# Design of Visual Observer Using Hi-speed Extraction Algorithm of Feature Points

FL seminar
Fujita Laboratory Satoshi Sunaga
November 12th, 2010



## Outline



- Introduction and Background
- Visual Observer with SURF
  - Visual Observer
  - SURF
  - Visual Observer with SURF
- Simulations and Experiments
- Conclusion and Feature Works



## Introduction



■ **Sensor Network**

Networked multiple wireless sensor nodes

- Humidity
- Temperature
- Light etc.

http://jp.makezine.com/blog/2008/09/bop_making_sense_of_space.html

■ **Visual Sensor Network**

A network consisting of spatially distributed smart cameras

■ **Advantage**

- Accurate measurement
- Robustness of failures
- Larger sensing area

■ **Applications**

- Monitoring
- Measurement

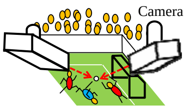



Fig.1 Monitoring

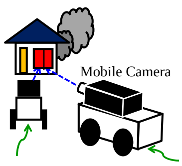



Fig.2 Measurement



## Previous Work and a Problem



**Visual Observer [1]**

Estimate object's relative positions and orientations in 3 dimension from the image information

**Networked Visual Observer [2]**

Construct camera network to run cooperative estimation

- Accurate measurement
- Robustness of failures

**Problem**

Use artificial feature points in experiment

**Defect**

- Unrealistic
- Weak measurement

**Speeded-Up Robust Features (SURF)**

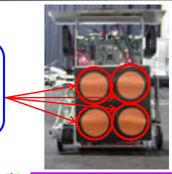
[1] M. Fujita, H. Kawai and M. W. Spong, "Passivity-based Dynamic Visual Feedback Control for Three Dimensional Target Tracking: Stability and L2-gain Performance Analysis," *IEEE Transactions on Control Systems Technology*, Vol. 15, No1, pp. 40-52, 2007.
[2] T. Nishi, "Three-Dimensional Motion Estimation based on Distributed Optimization Algorithms in Visual Sensor Networks," *Tokyo Institute of Technology bachelor thesis*, 2010.



## Visual Observer with Speeded-Up Robust Features



**Artificial Feature Points**



**Feature Points**

Extract feature points from visual image using special filters

**Advantage**

- unnecessity of artificial feature points
- Adaptiveness to variety objects
- Robustness to environmental disturbance

**Objective**

Consider the estimation problem using artificial feature points

Consider the estimation problem using feature points extracted by Speeded-Up Robust Features, not artificial feature points.



## Outline



- Introduction and Background
- Visual Observer with SURF
  - Visual Observer
  - SURF
  - Visual Observer with SURF
- Simulations and Experiments
- Conclusion and Feature Works

## Visual Observer[1]



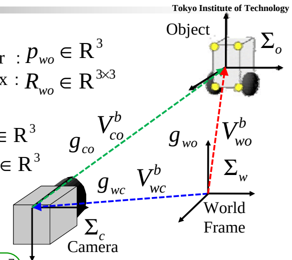
**Pose**

$$g_{wo} = \begin{bmatrix} R_{wo} & p_{wo} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}$$ Position Vector : $p_{wo} \in \mathbb{R}^3$; Rotation Matrix : $R_{wo} \in \mathbb{R}^{3\times3}$

**Body Velocity**

$$V^b_{wo} = \begin{bmatrix} v^b_{wo} \\ \omega^b_{wo} \end{bmatrix} \in \mathbb{R}^6$$ Body Linear Velocity : $v_{wo} \in \mathbb{R}^3$; Body Angular Velocity : $\omega_{wo} \in \mathbb{R}^3$

$$\hat{V}^b_{wo} = \begin{bmatrix} \hat{\omega}^b_{wo} & v^b_{wo} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}$$




**Kinematic Model**

$$\dot{g}_{wo} = g_{wo}\hat{V}^b_{wo}$$

$\wedge : \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

$\vee$ : Inverse effecter of $\wedge$

Relative Pose from camera to object $g_{co} = g_{wc}^{-1} g_{wo}$

**Relative Motion**

$$V^b_{co} := \left(g_{co}^{-1}\dot{g}_{co}\right)^\vee = -\mathrm{Ad}_{(g_{co}^{-1})} V^b_{wc} + V^b_{wo} \quad (1)$$

