plane:
  - camera pose: $\mathbf{p} = (p_x, p_y, \theta)^\top$
  - camera frame velocity: $\mathbf{v}_c = (v_x, v_y, \omega)^\top$
- Perspective projection



Projection equation

- Coordinates of a point of the plane in the camera frame:

$$\mathbf{X} = (X, Z)^\top$$

- Perspective projection x of a point $\mathbf{X}$ on the image line:

$$x = \Pi(\mathbf{X}) = \frac{X}{Z} = \frac{u - u_o}{f}$$

  - u is the point abscissa in pixel units
  - u_0 is the principal point
  - f is the focal length

- Use x as a feature for IBVS

Interaction matrix: $\dot{x} = \mathbf{L}_x v_c$

- World point movement (in camera frame)

$$\dot{\mathbf{X}} = \begin{pmatrix} \dot{X} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} -v_x + \omega Z \\ -v_y - \omega X \end{pmatrix}$$

- Derivative of a feature

$$\dot{x} = \frac{-v_x + x v_y}{Z} + (1 + x^2)\omega$$

- Interaction matrix (for a single point)

$$\dot{x} = \mathbf{L}_x v_c \text{ with } \mathbf{L}_x = \begin{pmatrix} \frac{-1}{Z} & \frac{x}{Z} & 1 + x^2 \end{pmatrix}$$

- Interaction matrix (k points observed)

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix}, \ \dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} v_c \text{ with } \mathbf{L}_{\mathbf{x}} = \begin{pmatrix} \mathbf{L}_{x_1} \\ \vdots \\ \mathbf{L}_{x_k} \end{pmatrix}$$

Control law

- Classical visual servo control law is

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_{\mathbf{e}}^{+}} \mathbf{e}$$

How to chose $\widehat{\mathbf{L}_{\mathbf{e}}^{+}}$?

- x and Z appear in $\mathbf{L}_e = \mathbf{L}_x$ expression
  - x is the point projection $\Rightarrow$ available at each time
  - Z is the point depth $\Rightarrow$ unavailable
- Estimate or observe Z at each time and compute $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}}^{+}$
- Set $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}^*}^{+}$, where $\mathbf{L}_{\mathbf{e}^*}$ is the interaction matrix at the desired position $\mathbf{e}^* = \mathbf{0}$ (i.e $\mathbf{x} = \mathbf{x}^*$)

Control law

- Classical visual servo control law is

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_{\mathbf{e}}^{+}} \mathbf{e}$$

How to chose $\widehat{\mathbf{L}_{\mathbf{e}}^{+}}$?

- x and Z appear in $\mathbf{L}_e = \mathbf{L}_x$ expression
  - x is the point projection $\Rightarrow$ available at each time
  - Z is the point depth $\Rightarrow$ unavailable
- Estimate or observe Z at each time and compute $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}}^{+}$
- Set $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}^*}^{+}$, where $\mathbf{L}_{\mathbf{e}^*}$ is the interaction matrix at the desired position $\mathbf{e}^* = \mathbf{0}$ (i.e $\mathbf{x} = \mathbf{x}^*$)

IBVS Stability Analysis

- Candidate Lyapunov function: squared error norm

$$\mathscr{L} = \frac{1}{2}\|\mathbf{e}(t)\|^2$$

- Its derivative is given by

$$\begin{aligned}\dot{\mathscr{L}} &= \mathbf{e}^\top \dot{\mathbf{e}} \\ &= -\lambda \mathbf{e}^\top \mathbf{L_e} \widehat{\mathbf{L_e}^+} \mathbf{e}\end{aligned}$$

- Sufficient condition for global asymptotic stability

$$\mathbf{L_e} \widehat{\mathbf{L_e}^+} > 0$$

3-DOF Line-scan IBVS Stability Analysis

With $k = 3$

- global asymptotic stability if $\mathbf{L_e}\widehat{\mathbf{L_e}^+} > 0$
- OK if $\widehat{\mathbf{L_e}^+}$ is not a too coarse approximation

In practice, more feature points are used.

