Human Motion Tracking and Recognition with Microsoft Kinect

Wenbing Zhao
Cleveland State University
Outline

- Introduction to the Kinect technology
- Kinect applications
- Human motion tracking with Kinect
- Demo
- Human motion recognition with Kinect
- Publicly available dataset
The Kinect Technology

- Launched in 2010 as an enhancement device for the Xbox 360 game console
  - “You are the controller”: Game playing using gestures and voices
  - 24 million units sold as of Feb 2013
- Software development kit (SDK) was released by Microsoft in late 2011
  - Third party SDKs were available before that
- Two generations of Kinect sensor
  - Kinect v1: Kinect for Xbox, Kinect for Windows
  - Kinect v2: Kinect for XboxOne
- What Kinect SDK provides
  - 2D color image frames
  - 3D depth image frames
  - 3D skeletal frames with a set of joints in each skeleton
Steps of Motion Tracking and Analysis with Kinect

- Kinect depth frames
- Kinect RGB frames
  - Human subject foreground extraction
  - Pose estimation
  - Skeleton estimation
    - Kinect skeleton frames
      - Motion recognition
      - Feedback to user
        - Actions triggered
Kinect Depth Sensing Technology

- The depth-sensing technology used in Kinect v1 was developed by PrimeSense.
- It uses structured light with a single infrared (IR) emitter and a single depth sensor for triangulation of depth in each pixel.
Kinect Depth Sensing Technology

- The depth-sensing technology used in Kinect v2 was based on time-of-flight measurement.
- The depth of each pixel can be calculated based on the phase shift between the emitted light and the reflected light.
- A clever design:
  - IR emitter is periodically turned on and off, and the output from the depth sensor is sent to two different ports when the light is on (port A) and off (B), respectively.
  - A-B: removes background ambient light => depth + high quality IR images.
IR and Depth Image Quality Comparison

(a) Kinect v1 IR Image

(b) Kinect v2 IR Image

(c) Kinect v1 Depth Image

(d) Kinect v2 Depth Image
# Feature Comparison of Kinect v1 & v2

<table>
<thead>
<tr>
<th>Feature</th>
<th>Kinect v1</th>
<th>Kinect v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth sensing technology</td>
<td>Triangulation with structured light</td>
<td>Time of flight</td>
</tr>
<tr>
<td>Color image resolution</td>
<td>640 × 480 30 fps</td>
<td>1920 × 1080 30 fps (12 fps low light)</td>
</tr>
<tr>
<td></td>
<td>1280 × 960 12 fps</td>
<td></td>
</tr>
<tr>
<td>IR image resolution</td>
<td>640 × 480 30 fps</td>
<td>512 × 424 30 fps</td>
</tr>
<tr>
<td>Depth sensing resolution</td>
<td>640 × 480 30 fps</td>
<td>512 × 424 30 fps</td>
</tr>
<tr>
<td></td>
<td>320 × 240 30 fps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 × 60 30 fps</td>
<td></td>
</tr>
<tr>
<td>Field of view</td>
<td>43° vertical</td>
<td>&gt; 43° vertical</td>
</tr>
<tr>
<td></td>
<td>57° horizontal</td>
<td>70° horizontal</td>
</tr>
<tr>
<td>Depth sensing range</td>
<td>0.4–3 m (near mode)</td>
<td>0.5–4.5 m</td>
</tr>
<tr>
<td></td>
<td>0.8–4 m (default mode)</td>
<td>Up to 8 m without skeletonization</td>
</tr>
<tr>
<td>Skeleton tracking (with full</td>
<td>Up to 2 subjects 20 joints per skeleton</td>
<td>Up to 6 subjects 25 joints per skeleton</td>
</tr>
<tr>
<td>skeleton)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-in gestures</td>
<td>None</td>
<td>Hand state (open, close, lasso) Hand pointer controls; lean</td>
</tr>
<tr>
<td>Unity support</td>
<td>Third party</td>
<td>Yes</td>
</tr>
<tr>
<td>Face APIs</td>
<td>Basic</td>
<td>Extended massively</td>
</tr>
<tr>
<td>Runtime design</td>
<td>Can run multiple Kinect sensors per computer;</td>
<td>At most one Kinect per computer;</td>
</tr>
<tr>
<td></td>
<td>One app per Kinect</td>
<td>Multiple apps share same Kinect</td>
</tr>
<tr>
<td>Windows store</td>
<td>Cannot publish to</td>
<td>Yes</td>
</tr>
</tbody>
</table>

* Source: University of Science and China Press*
Outline

■ Introduction to the Kinect technology
■ Kinect applications
■ Demo
■ Human motion tracking with Kinect
■ Human motion recognition with Kinect
■ Publicly available dataset
Kinect Applications

- Virtual Reality & Gaming
- Natural User Interface
- Education & Performing Arts
- Healthcare
- Retail
- Training
- Robotics Control & Interaction
- 3D Reconstruction
- Sign Language Recognition
- Physical Therapy
- Medical Operation
- Fall Detection
- Motion Recognition

