# Game Design Principles for the Intel® Play™ Me2Cam* Virtual Game System

Herman D'Hooge, Connected Products Division, Intel Corp.
Michael Goldsmith, Intel Labs, Intel Corp.

Index words: Consumer Products, Extended PC, Smart Toys, Computer Vision, Machine Vision, Video as Input, Gesture Recognition, Motion Detection, Video Image Segmentation, Virtual Blue Screening, Natural Interfaces, Human Computer Interfaces, Head Tracking, Hand Tracking, Usability

## ABSTRACT

Despite the long history of computer vision in academic and industry research circles, few real-world applications that use computer vision technology have actually been developed, let alone found their way into the homes of consumers. With the advent of fast processing, affordable and easy-to-install and use PC cameras, and advances in video-processing algorithms, the underlying technology has become sufficiently capable to enable certain classes of applications on average home PCs.

Natural interfaces have long been said to be the holy grail of computing that will revolutionize how people interact with computers. We've all heard claims that the keyboard and mouse will soon be a thing of the past, with people instead interacting with computing via speech and gesture. Whereas fulfilling these claims requires the accuracy of the technology to improve substantially, they usually ignore the more fundamental question of whether or not it is really desirable for people to use speech or gesture in the first place or under what circumstances it makes sense to do so.

This paper explores the *usage*–not the technology implementation–of computer vision in one commercial product developed jointly by Intel and Mattel. The product is the Intel® Play™ Me2Cam* Virtual Game System, designed for children aged four to eight. Informed by actual human behavior, the nature and limitations inherent in the technology have led the game designers to formulate a set of specific design rules that have guided the design of the application. This paper explores these design principles.

## THE INTEL® PLAY™ ME2CAM* VIRTUAL GAME SYSTEM

In October 1999, Mattel launched the Intel® Play™ Me2Cam* Virtual Game System, one of the first, if not *the* first, commercially available PC application products targeted for consumers that was built using computer vision technologies. The retail product includes a USB-connected PC video camera (the Me2Cam) with Windows* device drivers and the Virtual Game software, which consists of five computer games and some related activities. Once the application is launched, the player's physical motions control all activities of the software as observed by the Me2Cam camera, as detected by the computer vision software technology, and as interpreted by the specific application.

Like all tethered PC cameras, the Me2Cam camera is typically placed permanently on the computer's monitor facing towards the person sitting, or in this case standing and moving, in front of the computer monitor where the game activity is observed.

Throughout all activities, the player sees a mirror video image of himself or herself immersed in a virtual cartoon-like graphical world. Interaction with this world is through moving one's body or by touching objects in this virtual environment with either hands or the head. The



Intel Play and Me2Cam are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

virtual world responds to the player's actions in intuitive and interesting ways.

The five games are modeled and reached as physical places. The player can move from place to place on *Main Street* (Figure 1) by leaning slightly off-center to the left or right. This has the effect of moving the player either left or right thereby passing various places, just as one passes houses while walking on a street. An activity is selected and initiated by the player standing still (moving one's body back to center) for a few seconds in front of the place representing the desired activity.


**Figure 1: Main Street**

In the *Bubble Mania* game (Figure 2), the player finds himself standing in front of a giant bubble-making machine. The machine shoots bubbles into the room that come down in a slow-moving swirl around the player's body. If the player touches a bubble with a hand, the bubble pops and the player scores points. The objective of the game is to score as many points as possible before time runs out. To make the basic game more interesting, a number of bubbles with special behaviors are introduced as the game progresses. Some of these have desirable behavior whereas others are better avoided. For example, if a player pops a bubble with a red cross on it, the time clock will be replenished. However, if a player pops a jail bubble, the player's video image will be taken and shrunk, so it fits within the bubble where it will remain for several seconds while the bubble floats around, thereby wasting valuable time. If a player pops a firecracker bubble, the player's video image is broken into multiple pieces and scattered across the screen, again wasting time. Finally, if a player hits a bubble with his or her head, the bubble will bounce instead of pop, thereby providing the player a mechanism to move trouble bubbles out of the way.


**Figure 2: Bubble Mania game**

The *Snow Surfin'* game (Figure 3) places the player on top of a mountain on a surfboard, sliding downhill. The slopes have obstacles such as trees, roaming animals such as penguins, bears, and raccoons, and also ski jumps. The player steers the snowboard by leaning left and right, avoiding the obstacles that reduce the player's speed and prevent the player from getting to the bottom of the mountain before time runs out.


