Intuitive Information Transfer Techniques for Mobile Devices by Toss and Swing Actions

Koji Yatani[1] Koiti Tamura[1] Masanori Sugimoto[1]
Hiromichi Hashizume[2]

[1]Department of Frontier Informatics, University of Tokyo
[2]Multimedia Information Research Division, National Institute of Informatics

Outline of This Talk

- Backgrounds
- ***Toss-It*** Project
- Implementation
  - Overview
  - Hardware
  - Software
- Evaluations
- Conclusions and Future Plans

Backgrounds

- Mobile devices (cellular phones, PDAs) have rapidly penetrated into our daily lives

  e.g. cellular phones in Japan: 87 million (about 3/4 of total population) in March 2004.

- Multi-functional devices have recently gained popularity among people

  e.g. digital (video) camera, web, e-mail, movie, games, GPS, music player, calculator …

  → people have begun to use a mobile device like a personal computer.

- User interfaces of mobile devices still need to be improved

  e.g. problems of small screen real estate or awkward / bothersome input method, as compared with desktop/laptop computers

Backgrounds

Information transfer among multiple mobile devices in co-located situations

- Using a memory card

  Insert a card into or remove it from each device

- Using an infrared communication

  People come close to each other and place their devices to make connections

- Using a bluetooth network
- Using IP network

  Select / write a host name or an IP address of devices…

⬇

More intuitive or simple operations are required!!

Backgrounds

Examples of object transfer in the real world

- In a co-located situation, we can pass a physical object to a person as we would toss a ball
- In a face-to-face meeting by surrounding a table, we can distribute sheets of documents as we would deal cards

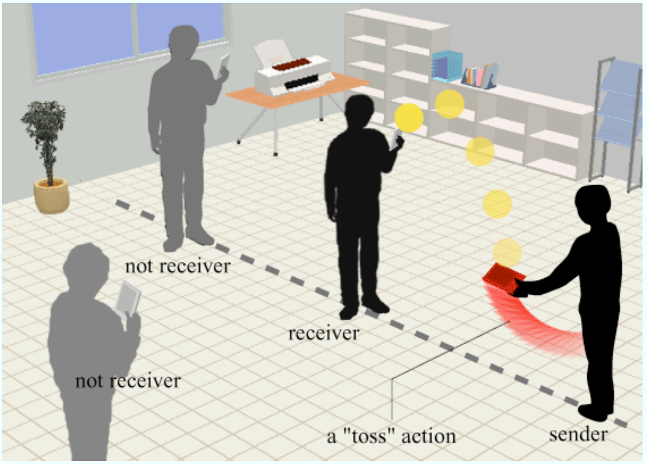
⬇

Challenge: Is it possible to do the same way for digital objects?

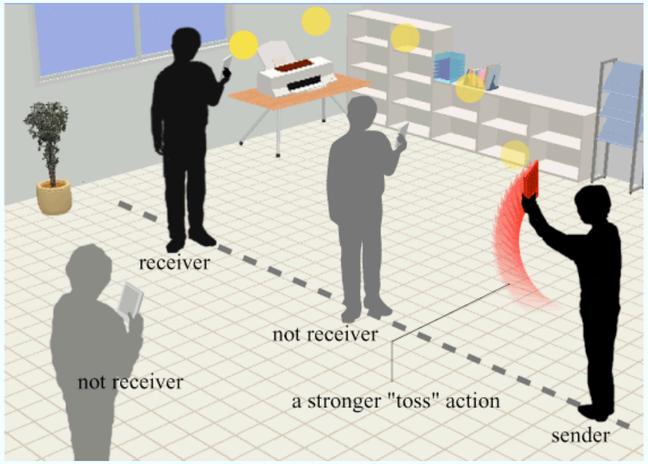
Toss-It Project: overview (1)

Goal: innovative and intuitive information transfer techniques *by fully utilizing the mobility of devices* (PDAs, in this study)

- A user can send information to a receiver nearby by conducting a *toss* action with his mobile device

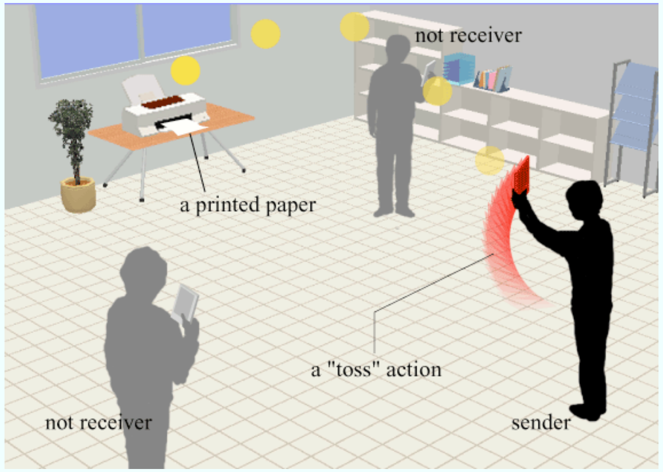


- A user can send information to a receiver beyond a person in-between by conducting a *stronger toss* action with his mobile device

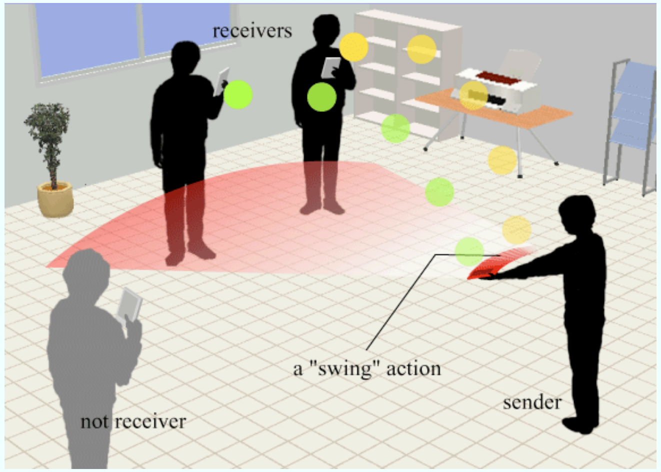


Toss-It Project: overview (2)

- A user can send information to an electronic device (e.g. a printer to print out a photo, or an LCD projector to project a document file) by conducting a *toss* action with his mobile device



- A user can simultaneously send information to multiple receivers around him by conducting a *swing* action with his mobile device



Toss-It Project: demo video

Toss-It Project:
design requirements and implementation (1)

[Requirements for *Toss-It*]

1. *Toss-It* must recognize users' toss and swing actions
2. *Toss-It* must automatically identify positions and orientations of devices (or users)
3. *Toss-It* must allow users to send selected information to and receive it from target devices via wireless network

Toss-It Project:
design requirements and implementation (2)