Cleveland State University

SCIENCE CHINA PRESS
## Kinect Applications In Healthcare

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical therapy and rehabilitation</td>
<td>Assessment of Kinect motion tracking accuracy for healthcare applications</td>
</tr>
<tr>
<td></td>
<td>An interactive game-based rehabilitation tool for adults with neurological injury</td>
</tr>
<tr>
<td></td>
<td>Full-body control in virtual reality applications</td>
</tr>
<tr>
<td></td>
<td>Integration of Kinect and a smart glove for patients with upper extremity impairment</td>
</tr>
<tr>
<td></td>
<td>An interactive rehabilitation system for disabled children</td>
</tr>
<tr>
<td></td>
<td>A rehabilitating program for young adults with motor impairments</td>
</tr>
<tr>
<td></td>
<td>Cognitive rehabilitation for Alzheimer’s patients using a Kinect-based game</td>
</tr>
<tr>
<td></td>
<td>A Kinect-based game for stroke rehabilitation</td>
</tr>
<tr>
<td></td>
<td>An exercise rehabilitation program for individuals with spinal cord injury</td>
</tr>
<tr>
<td></td>
<td>Integration of Kinect with rehabilitation robotics</td>
</tr>
<tr>
<td></td>
<td>Integration of inertial sensors with Kinect</td>
</tr>
<tr>
<td></td>
<td>An at-home exercise monitoring system</td>
</tr>
<tr>
<td></td>
<td>A Kinect-based intra-operative medical image viewer</td>
</tr>
<tr>
<td></td>
<td>A system that enables touchless controlling of medical images with hand and arm gestures</td>
</tr>
<tr>
<td></td>
<td>A real-time fall monitoring and detection system</td>
</tr>
<tr>
<td></td>
<td>Overcoming occlusions for human body fall detection</td>
</tr>
<tr>
<td>Medical operating room assistance</td>
<td>Fall detection based on Kinect skeletal data</td>
</tr>
<tr>
<td></td>
<td>Human fall detection using two Kinect</td>
</tr>
<tr>
<td></td>
<td>Capturing variations of stride-to-stride gait for elderly adults</td>
</tr>
<tr>
<td></td>
<td>Detecting falls and other abnormal events on stairs</td>
</tr>
<tr>
<td></td>
<td>Fall prevention in hospital ward environment</td>
</tr>
</tbody>
</table>
Exploration of medical images

- Controller-free exploration of medical image data: experiencing the Kinect, by Gallo, Placitelli, and Ciampi, Italy
- http://www.youtube.com/watch?v=CsIK8D4RLtY
## Kinect Applications in Virtual Reality and Gaming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>VR fundamentals</td>
<td>A markerless virtual reality system</td>
</tr>
<tr>
<td></td>
<td>A computational algorithm for skeleton animation</td>
</tr>
<tr>
<td></td>
<td>Mixed reality applications based on Kinect depth imaging</td>
</tr>
<tr>
<td></td>
<td>3D full human body scan using multiple Kinect</td>
</tr>
<tr>
<td>VR games</td>
<td>A game expressing “individuality”</td>
</tr>
<tr>
<td></td>
<td>A treadmill game based on Kinect depth maps</td>
</tr>
<tr>
<td></td>
<td>A HoloDesk game combining an optical display and Kinect</td>
</tr>
<tr>
<td></td>
<td>A game examining child-defined gestures</td>
</tr>
<tr>
<td></td>
<td>A cloud-hosted Kinect-based game</td>
</tr>
</tbody>
</table>
### Kinect Applications in Natural User Interface

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural user interface</td>
<td>Performing desktop tasks via arm gestures</td>
</tr>
<tr>
<td></td>
<td>3D object manipulation on a desktop display</td>
</tr>
<tr>
<td></td>
<td>3D control method based on Kinect</td>
</tr>
<tr>
<td></td>
<td>A 3D navigation user interaction system</td>
</tr>
<tr>
<td></td>
<td>A group meeting application based on Kinect</td>
</tr>
<tr>
<td></td>
<td>Controlling virtual globes via Kinect</td>
</tr>
<tr>
<td></td>
<td>Web browsing via natural user interfaces</td>
</tr>
<tr>
<td></td>
<td>Automatic camera control based on Kinect</td>
</tr>
</tbody>
</table>

### Kinect Applications in Education and Performing Arts

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Education and performing arts</td>
<td>A classroom teaching system with Kinect</td>
</tr>
<tr>
<td></td>
<td>A interactive music conductor generation system with Kinect</td>
</tr>
<tr>
<td></td>
<td>A puppetry control application with Kinect</td>
</tr>
<tr>
<td></td>
<td>A MotionDraw tool for enhancing art performance</td>
</tr>
</tbody>
</table>
## Kinect Applications in Robotics Control and Interaction