**Figure 3: Snow Surfin' game**

*Club Tune* (Figure 4) is all about movement and dance. The speed and intensity of a virtual band of musicians on stage is controlled by the amount of movement generated by the player. Stand still and the band will get bored and go to sleep. The player can chose from among several bands and types of music; the player also has a drum pad on the left for adding special drum effects during the activity.


**Figure 4: Club Tune Dance Hall**

The *Fun Zone* (Figure 5) activity is not a game with a specific goal. It moves the player's image through a series of special effects, which far surpass those of the familiar fun house of mirrors found in carnivals, shrinking, stretching, morphing, and transforming the player's video image as it passes through the stations on the way.

**Figure 5: Fun Zone activity**

Finally, for a true aerobic workout, there is *Pinball* (Figure 6). The virtual world is an actual pinball machine. The player's video image is shrunk in size and replicated five times as the actual bumpers and flippers of the pinball machine. As with regular pinball, points are scored by keeping the ball bouncing off various elements of the machine as long as possible. In this computer game, this is accomplished by hitting the ball with any body part thereby simulating the physics of the moving pinball.


**Figure 6: Pinball game**

## VIDEO AS INPUT TECHNOLOGIES

All activities in the product make use of a small number of computer vision technology components, sometimes referred to as *video as input* technologies. The Intel® Play™ Me2Cam* Computer Video Camera just captures and streams live video to the computer. This video stream contains everything that is in the field of view of the camera: the player's body, a section of a child's room with decorated walls, shelves with books and toys, etc.

To immerse the player's video image inside a virtual environment requires that the background be removed from the source video stream. The process or technology for determining which parts of a video stream are part of the foreground image and which are part of the background is referred to as *foreground-background segmentation*. For the purposes of the virtual game system, the foreground image is the player, and the background image is the image of the room–in other words, everything else.

The knowledge of which pixels in a video frame belong to the foreground and which belong to the background, then, allows the original video stream to be modified. This is done by removing all image elements that are considered to be part of the background and eventually substituting them with other images or graphics. This is often referred to as *virtual blue screening*. Instead of making the foreground/background determination based on a certain color in the background image (chroma keying), in the case of the Me2Cam, the determination must be made differently since the background can be arbitrary.

---

Intel Play and Me2Cam are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

In the *Bubble Mania* game, the behavior of a bubble is different depending on whether the bubble is touched by the player's head or a hand. The application must be able to determine the *x-y* location of the player's head and hands in order for it to be able to detect whether the head or the hand is touching the bubble; with this knowledge, it activates the appropriate behavior. A simple heuristic model of the human upper body is used by the computer vision *head-tracking* and *hand-tracking* algorithms to estimate the position of the player's head, left hand, and right hand.

In *Club Tune*, the amount of motion in the video stream is what drives the virtual world's behavior. The *motion-detection* algorithm provides a measure of the amount of motion in the video stream at a given point in time.

All applications in the Virtual Game System are built on these three technologies: (1) foreground-background segmentation, (2) head tracking and left/right hand tracking, and (3) motion detection.

This paper does not go into detailed descriptions of these algorithms or explore what makes one implementation superior. Instead, we focus on the design of the games and on the applications that use the technology.

## DESIGNED FOR USE IN REAL HOMES

The Intel® Play™ Me2Cam* Computer Video Camera is designed to be used by *real* people in their *real* homes and in whatever location and lighting conditions exist around their computers. Requiring consumers to move things around in their homes too much in order to use this product isn't realistic.

From an application design viewpoint, this adds requirements dictated by the *context of use* of the product and has significant implications for the product's design and engineering.

With the camera placed on top of the computer monitor, the source video stream is whatever happens to be in the field of view of the camera. Personal computers are located in a diverse range of home offices, children's rooms, or in the corners of family rooms. The background for the camera images can be made up of virtually anything, from white walls, to cluttered shelves with books and toys, to open rooms. This is why foreground-background segmentation must perform robustly with virtually any type of background.

The foreground-background segmentation includes an initialization phase each time the application is launched. During this phase, the player is asked to step out of the camera's field of view for a few seconds. This allows the software to record and register the background. After the initialization phase, the player can step back into the camera's field of view and play can begin. Segmentation will remove all pixels from any frame in the video stream that matches the registered background. What's left over is a video stream of just the foreground–in this case, the person playing the game.

