Technical Report
July 20, 2004

Martin Urban, murban@ncsa.uiuc.edu
Peter Bajcsy, pbajcsy@ncsa.uiuc.edu
Rob Kooper, kooper@ncsa.uiuc.edu

Automated Learning Group
National Center for Supercomputing Applications
605 East Springfield Avenue, Champaign, IL 61820

Recognition of Arm Gestures Using Multiple Orientation Sensors

ABSTRACT

We present a system for gesture recognition using multiple orientation sensors. We focus specifically on the problem of controlling Unmanned Aerial Vehicles (UAVs) in the presence of manned aircrafts on an aircraft carrier deck. Our goal was to design a UAV control with the same gesture signals as used by current flight directors for controlling manned vehicles. We have explored multiple approaches to arm gesture recognition, and investigated real-time and system design issues for a particular choice of active sensors. We describe several theoretical and experimental issues related to a design of a real-time gesture recognition system using the IS-300 Pro Precision Motion Tracker by InterSense. Our work consists of (1) analyzing several gesture recognition approaches leading to a selection of an active sensor, (2) scrutinizing sensor data acquisition parameters and reported arm orientation measurements, (3) choosing the most optimal attachment and placement of sensors, (4) measuring repeatability of our experiments using Dynamic Time Warping (DTW) metric, and (5) designing template-based gesture classification algorithms and robot control mechanisms, where the robot represents an UAV surrogate in a laboratory environment.

1 INTRODUCTION

With the current advancements of autonomous unmanned vehicles, there is a need to support a control of unmanned and manned vehicles without interfering with the current control mechanisms of manned vehicles. For instance, the current control mechanism for manned aircrafts is based on people, called flight directors or yellow shirts, performing gestures according to a pre-defined lexicon of gestures and pilots following the gesture corresponding commands. In order to avoid changes of standard control practices and accommodate newly developed unmanned aircrafts, a problem of unmanned vehicle control using standard control procedures arises. This problem motivated our work and development.

In this technical report, we focus specifically on the problem of controlling Unmanned Aerial Vehicles (UAVs) in the presence of manned aircrafts on an aircraft carrier deck. Our goal is to design a UAV control with the same gesture signals as used by current flight directors for controlling manned vehicles. Given the fact that such a system has to operate 24 hours a day in a noisy and harsh environment, for example, on a Navy carrier deck, our approach to this problem is based on arm gesture recognition. Speech recognition systems were not recommended due to a very noisy background environment. Thus, our objective is to explore multiple approaches to arm gesture recognition, and investigate real-time and system design issues for a particular choice of active sensors.

Our proposed gesture recognition system is based on IS-300Pro Precision Motion Tracker by InterSense [1], and an overview diagram in Figure 1 describes the entire system. An operator (a yellow shirt) performs a gesture, during which the tracker sensors transmit acquired data to the IS-300Pro base unit and then to a PC. Sensor outputs are analyzed and classified into corresponding commands (gesture name). Gesture commands are converted into a set of robot instructions and sent to a robot. The robot surrogate, Pioneer II Robot [4] representing a real UAV, executes robot instructions in our laboratory environment.

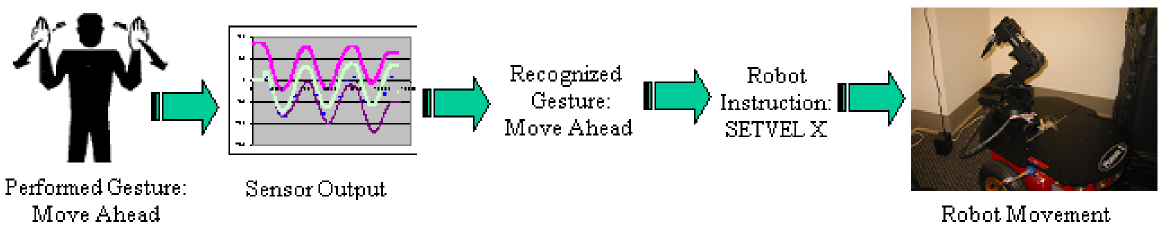

Figure 1: Flow diagram of a developed system for robot control using hand gestures.

In this technical report, we describe several theoretical and experimental issues related to a design of a gesture recognition system using the IS-300 Pro Precision Motion Tracker by InterSense. Our work consists of (1) analyzing several gesture recognition approaches in Section 2 leading to a selection of an active sensor, (2) scrutinizing sensor data acquisition parameters and reported arm orientation measurements in Section 3, (3) choosing the most optimal attachment and placement of sensors in Section 4, (4) measuring repeatability of our experiments using Dynamic Time Warping (DTW) metric in Section 5, and (5) designing template-based gesture classification algorithm and robot control mechanism in Section 6. Our work is summarized in Section 7 together with a list of challenges and an outline of future directions.

2 ARM GESTURE RECOGNITION APPROACHES

2.1 Overview of Gesture Recognition Approaches

Arm gesture recognition can be approached with using active or passive sensors, or a combination of both sensor types. An example solution using passive sensors would be a vision-based system. Single or multiple cameras acquire video stream that is processed and gestures are mapped into temporal signatures of changes in video frames [7]. This solution faces several challenges in such a harsh environment as the aircraft carrier deck and has to overcome changes in a flight director orientation, outdoor illumination (day and night), and possible occlusions of flight directors or recognizing the active (UAV specific) flight director among many directors on the deck. On the other side, this approach does not require any changes in the current control practices, or any changes in the flight director's equipment. One should be aware during a system design that any additional weight to the equipment worn by flight directors would increase fatigue of flight directors and hence additional weight is not desirable. This consideration imposes real-world constraints on systems with active sensors since they have to be worn.

Examples of solutions using active sensors would include gloves with bent sensors [8] or miniaturized accelerometers [9], [10]. For example, the cyberglove in [8] uses 18 distributed bent sensors embedded in a glove to capture finger articulation. Similarly, the advancement in Micro-Electro Mechanical Systems (MEMS) led to building a glove prototype at UC Berkeley [10]. Most of these solutions have been developed for indoor virtual reality (VR) applications and are not easily extensible to outdoor applications with highly uncontrolled environment.