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotics navigation and control</td>
<td>Robot navigation using Kinect and inertial sensors</td>
</tr>
<tr>
<td></td>
<td>Feasibility on using gestures to control industrial robots</td>
</tr>
<tr>
<td></td>
<td>A human imitation system</td>
</tr>
<tr>
<td>Interactively controlling robotics</td>
<td>Navigating a robot using hand gestures</td>
</tr>
<tr>
<td></td>
<td>A human–robot interactive demonstration system with a gesture recognizer</td>
</tr>
<tr>
<td></td>
<td>An athletic training speed skating system using Kinect</td>
</tr>
<tr>
<td>Robotics remote control</td>
<td>A Kinect on-board system that enables the control of velocity and attitude of a mobile robot</td>
</tr>
<tr>
<td></td>
<td>Controlling altitude of a quadrotor helicopter via Kinect</td>
</tr>
<tr>
<td></td>
<td>A real-time human imitation system for robotics</td>
</tr>
<tr>
<td></td>
<td>Tele-operating a humanoid robot using Kinect</td>
</tr>
<tr>
<td>What is Recognized</td>
<td>Feature Set</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Speech</td>
<td>Face and mouth</td>
</tr>
<tr>
<td>10 numbers in CSL</td>
<td>Depth and motion profiles</td>
</tr>
<tr>
<td>34 signs in Libras</td>
<td>Phonological structure</td>
</tr>
<tr>
<td>10 words in Libras</td>
<td>Hand shape</td>
</tr>
<tr>
<td>10 signs in ISL</td>
<td>Hand position and trajectory</td>
</tr>
<tr>
<td>20 words in CSL</td>
<td>Position and trajectory of right hand</td>
</tr>
<tr>
<td>34 words in CSL</td>
<td>Hand trajectories and hand shapes</td>
</tr>
<tr>
<td>25 signs in GSL</td>
<td>9-dimension joints based on Kinect joints</td>
</tr>
<tr>
<td>25 words in TSL</td>
<td>Hand positions, movement, and shapes</td>
</tr>
<tr>
<td>4 signs in PSL and 3 generic</td>
<td>Kinect joints</td>
</tr>
<tr>
<td>signs</td>
<td></td>
</tr>
<tr>
<td>111 words in TSL1</td>
<td>DCT coefficients</td>
</tr>
<tr>
<td>30 words in PSL1 isolated words</td>
<td>Kinect joints and hand shapes from color images</td>
</tr>
<tr>
<td>ASL alphabet</td>
<td>Hand shapes</td>
</tr>
<tr>
<td>10 SIBI words</td>
<td>Kinect joint orientation and features from depth images</td>
</tr>
<tr>
<td>24 ASL letters</td>
<td>Hand features from color and depth images</td>
</tr>
<tr>
<td>PSL2 alphabet</td>
<td>Hand angular pose</td>
</tr>
<tr>
<td>150 gestures</td>
<td>Arm postures and finger-related features</td>
</tr>
<tr>
<td>19 ASL words</td>
<td>Hand shapes, motion and pose</td>
</tr>
</tbody>
</table>
Sign Language Translation

- Sign Language Recognition and Translation with Kinect, by Chai et al, Chinese Academy of Sciences & Microsoft Research Asia
  - [https://www.youtube.com/watch?v=HnkQyUo3134](https://www.youtube.com/watch?v=HnkQyUo3134)
Kinect Applications in Retail, Training, & 3D Reconstruction

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail</td>
<td>Human behavior pattern recognition on products interaction</td>
</tr>
<tr>
<td></td>
<td>Augmented reality system for virtual handbags</td>
</tr>
<tr>
<td>Training</td>
<td>Back injury prevention</td>
</tr>
<tr>
<td></td>
<td>Musculoskeletal injury prevention</td>
</tr>
<tr>
<td>3D Reconstruction</td>
<td>3D human body reconstruction</td>
</tr>
<tr>
<td></td>
<td>Reconstructing 3D mesh skeleton</td>
</tr>
<tr>
<td></td>
<td>Real-time 3D reconstructing of moving human body</td>
</tr>
<tr>
<td></td>
<td>Integrate Kinect with high resolution webcam for 3D image reconstruction</td>
</tr>
</tbody>
</table>
Scanning 3D Full Human Bodies

- https://www.youtube.com/watch?v=OJZbmUVflYc
Kinect Application: Privacy-Aware Human Motion Tracking

- Patient privacy is protected by government regulations, such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- Patients cannot be videotaped without their explicit consent
- In the mean time, it could be very beneficial to monitor the activities of health caregivers: safe patient handling, training, etc.
- Inertial sensor based tracking preserves the patient privacy, however, this type of tracking is intrusive and cannot provide accurate and detailed information of the activities comparable to that of vision-based tracking (unless a large set of sensors attached to specific locations)
Privacy-Aware Human Motion Tracking

- Computer vision based tracking: non-intrusive, accurate, and detailed information human activities
  - Problem: a camera is non-discriminative – everyone in the view is tracked!

- What we want: track only consented human subjects when they interact with other people

- The proposed system ensures the tracking of only consented personnel: captured information regarding non-consented personnel in the view is discarded immediately
Privacy-Aware Human Motion Tracking: Hardware Components

- Microsoft Kinect sensor (or any equivalent programmable depth camera)
- Computer (to run the Kinect motion tracking application)
- Wearable device, such as smart watch (worn by the consented personnel)
- Smart phone (needed only if the wearable device could not directly communicate with the application running on the computer)
Privacy-Aware Human Motion Tracking: Software Components

- Pebble Application
  - Pebble Smart Watch
  - Pebble Mobile Application (PebbleKit JS)
  - pebble-js-app.js
  - Smart Phone
  - XMLHttpRequest
  - XMLHttpResponse (JSON object)

- Server Application
  - Skeleton Stream
  - Kinect
  - Computer
Discriminative Tracking Mechanism

- Simultaneous detection of a predefined gesture by both the wearable device worn by the consented personnel and the depth camera: **personnel registration mechanism**
- If detected: start to track the registered personnel
- If not, collected data (current frame) is discarded