To the segmentation algorithm, anything that does not match the registered background is considered foreground and displayed as if it is the person's video image; anything that does match the registered background is considered background and is removed. That means that anything in the image that changes over time relative to the background that was registered during initialization will be visible on screen. Flickering televisions or computer screens, other people or pets walking by, curtains waving in front of an open window, or clocks, will all become part of the foreground and may cause undesirable interference in the control of a specific game activity.

The set-up program of the Me2Cam helps the player in adjusting the play environment by identifying sources of motion in the background and suggesting actions for eliminating them and improving the game play experience.

Any sudden and dramatic change in the ambient room lighting on the background, such as someone turning a light on or off or clouds blocking/unblocking sunlight, will be perceived as different from the registered background scene and may cause the background to suddenly become visible. Also, bumping the camera, so that there is a slight shift in what is in the camera's field of view, can result in the whole camera image being interpreted as foreground.

These conditions do happen. If they happen frequently enough, they can easily become a nuisance and render the experience worthless. The application therefore has to deal with them and devise mechanisms to recover on the fly with minimal interruption to the actual game play. If foreground-background segmentation starts failing in the middle of game play (as observed by the player), he or she can step out of the camera's field of view for a few seconds. The computer vision recovery algorithm will then detect the lack of any motion and register the image as the new background from there on.

Some conditions are more difficult to design around. If the distance between the player and the background is short and lighting is frontal, harsh shadows of the player may be cast onto the background. These shadows will move as the player moves and appear as foreground.

---

Intel Play and Me2Cam are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Wearing a white shirt in front of a white wall may cause the player's image to match the registered background. If that happens, then the result is that the player's video image may have "holes" in the body where the background shows through. In the set-up program, players are advised to change clothing to increase foreground-background color contrast, which is more user friendly than requesting the player repaint the room.

When designing the Me2Cam, we envisioned localizing it for a number of countries and geographies outside of the US. Electrical power distribution in the majority of the target countries is either 50Hz or 60Hz (or both, like in parts of Japan). This is significant in that fluorescent light output tends to pulsate with the electrical power frequency. When captured with an imaging sensor, this causes horizontal banding in the image that may slowly scroll up or down the camera video image. Algorithms exist that eliminate this banding from the camera's image, but in order for these algorithms to work, the frequency needs to be known. Interestingly enough, there is no robust way to automatically determine the electrical frequency, so the user installing the product is being asked this obscure question. Again, this is an artifact of developing products for the real world.

Finally, studies done in people's actual homes have shown that the area around the computer in the home is dimly lit. When it's too dark, it will be difficult to make out details in the player's face. The set-up program again suggests adjustments to the room light level for an optimal play experience.

## ABOUT PC CAMERAS

Most high-resolution, high-frame-rate PC video cameras, common in video conferencing or used as webcams, use compression to reduce the bandwidth demands for moving video bits into the computer. The video stream is then decompressed by software running on the host PC. Compression/decompression introduces both latency (buffering, time to compress, time to decompress) and compression artifacts that may make the decompressed image appear noisy.

In a game where your physical motion controls the game behavior and the player uses visual on-screen feedback (closing the feedback loop), latency must be kept to a minimum. Furthermore, any noise or artifacts in the image used for segmentation may show up as foreground motion, potentially making it difficult to segment out static background since everything may appear to be moving.

For these reasons, the Intel® Play™ Me2Cam* Computer Video Camera uses both low resolution (120x180 pixels) and no compression. It operates at a high enough frame rate and a low enough end-to-end latency to make the games playable and responsive.

Most video conferencing cameras provide automatic and continuous adjustment of the camera's controls such as overall image brightness and white balance. If the player moves in front of the camera, the overall light level/brightness of the image is affected. If the auto adjustments are on, the camera will try to adjust the settings, thereby affecting the background's brightness. The adjustment will cause the background over time to deviate from the registered background, and it will show through as foreground. Once the game starts, auto camera settings must, therefore, be disabled.

Even with technology and precautionary measures in place to handle these real-world conditions, imaging limitations and computer vision technology are far from perfect. The next section takes a look at how the application or play pattern was designed such that these known technology limitations do not become application weaknesses. This art is what is referred to as play patterns or application design, and it is the field of expertise of interaction designers and product designers or toy designers. These applications require one to take a fundamentally human-centered viewpoint of the experience that's being created as opposed to a technology-centered one.