Our final goal: *Toss-It* is available anywhere

1. *Toss-It* must recognize users' toss and swing actions anywhere
2. *Toss-It* must automatically identify positions and orientations of devices (or users) anywhere

→ Using special equipment installed in an environment (e.g. installing cameras on a ceiling or a wall) for users' gesture / location / orientation recognition is inappropriate

→ Equipment for the recognition must be installed on each mobile device

3. *Toss-It* must allow users to send selected information to and receive it from target devices via wireless network anywhere

→ Using a server computer or a WLAN base station is inappropriate

→ Using P2P / Ad-hoc network technologies is preferable

Toss-It Project:
design requirements and implementation (3)

Our initial goal: to investigate whether information transfer by toss and swing actions is possible

1. *Toss-It* must recognize user's toss and swing actions anywhere

   → Main issue in this talk

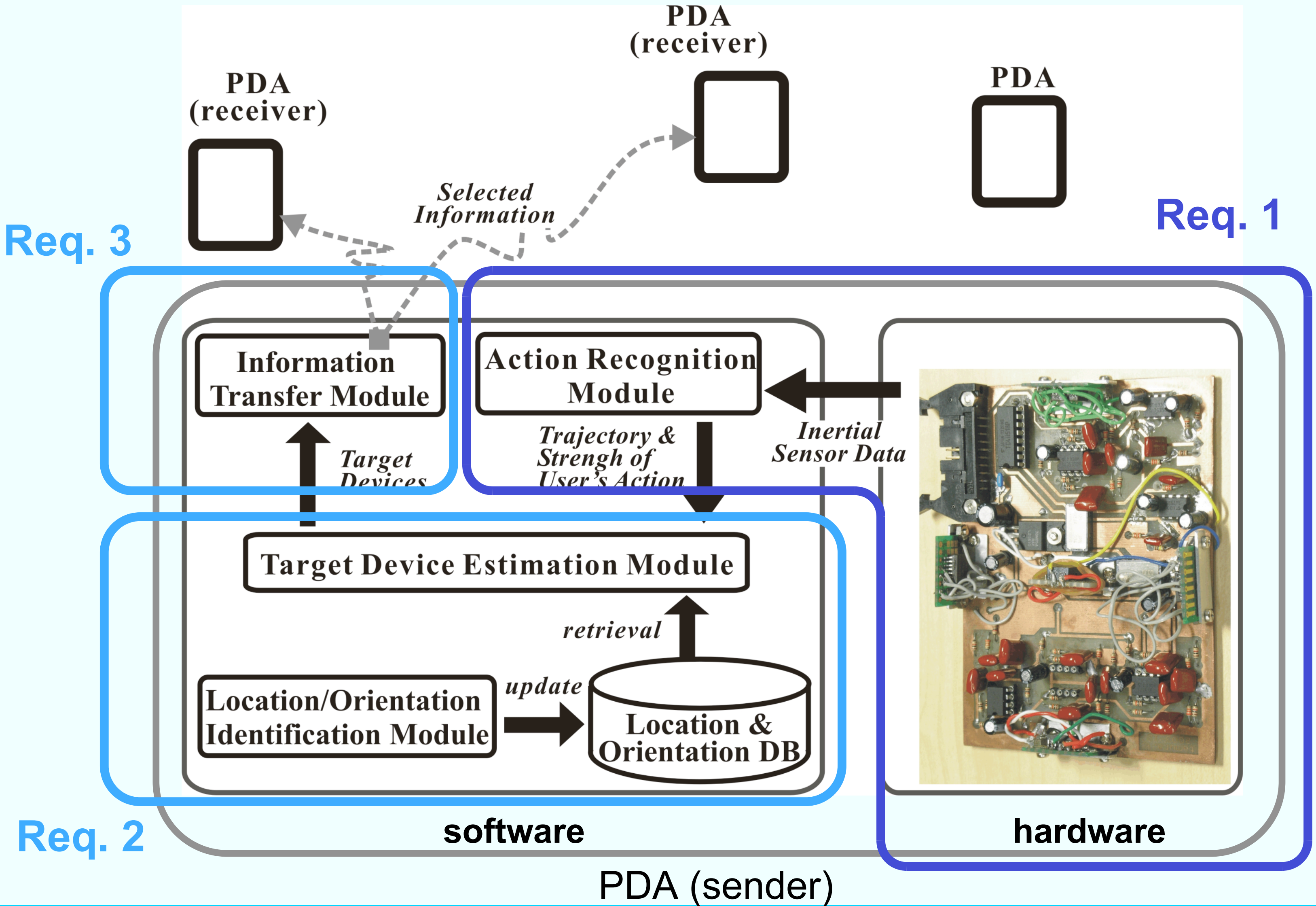
2. *Toss-It* must automatically identify positions and orientations of devices (or users)

   → We use a camera-based recognition system in this phase

3. *Toss-It* must allow users to send selected information to and received it from designated devices via wireless network