**Workflow of the registration mechanism:**
- Consented personnel walks in front of the camera
- He/she performs a predefined gesture with the wearable device
- Wearable device captures the gesture, sends a notification to the vision-side application
- Vision-based application examines the next frame to correlate the event to a specific skeleton detected
**Discriminative Tracking Mechanism**

- Predefined gesture: tapping the wearable device (worn on one wrist) with a finger => Distance between two wrists below a threshold
  - For each human skeleton detected by the programmable camera, calculate the distance between two wrists
  - The one with a distance below the threshold is the one to be tracked
  - Need to examine the current frame only

- To be more robust, on the vision-side, the entire gesture could be tracked to minimize the chance of tracking a wrong subject
  - Wrist-distance reduces to a minimum (< threshold) and then increases beyond the threshold
  - Require the caching of more frames
Discriminative Tracking Mechanism

- Practical issues in discriminative tracking:
  - Consented person might forget to register
  - The person might not be in the view of the camera when registering
  - The person might go out of the view of the camera => would require re-registration

- Reminder via the wearable device worn by the person
Registration Latency

- Mean round-trip time is 262ms: tolerable for caregiver registration.
- The skeleton frame examined by the server application will be 4-5 frames late most of the time.
Registration
Gesture Recognition

Tapping gesture:
• 1.5 seconds or so
• Wrist-to-wrist distance < 0.1m
A Use Case:

Safe Patient Handling Monitoring

- Nursing aids might not follow best practices when handling patients
- Privacy-aware vision-based human motion tracking could be used to help increase compliance
- Generate haptic notification via wearable device, together with a short descriptive message, when non-compliant activities are detected
- Such incidence, as well as statistics for correct activities are logged
Demo
Outline

- Introduction to the Kinect technology
- Kinect applications
- Demo
- Human motion tracking with Kinect
- Human motion recognition with Kinect
- Publicly available dataset
Human Detection and Tracking

- Object detection and tracking
  - Main approach: background subtraction => foreground humans
  - RGB-images sensitive to illuminating changes, even if camera is static
  - Depth data helps to establish a more stable background model

- Pose estimation: skeletal tracking
  - Per-pixel body part classification
  - Estimating body joint positions by computing a local centroids of the body part probability mass using mean shift mode detection
Object Detection and Tracking

- Depth data
  - Benefit: more robust against illuminating changes
  - Disadvantage: limited dimension => less discriminative in representing an object
  - May be beneficial to combine both depth and RGB data

- Common background subtraction algorithms
  - Gaussian mixture model (GMM)
  - Histograms of oriented gradients (HOG)
  - Pyramid HOG
  - Local binary patterns (LBP)
  - Local ternary patterns (LTP)
  - Census transform histograms (CENTRIST)
Pose Estimation

- Kinect SDK provides an advanced skeletal tracker => much simpler human gesture and activity recognition
- Core algorithm
  - Per-pixel body part classification: intermediate representation for human pose estimation
    - Invariant to human body shape and pose
    - Highly accurate and fast
  - Skeleton model is fitted to the hypothesized joint positions
    - Randomized decision forests
Outline

- Introduction to the Kinect technology
- Kinect applications
- Demo
- Human motion tracking with Kinect
- Human motion recognition with Kinect
- Publicly available dataset
Human Motion Recognition

- Human motion recognition aims to understand the semantics of the human gestures and activities

- Gesture
  - Typically involves one or two hands, and possibly body poses
  - Convey some concrete meaning, such as waving a hand to say goodbye

- Activity
  - A sequence of full body movements that a person performs, such as walking, running, brushing teeth, etc.,
  - Not necessarily conveys a meaning to the computer or other person
  - Rehabilitation exercises form a special type of activities
Approaches to Human Motion Recognition

- Template-based
  - Classification of an unknown gesture or activity is done by comparing with a pre-recorded template motion automatically via pattern recognition

- Algorithmic (rule) based
  - A gesture or an activity is recognized based on a set of manually defined rules
Approaches to Human Motion Recognition

- Motion Recognition
  - Template-Based Recognition
    - Direct Matching
      - DTW, etc.
  - Algorithmic Recognition
    - Model-Based Matching
      - Various Machine Learning Methods
    - Non ML-Based Kinematic Modeling
Algorithmic Recognition

- Algorithmic-based recognition is popular in gaming and healthcare applications
  - Gestures and/or activities are usually very well defined, relatively simple, and repetitive in nature
  - Each gesture or activity normally has a predefined starting and ending pose
  - For rehabilitation exercises, rules are primarily defined to assess the correctness of movements rather than to classify them
  - Rules are predominately expressed in terms of joint angles

- Limitations
  - Rules have to be carefully defined by experts and expressed in implementable form
  - Gesture/activity has to be simple enough to be defined in terms of a set of implementable rules
  - Parameters used in the rules for boundary conditions must be manually tuned
Algorithmic Recognition

- Algorithmic-based recognition is popular in gaming and healthcare applications
  - Gestures and/or activities are usually very well defined, relatively simple, and repetitive in nature
  - Each gesture or activity normally has a predefined starting and ending pose
  - For rehabilitation exercises, rules are primarily defined to assess the correctness of movements rather than to classify them
  - Rules are predominately expressed in terms of joint angles