## CHILDREN AGED FOUR TO EIGHT

The activities included in the product have been designed for use by children aged four to eight who are standing up and moving in front of the camera. These are small humans with limited computer literacy skills. Sure, they may be able to use a computer mouse, but mostly they just want to play and have fun.

These children have little patience for anything that doesn't work or requires a long learning curve to get started. They want instant gratification, but at the same time don't want the games to quickly become boring. Most cannot read and would not understand error messages, even if reported verbally. Further, they need to be verbally prompted through the program startup.

Finally, these children have reflexes and behaviors that are different from those of an adult and that are, in many ways, much more straightforward and logical.

---

## Game Design Principles

While the following principles may seem almost trivial, it is essential that the activity designer fully internalize what they are in order to design activities that deliver well within the constraints of the interaction medium. The rational/analysis is provided first, followed by a statement of the principles it supports.

The games are all controlled by the player's physical position or motion, and also by showing a real-time mirror video image on screen as captured by the camera and segmented by the foreground-background segmentation. For this to be effective at all times, the player has to stay within the field of view of the camera. A wider camera field of view picks up a broader scene that would allow the player to move a greater distance while still in the camera's field of view. However, a larger field of view does result in a proportionally smaller image of the player on the screen. This image could be digitally enlarged but that reduces the quality and the magic of recognizing one's personal features in the image.

While some left-right movement is possible, too much movement and the player strays outside of the camera's field of view or is too far removed from the screen to be able to see. That implies, for example, that "walking" past objects, as a metaphor for controlling on-screen scrolling, is not natural. The player just can't keep walking or very soon she'll be out of the picture. The player essentially has to remain in the same spot and merely lean left or right, or at best move a small step sideways in either direction. This leads to the first principle, P1.

**P1: The Player's Left/Right Movement is Very Limited**

Most computer monitors in the home are 15 inches or 17 inches in size. Being able to see oneself with enough detail on the screen from where one is standing is an essential aspect of this game. Moving too close to the camera will cause the player to be only able to see her forehead; moving too far from the screen will cause the player not to be able to see the screen or would cause the player to bump into a wall or furniture. Also, as the player moves forward or backward, the ratio of the player's image size on screen relative to the size of on-screen objects in the virtual world will change. This both destroys the designer's vision of the virtual world, and it also has the effect of having the player's image obscure areas of the virtual environment. Player motion is therefore to be confined to a plane at a fixed distance from the screen and camera. Hence, we have the second principle, P2.

**P2: The Player's Distance From the Camera is Essentially Fixed**

The camera's field of view is only able to capture the player's upper body, from the waist up. This restriction is due to the small size of the captured image. Capturing more of the player's body would result in the player's face being represented in too few pixels. That rules out the use of the player's legs or feet as a way to control the activities. It also implies that hands can be tracked only when they are not in front of the body.

The *Snow Surfin'* game, for example, places the player's upper body on a computer graphic bottom body that sits atop the snowboard to simulate a whole body. This also rules out games derived from sports such as soccer where foot action is key. This brings us to the third principle, P3.

**P3: Only the Player's Upper Torso is in the Game**

The player has to look directly at the computer screen to see what's going on. With the camera on top of the monitor, a player will always see herself looking forward. That has implications for the kinds of activities that make sense. Having a conversation with a cartoon character isn't natural if the character is depicted beside you.

For example, in the *Snow Surfin'* game, the player's on-screen representation is always moving towards the player. To be more exact, the player isn't really moving; instead the snow landscape around the player is scrolling. While it may seem more intuitive to be facing the other way, simulating a true first-person view for this game, children liked seeing their faces on screen and didn't mind at all that they were surfing in that direction. This brings us to the fourth principle, P4.

**P4: The Player Always Faces Forward**

The child will see his or her own image on screen. For many children the fact that they can see themselves on the screen in the game is a large portion of the appeal of the experience. The image acts as a mirror to the player since that is what all of us are intuitively expecting the behavior to be.

