Hybrid Tracking with Gravity Aligned Edges

Samuel Williams, Dr. Mark Billinghurst, Dr. Richard Green

Introduction

Augmented Reality

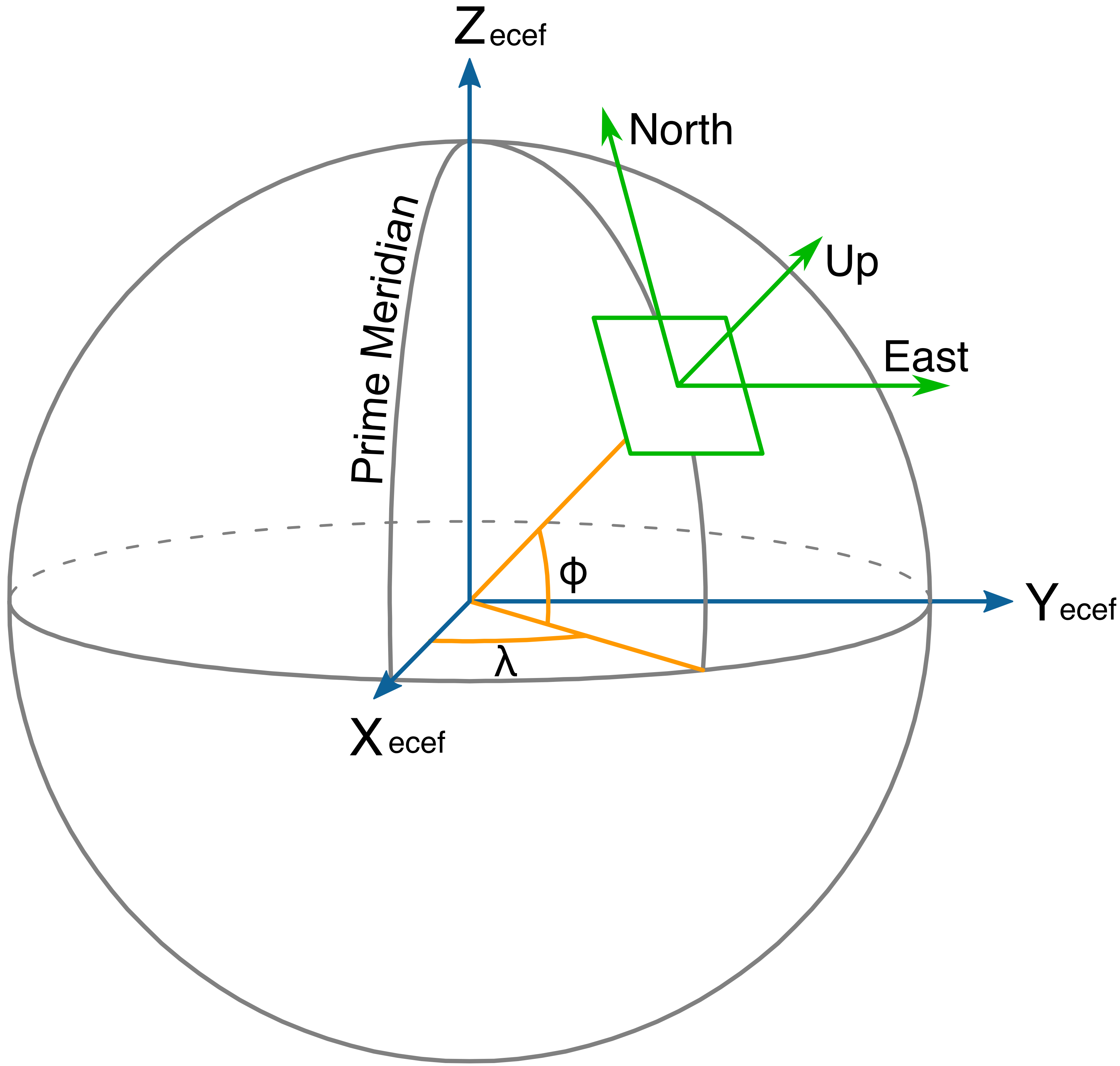
- Blend video frames and virtual content.
- Local tracking and global tracking.

Fundamental Issues

- Augmented reality systems deal with two fundamental technical challenges.
  - The camera’s position and orientation with respect to the real world.
  - The virtual object geometry and its accurate registration with the real world.

Camera Position

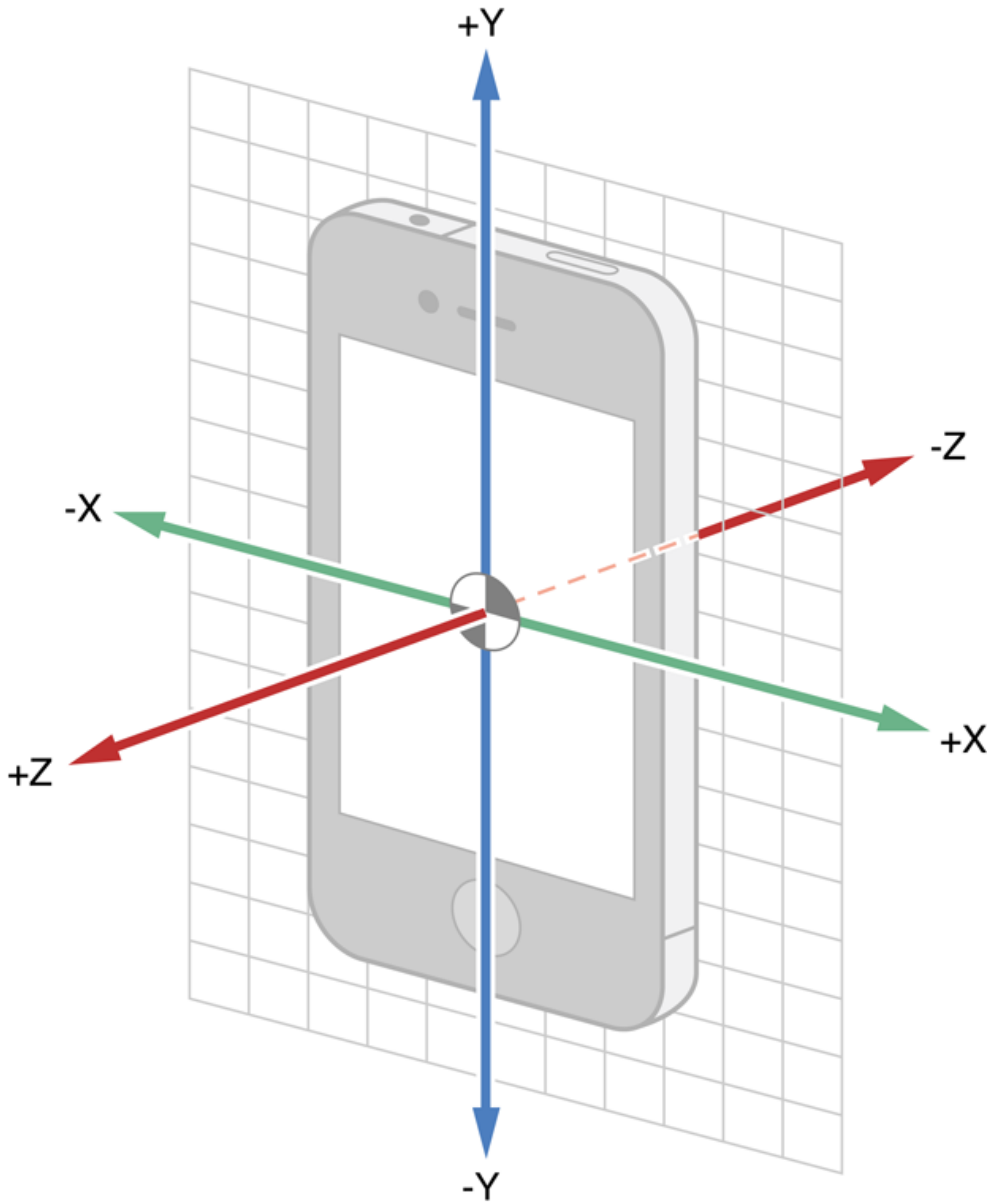
- World location in longitude, latitude and altitude, from GPS (WGS84).
- Relative change in position from starting point in <x,y,z> displacement, from local tracking (e.g. SLAM).



ECEF (Earth Centered Earth Fixed) and ENU (East North Up)

Camera Orientation

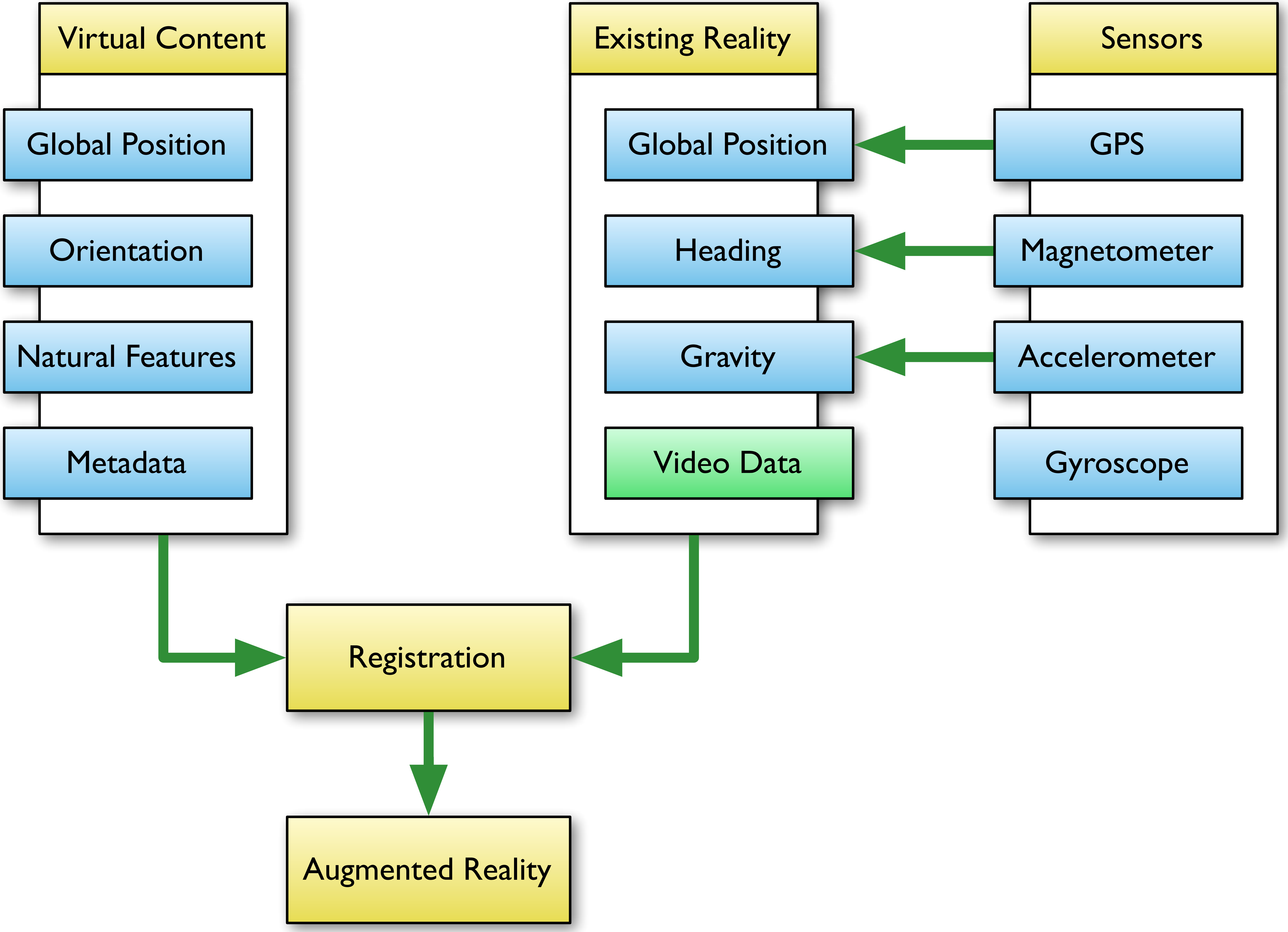
- Heading in degrees from north.
- Gravity as force vector <x,y,z> in g-force.