- Limitations
  - Rules have to be carefully defined by experts and expressed in implementable form
  - Gesture/activity has to be simple enough to be defined in terms of a set of implementable rules
  - Parameters used in the rules for boundary conditions must be manually tuned
Gesture Description Language

- GDL was developed for the purpose of recognizing hand and body gestures
- All rules are expressed in terms of one or more key frames except the final rule, which defines the gesture in terms of a sequence of basic rules
- During runtime, a gesture is recognized with the following steps executed in a loop
  1. When a new frame arrives, the new motion data is stored in a memory heap. The set of rules that have been satisfied so far are also stored in the heap
  2. Examine the new data to see if any new rule is now satisfied
  3. If a new rule is satisfied, the rule name is placed at the top of heap with a timestamp. If the final rule that defines a gesture is satisfied, then the gesture is recognized.
  4. If a new rule is satisfied in the previous step, go to step 2 to see if any other rule is now satisfied as well. Otherwise, go back to step 1 waiting for the next frame
Because GDL is designed to be based on a set of key frames, it is resilient to motion sensing errors.

As a tradeoff, it lacks the support for rules that depend on the entire trajectory of a gesture.

It also lacks a guideline as to how to identify the key frames for each gesture.
Real-Time Rule-Based Assessment for Rehabilitation Exercises

- We recently introduced a set of basic rule elements that can be used to define correctness rules for common rehabilitation exercises
- We also provided a guide on how to develop such rules
- The rules are encoded using eXtensible Markup Language (XML) for extensibility and customizability
- The rules have three different types:
  - Rules for dynamic movement
  - Rules for static poses
  - Rules for movement invariance
Specification of Correctness Rules

- Rules for dynamic movement
  - Each rule is expressed in terms of the sequence of reference configurations of a particular joint or body segment that delineate monotonic segments of each iteration

- Rules for static poses
  - Some exercises only involve stationary poses
  - It is also possible for some body parts to remain stationary at their desirable positions while other parts are moving in some other exercises

- Rules for movement invariance:
  - Each of which defines the requirement for a moving body segment that must be satisfied during every iteration of the exercise
Encoding of Rules

```xml
<CorrectnessRules>
  <ExerciseName> ... <ExerciseName>
  <DynamicRule> ... </DynamicRule>
  <DynamicRule> ... </DynamicRule>
  ...
  <DynamicRule> ... </DynamicRule>
  <StaticRules> ... </StaticRules>
  <InvarianceRules> ... </InvarianceRules>
</CorrectnessRules>

<DynamicRule>
  <Configuration> ... </Configuration>
  <Configuration> ... </Configuration>
  ...
  <Configuration> ... </Configuration>
</DynamicRule>

<Configuration>
  <Type> "JointAngle" </Type>
  <CenterJoint> "JointName" </CenterJoint>
  <DownstreamJoint> "JointName" </DownstreamJoint>
  <UpstreamJoint> "JointName" </UpstreamJoint>
  <Angle> "AngleValue" </Angle>
  <Tolerance> "ToleranceValue" </Tolerance>
  <MaxDuration> ... </MaxDuration>
  <MinDuration> ... </MinDuration>
</Configuration>
```
Example Correctness Rules

```xml
<CorrectnessRules>
  <ExerciseName>*Hip Abduction*</ExerciseName>
  <DynamicRule>
    <Configuration>
      <Type>*BoneOrientation*</Type>
      <DownstreamJoint>*HipCenter*</DownstreamJoint>
      <UpstreamJoint>*RightAnkle*</UpstreamJoint>
      <FrontalAngle>0</FrontalAngle>
      <SagittalAngle>15</SagittalAngle>
      <TransverseAngle>1</TransverseAngle>
      <FrontalAngleTolerance>5</FrontalAngleTolerance>
      <SagittalAngleTolerance>5</SagittalAngleTolerance>
      <TransverseAngleTolerance>-1</TransverseAngleTolerance>
    </Configuration>
    <Configuration>
      <Type>*BoneOrientation*</Type>
      <DownstreamJoint>*HipCenter*</DownstreamJoint>
      <UpstreamJoint>*RightAnkle*</UpstreamJoint>
      <FrontalAngle>0</FrontalAngle>
      <SagittalAngle>15</SagittalAngle>
      <TransverseAngle>1</TransverseAngle>
      <FrontalAngleTolerance>5</FrontalAngleTolerance>
      <SagittalAngleTolerance>5</SagittalAngleTolerance>
      <TransverseAngleTolerance>-1</TransverseAngleTolerance>
    </Configuration>
  </DynamicRule>
  <InvarianceRules>
    <Configuration>
      <Type>*JointAngle*</Type>
      <CenterJoint>*RightKnee*</CenterJoint>
      <DownstreamJoint>*HipCenter*</DownstreamJoint>
      <UpstreamJoint>*RightAnkle*</UpstreamJoint>
      <Angle>180</Angle>
      <Tolerance>5</Tolerance>
    </Configuration>
  </InvarianceRules>
</CorrectnessRules>
```
It is relatively simple to evaluate static rules and invariance rules because they must be satisfied by every frame supplied by the motion sensing device.

The calculation of the joint distance, joint angle, and body segment orientation is straightforward based on 3D vector math.