The use of passive and active sensors together was reported in the past [12] with the goal of combining advantages of both sensor types. For instance, placing fluorescent markers on tracked objects and illuminating them with known light sources is an example of a vision-based hybrid system that does not constraint moving subjects with heavy or bulky sensors and improves robustness of a standard vision based system in terms of motion detection and tracking.

We should also mention that the specific problem introduced in this section could have been approached by broadcasting video of synthesized gestures to the cockpit of manned aircrafts. A computer program driven by a flight director would create video of synthesized gestures. Pilots of manned aircrafts would recognize synthesized gestures the same way as they did in the past, and all unmanned vehicles would receive directly the de-coded (interpreted) commands. We developed video examples of synthesized gestures for test purposes. However, this solution, although very robust from gesture recognition viewpoint, is not acceptable by the end application because the person giving commands has to be present on the aircraft deck during the entire time of any vehicle navigation.

2.2 Proposed Approach and Sensing

While there are many approaches to gesture recognition, we chose to research and develop a solution with active sensors because of the end application requirements on performance robustness and reliability. By considering the importance of (a) system reliability in a highly varying environment (e.g., geometry, illumination, line of sight, temperature, and operator's fatigue) and (b) safety of navigation operations, the active sensing approach outperforms solutions based on passive sensing approach. As one part of our research, we surveyed and evaluated active sensors based on their (a) size, (b) weight, (c) cost, and (d) commercial availability. We considered three different solutions, such as, (1) virtual reality (VR) motion trackers [1], (2) global positioning systems (GPS) [11] and Micro-Electro-Mechanical Systems (MEMS) with tiny operating system (tinyOS) [9], [10]. The choice of the IS-300 Pro Precision Motion Tracker by InterSense, MA, for this work was primarily driven by its best sensing performance specifications and its commercial availability. For example, a spatial accuracy of GPS (around 3 m for the GPS with the Wide Area Augmentation System) and an extra development effort (building a glove with MEMS sensors) were considered as major drawbacks of the other two solutions. The cost of IS-300 Pro Precision Motion Tracker ($4,375 for the base unit plus $1,437 for each additional sensor), and the size and weight parameters (each sensor cube weighs 2.1 oz and measures 1.06"x1.34"x1.2") were at the borderline of being acceptable at the time of purchase. Nevertheless, the vendor has miniaturized the sensors and decreased their weight significantly since the time of purchase.

Given the choice of an active sensor, our approach to the problem of gesture recognition is based on (1) translating arm motion into a temporal sequence of orientation of angles, (2) describing a sequence of orientation angles with its characteristics, (3) building models of gestures in a lexicon using sequence characteristics of orientation angles, and (4) classifying sequences of orientation angles into gesture classes according to the developed gesture models in real time. The basic premise of our approach is an existence of a unique mapping between human gesture represented by arm motion and a temporal sequence of upper arm and forearm orientation angles. The existence of such a mapping is frequently used in the computer graphics community where arms are modeled as connected cylinders or ellipsoids, changing their orientation in a world coordinate system. Our overall approach is fundamentally robust to most environmental conditions on an aircraft carrier that makes the vision-based solution difficult. These conditions include variable lighting, occlusion in the line of sight, background clutter, fog, and hot engine exhaust. Distance from the director to the aircraft is not a factor either as long as the communication between the yellow shirt and a specific aircraft (manned or unmanned) can be established. Communication is clearly a problem, but our system requires very low bandwidth (only communicating high level commands at a frequency less than a few hertz).

3 SENSOR DATA ACQUISITION PARAMETERS

3.1 IS-300 Pro Parameters

The IS300 Pro Precision Motion Tracker is shown in Figure 2 and it was developed for head movement tracking in virtual reality systems. The base unit of IS300 can track up to four sensor inertia cubes. We have scrutinized (1) acquisition rate (maximum tracking rate is 1200° per second, update rate is up to 500Hz), (2) measurement accuracy (RMS angular resolution is 0.02°, RMS angular accuracy is 1.0°, and RMS dynamic accuracy is 3.0°), (3) temperature range (0°C to 50°C), and (4) ruggedness (shock sensitive), in addition to size, weight and cost evaluation criteria mentioned before. All parameters were adequate for our application except from the sensor ruggedness. However, the ruggedness was not our major concern at this time. By using the IS300 Pro Precision Motion Tracker, we have also avoided the issues related to multiple sensor synchronization because the IS300-Pro base unit handles four sensors simultaneously.

a b c

Figure 2: a) IS300 Pro base device, b) single Inertia Cube, and c) base device with 4 cubes on arm bands

3.2 Selection of Reported Orientation Measurements

The last parameter scrutinized before running any experiments was the choice of reported orientation measurements, such as, (1) a 3x3 rotation matrix, (2) three Euler angles (yaw, pitch and roll), and (3) four-element quaternion. First, we decided not to use the rotation matrix because (a) it can be constructed from the Euler angles or quaternions, and (b) it requires transmitting larger number of bytes (matrix entries) than the other two representations and hence adds unnecessary communication and computational cost. Second, we evaluated the pros and cons of Euler angles and quaternions. The major advantage of Euler angle representation over quaternion representation is its easy comprehension for humans. The disadvantage of Euler angle representation is its singularity point when yaw is near 90 degrees. Another advantage of quaternion representation is the mathematical simplicity when performing rotation of one quaternion by another. Third, we have investigated the transformation uniqueness between rotation matrices derived from Euler angles or quaternions, Euler angles, and four-element quaternions. This seemingly trivial issue is complicated by the fact that the IS300 Pro reports angular values in left-hand coordinate system while all computer graphics and java3D libraries use right-hand coordinate system (see Figure 3). We have investigated conversions of (a) Euler angles to rotation matrix to Euler angles, (b) quaternions to rotation matrix to Euler angles, and (c) rotation matrix to Euler angles and quaternions. We have developed two different methods, GEMS [6] and TRTA [5], for this purpose. To the best of our obtained knowledge, we could not find a method that would recreate identical angles in the above (a), (b) and (c) transformations. An example of discrepant results is shown in Figure 4.

