Abstract: Methods, systems, and computer readable media for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays are disclosed. According to one aspect, a method for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays includes calculating a desired image, calculating an error image as the difference between the desired image and an image currently being perceived by a user, identifying as an error portion a portion of the error image having the largest error, updating a portion of a projected image that corresponds to the error portion, and recalculating the image currently being perceived by a user based on the updated projected image.
DESCRIPTION
MINIMAL-LATENCY TRACKING AND DISPLAY FOR MATCHING REAL
AND VIRTUAL WORLDS

PRIORITY CLAIM
This application claims the priority benefit of U.S. Provisional Patent Application Serial No. 62/075,221, filed November 4, 2014, which is incorporated by reference herein in its entirety.

GOVERNMENT INTEREST
This invention was made with government support under Grant No. IIS-1423059 awarded by the National Science Foundation. The U.S. Government has certain rights in the invention.

TECHNICAL FIELD
The subject matter described herein relates to methods and systems for providing augmented reality devices. More particularly, the subject matter described herein relates to minimal-latency tracking and display for matching real and virtual worlds in head-worn displays.

BACKGROUND
For the past two decades, researchers have demonstrated the promise of augmented reality (AR) to allow society to reach new levels of capability and efficiency in areas as diverse as medicine [22], manufacturing [5], maintenance [17], navigation [16], and telepresence [51]. Although to date this technology has been primarily confined to the lab, new advances in mobile processors, sensors and displays offer the promise of mainstream use of AR. In particular, optical see-through head mounted displays (HMDs) offering augmented overlays over one’s natural vision are beginning to become practical.

Figure 1 illustrates several optical see-through glasses models that would benefit from the systems and methods described herein. The formerly bulky designs and narrow fields of view are being superseded by high-performance devices in comfortable and compact form factors that approach
those of ordinary eyeglasses such as those shown in Figure 1. However, such systems still cannot offer augmented imagery registered to real-world objects due to a fatal flaw - latency.

Figure 1 illustrates various optical see-through glasses that can approach the form factor of ordinary eyeglasses, enabling extended use and mobile augmented reality applications. The left hand image in Figure 1 is the Google Glass display. The center image is the Lumus DK-32 display. The right image illustrates a future augmented reality glasses concept with multiple integrated tracking cameras.

Optical see-through displays offer a direct and undegraded view of the real environment that is suitable for extended use throughout one’s daily activities. However, this desirable attribute comes at a cost; unlike video see-through displays, which allow synchronization of real and virtual objects in software through a combined video signal, optical see-through designs must rely on very low display latency to keep the virtual and real aligned [62]. The latency of today’s AR systems, even those optimized for low latency, extends beyond a simple annoyance or distraction and renders many optical see-through applications unusable (see Figure 2).

Figure 2 illustrates simulation of latency for a surgery application with an augmented reality overlay of organs. The images show scalpel location and corresponding location of augmented reality overlay that should appear under the scalpel. The displacement of the augmented reality imagery from the scalpel imagery is due to the latency of the tracked head movement for a head moving at a moderate speed of 50°/sec with imagery at arms length (60 centimeters). The left image shows 100 ms latency, typical of an ordinary AR system. The middle image shows 50 ms latency, typical of an AR system designed for low latency. The right image shows 1 ms latency, the expected performance of the design described herein. 1 ms latency assumes a 60 Hz camera update rate for a 1,000 line camera and a total budget of 60 scanlines latency for camera acquisition, tracking, rendering, and display.

Unfortunately, latency occurs throughout all components of an AR system and thus is not easy to reduce [36]. Tracking cameras process data
in whole-frame intervals and apply processing to reduce the effects of rolling shutters. Modern graphics hardware uses deep pipelines to achieve high performance in exchange for increased latency, and video image scanout uses one or more frame buffers to prevent image tearing. Displays provide on-board image scaling and enhancement which introduce additional delays. The sum of these latencies typically reaches several frames.

Accordingly, in light of these disadvantages associated with conventional AR systems, there exists a need for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays.