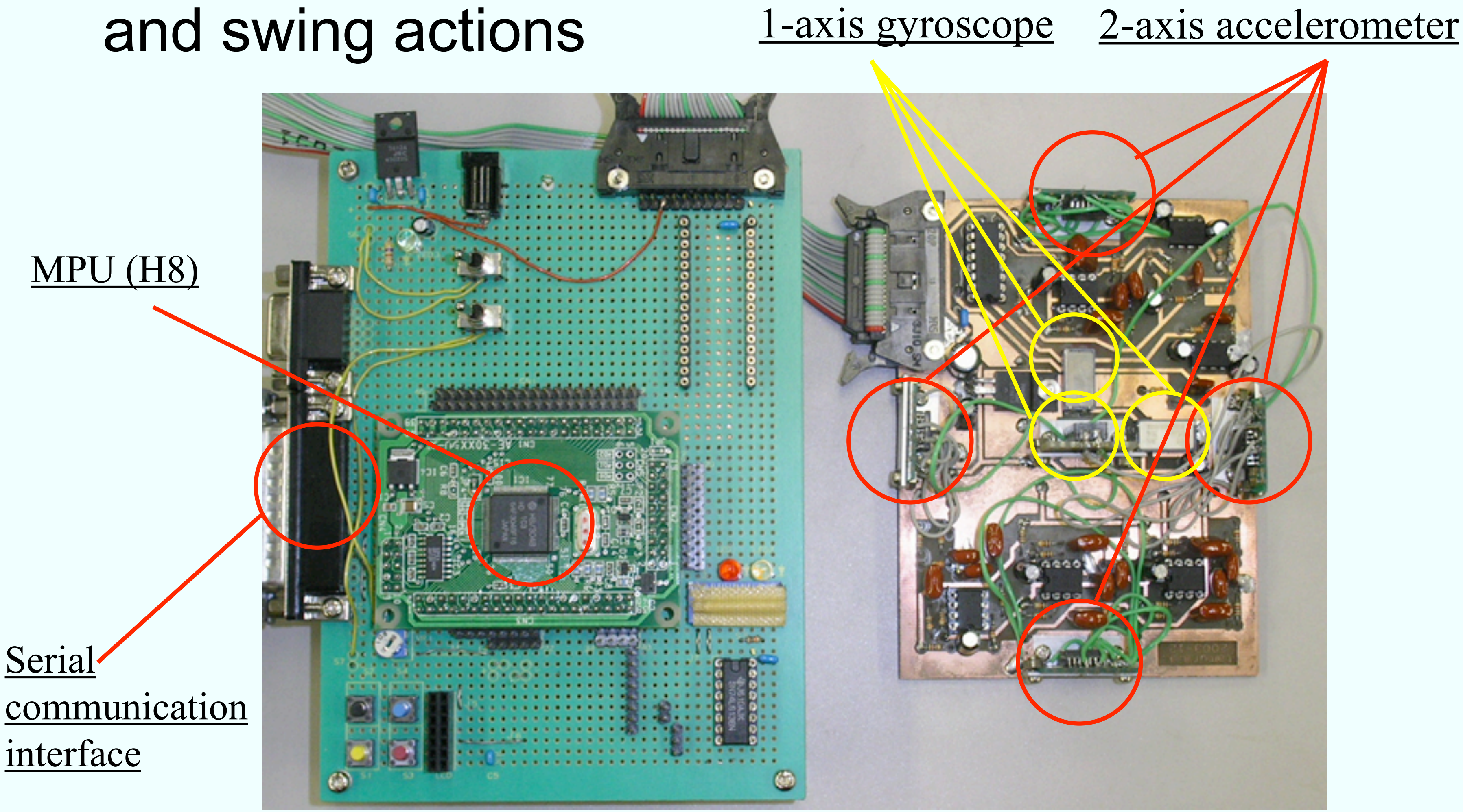
   → We use a server computer and a WLAN base station in this phase

Implementation: overview

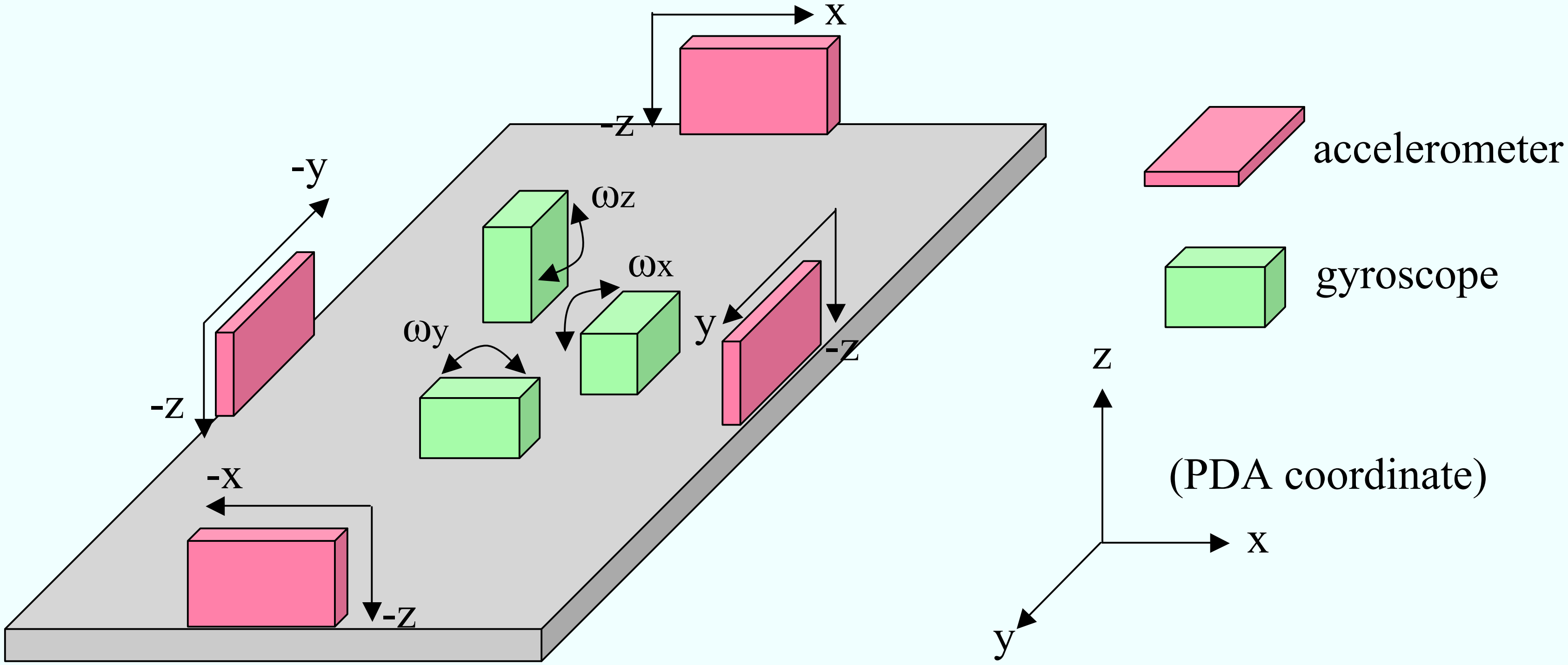


Implementation: hardware (1)

Prototype circuit board for capturing user's toss and swing actions



Implementation – hardware (2)