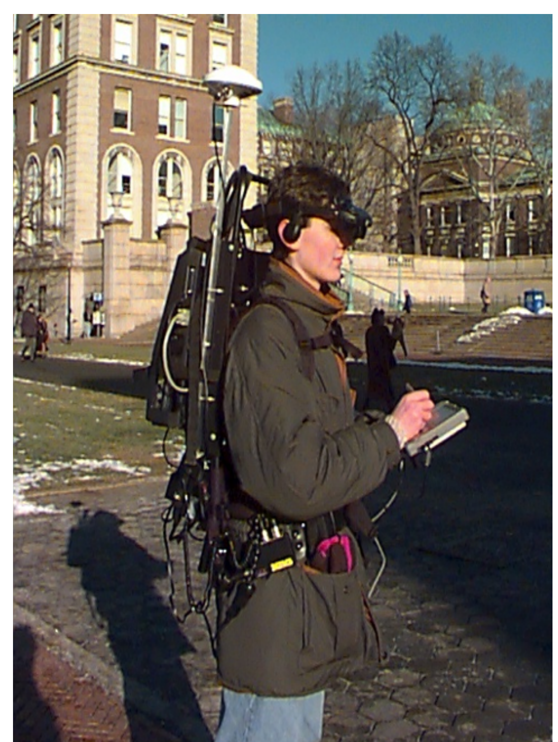
Device Frame of Reference

Registration

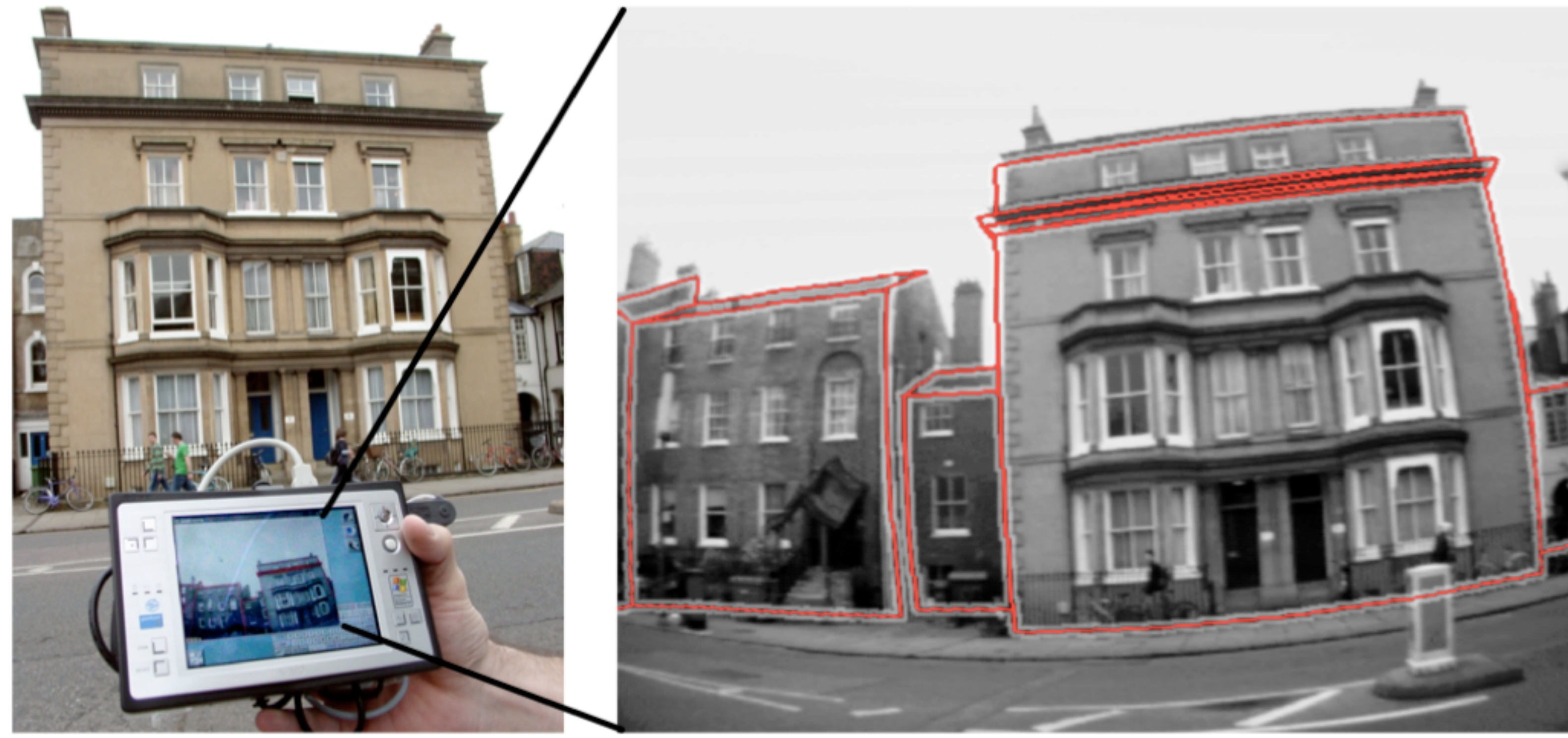
- Virtual content can be registered:
    - with fiducial markers,
    - with natural feature tracking,
    - with associated world coordinates.


Virtual Content
Global Position
Orientation
Natural Features
Metadata
Existing Reality
Global Position
Heading
Gravity
Video Data
Sensors
GPS
Magnetometer
Accelerometer
Gyroscope
Registration
Augmented Reality

Existing Research

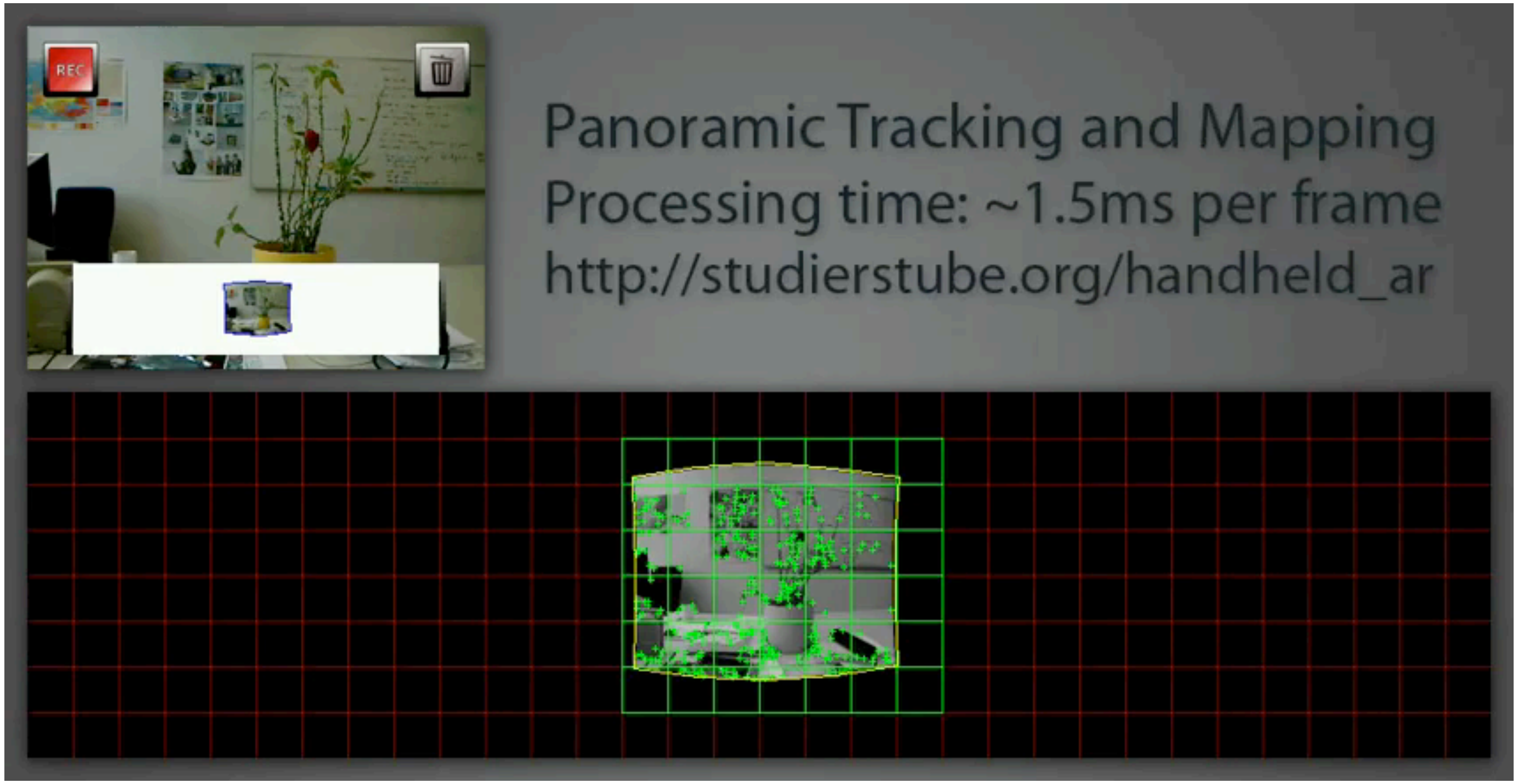


“Going Out”



- Model-based outdoor augmented reality.
- Requires 3D model of environment (vector geometry, point cloud).