With $k > 3$

- $\mathbf{L_e}\widehat{\mathbf{L_e}^+}$ is not full rank (it at most of rank 3)
- configurations such that $\mathbf{e} \in \mathrm{Ker}\widehat{\mathbf{L_e}^+}$ correspond to local minima
- local asymptotic stability in a neighborhood of $\mathbf{e} = \mathbf{e}^* = 0$ if $\widehat{\mathbf{L_e}^+}\mathbf{L_e} > 0$
- *Size of the neighborhood in unknown... but large in practice*

Attraction domain computation

Convergence domain computation

- Find the attraction domain of the desired position
- Compute the attraction domain of the desired position **in the camera pose space**
  - Dimension 3, regardless of the number of points
  - Enables to derive a safe working area in the world metric coordinate frame

Camera pose evolution model

$$\dot{\mathbf{p}} = \begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{v}_c$$

- camera body frame speed $\mathbf{v}_c = (v_x, v_y, \omega)^\top$

$$\mathbf{v}_c = -\lambda \mathbf{L}_{\mathbf{x}^*}^{+}(\mathbf{x} - \mathbf{x}^*)$$

- scan-line points $\mathbf{x} = (x_1, \ldots, x_k)^\top$

$$x_i = \frac{X_i}{Z_i} = \Pi(\mathbf{X}_i) = \Pi\left(\underbrace{{}^c\mathbf{T}_w(\mathbf{p})}_{\text{world to camera transform}} \left(\overbrace{{}^w\mathbf{X}_i}^{\text{world point}} \right) \right)$$

Visibility constraint

- Points have to be in the camera field of view to be observed
- We consider a classical IBVS policy: all the points have to remain in camera the field of view, otherwise, the system stops and fails
- The visibility constraint is expressed as

$$\forall i \in \{i, \ldots, k\}, x_i \in [-fov, +fov]$$

(where *fov* represents the field of view on the image line).

Capture basin approximation method

- **Capture basin** C
  Set of poses from which a target set T is reached in a finite time, without leaving the set of acceptable configurations K.

$$C = \{\mathbf{p} \mid \exists t \geq 0,\ \varphi(t, \mathbf{p}) \in T \text{ and } \varphi([0, t], \mathbf{p}) \subseteq K\}$$

- K is defined by the points visibility constraint (field of view)

$$K = \left\{\mathbf{p} \in \mathbb{R}^3 \mid \Pi(\mathbf{X}(\mathbf{p})) \in [-fov, +fov]^k\right\}$$

- If T is in the attraction basin of the desired position and is stable with respect to K, i.e $\forall \mathbf{p} \in T, \lim_{t \to \infty} \varphi(t, \mathbf{p}) = \mathbf{p}^*$ and $\forall \mathbf{p} \in T, \forall t \geq 0, \varphi(t, \mathbf{p}) \in K$, then this also applies to C

Capture basin computation

- Point implications
  - $\mathbf{x} \in T \Rightarrow \mathbf{x} \in C$
  - $\mathbf{x} \notin K \Rightarrow \mathbf{x} \notin C$
  - $\exists t \geq 0,\ \varphi(t, \mathbf{x}) \in C \text{ and } \varphi([0,t], \mathbf{x}) \subseteq K \Rightarrow \mathbf{x} \in C$
  - $\exists t \geq 0,\ \varphi(t, \mathbf{x}) \notin C \text{ and } \varphi([0,t], \mathbf{x}) \cap T = \emptyset \Rightarrow \mathbf{x} \notin C$
- Interval implications
  - $[\mathbf{x}] \subseteq T \Rightarrow [\mathbf{x}] \subseteq C$
  - $[\mathbf{x}] \cap K = \emptyset \Rightarrow [\mathbf{x}] \cap C = \emptyset$
  - $\exists t \geq 0,\ \varphi(t, [\mathbf{x}]) \subseteq C \text{ and } \varphi([0,t], [\mathbf{x}]) \subseteq K \Rightarrow [\mathbf{x}] \subseteq C$
  - $\exists t \geq 0,\ \varphi(t, [\mathbf{x}]) \cap C = \emptyset \text{ and } \varphi([0,t], \mathbf{x}) \cap T = \emptyset \Rightarrow [\mathbf{x}] \cap C = \emptyset$

(Lhommeau et al, 2011)