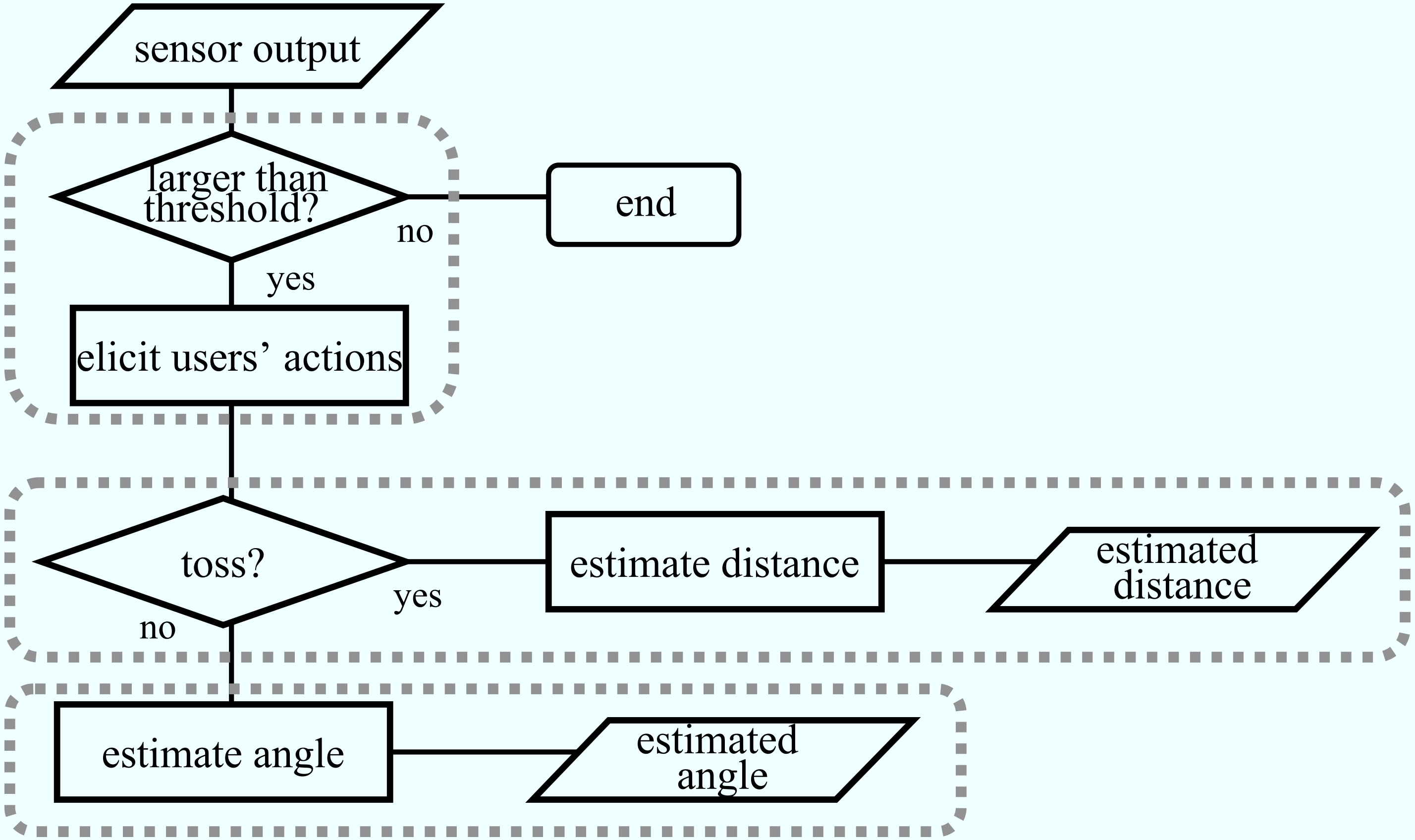


- four 2-axis accelerometers (Analog Devices ADXL210)
- three 1-axis gyroscopes (Murata ENC-03J, ENC-03M)

→ 11 degrees of freedom (> 6 degrees of freedom): Redundant design for capturing a user’ s quick action as accurate as possible

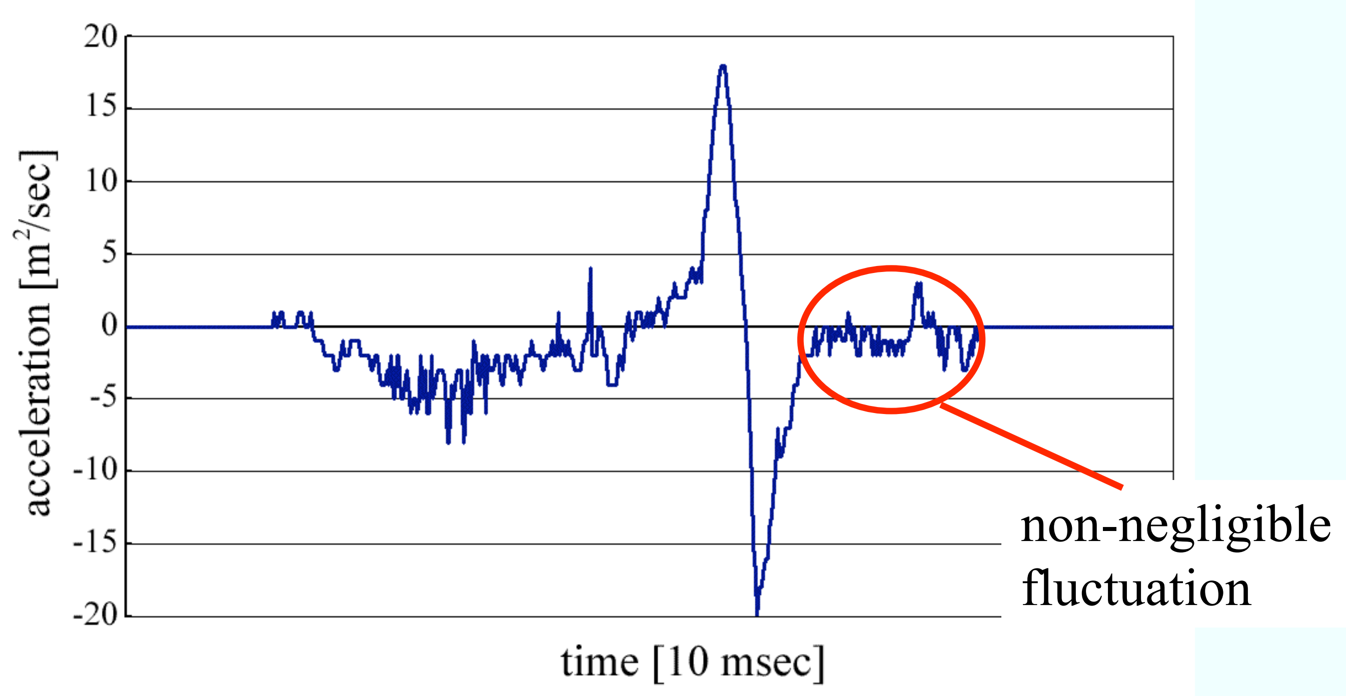
- low pass filter for eliminating high frequency noises (cut-off frequency: 1kHz)