“Panorama Tracking”



- Uses SLAM to track local orientation (3D) and position.


2degrees
6:37 pm

Problems

- Latency
- Processing
- Registration
- Precision
- Accuracy


No SIM
12:34 PM
37° NE
318° NW
43°31'19'' S, 172°34'59'' E
2degrees
3:57 PM
267° W
43°31'18'' S, 172°34'59'' E
Needle: 10 ° N
Heading: 350 ° N

Mobile Sensors

Sensor	Absolute Error	Relative Error	Latency
GPS	±20m	±10m	≤ 10s
Compass	±20°	±5°	≤ 2s
Accelerometer	±10° ±0.1g	±1° ±0.1m/s	≤ 20ms
Gyroscope	~0°	±0.1°/s	≤ 20ms

These are ball-park average to worst-case measurements.

Image Processing

- Memory and processing requirements.
- Failure cases - **drift**, **jittering**, **drop-outs**.

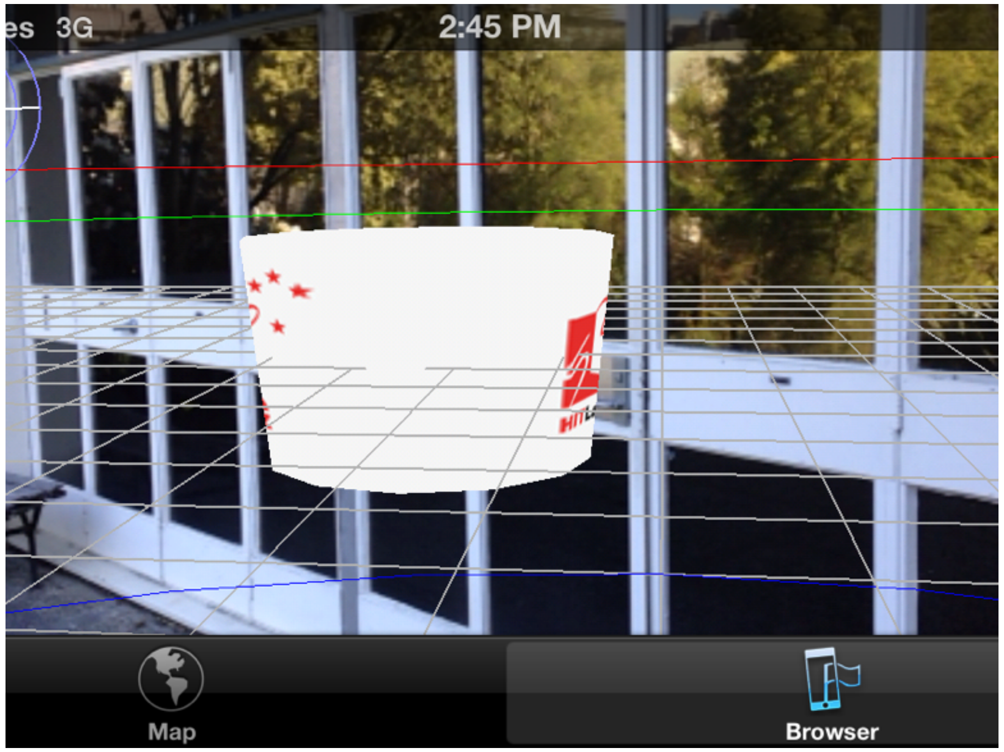
Goals

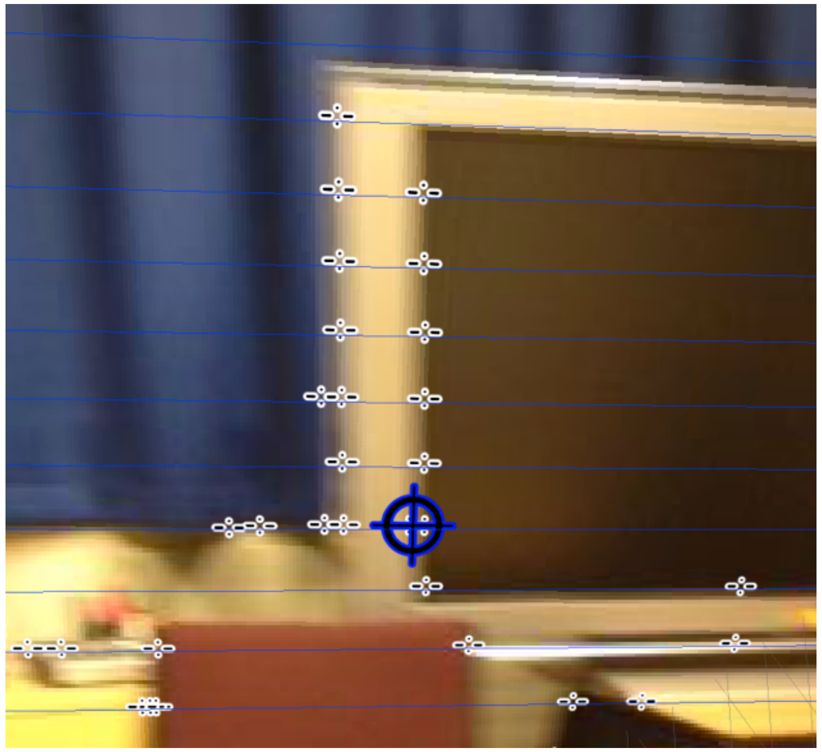
- Improve accuracy by combining sensor data to reduce error.
- Hybrid tracking to reduce performance requirements
- “Better” outdoor augmented reality for the end user.

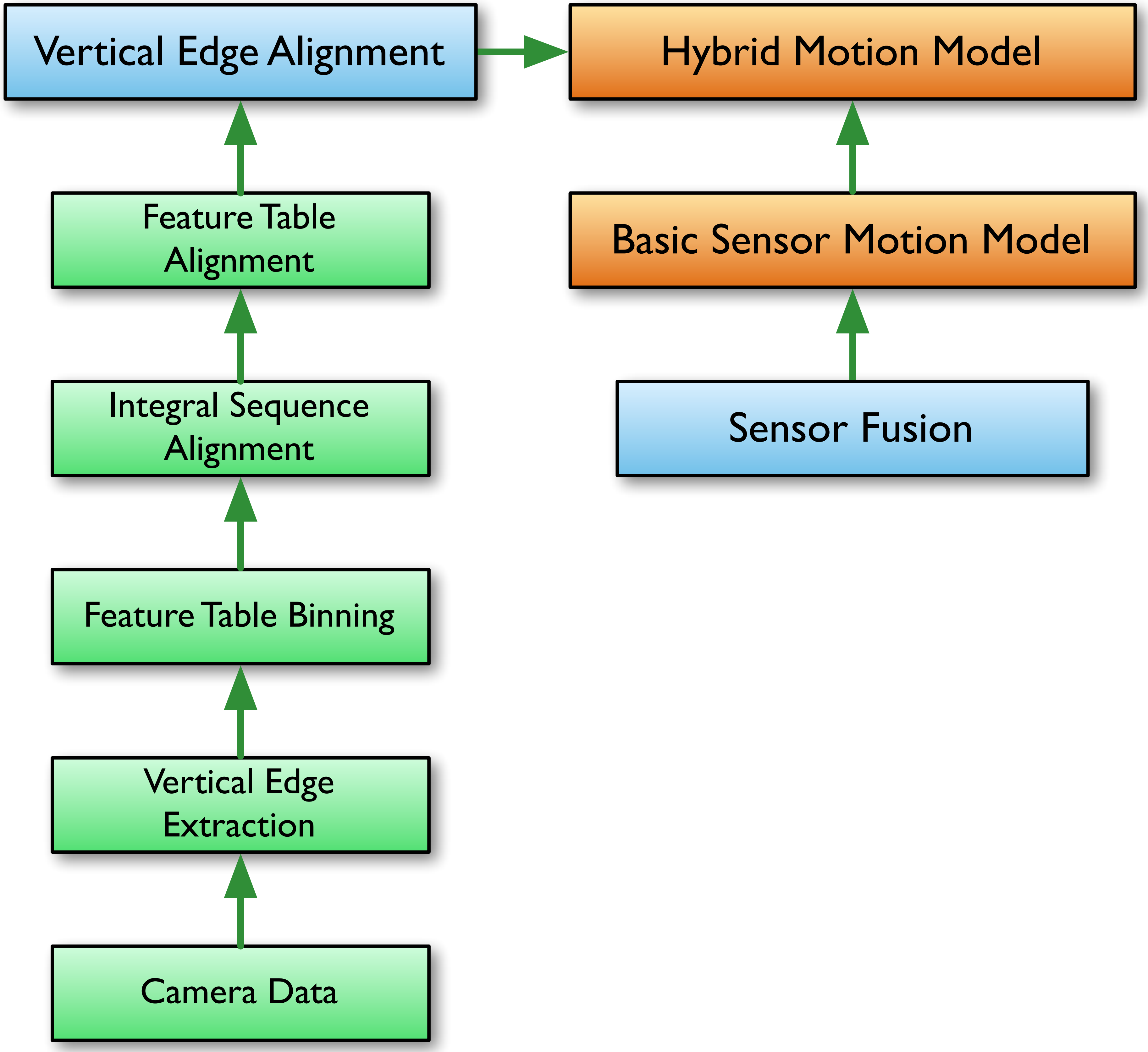
Hybrid Tracking

- By how much can we improve the stability and responsiveness of tracking by combining multiple sensors?
- Sensor data can provide cheap motion estimates.