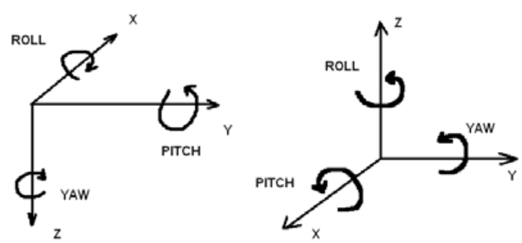

Figure 3: Left-hand (left) and right-hand (right) coordinate system.

Based on our scrutiny, we decided to directly acquire Euler angles, and avoid any angular and coordinate system transformations by modeling gestures directly with a combination of absolute angles (roll and pitch) and relative angles (yaw), work in the left-hand coordinate system. The singularity point in Euler angle representation was compensated by an appropriate design of our classification algorithm.

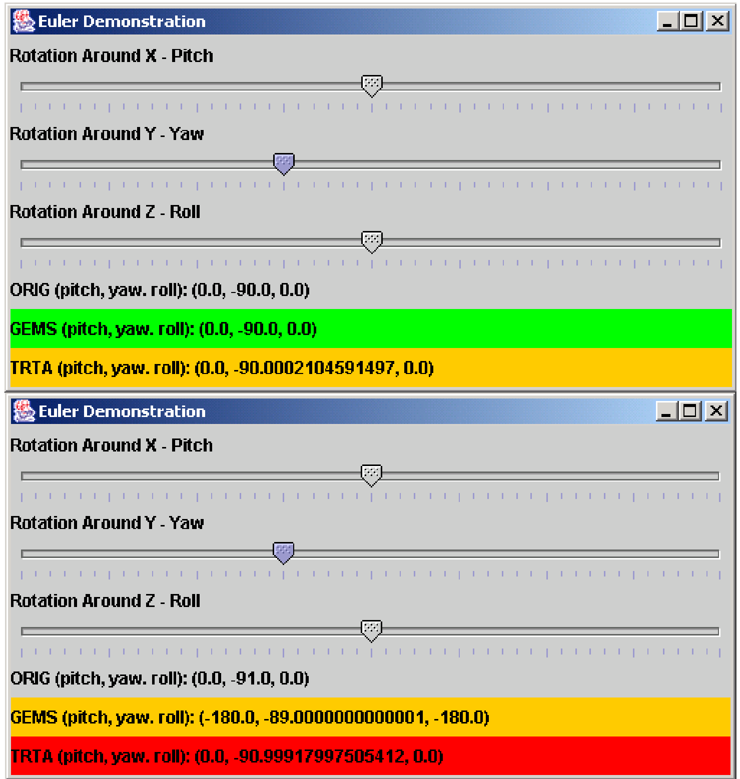

Figure 4: The problem of mapping between Euler angles, quaternions and rotation matrices.

4 ATTACHMENT AND PLACEMENT OF SENSORS

Sensor placement is crucial for obtaining repeatable and gesture unique measurements. Repeatability of measurements was improved by a thorough design of tight attachment mechanisms between sensors and a sensor base, and a sensor base and human arm. Sensors were attached firmly to a plastic flat board by two screws and the board edges had openings for Velcro armband strips, see Figure 2c. To assure minimum movement of the sensors it is recommended to place armbands tightly around skin rather than around any sleeves or other clothing.

The issue of acquiring gesture unique measurements was addressed by investigating (1) different number of sensors per arm (two or three sensors per arm), (2) variable sensor locations and (3) several sensor orientations on a forearm or upper arm. The possible sensor placements are illustrated in Figure 5.

We represent the human arm by three connected almost rigid segments corresponding to the upper arm, lower arm, and wrist. A sensor mounted on each of these segments captures the full range of motion of that segment and three sensors together capture the full range of motion of the arm. However, the fact that the IS300 Pro unit can handle at most four sensors limits us to only two sensors per arm. A closer analysis of the gesture lexicon suggests that the wrist segment gives the least amount of information about the whole arm orientation. Furthermore, through numerous experiments of collecting gesture data with three sensors on one arm, as in Fig. 3c, we determined that the user moves the wrist joint too rapidly, and also makes slight involuntary movements that are not part of the target gesture. From this we conclude that two sensors per arm, mounted on the upper and lower arm are adequate for majority of the gestures in the lexicon. The placement of the sensors along their respective arm segments is also crucial. Placing the lower arm sensor near the wrist end allows for greater range of angles to be captured than by placing the sensor near the elbow, and this placement also gives a perfect representation for the facing direction of the palm. A physical support for greater rotational range at the wrist is due to the lower arm being composed of two parallel bones. To capture the biggest rotational range in the upper arm the sensor is placed near the elbow; because the flesh and muscles near the shoulder do not move with the upper arm bone.

The starting rotational orientations were also considered. Either the sensors point sideways away from the body as in Figure 6a, or they point forward, Figure 6b. The sideway pointing location is preferred, because the upward pointing sensor location hindered the movement of the arm as the lower arm would hit the upper arm's sensor when the wearer bent the elbow more than 90 degrees. Another reason for choosing the sideways placement is that the armbands rotate slightly towards the sideway position with each movement of the arm, thus eventually fully rotating from the upward placement orientation to the sideways placement orientation.

To prevent the weight of the wires from pulling on the sensors and making them move, it is helpful if the wearer holds the wires in his hands allowing just enough wire to be between the hand and sensor for free movement in all directions.