Transform from $V^b_{wc}$ to $V^b_{wo}$ : $\mathrm{Ad}_{(g_{co}^{-1})}$



## 付録:ビジュアルオブザーバ [1]

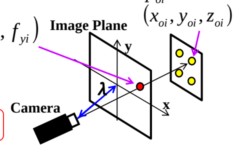


**Output**

Perspective Projection Model

$$f_i = \begin{bmatrix} f_{xi} \\ f_{yi} \end{bmatrix} = \frac{\lambda}{z_{ci}} \begin{bmatrix} x_{ci} \\ y_{ci} \end{bmatrix} \quad (p_{ci} = R_{co} p_{oi} + p_{co})$$

$f_i = (f_{xi}, f_{yi})$

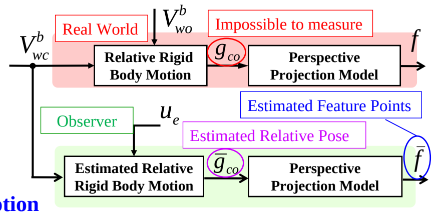



$g_{co}$ s.t. $R_{co}, p_{co}$ : Impossible to measure

Observer Model
Using feature points $f$

Estimated Relative Pose

$$\bar{g}_{co} = \begin{pmatrix} \bar{R}_{co} & \bar{p}_{co} \\ 0 & 1 \end{pmatrix}$$




**Estimated Relative Motion**

$$\bar{V}^b_{co} := -\mathrm{Ad}_{(\bar{g}_{co}^{-1})} V^b_{wc} + u_e \quad (1)$$

$u_e$ : Observer Input



## 付録:ビジュアルオブザーバ[1]



**Objective**

$\bar{g}_{co}$ is nearly equal to $g_{co}$

**Theorem**

If $V^b_{wo} = 0$ , then the equilibrium point $e_e = 0$ for the closed-loop system (2) and (3) is asymptotic stable.

$\bar{g}_{co} \to g_{co} \quad (t \to \infty)$

**Dynamics of Estimation Error**

$$V^b_{ee} = \left(g_{ee}^{-1}\dot{g}_{ee}\right)^\vee = -\mathrm{Ad}_{(g_{ee}^{-1})} u_e + V^b_{wo} \quad (2)$$

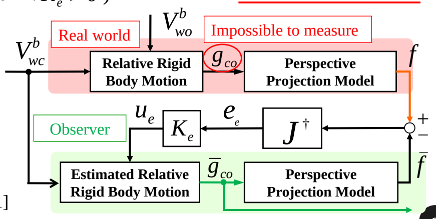
**Observer Input**

$$u_e = K_e e_e \quad (3) \qquad (\text{Gain} : K_e > 0)$$

$J$ : Image Jacobian

$g_{ee}$ : Estimation Error $g_{ee} = (\bar{g}_{co}^{-1} g_{co})^\vee$

$e_e$ : The Vector of Estimation Error derived from $f$ [1]






## Speeded-Up Robust Features (SURF) [3]



**Speeded-Up Robust Features (SURF) [3]**

- One of feature extraction methods
- Simple and fast computation
- Robust to rotation, scale, and and illumination change

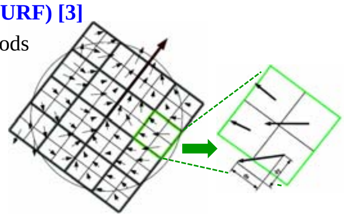


Fig.3 Descriptor with 16 Sub Region

**Procedure**

1. Detection of feature points
   Hessian matrix (Find corner of the image)
2. Derivate the orientation and features
   Calculate gradient $dx,\ dy$
   Approximation with filter

Feature : $4 \times 16 = 64$ dimensions

Each sub region : $\left(\sum dx \quad \sum|dx| \quad \sum dy \quad \sum|dy|\right)$

[3] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding(CVIU)*, Vol.110, Issue 3, pp.346-359, June 2008.