Vertical Edge Alignment

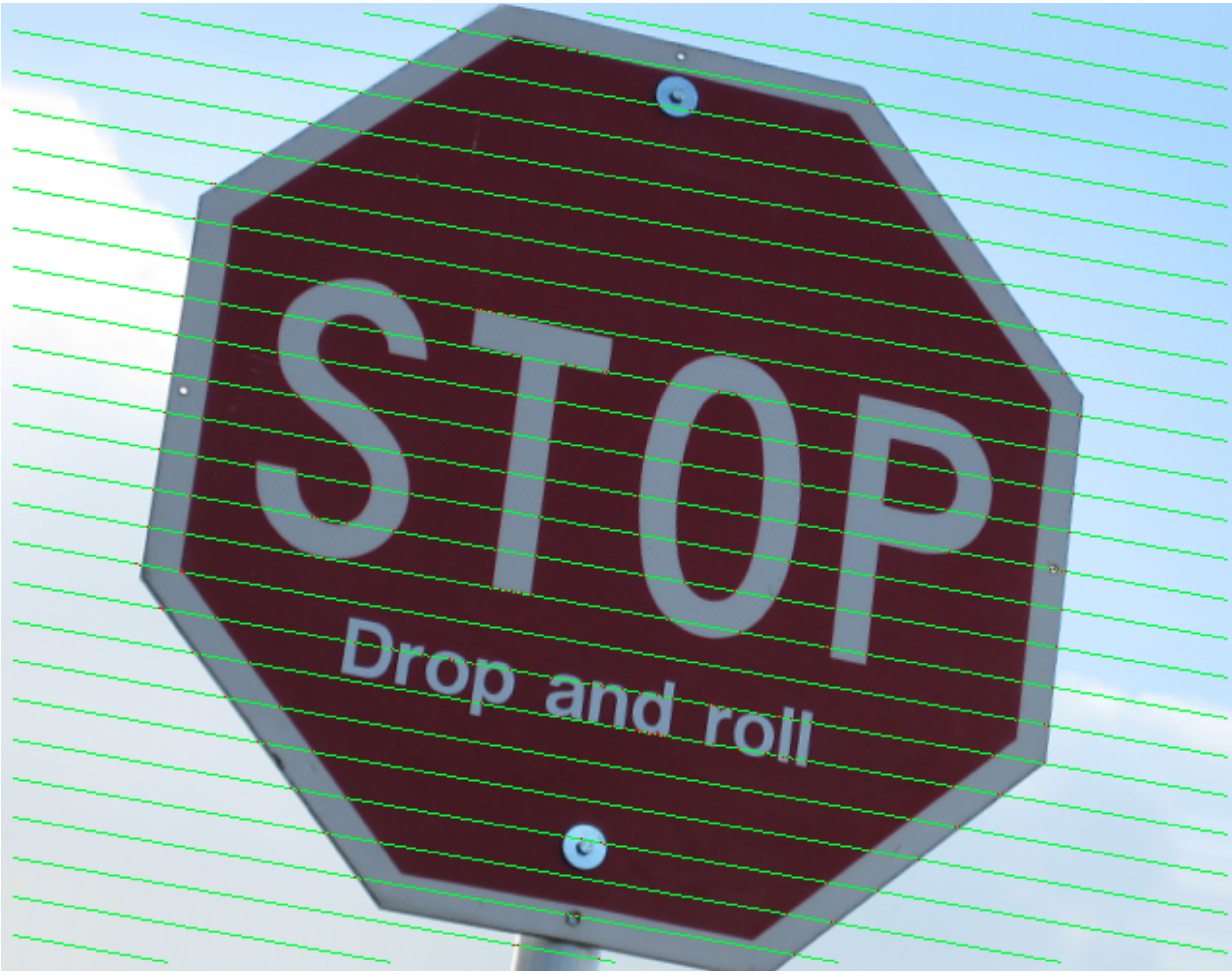


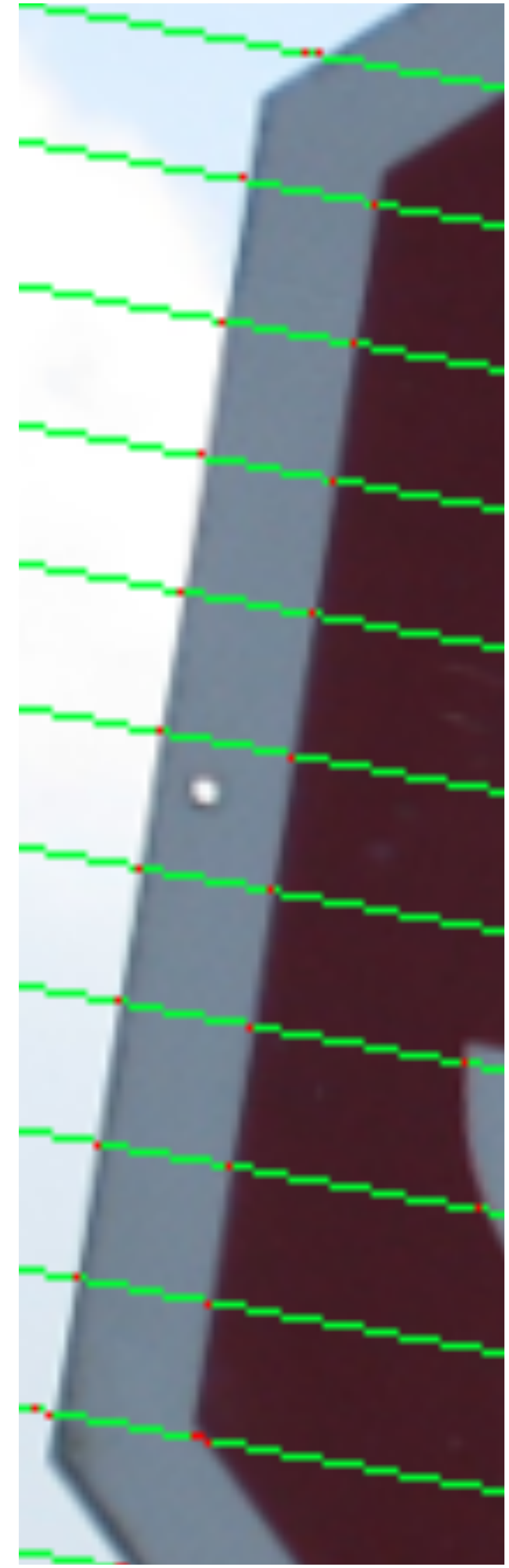


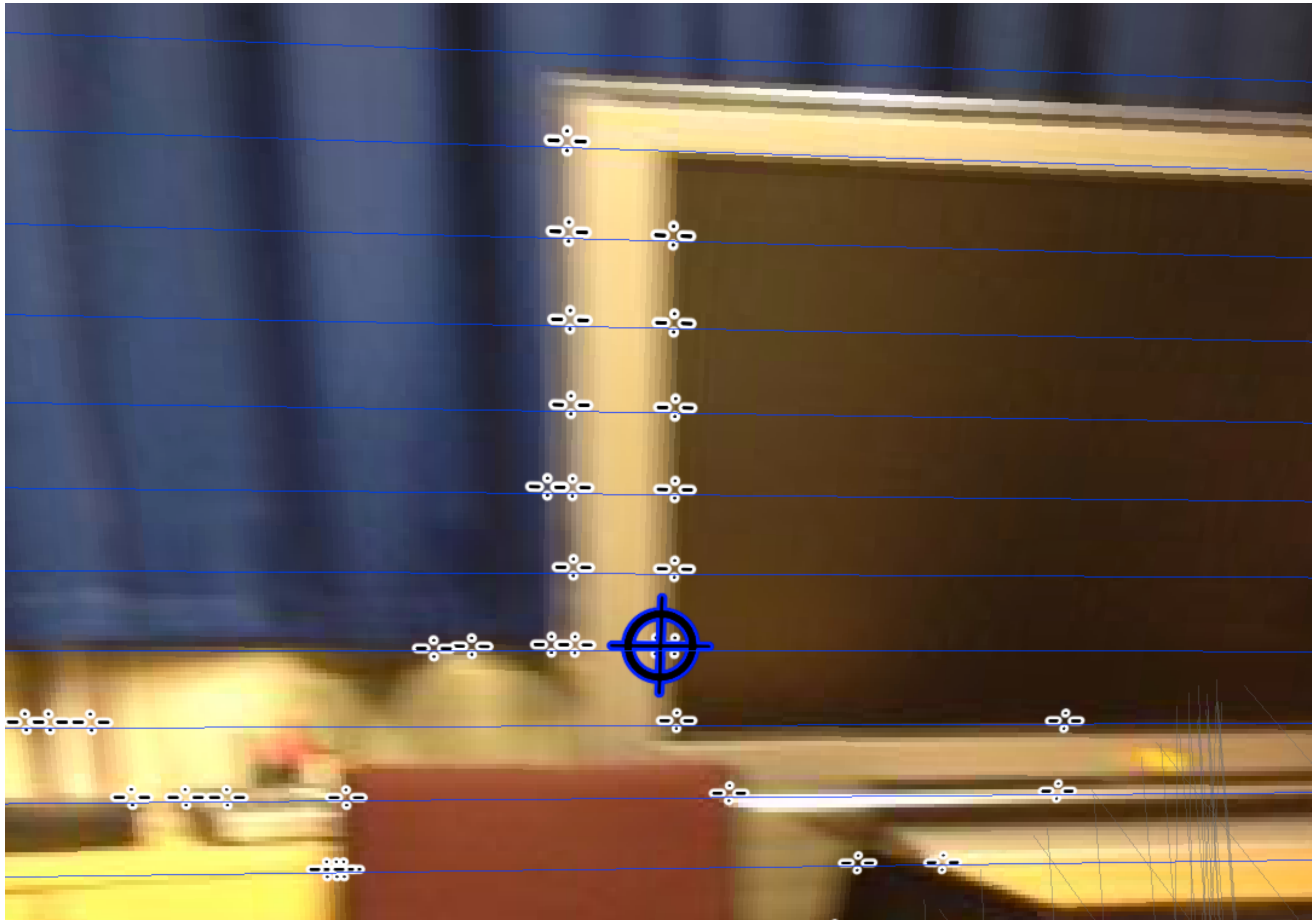


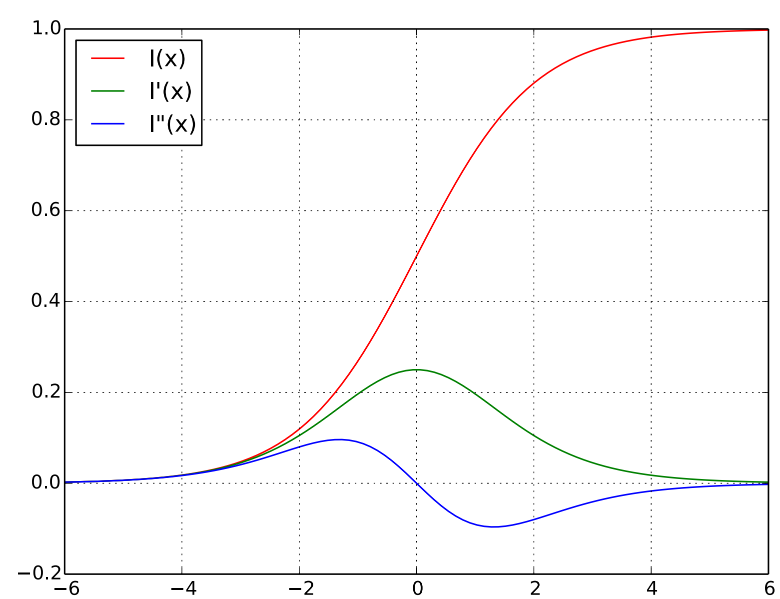
Vertical Edge Alignment
Hybrid Motion Model
Feature Table Alignment
Basic Sensor Motion Model
Integral Sequence Alignment
Sensor Fusion
Feature Table Binning
Vertical Edge Extraction
Camera Data