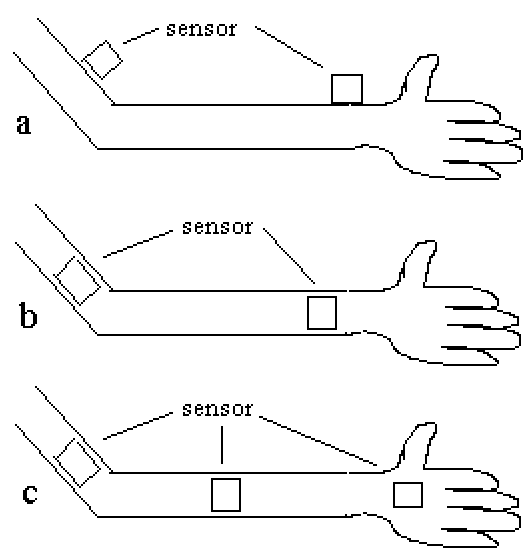

Figure 5: Possible sensor placements on right arm (mirror image placement on left arm). a) 2 sensors on top of arm, b) 2 sensors on side of arm, c) 3 sensors on side of arm.

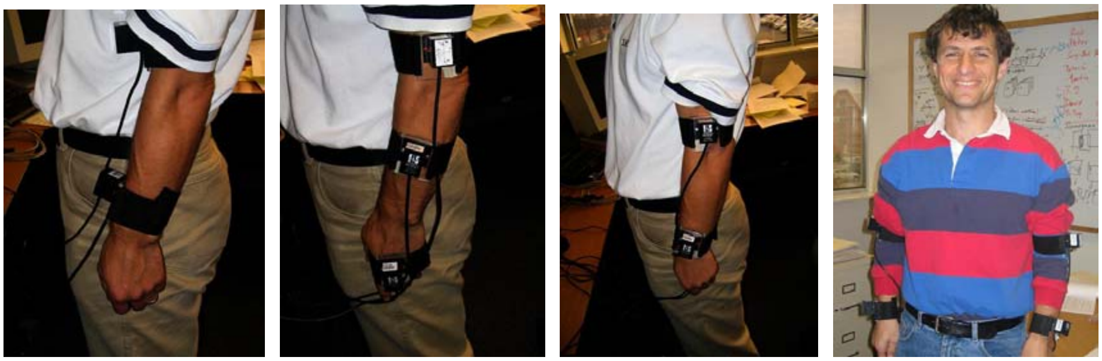

Figure 6: Tested orientations: 2 sensors sideways, 2 sensors up, and 3 sensors sideways. Last picture shows chosen sensor placement orientation (2 sensors sideways on both arms).

5 REPEATABILITY ASSESSMENT

In order to show repeatability among gestures, we need to understand how the Euler angles behave under different sensor motions. For this we analyzed the outputs of individual sensors against each other, as well as the combined output of the sensors in a gesture versus the output of the sensors in a repeat of the same gesture, or another gesture. The motivation for showing good repeatability among gestures is gesture recognition.

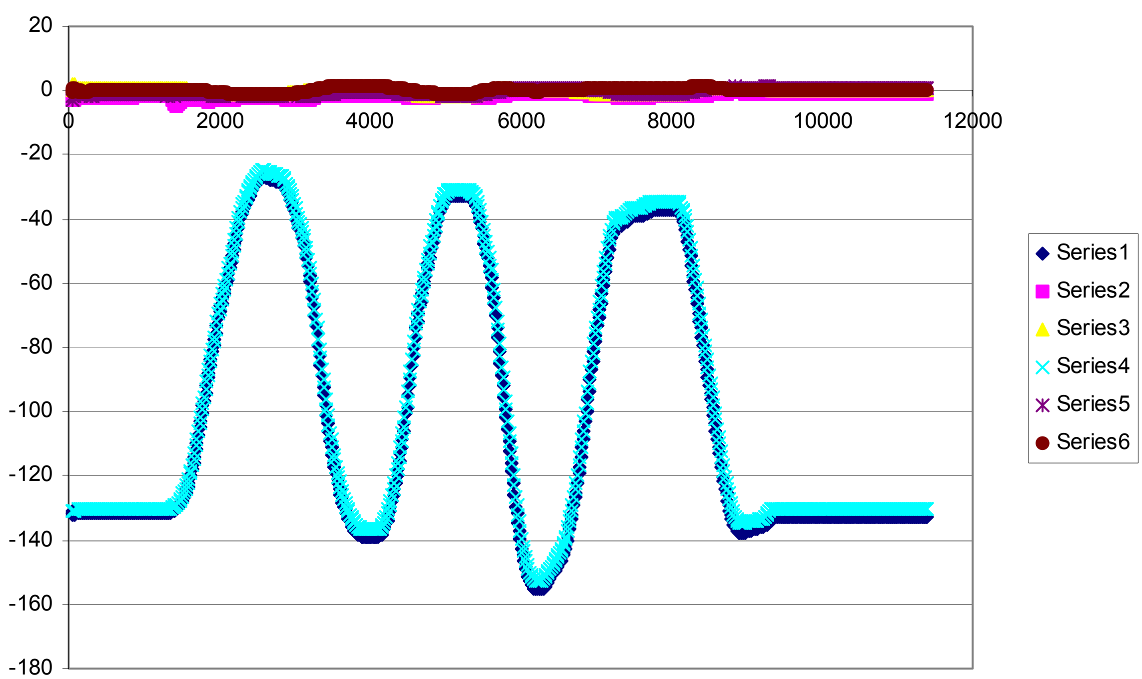

Figure 7: Book cover experiment with two sensors going through the same motion (3x repeated 90° yaw rotation). The light blue and dark blue sinusoids are the yaws for the two sensors. The four nearly constant lines at 0 are the pitches and rolls.

5.1 Sensor Repeatability Analysis

First we tried to see how reliable each of yaw, pitch and roll are, and also how close are the results from different sensors experiencing the same motion. To get the sensors to experience the same motion we attached two sensors to the cover of a hardback book, and positioned the book on a flat horizontal surface such that fully opening and closing the cover repeatedly, or rotating the book would only change one of the three Euler angles, while keeping the other two constant. Visualizing the results of each experiment showed that both sensors reported almost identical angle measurements.