Capture basin computation

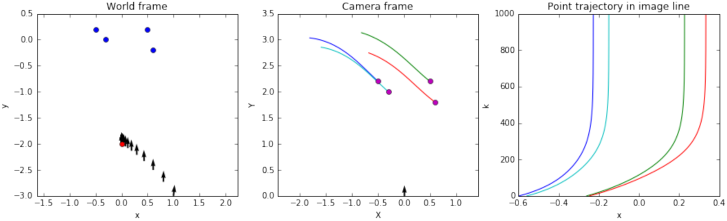
Enclose C between an inner approximation C^- and an outer approximation C^+

- Starting from a box (initial search domain)
  - Use interval constraint propagation to enforce the visibility constraint K (with Ibex)
  - Try to compute guaranteed integration (with CAPD)
  - If integration fails, bisect the box and put in C^+
  - Decide if the box is inside or outside the capture basin using previous relations:
    if inside, add to C^- and C^+ / discard if outside / put in C^+ otherwise
- Iteratively reprocess C^+ boxes not in C^- until fixed point

Results

Studied example

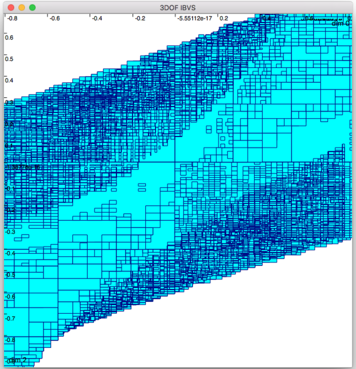
- 4 points: (-0.5, 0.2), (0.5, 0.2), (0.6,-0.2), (-0.3,0.0)
- desired pose $\mathbf{p}^* = (0, -2, 0)$
- desired image $\mathbf{x}^* = (\frac{-5}{22}, \frac{5}{22}, \frac{1}{3}, \frac{-3}{2})$
- Target is chosen as a small neighborhood of $\mathbf{p}^*$ (radius 0.005,0.005,00125)



Results

Computation in the domain
$([p_X],[p_y],[\theta]) = ([-1.1, 1.1], [-2.8, -0.6], [-1.08, 1.08])$
5cm

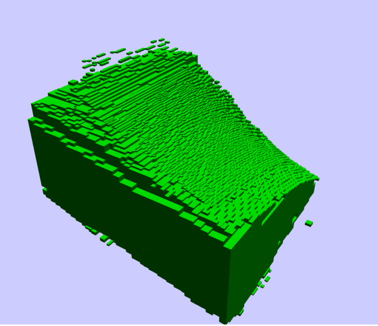
p_x, θ projection of the computed set



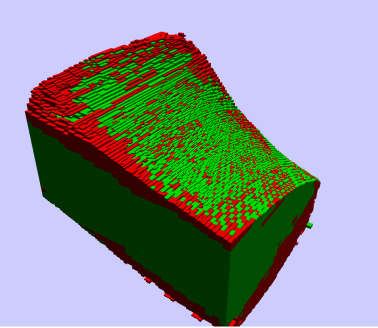
Results

$$([p_X],[p_y],[\theta]) = ([-1.1, 1.1],[-2.8,-0.6],[-1.08, 1.08])$$

C^- before last iteration



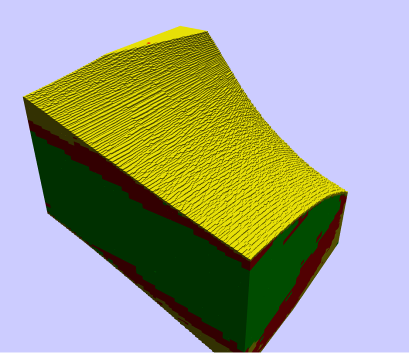
C^- last iteration result



Results

$$([p_X],[p_y],[\theta]) = ([-1.1,1.1],[-2.8,-0.6],[-1.08,1.08])$$

C^+ after last iteration



Summary

- Guarantee a convergence domain for 3DOF IBVS with line-scan camera
- Use of a capture basin approximation method with guaranteed integration : long computation time

- Outlook
  - Compute a neighborhood included in the attraction domain of the asymptotically stable equilibrium: study in image space and pose space
  - Move to a more realistic IBVS problem (classical 2D camera, 6DOF)