Kinect Identity: Technology and Experience

Tommer Leyvand, Casey Meekhof, Yi-Chen Wei, Jian Sun, and Baining Guo,
Microsoft

Kinect Identity, a key component of Microsoft's Kinect for the Xbox 360, combines multiple technologies and careful user interaction design to achieve the goal of recognizing and tracking player identity.

Controller-less immersion has been the Holy Grail of game designers and developers for many years now. One of the primary challenges is how to seamlessly track and successfully recognize an individual's identity during game play to ensure a smooth and natural user interface.

So far, no perfect solution exists. Nevertheless, the identity technology behind Microsoft's Kinect for the Xbox 360 provides enhanced gaming and entertainment experiences by combining multiple technologies based on the use of RGB cameras, depth-sensing, and careful user interaction design.

IDENTITY TRACKING TECHNIQUES

The Kinect system tracks identity in two ways:

- *biometric sign-in*, where the system learns a player's appearance over time and signs that person in when he or she is in view; and
- *session*, in which the system remembers who's who during a particular game session, for example, player 1 versus player 2, along with their scores.

To maximize the chances of creating a successful identity tracking system, Kinect's developers experimented with a set of independent identification technologies, selecting a set that, when combined, created a complete picture. Each of the selected technologies needed to be robust (at least for short-term changes), non-CPU and memory intensive, and as independent as possible from the others.

The final set consisted of three techniques: face recognition, clothing color tracking, and height estimation.

We can break down the most important of these—facial recognition—into three subtasks:

- determining the face's location and size;
- aligning the face to "normal" coordinates, that is, head straight up, facing the camera; and
- extracting the facial signature to capture the face's microstructure.

Adding clothing color and the player's height to facial recognition technology fuses multiple characteristics to form identity recognition.

At runtime, the game asks the system to remember a new player, so the system gathers *prints*, or signatures, for each of the unidentified skeleton's characteristics—face, clothing, height—each of which provides a response of "positive" (for example, shirt color matches the known print), "negative" (stored shirt color is blue but unidentified shirt color is green), or "unknown" (too close to call for this particular characteristic).

When the game attempts to determine whether a new skeleton is already known, the system runs through all existing candidates and produces a "truth table" that gives a recommendation for each characteristic. In the example shown in Table 1, when the face characteristic is compared, #1, #3,

 Published by the IEEE Computer Society

Table 1. A truth table for identifying a new player in a game.

Characteristic	Enrolled ID #1	Enrolled ID #2	Enrolled ID #3	Enrolled ID #4
Face	Positive	Negative	Positive	Unknown
Clothing color	Unknown	Unknown	Negative	Negative
Height	Positive	Positive	Unknown	Unknown

and #4 are candidates. However, since #1 and #3 are both positive, they are treated as unknowns. Comparing color eliminates #3 and #4 as options because of the negative responses. Although #1 is left as an option, the process continues because a positive response is still missing. Finally, when height is compared, #1 has a positive response. Since it is the only positive response in the candidate set, a successful identification has been made.

For the system to successfully identify the currently unidentified skeleton as a previously enrolled entity, the following must hold true:

- At least one positive response must be returned, and no negative responses are allowed (except for a very strong face recognition match).
- Starting from the face, then clothing color, and finally height, only one candidate can be positive; in the case of multiple matches, the system deduces that the match can't be fully trusted, so it treats it as unknown.

The system then processes the results of the truth table to produce the final result. Interestingly, some characteristics, such as height, are excellent for rejecting a clear mismatch, but by themselves, these characteristics aren't accurate enough to accept a match: many people are the same height, so height by itself isn't enough to identify a person.

CHANGES OVER TIME

Of course, identification technology also must be able to adapt to changes in physical appearance over time. Session identity has a short time frame, and players don't tend to change their clothes, rearrange their hairstyle, grow a beard, or decide to switch between wearing contacts or glasses in the middle of a game, but they do change their facial expressions, strike different poses, or change the lighting in that span. Biometric identity is even more challenging because appearance will almost certainly change across different gaming sessions (players are likely to wear different clothes, perhaps someone got a haircut between sessions), as will lighting conditions (playing in the afternoon or at night).

To work around these issues, the system associates prints with as many environmental details as possible, such as where they were captured and what the lighting levels were like. The system then takes these into account when matching a new player against a set of known players. For biometric sign-in identity, Kinect Identity relies only on facial recognition. It asks players to move around the play space to capture the various local environmental details and then reruns the tool to capture more data if the players significantly alter their appearance or the lighting changes.