To assess individual sensor repeatability in real arm motions we experimented with mounting all the sensors on the lower arm and performing various movements to see how well the sensors' outputs compare to each other. One sensor placement was with the sensors on side of the arm and the other with the sensors on top of the arm as in Figure 9. The performed motions were raising and lowering the arm forward or sideways. The graphs of a repeated motion of raising the arm sideways from vertical to horizontal position are shown in Figure 7. From this experiment and the book cover experiment we conclude that all the sensors indeed report almost identical readings when experiencing equal motion and therefore they are reliable.

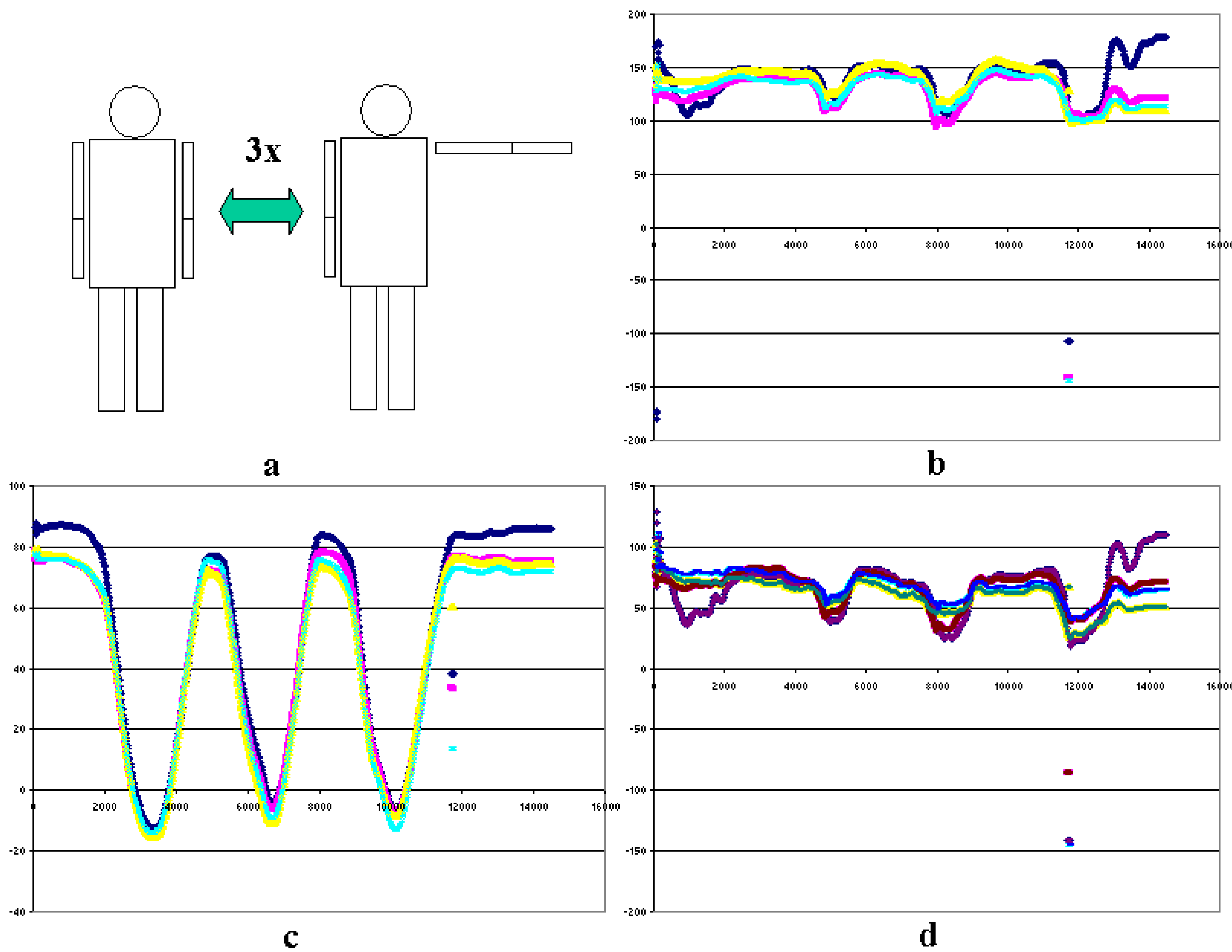

Figure 8: Comparison of Euler angles acquired while moving arm sideways up three times (see (a)) with four sensors attached as shown in Figure 9b. Graphs showing measured b) yaw, c) pitch, and d) roll angles.

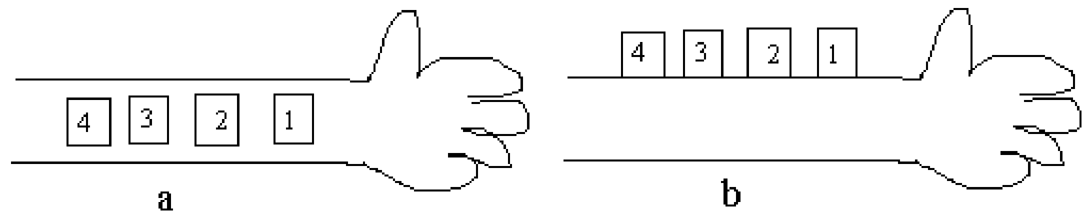

Figure 9: Four sensors in a row on lower arm placements, a) on side of arm, b) on top of arm.