The activities are designed with a clear reason why the player sees himself in the game. This cannot be a keyboard and mouse replacement. If an activity is easier performed with a mouse, then using one's body motion instead can quickly become a frustrating experience. We, therefore, defined activities or worlds that fundamentally require a physical and mechanical behavior: i.e., the hand and body action are the natural way to interact. Popping falling bubbles, hitting a pinball, and leaning to the left or right to steer a snowboard are all physical behaviors. This leads to the following two related principles, P5 and P6.

**P5: Body Motion and Vision Cannot be a Keyboard or Mouse Replacement**

The state of the art in computer vision is approximately where speech recognition was a decade or more ago. The technology is not 100% robust. Attempts to replace the keyboard with speech, even with today's speech engines, are still largely unsatisfying, except in special cases. In both speech recognition and computer vision, there must be a compelling need for the input modality. In speech, this is frequently a hands-free/eyes-free requirement. For computer vision in games, the motivation is fun; and that fun is what allows the player to tolerate, and even enjoy computer vision.

**P6: There Must be a Valid Reason for the Player's Image to be in the Game**

The resolution or precision of a computer mouse or typical pointing device as operated by a typical player is about a few screen pixels squared. That makes it possible for mouse-controlled computer applications to have large numbers of controls packed on a single computer screen.

In contrast, the resolution achieved with hand or head interaction with objects in a virtual world is very low. The relatively large on-screen hand size, user's imprecision of hand orientation and movement, non-zero latency, and imprecision in the vision technology, all contribute to making this a low-bandwidth user interface. The minimum size of on-screen objects being controlled is roughly of the same order of magnitude as the hand controlling it. Thus, we have the seventh principle, P7.

**P7: The User Interface Resolution is Very Low**

In the design of the Intel® Play™ Me2Cam* Virtual Game System's activities, the resolution of the user interface implies that only a small number of distinct active controls can exist on any given screen and that they are spatially separated. For example, the floating bubbles are the active controls in the *Bubble Mania* game. Intersecting one's hand with the bubble activates a bubble's popping behavior. The hand is a more-or-less blob-shaped object with an approximate *x-y* center position, and a bubble is roughly the same. When both *x-y* positions are within some threshold, a collision occurs, activating the bubble.

Although popping bubbles for points in a child's game is far from mission critical, the illusion of the virtual world has to be sufficiently close to physical world bubble popping for this to be believable and not become frustrating.

It is a fact of human anatomy that our hands connect to our arms and our arms connect to our bodies at the shoulders. Human heads are fairly fixed to the middle and top of the human upper torso. The spatial range of where head, hands, and body can be moved in the virtual world on the computer screen is severely limited. Game design must take this into account when placing objects in the virtual world.

The most logical location of the objects that can be activated is in a semi circle around the player's body and above the head. They must be close enough so they can be reached; yet, they can't intersect with the body.

On the other hand, controls that may have undesired effects if activated at the wrong time should be harder to reach or activate. Exiting an activity requires standing motionless on the exit control for a few seconds– something that is unlikely to happen accidentally in an energetic game. This brings us to the eighth principle, P8.

**P8: Common User Interface Objects Must be Easy to Reach; Infrequently Accessed Controls Must Require a More Deliberate Action to Activate**

It has to be instantly clear to the player what is expected of her and how she should interact with the environment. While this is good user-interaction design in general, it is especially true in the case of a game for young children.

Game designers deliberately rejected the idea of using specific gestures for control. Gestures, no matter how basic, would have to be learned by the player and be such that they would be unlikely to occur during normal game play. That doesn't match with the need for instant gratification. Also, some game play can become pretty involved, as when trying to achieve a new high score, so accidentally activating the "exit" gesture command would be highly undesirable. And this brings us to the ninth principle, P9.

**P9: Intuitive Interface; No Learned Gestures**

Having each virtual world be based on a familiar physical world setting creates instant familiarity with what seems fun and logical activities within that world. When you see bubbles floating around you, your normal reaction is to pop the bubbles with your hands. When you see yourself

---

on top of a snowboard that starts sliding downhill, your reaction is to take control of the snowboard and try to avoid obstacles by leaning left and right. When you see yourself inside a pinball game, the thing to do is to keep hitting the ball into the game when it comes within reach. Just to be sure though, at the beginning of each game, a two-sentence voice announcement tells the player what to do.

Inter-game navigation uses the same physical world metaphors as the games themselves. For example, exiting a specific game is done by standing in front of a clearly marked exit sign that is out of the way of regular play.