## Derivation of Object’s Motion from Feature Points



**Note :** Use some object’s feature points ( Diminish calculations of Jacobian )

### Derive Target Object’s Motion

- Derive feature-points-set coordinates from object’s feature points and captured image’s feature points
  - About target’s features
    - ➡ Use feature points which can easily be captured
    - ➡ Decide that feature points before estimation of object’s motion
- Send coordinates about feature-points-set coordinates
- Calculate object’s motion



## Estimation of Motion about an Object



Step 1. Setting object frame
Step 2. Decide feature points of a target object
Step 3. Matching feature points of camera vision
Step 4. Derive feature-points-set
Step 5. Estimate motion of the object by Visual Observer

## Outline



- Introduction and Background
- Visual Observer with SURF
  - Visual Observer
  - SURF
  - Visual Observer with SURF
- Simulations and Experiments
- Conclusion and Feature Works

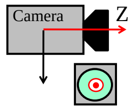


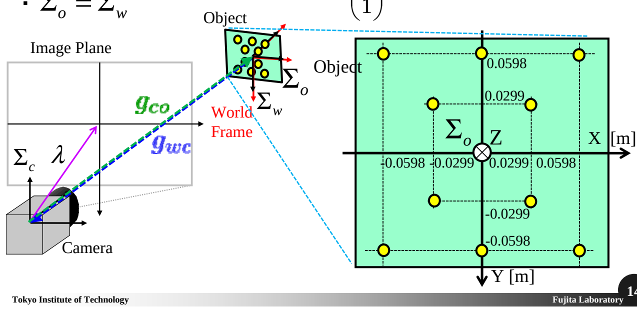
## Simulation of Visual Observer



### Setting

- Focus length : $\lambda = 0.0033$  $p_{co} = (0.25, 0.12, 1)$
- Relative Pose : $g_{co} = (p_{co}, e^{\hat{\xi}\theta_{co}})$  $\xi_{co} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, $\theta_{co} = 1[\text{rad}]$
- $\Sigma_o = \Sigma_w$



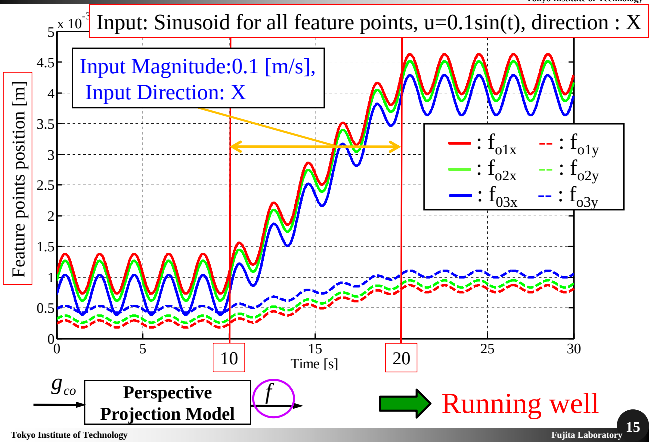






## Simulation : Object Feature Point on Image Plane



Input: Sinusoid for all feature points, u=0.1sin(t), direction : X



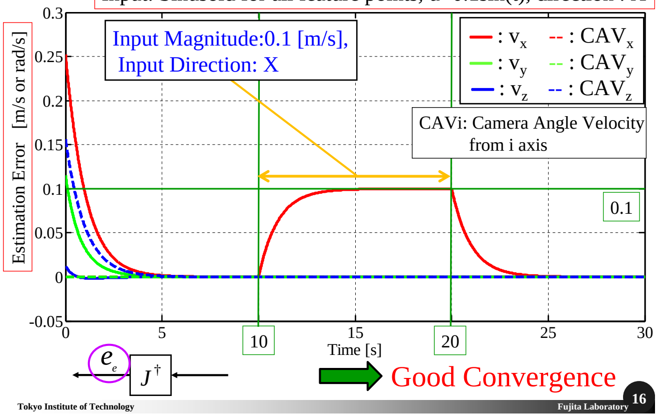

Running well



## Simulation : Estimation Error



Input: Sinusoid for all feature points, u=0.1sin(t), direction : X




Good Convergence



## Program



### Spec. of Program

- Compare with captured present feature points and advanced target's feature points
- Save feature coordinates from camera and target's coordinates
  - ➡ Possible to match feature points
- Set calibration time span
  - ➡ Get upper 10 target's feature points with high probability