SUMMARY

According to one aspect, the subject matter described herein includes a system for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays. The system includes a display module for displaying a projected image to a user, a rendering module for calculating a desired image, and a processing module for calculating a user-perceived image based on the projected image and a perception model. The processing module receives from the rendering module the desired image, calculates an error image as the difference between the desired image and the user-perceived image, identifies as an error portion a portion of the error image, and updates a portion of the projected image that corresponds to the error portion.

According to another aspect, the subject matter described herein includes a method for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays. The method includes calculating a user-perceived image based on a projected image and a perception model, calculating a desired image, calculating an error image as the difference between the desired image and an image currently being perceived by a user, identifying as an error portion a portion of the error image, and updating a portion of a projected image that corresponds to the error portion.

The subject matter described herein can be implemented in software in combination with hardware and/or firmware. For example, the subject matter described herein can be implemented in software executed by a
processor. In one exemplary implementation, the subject matter described herein can be implemented using a non-transitory computer readable medium having stored thereon computer executable instructions that when executed by the processor of a computer control the computer to perform steps. Exemplary computer readable media suitable for implementing the subject matter described herein include non-transitory computer-readable media, such as disk memory devices, chip memory devices, programmable logic devices, and application specific integrated circuits. In addition, a computer readable medium that implements the subject matter described herein may be located on a single device or computing platform or may be distributed across multiple devices or computing platforms.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the subject matter described herein will now be explained with reference to the accompanying drawings, wherein like reference numerals represent like parts, of which:

Figure 1 illustrates several optical see-through glasses models that would benefit from the systems and methods described herein;

Figure 2 shows the results of a simulation of latency for surgery with augmented reality overlay, showing the unacceptable latency that is characteristic of prior art approaches;

Figure 3 shows an exemplary tracking camera cluster according to an embodiment of the subject matter described herein;

Figure 4 illustrates graphs of the performance of a fast tracking algorithm according to an embodiment of the subject matter described herein;

Figure 5 is a block diagram illustrating an exemplary system for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays according to an embodiment of the subject matter described herein;

Figure 6 is a flow chart illustrating an exemplary process for minimal-latency tracking and display for matching real and virtual worlds in head-
worn displays according to an embodiment of the subject matter described herein;

Figures 7A through 7E illustrate the operation of systems and methods according to an embodiment of the subject matter described herein using a simple graphic image; and

Figures 8A through 8K illustrate the operation of systems and methods according to an embodiment of the subject matter described herein using a complex image.

DETAILED DESCRIPTION

In accordance with the subject matter disclosed herein, systems, methods, and computer readable media for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays are provided. The subject matter described herein includes a new method of reducing latency that takes advantage of the principles of bufferless input/output and "minimal-latency" processing throughout all main components of an AR system: tracking, rendering and display. Tracking will take advantage of the native line-sequential exposure (or "rolling shutter") of the inexpensive CMOS cameras found in mobile devices; each new line read out of such a camera produces a new one-line "snapshot" of the scene. When a new data line arrives from a cluster of such cameras in various orientations, it will be fed immediately to an efficient tracking algorithm to provide a new estimate of the user's pose. At the same time, a minimal-latency cascade of successively simpler and faster Tenderers and warpers will draw the augmented scene, reflecting the most recent tracking pose. Each of these components will be designed to operate flexibly at independent rates, with the goal of minimizing the time between when a tracking observation is made and when the scene data relying on that observation is displayed.

Reference will now be made in detail to exemplary embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.
Figure 2 shows the results of a simulation of latency for surgery with augmented reality overlay, showing the unacceptable latency that is characteristic of prior art approaches. As shown in Figure 2, we estimate that an end-to-end latency of \(~1\text{ms}\) is necessary to keep real and virtual objects nearly aligned under moderate head movement (507sec). Using a 60 Hz, 1000 line camera, we expect that 1 ms latency is achievable with our approach. 1 ms will allow for a total budget of 60 scanlines of latency for camera acquisition, tracking, rendering, and display.

Using this approach, one can design and build a complete AR system based on modified commodity hardware: a cluster of tracking cameras, a computing platform, and an optical see-through HMD. Such a system can operate with extremely low latency—an unprecedented 1-2 orders of magnitude below existing systems—thus removing a significant barrier to the deployment of many AR applications. Such a system can facilitate more widespread adoption of AR, allowing the public to realize the many anticipated benefits of augmented reality.