5.2 Gesture Repeatability Metric

In order to quantitatively evaluate repeatability, we propose to use a Dynamic Time Warping (DTW) based metric that has been used in speech recognition domain [3] for matching words (sounds). The DTW algorithm accounts for different rates of the said words, which correspond to individual gestures in our case. First, the DTW algorithm finds the difference between two recordings of gestures one angle at a time, resulting in a numerical error for each angle. Second, the algorithm compares gesture-A (x-axis) with gesture-B (y-axis) by going only forward in time and making the best match at each sample pair (i, j), where i is a time sample from gesture-A and j is a time sample from gesture-B. To find the smallest value at each (i, j), the local distance is calculated first between the samples of gesture-A(i) and gesture-B(j), and then added to the lowest cumulative global distance from one of the three possible previous coordinates according to equation (1).

$$D_{i,j} = d_{i,j} + \min\left(D_{i-1,j}, D_{i-1,j-1}, D_{i,j-1}\right) \qquad (1)$$

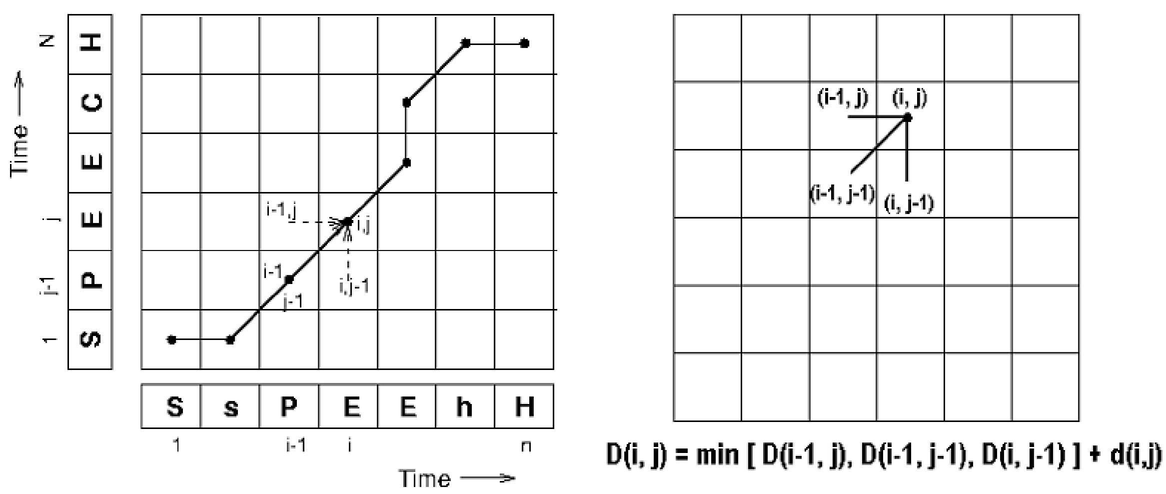

Figure 10: Illustration of DTW algorithm used for comparing two instances of the word "speech". In our case the letters are replaced by angle measures. The picture shows the shortest global path from beginning to end, as well as the calculation of error at coordinates (i, j).

Figure 10 shows an illustration of DTW error computation for the in word "speech". In our case, the local distance $d_{i,j}$ is calculated according to (1) between one Euler angle from gesture-A and the corresponding Euler angle from gesture-B. $D_{i,j}$ is the overall error at times i and j for the chosen Euler angle in the two gestures. The final error E_{DTW} between two gestures for one Euler angle is the last computed $D_{i,j}$, where i and j are the final samples in their respective gestures. E_{DTW} corresponds to the upper right corner of the illustration in Figure 10. By considering one Euler angle at a time, we obtain 12 different global errors E_{DTW} (3 angles for each of the 4 sensors). The total DTW based error E_T for a pair of gestures is then computed by summing all global errors according to (2).

$$E_T = \sum_{i=1}^{12} E_{DTW}(i) \qquad (2)$$

Table 1: Rankings of 20 gestures from lexicon, from 1 being most similar to others to 20 being most different from others. All directions refer to the orientation of the pilot in the aircraft, unless otherwise noted.

Rank	Gesture name
1	Turn To Right
2	Turn To Left
3	Launch Bar Up
4	Up Hook
5	Down Hook
6	Move Ahead
7	Disengage Nose-gear Steering Left
8	Fold Wings
9	Launch Bar Down
10	Move Back
11	Spread Wings
12	Engage Nose-gear Steering Left
13	Pivot To Left
14	Pivot To Right
15	I Have Command (Yellow shirt's left arm is up)
16	Brakes
17	Slow Down
18	Pass Control (To yellow shirt's left)
19	Stop
20	Slow Down Engines on Right

5.3 Experimental Results

While conducting gesture comparisons, an arbitrary percentile of the same gesture's recordings can be used as training data or as templates for classification. To select the best templates, six runs of each gesture were recorded and the three best were chosen by computing total DTW based errors. All-possible combinations of triplets (6 choose 3) were evaluated by summing up the three pair-wise total errors for each triplet. For example, given the triplet of gesture sets 1, 2, and 3, the sum of total errors is equal to $E_T(1,2) + E_T(1,3) + E_T(2,3)$ (see Table 2). Minimization of the sum of total errors leads to the optimal selection of training data.

Table 2: Chart with total DTW errors of 6 "Move Ahead" trials, showing that gestures 2, 3, and 6 are the most alike.

	2	**3**	**4**	**5**	**6**
1	21847	22242	26555	32440	28448
2		10783	18124	19220	16395
3			19880	18179	16393
4				22184	16918
5					18786

We also compared all 20 gestures from the NAVY lexicon [2] to each using the DTW based metric. The results showed that gestures “turn to left” and “turn to right” appear to have the highest similarity to the rest of the gestures and hence might be misclassified more likely than other gestures. However, the E_T values for these two gestures are larger than the E_T values for gesture repetitions by a factor of at least three. The E_T value for a pair of the gestures “slow down” and “slow down engine on indicated side” leads to the largest value among E_T values for all pairs of gestures and therefore these two gestures should be classified with the highest confidence. We also ranked all 20 NAVY gestures based on the global error E_T. The ranking of gestures from least repeatable to most repeatable is shown in Table 1 and the error values are shown in Table 3. A lower value of E_T means that a gesture is more similar to the rest and thus less repeatable.