Implementation :
software for toss or swing actions



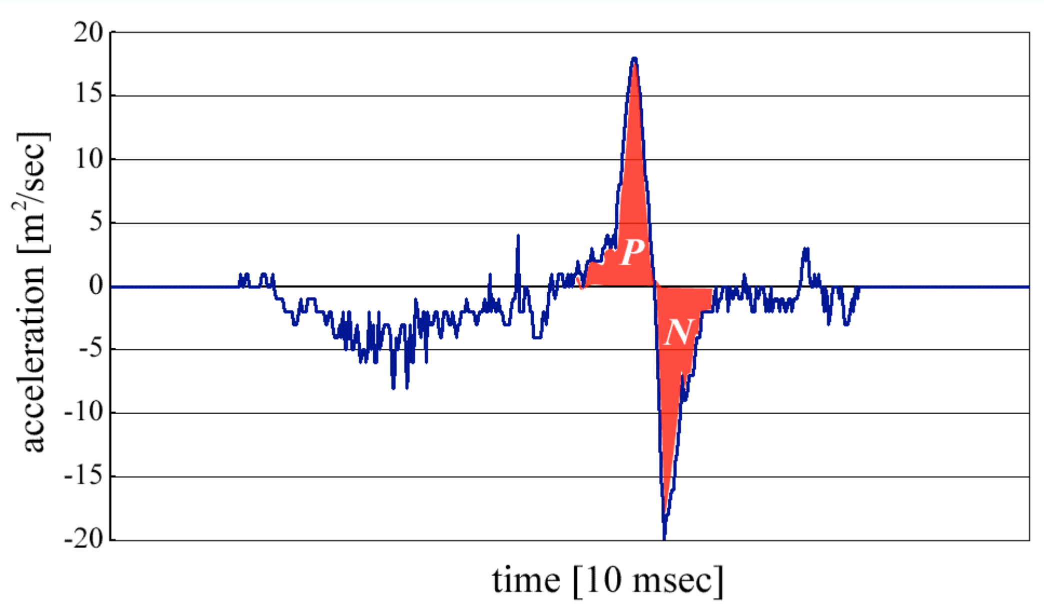
Implementation :
software for eliciting users' actions (1)

- Several experiments have indicated that non-negligible fluctuation occurs in the output data just after a user's action has been completed



Implementation :
software for eliciting users' actions (2)

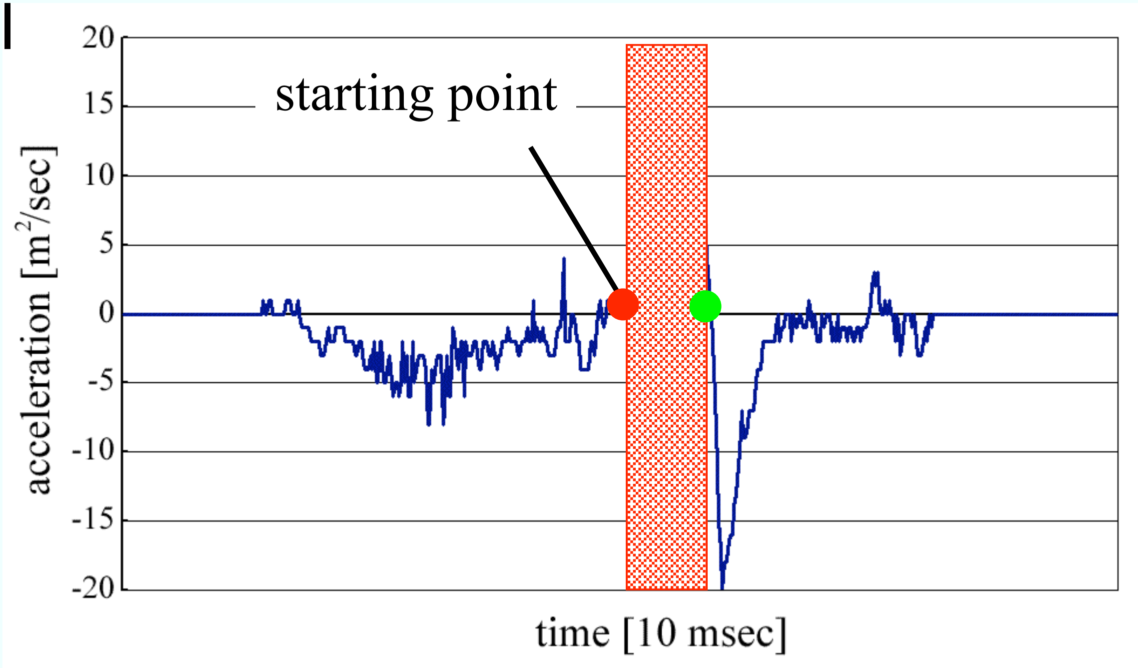
- Through evaluations of several algorithms in different settings, an assumption has been made: a toss or swing action starts and ends in a state of rest
- This assumption justifies the idea that the area of the positive part (*P* in the figure) is equal to that of the negative part (*N* in the figure)



Implementation :
software for eliciting users' actions (3)

[Algorithm]

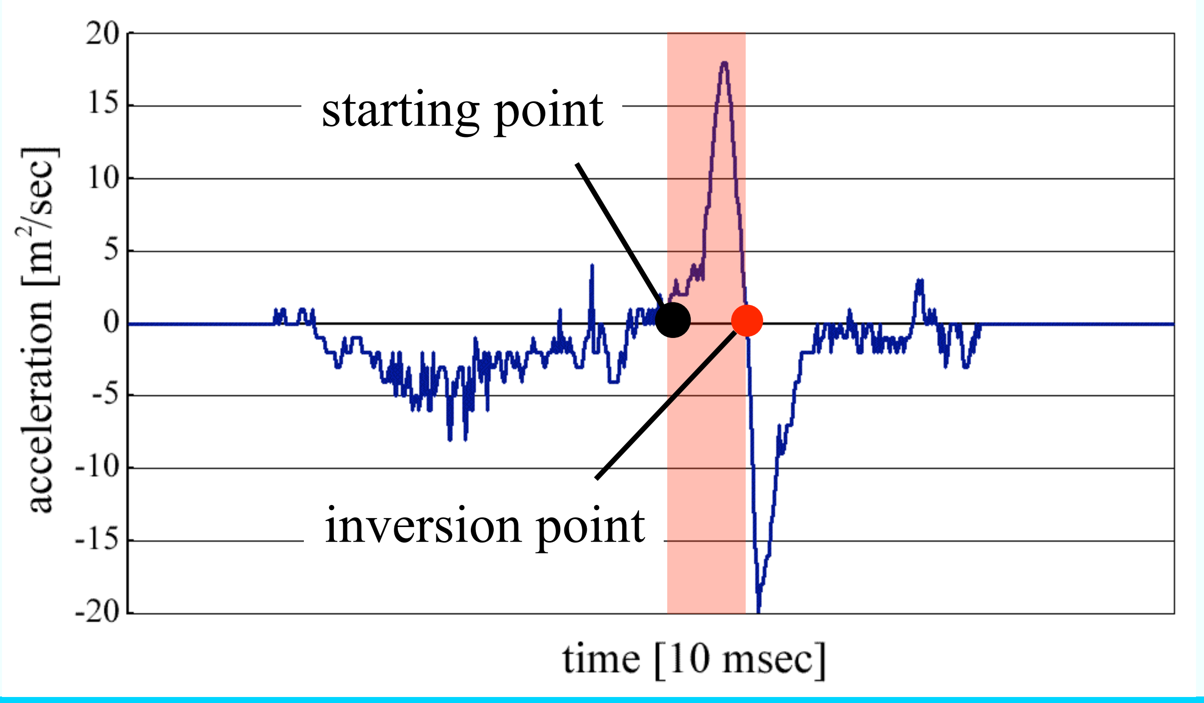
1. Search an intersecting point of the output data curve and the zero acceleration line
2. Calculate the integral of the acceleration values between the intersecting point and the previous intersecting point, named the "starting point"