Each dynamic rule is assessed in realtime by a finite state machine.
The number of states are determined by the number of monotonic segments in the rule, i.e., the number of reference configurations.

Given k reference configurations, C1, C2, ..., Ck, there are k number of states, S1, S2, ..., Sk, and each state Si is initiated by the detecting of the corresponding reference configuration Ci.

Hence, the finite state machine is transitioned to Si on detecting configuration Ci and it will stay in state Si until the next reference configuration Ci+1 specified in the dynamic rule is detected.
Dynamic Rule Assessment: Implementation Issues

- A key mechanism is the detecting of actual monotonic segments in realtime while the patient is doing the exercise by tracking the change of the variables of interest
  - E.g., joint angle, joint distance, or body segment orientation angles
- When a change of sign in speed is detected (i.e., from increasing to decreasing, or vice versa), the current monotonic segment has just ended
- Only at this point, the condition for the next reference configuration is checked
- If the condition is not met, then an error is detected and the patient will be informed via appropriate feedback
- When this happens, the finite state machine goes back to the initial state and waits for the first frame that satisfies the condition in C1
Dynamic Rule Assessment: Implementation Issues

- At state $S_i$, there may be three types of events:
  - Event $e_1$: The arrival of a frame that does not yet satisfy the condition in $C_{i+1}$ and the elapsed time in the current state is smaller than $MaxDuration$, if one is specified. The finite state machine stays at the current state $S_i$ upon this type of events.
  - Event $e_2$: The arrival of the first frame that satisfies the condition in $C_{i+1}$ and the elapsed time in the current state is larger than $MinDuration$, if one is specified. The finite state machine transitions to state $S_{i+1}$ as a result of $e_2$.
  - Event $e_3$: The detection of an error, which could be any of the following:
    - Elapsed time at the current state is too short upon receiving a frame that satisfies the condition for $C_{i+1}$
    - Elapsed time at the current state is too long
    - The current monotonic segment ends too early or too late (i.e., not at the specified target value)
Dynamic Rule Assessment: Implementation Issues

- The addition of the mechanism to detect the actual monotonic segments at runtime makes the system vulnerable to motion sensing errors and small movement errors from the patient.

- For each finite state machine, we keep track of the maximum and minimum values of the variable of interest in each state.
  - For a monotonic segment with increasing values, we delay declaring the end of the segment until the current value is smaller than the last seen maximum value by a predefined heuristic value.
  - Similarly, for a monotonic segment with decreasing values, we delay declaring the end of the segment until the current value is larger than the last seen minimum value by a predefined heuristic value.
Direct-Matching-Based Recognition

- The unknown gesture or activity is directly compared with a set of templates
- Dynamic Time Warning (DTW): most well-known technique to analyze the similarity between two temporal sequences that may vary in time and speed by finding an optimal alignment between them
- Other methods exist: maximum correlation coefficient, earth mover’s distance
Dynamic Time Warning

- One sequence is an unknown sequence to be classified
- The other sequence is a pre-classified reference sequence (i.e., the template, also referred to as the exemplar)
- The difference between the two sequences is expressed in terms of the distance between the two
- DTW has been used to recognize hand gestures
  - Using Kinect depth frames
  - Using L/R hands/wrists joints from Kinect skeleton frames
- DTW has also been used to assess the quality of rehab exercises done at home
  - Trajectory and speed of individual joints are used as feature vectors and are compared separately
- DTW has been used in a 3D signature scheme to compare a test signature with a reference signature
  - Finger tip positions, velocity, acceleration, etc.
Other Direct Comparison Methods

- **Maximum correlation coefficient**
  - MCC used to classify unknown gestures based on a single exemplar gesture set
  - Finding a known gesture that has the maximum correlation coefficient of the corresponding feature vectors

- **Earth Mover’s Distance**
  - EMD used to calculate the similarity in hand shapes between an unknown static hand gesture and a set of templates
Non-Machine-Learning-Based Kinematic Modeling

- MotionMA: A system to assess rehab exercises
- Kinematic model using exemplar data: collection of joint angles
  - Training data first filtered using a low-pass filter to remove noise
  - Feature data is extracted on zero derivatives (i.e., peaks, valleys, and inflexion points)
  - Feature data merged using k-means clustering
  - Merged data serves as the model for the gesture and is used to identify static and dynamic axes
  - Enables the system to monitor violations in static axes continuously in realtime, and to count repetitions for dynamic joints
Machine-Learning-Based Recognition

- ML-models are used to capture the unique characteristics of a gesture or activity
- Typically need large amount of training data
- May need to tune model parameters for best recognition
- Recognition typically framed as a classification problem
- Could also be formulated as a regression problem: has the benefit of determining the progress of a gesture/activity
- Most research work uses joint data from Kinect skeleton frames; some relies on Kinect depth data directly
Hidden Markov Model for Human Motion Recognition

- The HMM is perhaps the most popular model used for motion recognition with Kinect data
- HMM is applicable to any dynamic system that is governed by a Markov chain where only its output values are observable and not its states
- A system that is modeled by HMM is defined by the following parameters:
  - (1) Number of states
  - (2) Number of distinct observation symbols per state
  - (3) Initial state distribution vector
  - (4) The state transition probability distribution matrix, which defines the probability of the transition from one state to another
  - (5) the observation probability distribution matrix, which defines the probability of observing each output symbol given each state
Hidden Markov Model for Human Motion Recognition