Table 3: A gesture dissimilarity matrix formed by comparing all pairs of gestures from the NAVY lexicon using the proposed DTW metric. Small values indicate high similarity. Color-coded entries show values below 70,000 (green); or below 60,000 (yellow); or below 50,000 (orange).

		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Brakes	1	80	90	128	94	87	87	85	81	108	108	95	99	120	168	100	110	80	67	70
Disengage Nosegear	2		65	71	81	78	85	61	67	78	104	67	96	96	143	92	142	54	63	67
Down Hook	3			72	73	97	69	60	85	69	102	99	82	104	102	76	124	72	66	48
Engage Nosegear	4				94	85	105	67	75	91	106	97	107	103	129	100	139	47	66	85
Fold Wings	5					115	65	46	78	92	94	75	101	93	157	67	92	77	73	72
I Have Command	6						106	85	97	88	101	91	110	112	77	109	165	63	81	74
Launch Bar Down	7							52	71	94	108	110	71	123	122	80	97	95	81	44
Launch Bar Up	8								62	79	104	93	93	94	113	72	101	69	60	58
Move Ahead	9									70	91	82	82	90	145	74	106	63	62	81
Move Back	10										110	84	77	78	103	80	142	66	60	93
Pass Control	11											110	116	107	152	105	105	97	112	95
Pivot to Left	12												108	87	127	96	137	40	85	104
Pivot to Right	13													109	126	86	140	73	44	71
Slow Down	14														150	68	106	86	97	126
Slow Down Engines on side	15															152	220	123	96	88
Spread Wings	16																88	76	70	91
Stop	17																	126	126	127
Turn to Left	18																		41	81
Turn to Right	19																			69
Up Hook	20																			

5.4 Data Visualization

The data is captured in sets, with each set containing a timestamp (milliseconds from power on or last reset), and 3 (Euler angles) or 4 (quaternion) values for each of the four sensors. The baud rate of the connection between the PC and the base unit determines the length of the time intervals, which are about 5-10 milliseconds at the highest setting of 115,600 baud. The lowest available transmission rate is 9600 baud. However since the human arm cannot make significantly large movement changes in such small time intervals, it is reasonable to down-sample the collected data to about 10 samples per second. This will allow for faster processing times during various calculations such as classification.

To help us analyze the results we created our own software for data visualization in real time capture, which allows us to detect any glitches in the captured data by eye. Our software allows us to choose and display any or all of the angles and sensors in multiple windows, as opposed to the IS300 demo software, that came on a CD with the hardware, only showing the orientation of one inertia cube at a time. An example of visualization software is in Figure 11.

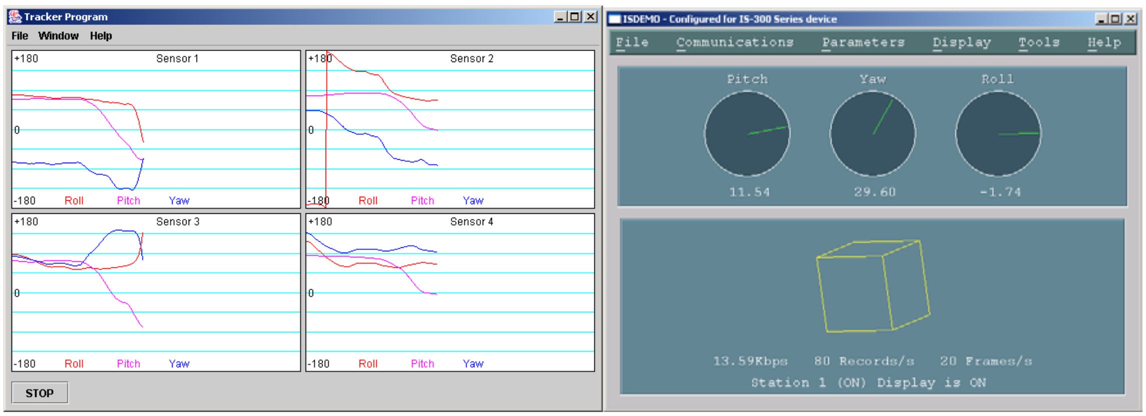

Figure 11: Example of our developed real time visualization (left) vs. IS300 Demo (right).

6 CLASSIFICATION AND ROBOT CONTROL

6.1 Classification

One attempted method of classification was a template-based method using the DTW. Six repeated recordings were taken for each of 11 different gestures (turn to right, turn to left, launch bar up, move ahead, pivot to right, pivot to left, brakes, slow down, pass control, stop, slow down engines). In each set of six the DTW error was calculated pair-wise, then for each of the six recordings the sum of the DTW errors versus the other five was calculated. From each set the gesture with the lowest sum was chosen as the template for that gesture. Then four more recordings of each of the eleven different gestures were collected and compared to the eleven templates. The recordings were classified to be the template with the smallest matching DTW error. These 44 tests had 91% accuracy. Also it was found that all the best matching templates gave DTW error values smaller than 65,000. This means that 65,000 can be used as a good threshold value to mark gestures as unknown. Results are shown in table 4.

When only five of the eleven templates were used (best five according to gesture rankings: brakes, slow down, pass control, stop, slow down engines) in conjunction with the threshold of 65,000 the accuracy of the classification was 95%. Results are in Table 5. One setback of this classification method is that it is very time consuming, as the classification time grows linearly with the number of templates used. This makes it practically unusable for real time classification.

Table 4: Result of gesture classification with 11 templates and four repetitions of each of the 11 gestures. Green numbers show correct classification, orange shows incorrect classification.

		1	2	3	4	5	6	7	8	9	10	11
TurnToRight	1	4										
TurnToLeft	2		0				4					
LaunchBarUp	3			4								
MoveAhead	4				4							
PivotToRight	5					4						
PivotToLeft	6						4					
Brakes	7							4				
SlowDown	8								4			
PassControlMyLeft	9									4		
Stop	10										4	
SlowDownEngines(MyLeft)	11											4

Table 5: Result of gesture classification with 5 templates and four repetitions of each of the 11 gestures, with a threshold to determine unknown gestures.