Implementation:
software for eliciting users' actions (4)

[Algorithm]

3. If the integral value is greater than a specified threshold, regard the intersecting point as the "inversion point" and begin to calculate the integral of the acceleration values from the inversion point

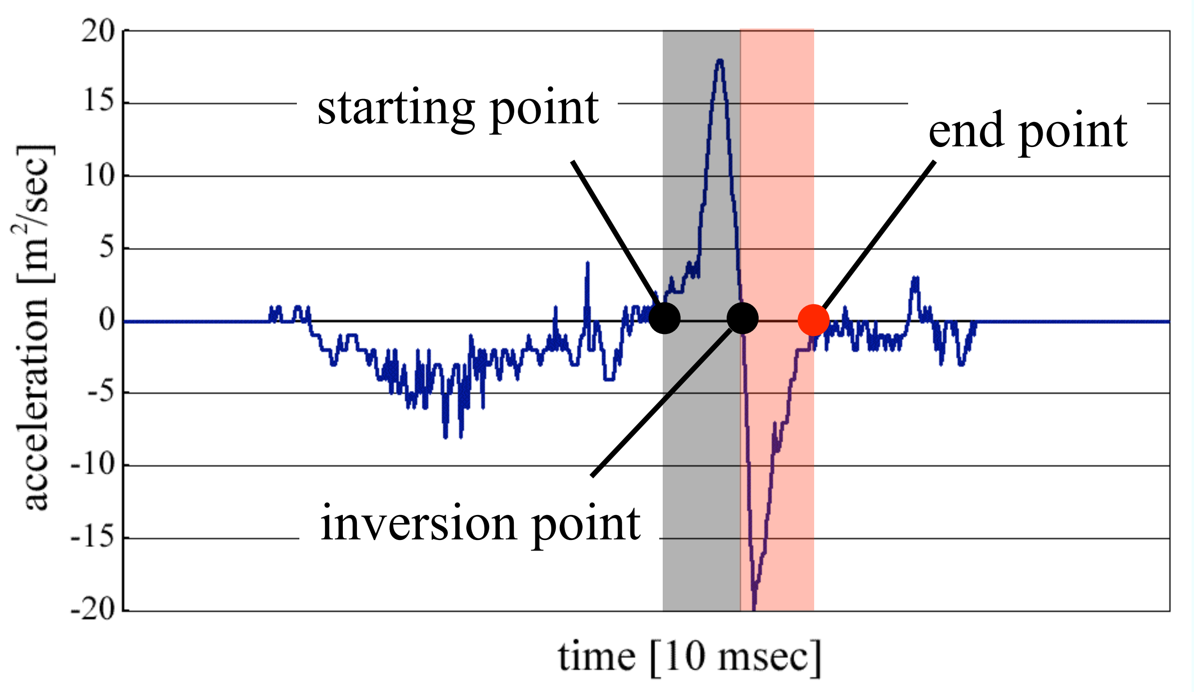


Implementation:
software for eliciting users' actions (5)

[Algorithm]

4. When the summation of the two integral values (the integral between the starting and inversion points, and the integral from the inversion point) becomes approximately zero, stop the calculation and regard the current point as the "end point"

5. Decide the starting and the end point that corresponds to a user's action



Implementation:
software for recognizing toss actions (1)

[Estimating the distance]

- Estimate the strength of a toss in order to determine how far “tossed” information travels and which device receives the information
- Through the evaluations of several algorithms in different settings, assumptions for simplifying the estimation were made
  - A “tossed” object is released at the maximum speed
  - A toss action is started at the vertically downward position to the floor and finished without a follow through
- Calculate the launch angle by integrating the output data from the gyroscopes

Implementation:
software for recognizing toss actions (2)

[Algorithm]

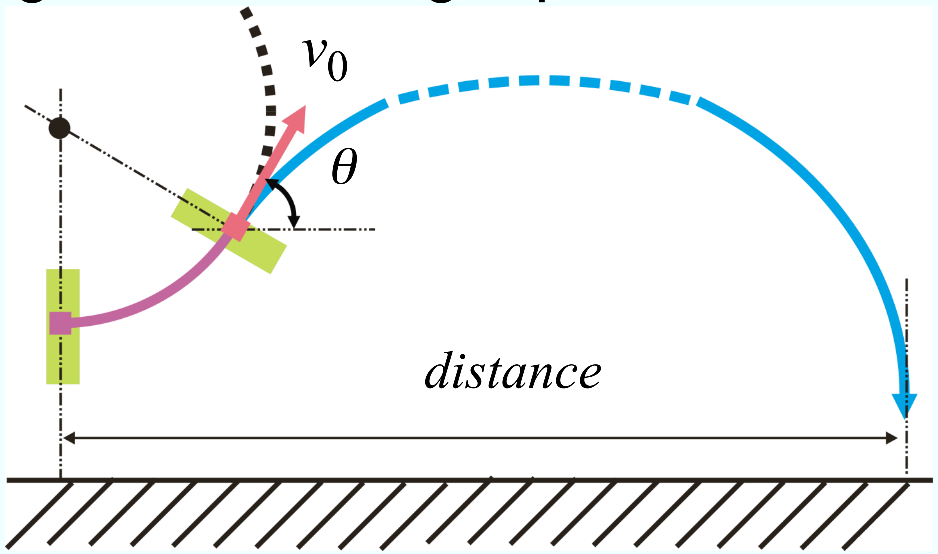
1. Regards a point of the maximum velocity as a release point of "tossed" information
2. Calculate the launch angle through the outputs of the gyroscopes
3. Estimate the distance through the following equation :

$$distance = \frac{v_0^2 \sin 2\theta}{g}$$

v_0 : maximum speed

θ : launch angle

g : gravity acceleration



Implementation:
software for recognizing swing actions (1)

[Estimating the trajectory of swing actions]