Vertical Feature Extraction


STOP
Drop and roll







$$\mathtt{LoG}_5(I,x) = \begin{bmatrix} -1 \\ -1 \\ 4 \\ -1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} I(x-2) \\ I(x-1) \\ I(x) \\ I(x+1) \\ I(x+2) \end{bmatrix}$$

Feature Table Binning

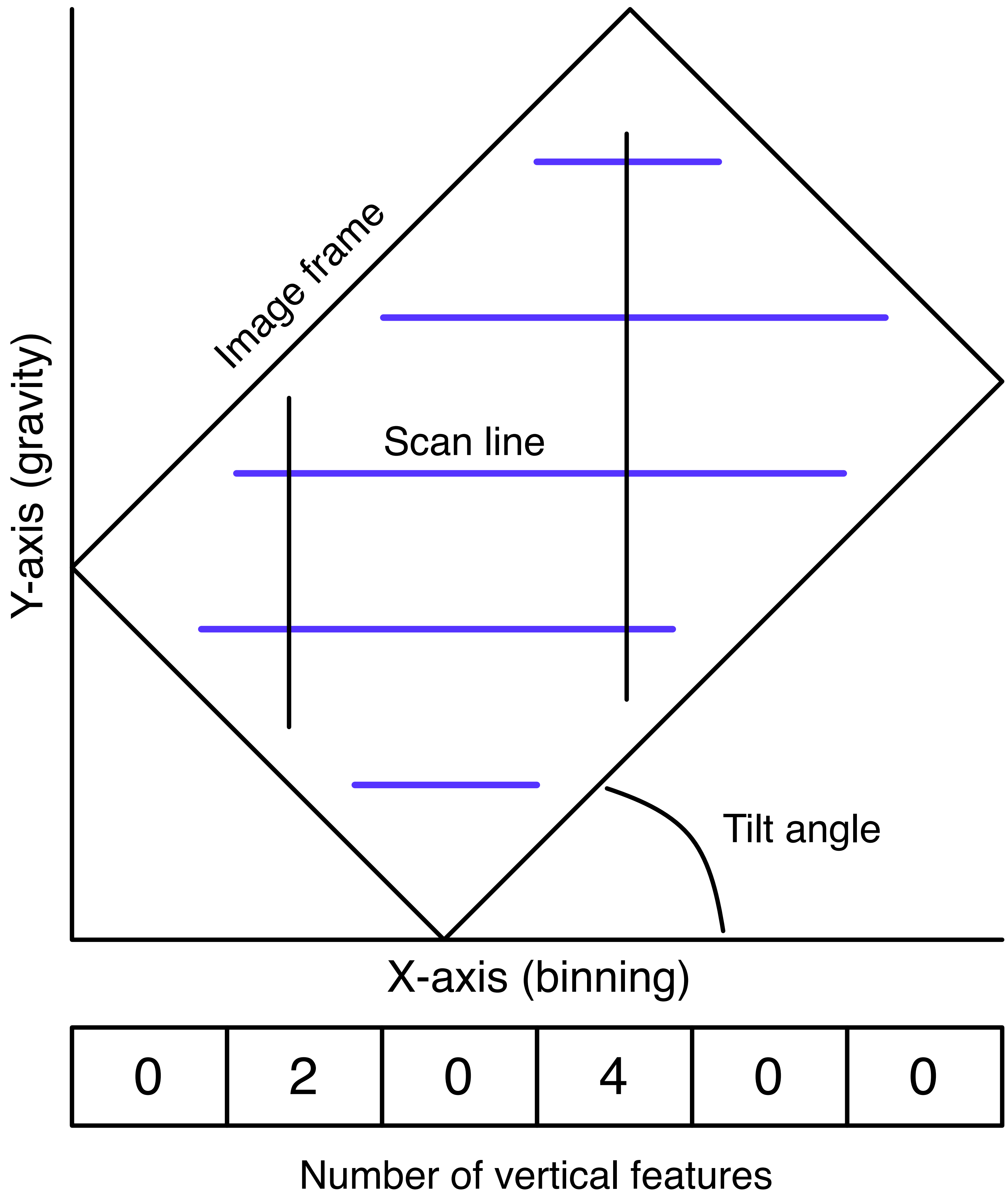
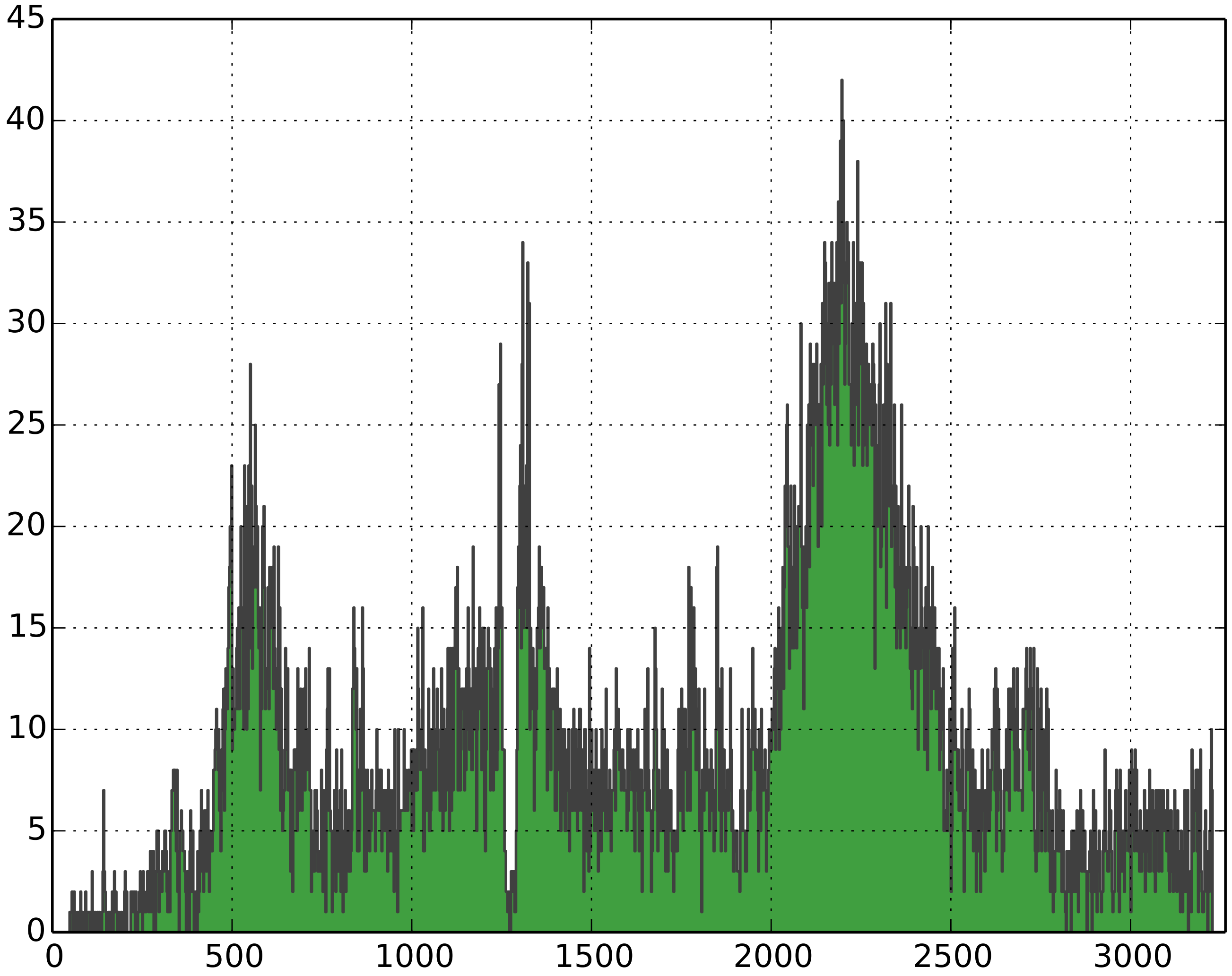


Image frame
Scan line
Tilt angle
Y-axis (gravity)
X-axis (binning)
0
2
0
4
0
0
Number of vertical features