		1	2	3	4	5	Unknown
Brakes	1	4					
SlowDown	2		4				
PassControlMyLeft	3			4			
Stop	4				4		
SlowDownEngines(MyLeft)	5					4	
TurnToRight	6			1			3
TurnToLeft	7						4
LaunchBarUp	8			1			3
MoveAhead	9						4
PivotToRight	10						4
PivotToLeft	11						4

6.2 Robot Control

The final, and probably the simplest, step is to create a mapping between the gesture lexicon and robot movement instructions. Few examples are shown in Table 6.

Table 6: Examples of some gesture to robot instruction mappings.

Gesture	Robot Instruction
Move Ahead	SETVEL2 40 40
Turn To Left	SETVEL2 30 40
Turn To Right	SETVEL2 40 30
Brakes	STOP
Pivot To Left	SETVEL2 –30 30
Pivot To Right	SETVEL2 30 –30
Slow Down	MULTVEL 0.8
Move Back	SETVEL2 –40 -40
Slow Down Engines on Left	MULTVELL 0.8

7 CONCLUSION

7.1 Summary of Current Work

In building our system we first considered the advantages and disadvantages of existing technologies that could be used in gesture sensing in Section 2. Some technologies we considered such as the cyberglove [8] or the MEMS glove from UC Berkeley [8] were more appropriate for hand and finger gestures than whole arm gestures. Passive sensing such as video recognition was also considered but due to the possibility of bad visibility on Navy aircraft carriers we decided to use an active sensor method. This led us to the IS-300Pro Precision Motion Tracker by InterSense [1]. In Section 3 we discussed the IS-300Pro unit's parameters. The unit can have up to four cubes connected to it, and these InertiaCubes can detect their orientation in 3D space. These orientations can be reported to the computer, which is connected to the base unit, in the form of Euler angles, quaternions or rotation matrices. We chose Euler angles, because they can be easily understood and interpreted by humans, and also because they require less bandwidth when sending packets from the IS-300Pro base unit to the computer. We tested different attachment locations of the InertiaCubes to the human user's arms in Section 4. The attachment locations chosen were two sensors per arm, one located on the upper arm near the elbow, and the other on the lower arm near the wrist. The facing of the sensors is sideways away from the user when the arms are held vertically next to the body. In Section 5 we introduced the metric called Dynamic Time Warping for measuring the repeatability of these gestures. We also ranked the 20 gestures from the Navy lexicon [2] according to similarity to each other. Finally, in Section 6 we use the DTW to form a template based method of arm gesture recognition, and presented some test results of this system. The robot movement to gesture mappings were also presented in Section 6.

7.2 Challenges

During the experiments, we have resolved several challenges related to (1) yaw variation due to flight director's orientation, (2) angular offset due to sensor attachment, and (3) singularity points of Euler angles. First, processing only relative values compensated the yaw variation. Second, periodically running repeatability experiments and mending any loose sensor attachment detected the angular offset. Third, an appropriate gesture modeling compensated the occurrence of singularity points.

In the real environment of an aircraft carrier there are challenges related to (1) a UAV following gestures from a flight director that is not located within its line of sight, (2) approximately ±6° orientation variation due to back and forth rocking of the aircraft carrier, and (3) gestures deviating from a lexicon that are caused by flight director's fatigue. First, video recognition could be used to determine whether an active flight director is in the UAV's line of sight. Second, the orientation sensors of each flight director could be normalized to the orientation of the aircraft carrier. Third, a system detecting continuously increasing deviations of performed gestures from lexicon-defined gesture could be used for alerting flight directors about their fatigue.

7.3 Future Work

In the future it would help if the sensors became wireless. This could solve some problems a flight director (an operator) might have with tangled wires, as well as remove the need for an operator to wear the base device on his/her body. More reliable and smaller sensors would also be helpful in improving the repeatability and successful recognition rates.

The hand gesture tracking can be used anywhere where communication by sound is impossible, either due to requirements of silence, such as in covert commando operations, or in loud places, such as construction sites. Another future application of this technology could find the use by deaf and mute persons in order for them to communicate with people that do not understand sign language, for instance, by connecting the gesture recognition technology to a voice synthesizer.

8 REFERENCES

[1] Intersense web site: http://www.intersense.com/

[2] US Navy, "Field Manual FM1-564 Appendix A", web site: http://www.adtdl.army.mil/cgi-bin/atdl.dll/fm/1-564/AA.HTM

[3] Wrigley Stuart N. "Speech Recognition by Dynamic Time Warping", web site: http://www.dcs.shef.ac.uk/~stu/com326/

[4] ActivMedia robots web site: http://robots.activmedia.com

[5] Gregory G. Slabaugh, "Computing Euler angles from a rotation matrix" (denoted as TRTA implementation from: http://www.starfireresearch.com/services/java3d/samplecode/FlorinEulers.html.

[6] Paul Heckbert (editor), Graphics Gems IV, Academic Press, 1994, ISBN: 0123361559 (Mac: 0123361567), (the GEMS method refers to a particular implementation of "Euler Angle Conversion" by Ken Shoemake, shoemake@graphics.cis.upenn.edu)

[7] Bobick A.F. and J. W. Davis, " The Recognition of Human Movement Using Temporal Templates," IEEE Trans. On PAMI, VOL. 23. NO. 3, March 2001, pp. 257-267

[8] Kessler G. D., L.F. Hodges, N. Walker, "Evaluation of the CyberGlove as a Whole Input Device,"

[9] Smart Dust Project at http://www.cs.berkeley.edu/~awoo/smartdust/

[10] Hollar, S. E., "COTS Dust," Master Thesis at University of California, Berkeley, Fall 2000.

[11] Global Positioning Systems, The GPS Store, http://www.thegpsstore.com/site/

[12] Moeslund Thomas B., Granum Erik, "A Survey of Computer-Vision Based Human Motion Capture," Computer Vision and Image Understanding 81, 231–268 (2001)