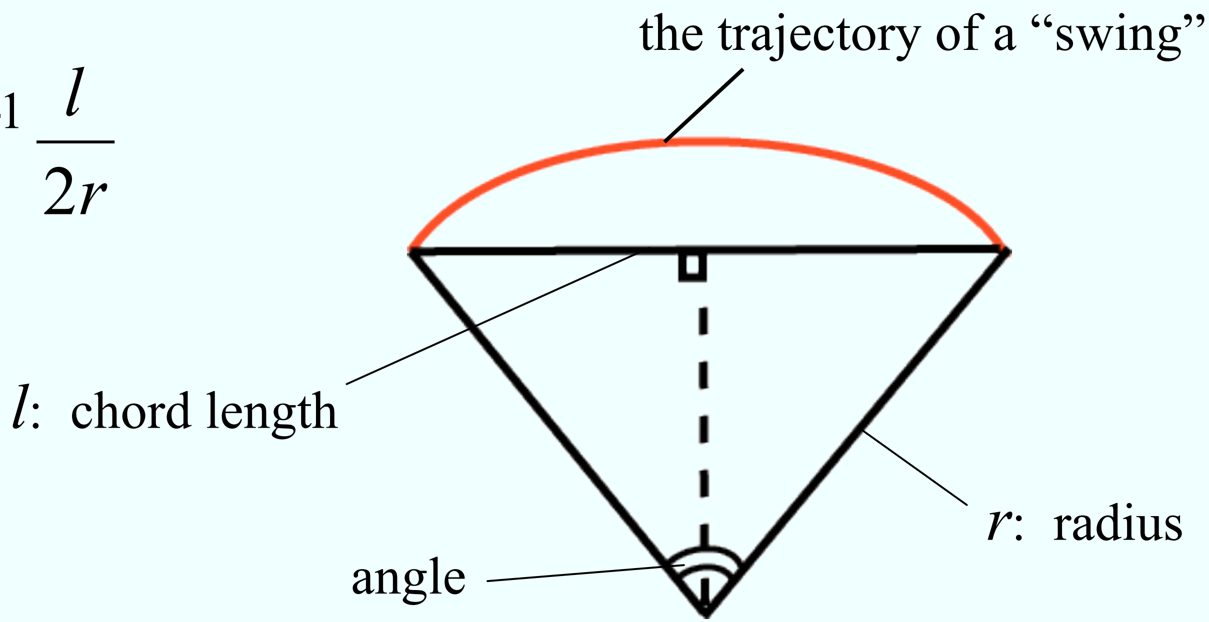
- Estimate how many degrees a user has swung his PDA, in order to determine who the receivers are
- Assumptions for simplifying the estimation were made:
  - A trajectory of a swing is an arc
  - A radius of an arc is decided based on each user's arm length
  - A swing consists of horizontal moves only

Implementation:
software for recognizing swing actions (2)

[Algorithm]

1. Calculate the chord length of an arc from the estimated trajectory
2. Calculate the angle through the following equation:

$$angle = 2\sin^{-1}\frac{l}{2r}$$



Implementation: Transformation from the PDA to the world coordinates

[Algorithm]

1. Calculate the rotation matrix $\boldsymbol{R}(\boldsymbol{k})$ through the output of gyroscopes
2. Calculate an acceleration vector in the PDA coordinate by eliminating the influence of the gravitational acceleration
$$\boldsymbol{a}_{PDA}(k) = \boldsymbol{a}_{sensor}(k) - \boldsymbol{R}^{-1}(k)\boldsymbol{g}_0$$
3. Calculate a velocity vector in the world coordinate (T : sampling time)
$$\boldsymbol{v}_{abs}(k) = \boldsymbol{v}_{abs}(k-1) + T\boldsymbol{R}(k)\boldsymbol{a}_{PDA}(k)$$
4. Calculate a position vector in the world coordinate
$$\boldsymbol{p}_{abs}(k) = \boldsymbol{p}_{abs}(k-1) + T\boldsymbol{v}_{abs}(k)$$

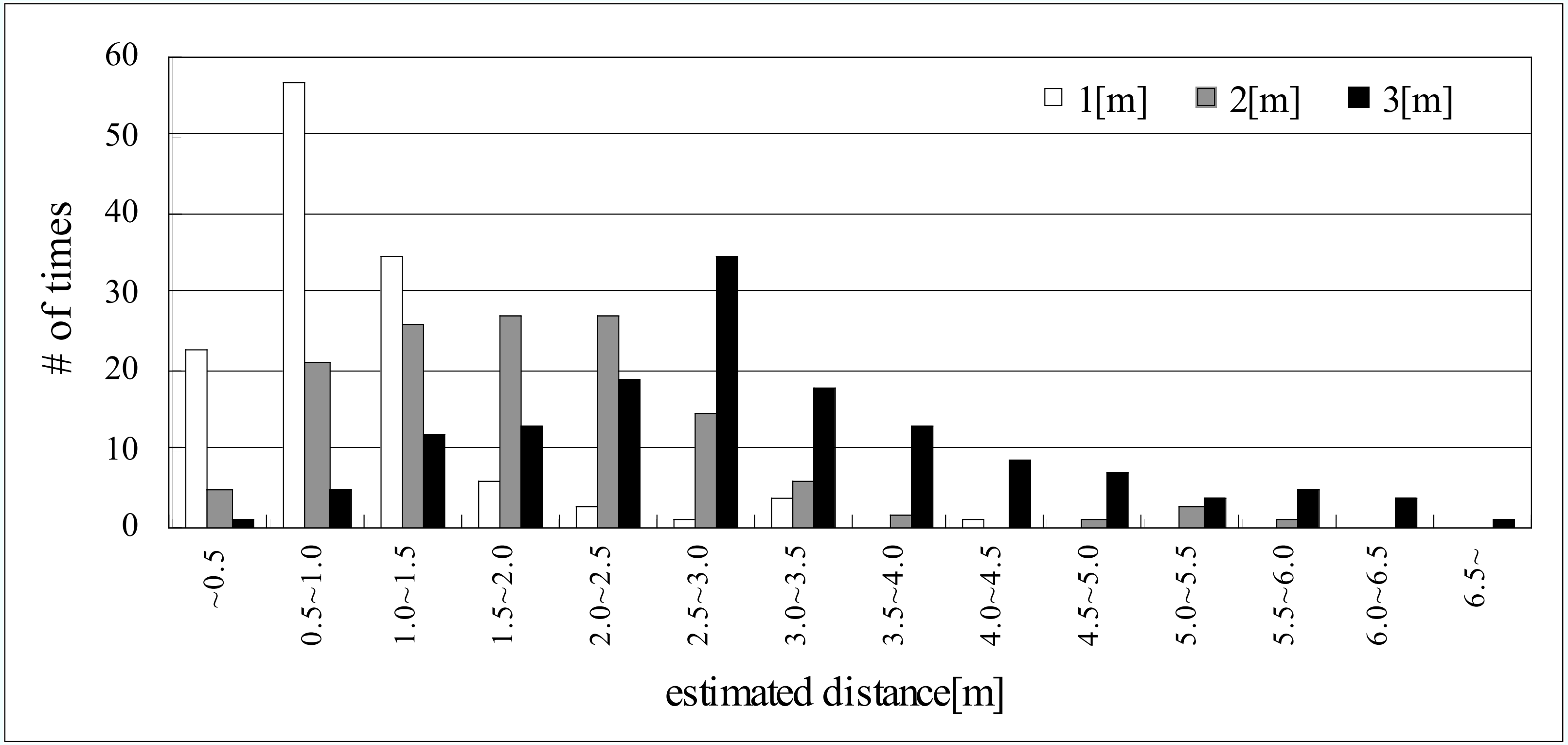
($\boldsymbol{a}$:accelerometer $\boldsymbol{v}$:velocity $\boldsymbol{p}$:position)