### However…

Impossible to get 10 feature points pairs

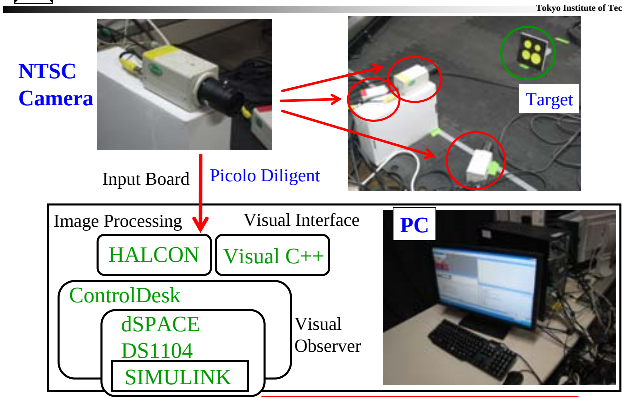
⬇

Improve Visual C++ program



## Experiment Framework






**Impossible to capture image!**

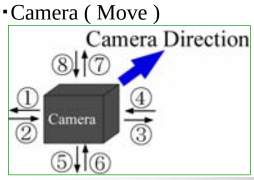
## Captured feature points on dSPACE (Control Desk)



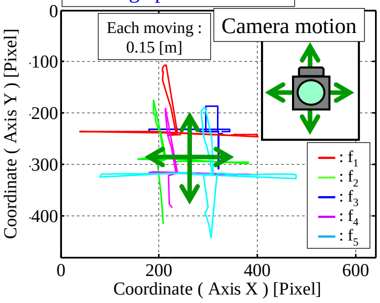
### Setting ( Linear Moving )

- Send feature points data only
- Simulation time: 45 [s]
- Target: e-nuvo Fixed

- Camera ( Move )




Feature point trajectories on image plane




Not so bad



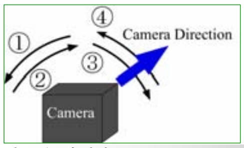
## Captured feature points on dSPACE (Control Desk)



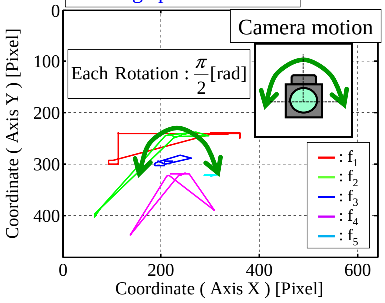
### Setting ( Rotation )

- Send feature points data only
- Simulation time: 30 [s]
- Target: e-nuvo

- Camera motion




Feature point trajectories on image plane




Not so bad



## Outline



- Introduction and Background
- Visual Observer with SURF
  - Visual Observer
  - SURF
  - Visual Observer with SURF
- Simulations and Experiments
- Conclusion and Feature Works



## Conclusion and Feature Works



### Conclusion

- Build program with SURF in OpenCV
- Study theory of Visual Observer
- Do distinct element simulations
- Do basic experiment with Web camera

### Feature Works

◎ Improve C++ program

◎ Simulate Visual Observer

◎ Construct visual interface

○ Study about theory multi camera network and develop that environment



## Reference



[1] M. Fujita, H. Kawai and M. W. Spong, “Passivity-based Dynamic Visual Feedback Control for Three Dimensional Target Tracking: Stability and L2-gain Performance Analysis,” *IEEE Transactions on Control Systems Technology*, Vol. 15, No1, pp. 40-52, 2007.

[2] T. Nishi, ”Three-Dimensional Motion Estimation based on Distributed Optimization Algorithms in Visual Sensor Networks,” *Tokyo Institute of Technology bachelor thesis*, 2010.

[3] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, “Speeded-Up Robust Features(SURF),” *Computer Vision and Image Understanding(CVIU)*, Vol.110, Issue 3, pp.346-359, June 2008.

[4] R. Murray, Z. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

[5] H. Kawai, “Visual Feedback Control”, *Fujita Laboratory* , 2003.

# Appendix

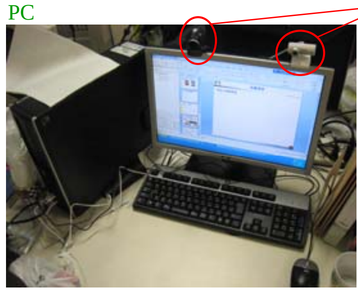
## Equipments



# Experimental Environment



## In Student Room




Developing Environment

- Microsoft Visual C++ 2005
- OpenCV 1.1 pre1

Test of cameras



# New Experimental System