As part of remembering a person and trying to recognize a new skeleton, the system captures a series of frames to procure more of the possible variations, usually when the conditions are suitable or, better still, optimal, for recognition. For example, it's better to skip frames when the player isn't facing the sensor or when the face is occluded. Kinect also gives players feedback and an opportunity to fix the capture conditions, which greatly improves the speed and accuracy of identification.

Naturally, players might take off a layer of clothing during a single gaming session, which will produce a false negative if the new layer of clothing is a different color than the system previously matched with that player. If the returned response is always to reject a clothing color mismatch, then a false negative will always be produced when someone takes off a piece of clothing during a gaming session. However, never rejecting in this scenario would discard a lot of valuable information. This is another reason why relying on other characteristics is so crucial to identity in Kinect: face and height will provide a positive response and override the clothing color mismatch (under certain circumstances).

How does Kinect differentiate between identical twins? The truth table has a better chance of succeeding if it knows that it's dealing with two people who look exactly the same as opposed to knowing only one twin and matching the other incorrectly. If Kinect sees them concurrently, or if they have two separate Kinect identity profiles, the system stands a better chance of telling the two apart. This is further enhanced during a single session, when clothing differences can help differentiate between the two. But keep in mind that if a human has a difficult time telling identical twins apart, so will Kinect!

CHALLENGES

One of the most challenging aspects of developing Kinect Identity involved accuracy—both measuring and regressing—when making changes to the code or algorithm. Accuracy is extremely important because the environmental and personal conditions in a working environment aren't representative of the customer base.

Kinect's developers focused on two requirements for measuring accuracy: the data should be representative of real-world environments, and they needed a lot of it. For the latter requirement, developers used various beta programs to collect as much data as possible: the data-capturing tool itself, tools for tagging the ground-truth, and testing tools to train the algorithm. To make things even more challenging, some changes that directly affected the image frame (camera settings such as exposure or zoom) invalidated the old dataset and required gathering new data with updated settings.

Developing identification technology is a difficult task, but implementing a good user experience around it is equally, if not more, challenging. How do you design an experience around a system that's never quite sure if it's right? The best way to do so is to design how everything *should* work, assuming identity doesn't exist, and then look for shortcuts—areas that can be sped up using identity.

The "wow" factor is very important for Kinect, so its developers carefully considered the options for handling positive results (the system believes it has matched the skeleton to a known user), negative results (the system doesn't recognize the person), and the failed operation (the system is unable to complete). Because the positive result has the greatest potential to wow, developers put the most emphasis on it by using a "trust but verify" approach: instead of asking users to confirm a positive, the system shows them the result with the assumption that it's correct, and the player has the option to correct it.

To minimize damage due to trusting positive results, Kinect's developers tuned the system to reduce false positives as much as possible. In case of a mistake, the system lets players correct their identity or run through the Kinect Identity process to reduce the chances of future misidentifications. A player who wants to play as another player has two options:

- *taking control*—a player steps in and starts controlling whatever is going on; the new player isn't required to have a recognized identity, and anything that happens is attributed to the active profile; and
- *assuming an identity*—the new player tells the system to recognize him or her as the old player; the player assumes the selected identity.

The challenges include deciding when to use which approach and communicating to the player which approach is currently in use. The key is clear and consistent feedback. A common pattern that tends to work is determining identity prior to starting game play to review the active profiles and then letting users jump in and take control without changing their identity. An important decision in Kinect's development was to treat profile selection as an identity operation, meaning that whenever a user selects a profile in the sign-in dialog, the player's identity is associated with that profile for the session.

As with any system, it isn't just one or two parts that make Kinect great, but rather the fusion of a variety of technologies, techniques, methodologies, and ideas into one design. It took the collaboration of researchers, developers, and product managers from all over the globe to get Kinect as close as any researchers have come to developing a fully controller-free gaming system. C

Acknowledgments

We are grateful to Matt Callcut for his writing assistance. Kinect Identity was developed by the Xbox Kinect Identity team.

Tommer Leyvand *is a principle software development lead for the NUI Platform, Xbox Platform R&D. Contact him at tommerl@microsoft.com.*

Casey Meekhof *is a senior program manager for the NUI platform, Xbox Platform R&D. Contact him at cmeekhof@microsoft.com.*

Yi-Chen Wei *is an associate researcher in computer vision at Microsoft Research Asia. Contact him at yichenw@microsoft.com.*

Jian Sun *is a senior researcher in computer vision at Microsoft Research Asia. Contact him at jiansun@microsoft.com.*

Baining Guo *is a principal researcher at Microsoft Research Asia. Contact him at bainguo@microsoft.com.*