Evaluations (1)

- *Toss-It* could completely distinguish between a "toss" (vertical move) and a "swing" (horizontal move)
- User studies on how accurately *Toss-It* could recognize receivers with toss and swing actions.
- Six subjects were asked to conduct the following tasks:
  - conduct a toss action to send information to devices placed at three different locations (1[m], 2[m], and 3[m] away from a subject)
  - conduct three different horizontal swing actions with their PDAs (45[deg], 90[deg], and 135[deg] around a subject)
- Each subject repeated toss and swing actions 25 times for each of the three locations and three angles, respectively

Evaluations (2)

[Experimental results for toss actions]



Evaluations (3)

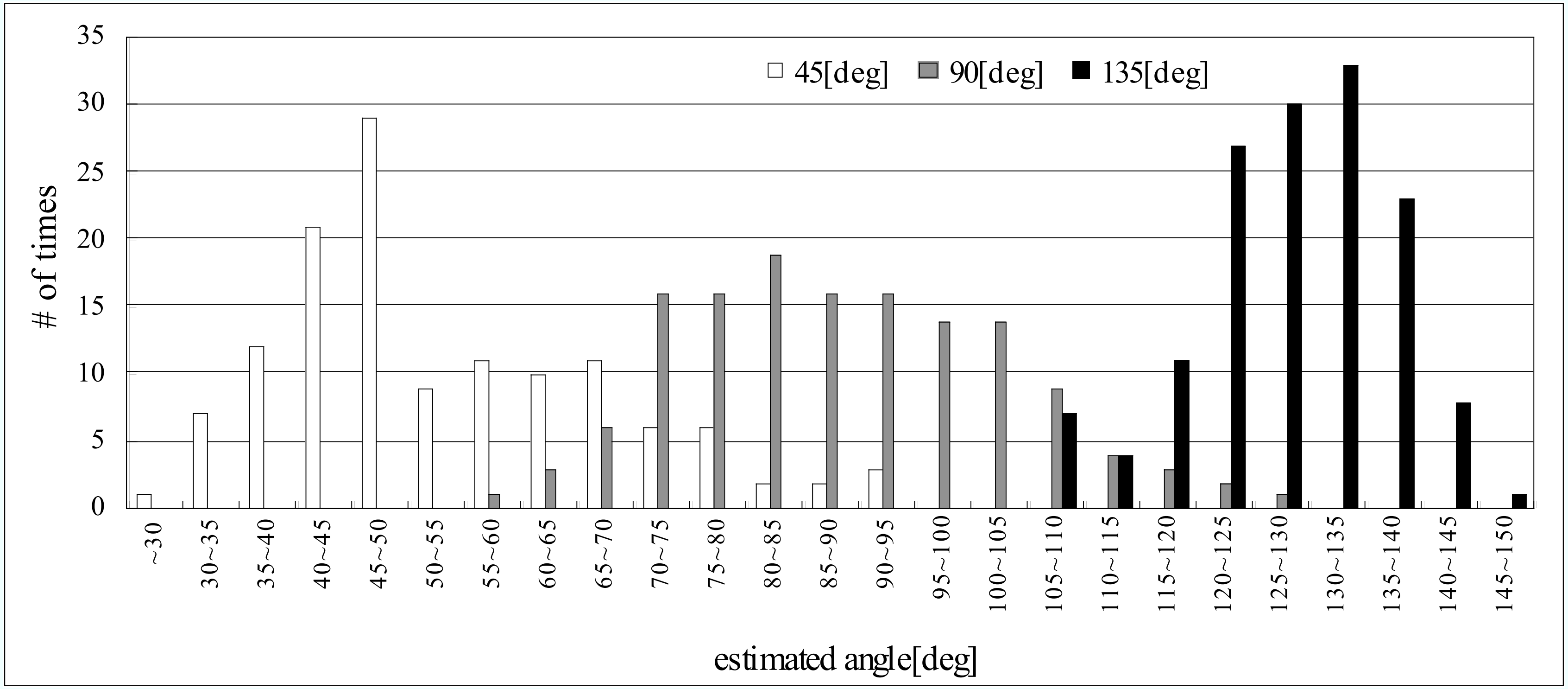
[Summary of the results]

Target distance [m]	Average [m]	Standard Deviation
1	0.995	0.683
2	1.91	1.01
3	3.02	1.35

- The average of the estimated distance is very close to each of the target distances
- The variation/standard deviation of the estimated distance becomes larger, as the corresponding target distance is longer.
- Differences between estimated distances by toss actions for two target distances (1[m] - 2[m], and 2[m] - 3[m]) proved to be statistically significant by Welch's paired t-test (two-tailed, $p<.01$)

Evaluations (4)

[Experimental results for swing actions]



Evaluations (5)

[Summary of the results]

Target angle [deg]	Average [deg]	Standard Deviation
45	54.2	14.8
90	89.6	14.3
135	129	8.80

- The difference between the average of the estimated angle and the corresponding target angle is less than 10 degrees.
- The standard deviations of the estimated angles are less than 15.
- Differences between estimated angles by swing actions for two target angles (45 [deg] - 90 [deg], and 90 [deg] - 135 [deg]) proved to be statistically significant by Welch’s paired t-test (two-tailed, $p<.01$)

→The results of these user studies show that information transfer by toss and swing actions is possible, although the estimated values are not always accurate.

Conclusions and Future Plans

- Intuitive information transfer techniques for mobile devices: concept, design, implementation and preliminary evaluations
- Our idea (toss and swing actions for information transfer) has been proved to be possible (not sure if it is useful !!)
- More accurate estimation for users' actions is necessary
- Utilizing contextual information or receivers' gesture.
- Evaluations in a realistic setting by using location / orientation recognition technologies is necessary.
- Privacy and security issues
- Development and implementation of the location /orientation recognition system and P2P/ ad hoc network technologies

Thank you for your attention!!