Derenzy Pl


45
40
35
30
25
20
15
10
5
0
0
500
1000
1500
2000
2500
3000

Integral Sequence Alignment

Fast Integral Sequence Alignment

$$u = [3, 7, 8, 7, 6, 0, 0, 7, 5, 3]$$

$$v = [7, 7, 0, 0, 7, 5, 4, 0, 1, 5]$$

Integral Sequences

Sum of squared differences

$$(u * v)(k) = \sum_{i=-\infty}^{\infty} [u(i) - v(i-k)]^2$$

Offset

Offset estimate

$$(u * v)(k, e) = (k - e)^E + \sum_{i=-\infty}^{\infty} [u(i) - v(i-k)]^2$$

Error bias

Estimate Alignment

Alignment Error

$$\texttt{FISA}(u, v, e) = \min_{k=-n/2}^{n/2} (u * v)(k, e)$$

Integral Sequences

Find the minimum

Fast Integral Sequence Alignment

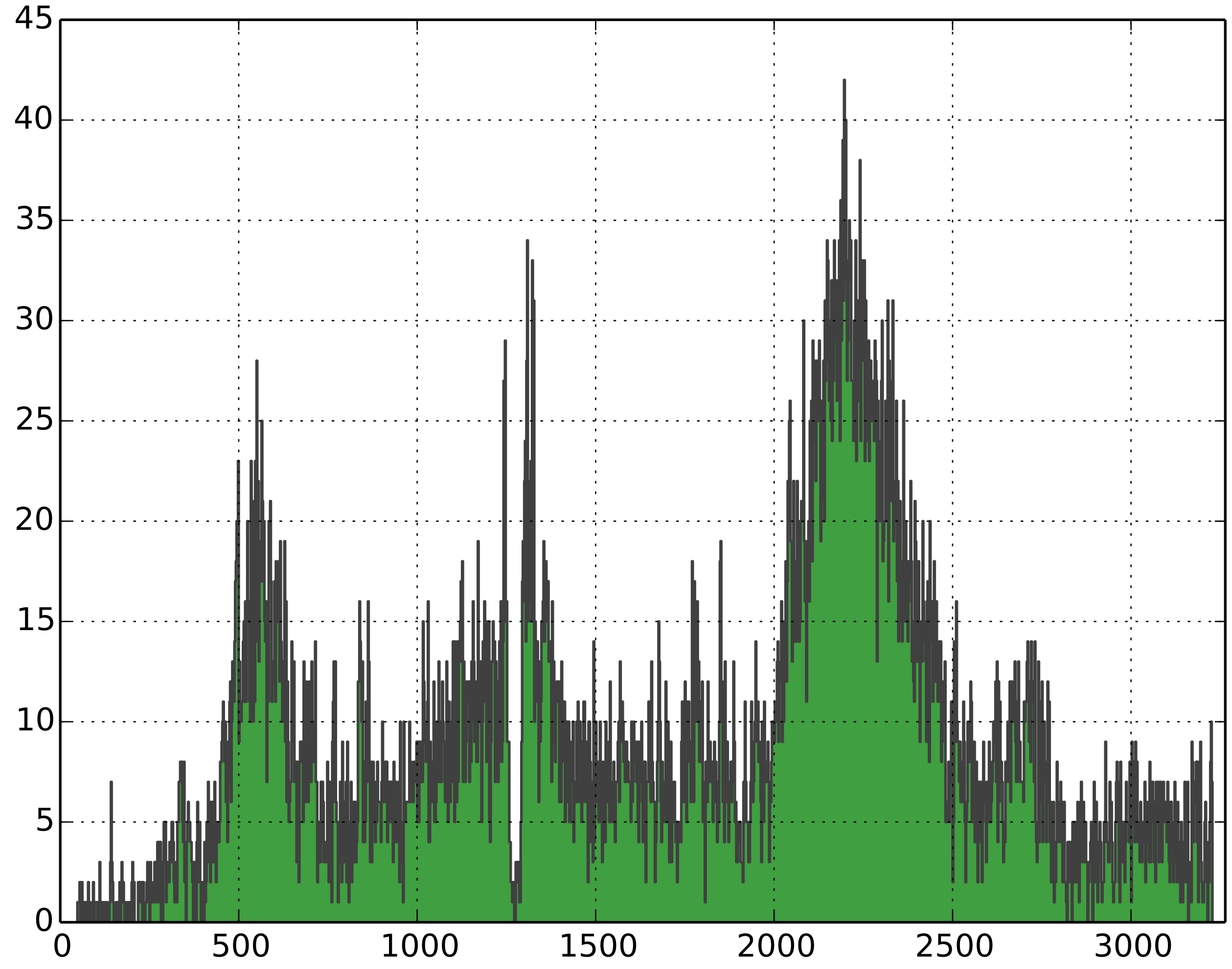
Table 4.1: FISA Performance Results.

	Linear (µs)		Heap (µs)	
n	**Left-Right**	**Outward**	**Left-Right**	**Peaks**
8	0.552	**0.484**	3.628	6.629
16	1.716	**1.357**	10.721	10.489
32	4.914	**2.878**	22.536	20.126
64	16.560	**9.496**	46.687	34.780
128	56.323	**32.793**	94.913	60.527
256	176.742	112.915	201.484	**103.439**
512	733.457	430.844	492.797	**205.429**
1024	2987.51	1713.51	1253.59	**365.713**

Feature Table Alignment

Feature Table Alignment

- Match features in the bins and compute precise alignment of vertical edges.



Evaluation

- Synthetic tests, performance tests.
- Tracking and registration will be compared using pre-recorded data sets.
- User evaluation impact on usability and precision.

Synthetic Tests

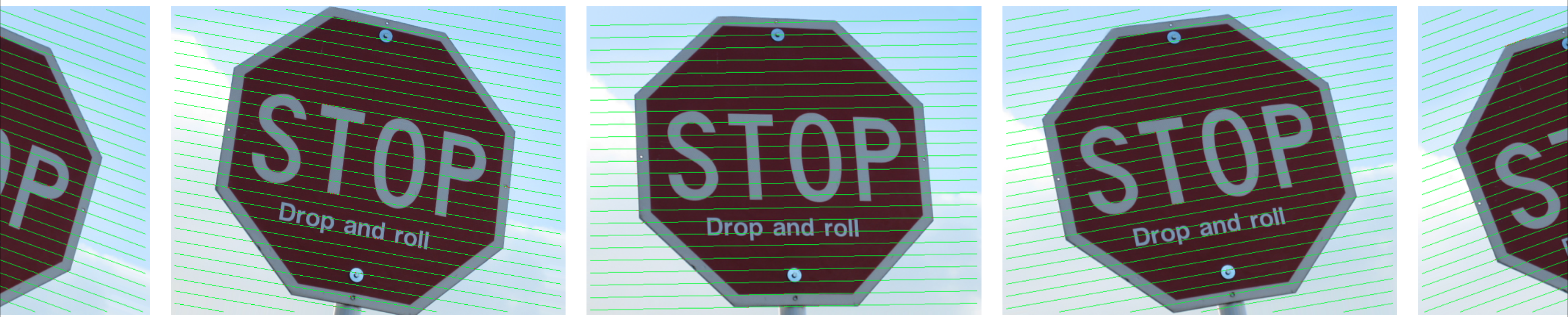


Table 3.2: Rotation Performance and Accuracy.

			Output Error (px)		
dy (px)	**Features**	**Alignment**	**Mean**	**S.D.**	**S.E.**
5	12.13ms	303.9µs	-0.0026	0.019	0.003
10	6.29ms	143.3µs	-0.0045	0.028	0.0044
15	4.24ms	109.9µs	-0.0082	0.04	0.0062
20	3.27ms	119.5µs	-0.023	0.046	0.0071
25	2.57ms	125.8µs	-0.0099	0.066	0.01
30	2.08ms	126.5µs	0.16	0.64	0.099

Performance

Table 6.2: Real World Performance Comparison

		Frame Time				
Resolution	**dy** (px)	**Mean**	**S.D.**	**S.E.**	**Max**	**FPS**
480 × 360	5	14.9ms	2.9ms	0.1ms	24.8ms	67.1
480 × 360	10	5.7ms	3.5ms	0.1ms	15.7ms	177.0
480 × 360	15	2.9ms	2.4ms	0.1ms	11.8ms	344.7
480 × 360	20	1.9ms	2.0ms	0.1ms	9.6ms	515.0
640 × 480	10	16.0ms	3.2ms	0.1ms	24.4ms	62.5
640 × 480	15	11.4ms	4.1ms	0.2ms	21.4ms	87.9
640 × 480	20	8.6ms	4.3ms	0.2ms	17.5ms	115.8
640 × 480	30	5.4ms	3.6ms	0.1ms	13.7ms	183.7
1280 × 720	20	29.1ms	3.6ms	0.1ms	38.2ms	34.3
1280 × 720	30	20.6ms	4.0ms	0.2ms	30.2ms	48.5
1280 × 720	40	16.0ms	3.9ms	0.2ms	26.2ms	62.3
1280 × 720	80	8.8ms	4.0ms	0.2ms	21.7ms	114.1

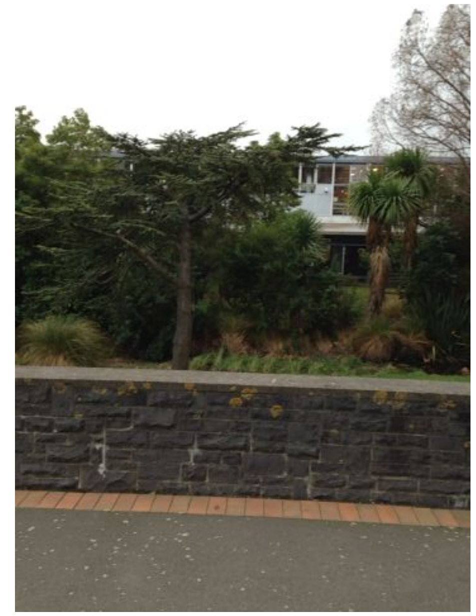
Comparison with ORB

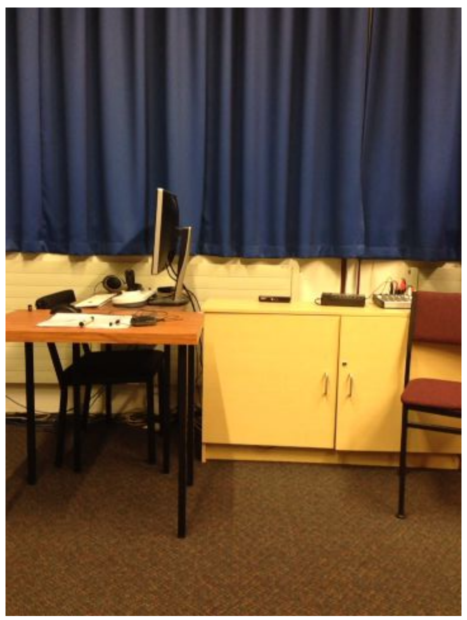
TABLE 3.4: Image Alignment Performance Comparison on iPhone 5