Having a virtual world based on real-world situations creates instant familiarity. Since human body motion is subject to the laws of physics, so should the behavior within the virtual worlds, at least to some degree. Again, the physical behavior creates familiarity and meets intuitions of how things work. Hence, we have the tenth principle, P10.

**P10: Virtual Worlds have Familiar Physical Behaviors**

It would, however, not fully exploit the potential of a PC-based gaming system if the virtual environment merely simulated the physical world to the letter–especially in a game. Certain actions can trigger behaviors that are simply impossible in the real world but contribute tremendously to the magic of the experience. Being shrunk and captured inside a bubble, in the *Bubble Mania* game, and seeing yourself float away is magic; moreover, it exploits the full potential of the designer's imagination and the medium. Furthermore, it adds an element of discovery for the player who is eager to find out what will happen if he does either this or that. This brings us to the eleventh principle, P11.

**P11: Virtual Worlds have Surprising Behaviors**

The amount of floor and airspace around the home computer can cause confusion for the computer vision technology. The games were designed for a single player in the camera's view at any one time. Moreover, the vision algorithm is optimized for a single head and, at most, two hands. If more than one player appears in the camera's field of vision, the computer will randomly alternate amongst the visible body parts. This can cause the game to malfunction in various ways, such as game objects not correctly interacting with the player or the player's position randomly jumping around the screen. We, therefore, have the twelfth principle, P12.

**P12: Only One Player at a Time**

While some of these principles may seem obvious, enthusiastic application designers may get carried away. They might assume that computer vision interaction technology can deliver far more powerful user experiences than it can in practice. The Me2Cam game designers at the Smart Toy Lab know; they've been there. During our exploration, numerous activity ideas had to be abandoned because they ultimately violated one or more of these principles, which at the time, we had yet to discover and articulate.

## CONCLUSIONS

"*It's the application, stupid*" was a popular catch phrase around Intel's technology labs for a while. It was used to remind technologists and engineers that technology itself is not important; what ultimately matters is what the technology is used for. This paper provides an application analysis case study, from the application designer's, not the implementer's viewpoint for the Intel® Play™ Me2Cam* Virtual Game System.

It is the application designer's role to understand the strengths and weaknesses of the underlying technologies and to take an unwavering human-centered perspective in defining the experience. In the mind of a skilled application designer, a well-designed application cannot promise and deliver more than what the technology can provide.

Computer vision will never replace today's user interface, definitely not for current applications. However, when used appropriately, it does have a place. It can either augment existing interfaces or, as this paper illustrates, provide different experiences ("new uses, new users") altogether.

## ACKNOWLEDGMENTS

The authors thank the toy designers Brett Bogar (Mattel) and Frank Brown (Mattel, now Intel), Anne Paper (Mattel), Melanie Goldstein (Mattel, now Intel), Kendal Miller, Terry Ycasas, Bill DeLeeuw, Judy Goldstein, Cory Cox, Raj Hazra, David Linn, the people from the now-defunct ePlanet: Emily Albertson, Jamie Christini, Katie Franz, and John Gomes, and many others for turning the idea of a computer vision game for children into a shipping product.

## REFERENCES

www.intelplay.com

---

http://www.intel.com/ial/home/digentertain/immerse.htm

## AUTHORS' BIOGRAPHIES

**Herman D'Hooge** manages the Design Center for Intel's Connected Products Division. Prior to that, he managed New Concept Development at the Intel Smart Toy Lab which he co-founded in early 1998. During his 20 years with Intel he has held various positions in research, development, product architecture, industry evangelism, and management in areas ranging from multi-processor computer architectures, PC system architecture, operating systems, computer security, fault tolerance, distributed systems, computer telephony integration, and new computing applications research. He holds an M.S. degree in Electrical Engineering and an M.S. degree in Computer Science, both from the University of Ghent, Belgium. His e-mail is herman.d.dhooge@intel.com.

**Michael Goldsmith** is a Staff Software Engineer in Intel Labs. He currently works on turning Video-As-Input technology into real-world applications. During his 10 years at Intel he has held various positions in development, software architecture, and management in areas ranging from operating systems, compilers, speech-based applications, PC-based video conferencing, and Video-As-Input applications. He is the creator of the Pinball activity. He holds a B.S. degree in Computer Science from the Massachusetts Institute of Technology. His e-mail is michael.a.goldsmith@intel.com.