- The first three parameters must be determined manually and they are application dependent.
- The last two parameters are determined based on training data.
- The larger the state size and the observation symbol size, the larger amount of training data is required.
- Once all parameters are set, one can perform classification by calculating the most likely state sequence given a sequence of output values.
- Because a human motion consists of a sequence of poses, it is quite natural to use HMM to model a gesture/activity for the purpose of recognition.
- The number of states and number of output symbols depend on the motions to be recognized.
Hidden Markov Model for Human Motion Recognition: Example Research

- Dynamic features: derived from physics-based representation of the human body, such as torques from some joints => lower dimension than kinematic features
- Orientation of hand centroid extracted from Kinect depth frames in 2d frontal plane
- Hierarchical maximum entropy Markov model: to recognize sophisticated gesture/activity with sub-activities
- Apply HMM to individual features, and combine output of HMM using fuzzy logic rules
HMM for Human Motion Recognition

- One-shot HMM learning and realtime classification
  - One state per frame
  - Each state \( i \) corresponds to a sample in the training data and is associated with a Gaussian probability distribution \( b_i \)
  - \( b_i \) used to compute the probability of an observation \( O \)

\[
b_i(O) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp \left[ - \left( \frac{(O - \mu_i)^2}{2\sigma_i^2} \right) \right]
\]

- \( \mu_i \): \( i \)th sampled value associated with state \( i \)
- Tolerance: \( \sigma_i \) standard deviation between recorded references and performance
- Recognition: follows standard forward procedure to HMM
- Latency: how many frames to consider
HMM for Human Motion Recognition

- One-shot HMM learning and realtime classification
  - Recognition: follows standard forward procedure to HMM
  - Requires the computation of a distribution $\alpha_i(t)$, which corresponds to the probability distribution of the partial observation sequence until time $t$ and state $i$.
  - This distribution is estimated iteratively in real time each time a new observation is received and makes it possible to compute two important values:
    - The time progression of the sequence, which is related to the recorded example
    - The likelihood relative to each reference gesture
  - These likelihoods are computed by averaging “instantaneous” likelihoods, referred to each coming observation
  - The average is computed using a sliding window, whose size depends to the number of frames taken into account
  - Latency: how many frames to consider. More frames => higher recognition accuracy
Artificial Neural Networks for Human Motion Recognition

- ANNs refer to a collection of statistical learning algorithms inspired by biological neural networks.
- An ANN models the system as a network of neurons with several layers. The first layer consists of input neurons that send signal to the second layer of neurons.
- The last layer consists of output neurons, which takes input from other neurons.
- There could also be intermediate layers.

Artificial Neural Networks for Human Motion Recognition

- Prior to training the model, the number of input and output neurons, as well as the activation function must be determined based on the recognition problem.
- The number of input neurons depends on the dimension of the feature set,
- The number of output neurons depends on the number of classes to be recognized,
- The total number of neurons needed is typically a tunable parameter.
- Once the network topology is decided, the weights of the interconnections can be learned with pre-labeled training data.
Artificial Neural Networks for Human Motion Recognition: Example Research

- Multiplayer perceptron (MLP) model: classify static gestures using Kinect data
- NN-based additive nonlinear autoregressive exogenous (NN-ANARX) used to determine the quality of a rehab exercise in terms of the difference between observed motion and the predicted motion with the trained model
- Complex-valued neural network (CVNN): used for static hand gesture recognition
  - Hand tree: length and angles of the lines that form the tree
  - Output: 26 English alphabet letters
Support Vector Machines for Human Motion Recognition

- The SVMs are supervised learning models for linear as well as nonlinear classification.
- For linear classification, the training data is used to determine a plane that separates the data belonging to different classes as further away as possible
  - This plane then can be used to classify unknown data into one of the two classes
- For nonlinear classification, a kernel function is used to make higher dimension classifications (the plane derived from the training data is referred to as hyperplane)
- The key advantage of SVM is that it guarantees maximum-margin separation using relatively little training data
Support Vector Machines for Human Motion Recognition: Example Research

- Segment a gesture into a sequence of units and formulate the gesture analysis problem into a classification task using SVM
  - Distinguish rest positions from a gesture unit
  - Feature vector: L/R hand, L/R wrist, head and spine joints
- SVM was used to identify key poses in a sequence of body motion where the joint angles are used as features
  - Actual gesture recognition was done via a decision forest.
- SVM was used to classify static gestures such as stand, sit-down, lie down using skeleton joints as feature vector
Decision Tree/Forest for Human Motion Recognition

- A decision tree consists of a collection of nodes connected to a tree structure
  - Each internal node (often referred to as the split node) in the tree represents a test on one of the features with a threshold, and each branch represents the outcome of the test
  - A leaf node in the tree represents a class label
- A decision can be taken using the decision tree by computing all attributes
  - The test at the split node is essentially a weak classifier
  - Hence, a decision tree is an ensemble of weak classifiers on different features, which could lead to a better overall classification than any individual weak classifier
- A decision tree is constructed using pre-classified training data
Decision Tree/Forest for Human Motion Recognition

- To implement a multi-class classifier, a collection of decision trees is usually used.
- The collection of decision trees are referred to as a decision forest.
- To reduce the correlation among the trees in a decision forest, a random subset of the features is selected at each split during the learning process.
  - This method is referred to as randomized decision forest, or randomized forest for short.