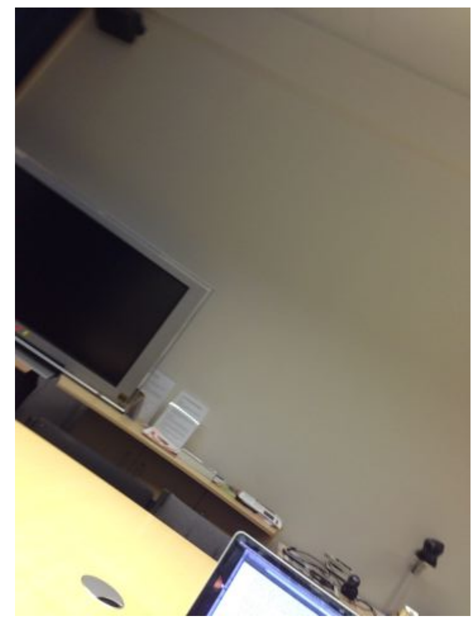
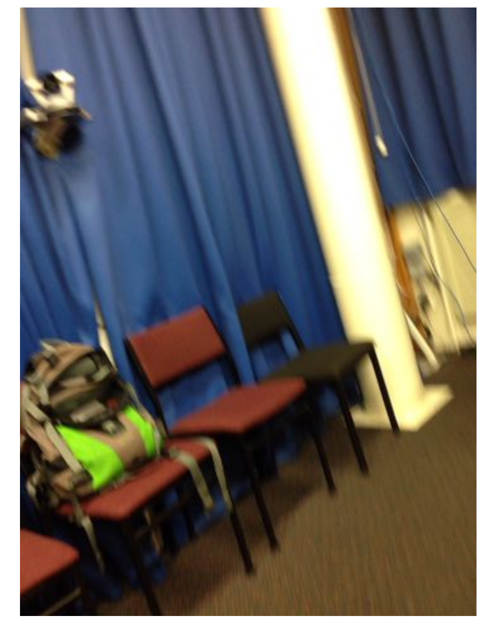
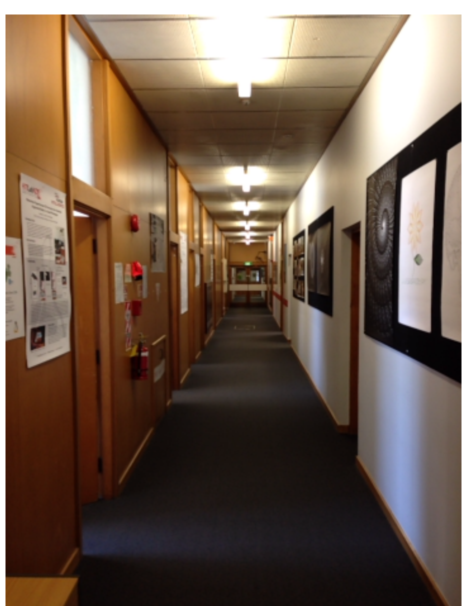
	ORB/LK Optical Flow		Proposed Implementation	
X Offset (px)	**Features**	**Alignment**	**Features**	**Alignment**
-20	112ms	309ms	3.84ms	55.0µs
-10	107ms	318ms	3.62ms	54.2µs
0	112ms	310ms	3.60ms	55.1µs
10	107ms	314ms	3.77ms	54.6µs
20	108ms	322ms	3.60ms	53.9µs
	Error: $\pm 0.00005px$		Error: $\pm 0.05px$	

Tracking Points







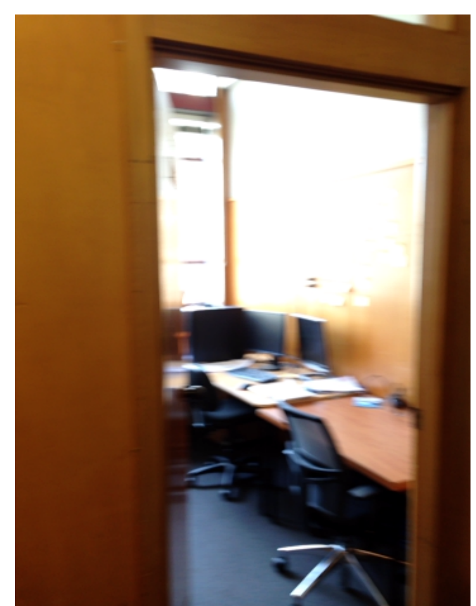
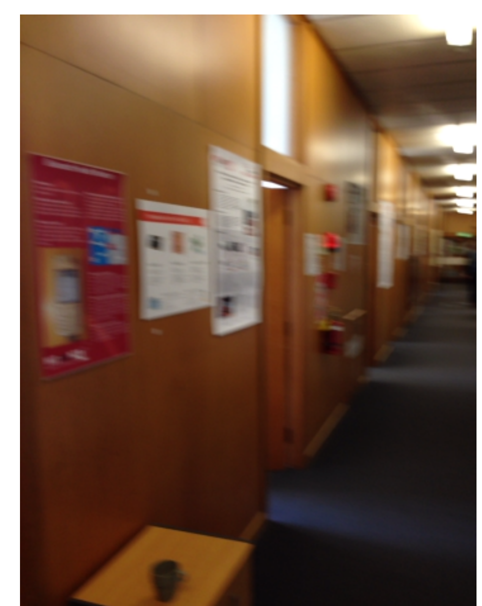
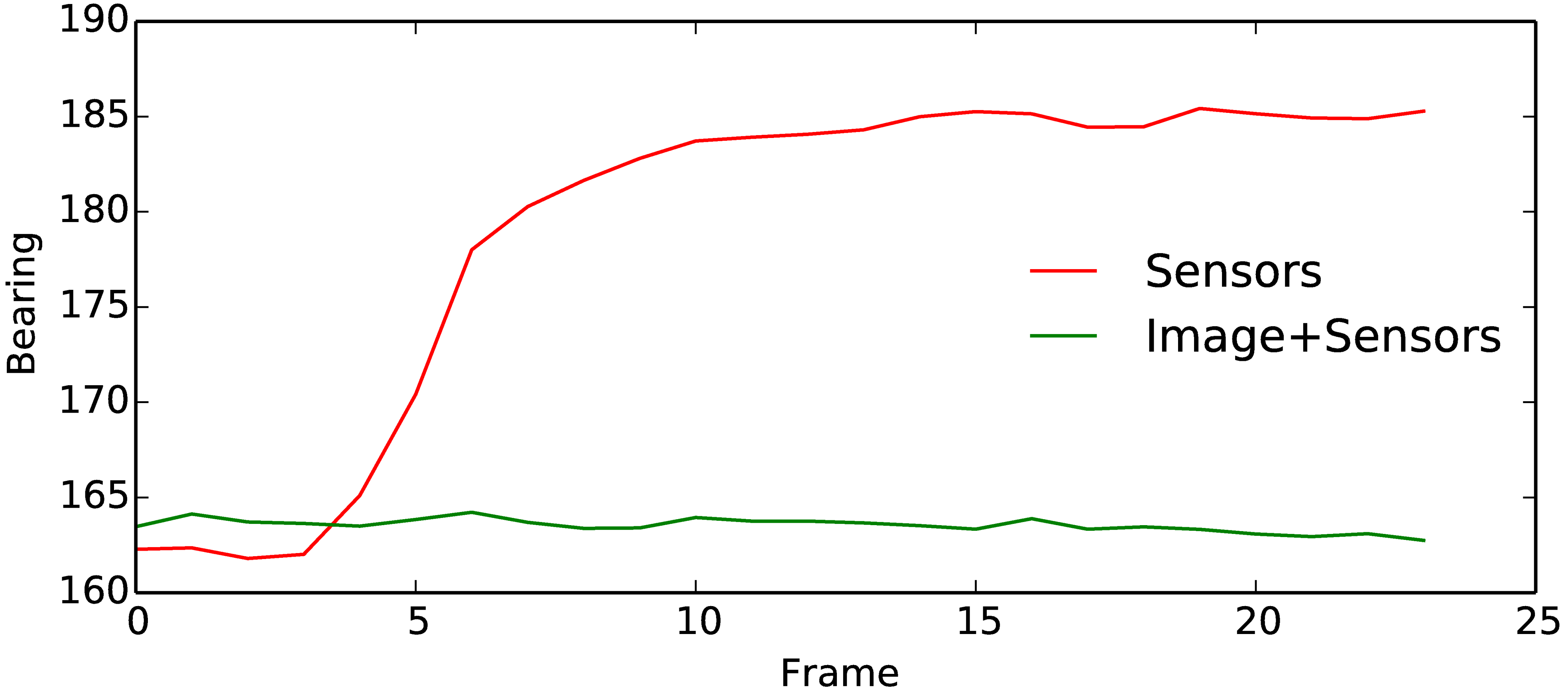
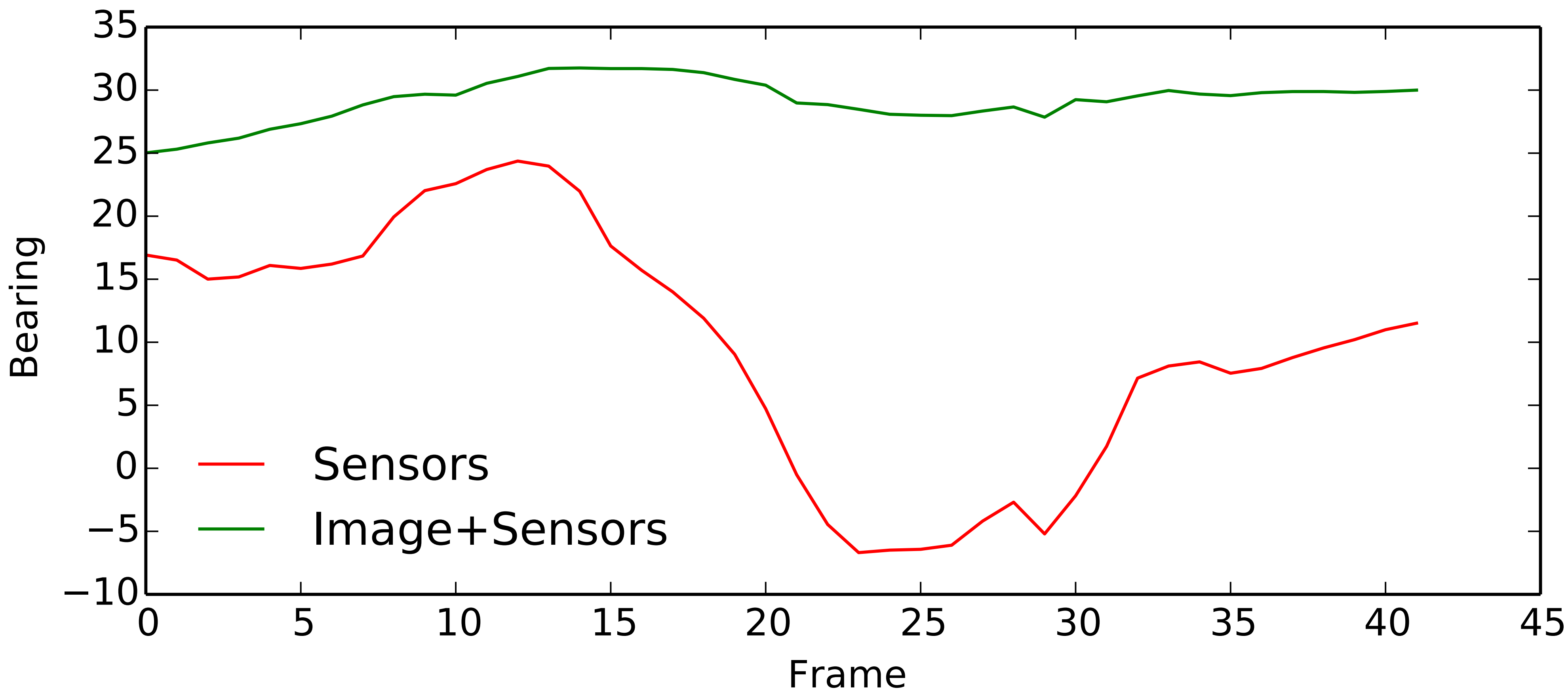
Room 125



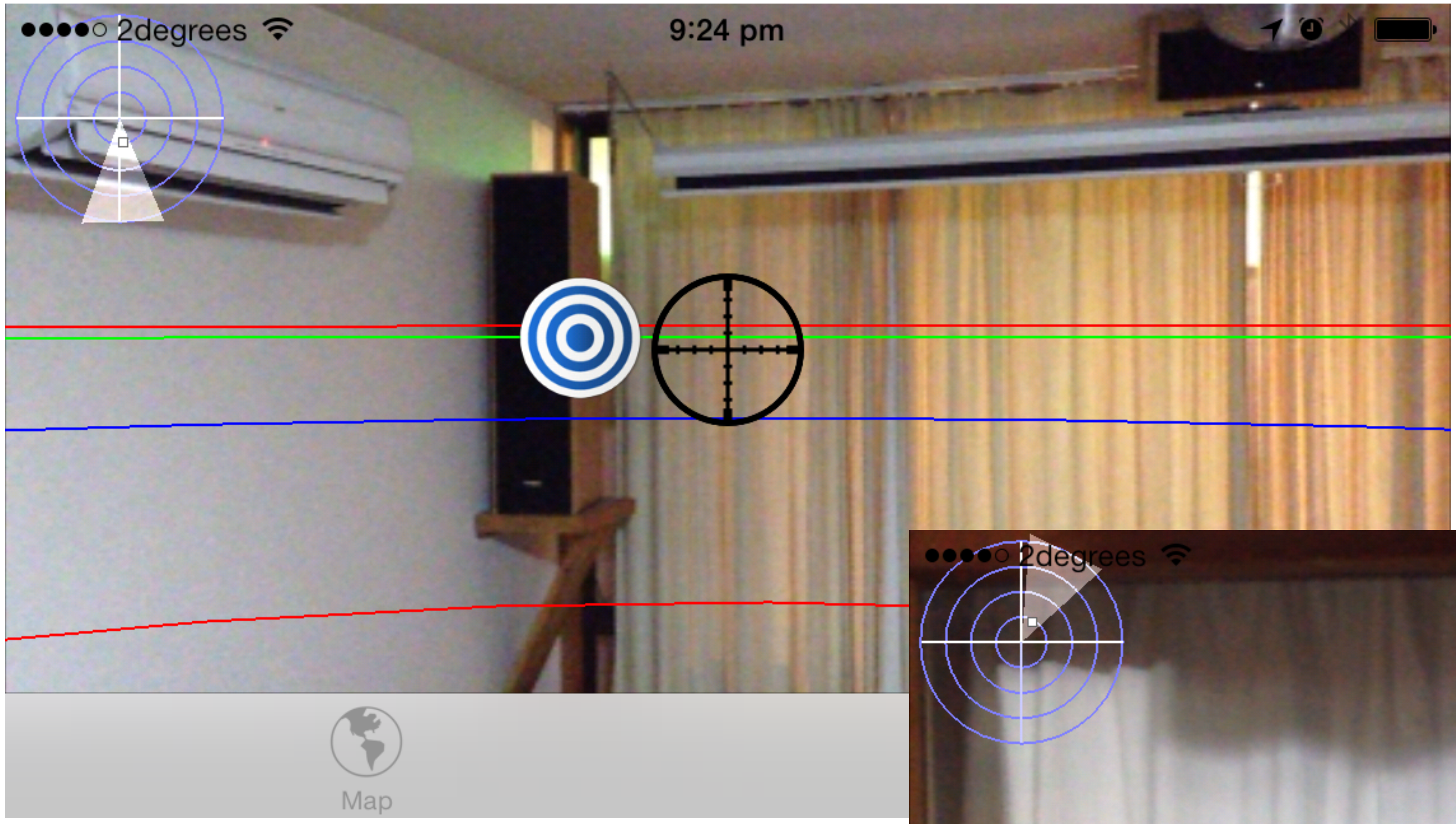
Table 6.1: Relative Bearing Accuracy

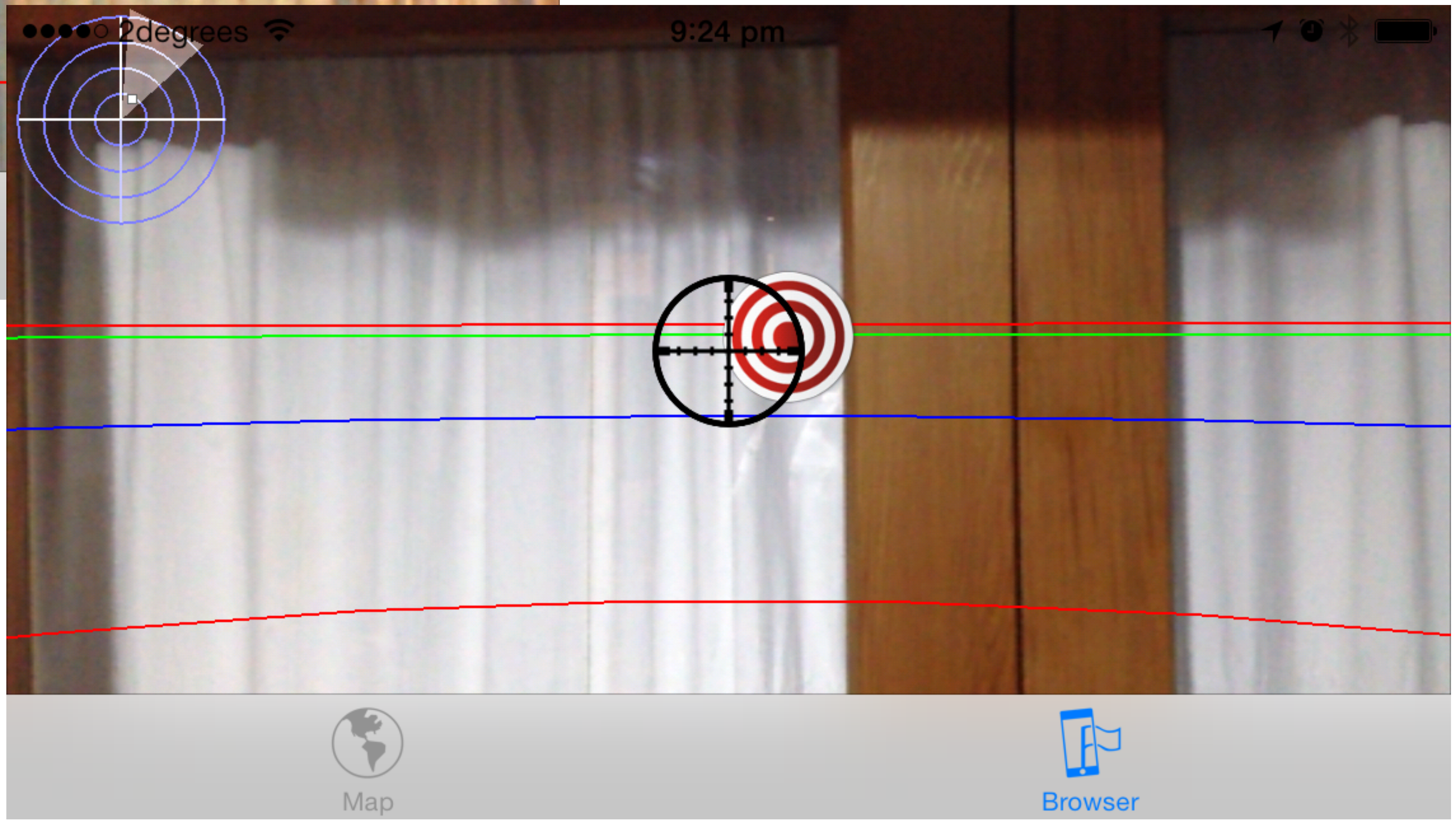
	Sensor Tracking		Hybrid Tracking		
Data Set	**S.D.**	**S.E.**	**S.D.**	**S.E.**	**Improvement**
2013A0	0.37°	0.08°	0.29°	0.06°	1.3×
2013A3	8.88°	1.81°	0.35°	0.07°	25.4×
2013A4	2.82°	0.54°	0.64°	0.12°	4.4×
2013A5	5.93°	1.12°	0.85°	0.16°	7.0×
2013B0	2.65°	0.38°	0.76°	0.11°	3.5×
2013B1	2.24°	0.26°	2.22°	0.25°	Negligible
2013B2	9.52°	1.47°	1.70°	0.26°	5.6×
2013C0	2.93°	0.49°	0.69°	0.12°	4.24×
2013C1	2.55°	0.43°	0.87°	0.15°	2.9×
2013C2	8.34°	1.74°	7.98°	1.66°	Negligible
2013D0	1.17°	0.28°	0.88°	0.21°	1.3×
2013D1	1.83°	0.47°	0.48°	0.12°	3.8×
2013D2	1.29°	0.30°	0.18°	0.04°	7.2×


190
185
180
175
170
165
160
Bearing
0
5
10
15
20
25
Frame
Sensors
Image+Sensors


35
30
25
20
15
10
5
0
−5
−10
Bearing
0
5
10
15
20
25
30
35
40
45
Frame
Sensors
Image+Sensors

User Study





User Study

TABLE 6.3: User Study Performance Comparison

User ID	Tracking Method	Mean	S.D.	S.E.
All	Sensor Tracking	16.53s	7.02s	2.34s
All	Hybrid Tracking	7.09s	3.31s	1.10s

The results in Table 6.3 clearly show the benefit of our proposed hybrid tracking algorithm. A paired t-test was used to compare the different algorithms. There was a significant difference in the average task completion time for sensor tracking ($M = 16.53s$, $SD = 7.02s$) and hybrid tracking ($M = 7.09s$, $SD = 3.31s$) conditions: $t(8) = 5.0089$, $p < 0.0010$.

Conclusion

- Algorithm works well, helps to improve user performance.
- All data sets and source code published on GitHub.

Questions?