![Decision Tree Diagram]

![Decision Forest Diagram]
Decision Tree/Forest for Human Motion Recognition: Example Research

- Decision forest was used to identify gestures in realtime
  - A gesture was modeled as a sequence of key poses
  - During training, a decision forest is constructed based on key poses
  - Each path from a leaf node to the root represents a gesture
  - Gesture recognition is reduced to a simple searching problem based on the decision forest

- Randomized decision forest was used in fall detection
  - Aim: to recognize skeleton shape deformation caused by falling
  - Recognition accuracy is low due to changes in orientation of the body during movement
Adaboost for Human Motion Recognition

- Adaboost refers to a meta-algorithm for machine learning called Adaptive Boosting
  - Works with a set of lower level classifiers, and selects the most optimal ones that lead to a more accurate classification.

- The learning step of Adaboost is not to fit unknown parameters for a model, but instead, to find the best lower level classifiers
  - Hence, weak classifiers such as decision dumps (i.e. 1-level decision tree) can be used with Adaboost to form a strong classifier that produces highly accurate classification.

- A benefit for using Adaboost is that it can be used to facilitate knowledge discovery in that the user can see which lower level classifiers are most appropriate for each gesture

- Adaboost requires high quality training data to achieve good classification accuracy

- Adaboost was used to provide categorical gesture recognition in Microsoft Kinect v2 SDK
  - Complex gesture must be decomposed into a set of simple actions
Regression-based Method for Human Motion Recognition

- Kernel regression: maps an input variable to an output value by averaging the outputs of a kernel function based on a set of predefined set of data and the input variable.

- Least squares regression: aims to fit parameters for a linear or nonlinear function with minimum squared errors.

- Randomized decision forest regression (or regression forest):
  - Output label to be associated with an input data is continuous.
  - Training labels are continuous.
  - Used for non-linear regression of dependent variables given independent input.
Regression-based Method for Human Motion Recognition: Example Research

- **Kernel regression:**
  - The feature set used in the study includes the distances of the joints with respect to the spine joint, the displacements vectors of the joints with respect to the spine, and those with respect to the parent joint.
  - A gesture is defined as a sequence of control poses. Via principal component analysis (PCA), the set of feature vectors together with their classification are mapped to a one-dimensional signal.
  - Given a test feature vector of an unknown gesture, the kernel regression mapping is used to produce a one-dimensional value with the trained data.
  - This value predicts the state of the gesture.
  - The category of the unknown gesture is obtained via an arg max operation on a Gaussian kernel with respect to the unknown gesture and each of the labeled gesture in the training set.
Regression-based Method for Human Motion Recognition: Example Research

- Least squares regression:
  - Build a product manifold representation based on the Kinect depth data, where a gesture would be located as a point on the product manifold
  - Least squares regression is used to produce a smooth decision boundary for classification
  - A gesture has a unique underlying geometry
  - A main advantage of this approach is that it works for a small training dataset

- Randomized decision forest regression (or regression forest)
  - Used in Kinect v2 SDK to determine the progress of a gesture
  - Meaningful only when the current gesture has already been classified
Outline

- Introduction to the Kinect technology
- Kinect applications
- Human motion tracking with Kinect
- Human motion recognition with Kinect
- Publicly available dataset
Publicly Available Kinect Datasets

- These datasets may be very valuable resources for other researchers to carry out additional computer vision and motion analysis work.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dataset Webpage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor scenes(^{107,108})</td>
<td><a href="http://cs.nyu.edu/">http://cs.nyu.edu/</a> silberman/datasets/</td>
</tr>
<tr>
<td>Human actions(^{64,132,133,141})</td>
<td><a href="http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/">http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/</a></td>
</tr>
<tr>
<td>Hand gestures(^72)</td>
<td><a href="http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm">http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm</a></td>
</tr>
<tr>
<td>Object tracking(^{110})</td>
<td><a href="http://tracking.cs.princeton.edu/dataset.html">http://tracking.cs.princeton.edu/dataset.html</a></td>
</tr>
<tr>
<td>3D reconstruction(^{139})</td>
<td><a href="http://sun3d.cs.princeton.edu/">http://sun3d.cs.princeton.edu/</a></td>
</tr>
<tr>
<td>Category modeling(^{145})</td>
<td><a href="http://shiba.iis.u-tokyo.ac.jp/song/?page_id=343">http://shiba.iis.u-tokyo.ac.jp/song/?page_id=343</a></td>
</tr>
<tr>
<td>3D scans(^{30})</td>
<td><a href="http://vcl.iti.gr/3d-scans/">http://vcl.iti.gr/3d-scans/</a></td>
</tr>
<tr>
<td>Human actions(^{138})</td>
<td><a href="http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html">http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html</a></td>
</tr>
<tr>
<td>Scene in hall way(^{73,111})</td>
<td><a href="http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html">http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html</a></td>
</tr>
<tr>
<td>Various objects(^{53,54})</td>
<td><a href="http://kinectdata.com/">http://kinectdata.com/</a></td>
</tr>
</tbody>
</table>