2 Minimal-latency Visual Tracking

In order for augmented reality to be convincing to the user, robust and reliable tracking of the camera (and thus user and HMD motion) is provided. While robust tracking alone is sufficient for video augmentation, targeted optical see-through systems also need very low end-to-end system latency to achieve high-quality augmentation. In environments that are not instrumented, a natural choice is the inside-out tracking with wearable (HMD-mounted) cameras. For camera-based tracking, a natural boundary on the system latency is given by the frame rate of the camera, i.e. the system latency is at least as big as the time between frames. The subject matter described herein takes advantage of the fact that rolling shutter cameras provide a high-frequency sampling of the scene on a per-image-row basis.

We use a novel high-frequency, low-latency tracking approach that leverages the rolling shutter present in modern CMOS cameras. This approach is described in detail in commonly owned International Patent
Application Serial No. PCT/US14/26349, filed on March 13, 2014, and hereby incorporated by reference in its entirety. For typical high-definition (HD) cameras, this approach can theoretically reach sampling rates of 30 kHz. In practice, sampling rates in the range of a few kHz are common. As the presented tracking is incremental in nature, we combine it with a global visual mapping approach to overcome the inherent drift of the incremental approach.

Next, we review "rolling shutter" cameras and their tracking-relevant characteristics.

2.1 The Rolling Shutter

In traditional CCD sensors (and global shutter cameras in general), all rows of pixels within a frame begin exposure at the same time. For each frame, all rows of the frame begin integration at the same time. Based upon the exposure settings, all rows conclude integration at the same time. Next, the image is read out line by line to the host device and integration begins again for the next frame. In contrast, typical CMOS image sensors commonly found in handheld devices. Instead, these sensors employ rolling shutters [47]. A rolling shutter causes each row of a frame to begin its exposure process at a slightly later time than the previous row. Hence each row of the resulting image can be considered a "snapshot" of the scene at a slightly different point in time; for moving cameras, the scene is at a different spatial position and orientation when each row is exposed. As soon as a rolling shutter has finished exposing a particular row of pixels, that row is read out into line memory [24]. By considering multiple samples from a rolling shutter image, our method leverages temporally uniform distributed appearance sampling of the scene to achieve KHz-rate tracking with low latency.

2.2 Related Work in Camera Tracking

Tracking camera motion using video is a well-established research area in computer vision [54, 55, 53] and robotics [11, 12, 52], and it is frequently leveraged by AR systems for pose estimation. There are two
groups of tracking classes: the first class of methods uses sparse salient features for camera motion estimation [55], while the second directly uses the global scene appearance.

The salient-feature-based methods deploy tracking or matching of the salient sparse features in the image to compute the camera position from the feature motion (often termed structure from motion). Alternatively, simultaneous localization and mapping (SLAM) is used to obtain the camera position through a filtering approach [111]. The feature-based methods have been shown to allow real-time tracking at frame rate through fast RANSAC [18] techniques, such as our real-time RANSAC [61]. However, these techniques require the full frame to be present at motion estimation time, so the tracking latency is at least as long as it takes to acquire one frame from the camera.

The systems and methods described herein fall in the second class of appearance-based tracking. These methods use global image appearance to formulate the tracking problem in the camera parameters [30, 26]. Murase and Nayar [48] project the appearance into a hyperspace parameterized in the illumination and object position; this requires object segmentation and illumination determination techniques. Other methods represent objects in separate linear subspaces for different illuminations [4, 25, 3] to overcome the illumination constancy requirement. However, these methods require non-scalable efforts to obtain the linear subspaces or offline learning processes [14]. Our method overcomes all of these restrictions by simultaneously tracking and sampling the local appearance manifold.

2.3 Vision-based Tracking with a Rolling Shutter

The tracking approach described herein extends our preliminary approach [69], which leverages a local linearization of the appearance manifold of the scene. Similar to our preliminary work, the presented approach performs real-time tracking even in highly complex and irregular scenes. The major drawback of our preliminary approach [69] is the sub-pixel motion restriction; our presented approach overcomes this limitation through high-frequency rolling shutter sampling.
A scene's appearance manifold is spanned by the scene appearance as captured by the images under the possible camera positions and orientations. Each image is considered to be a vector \( I \in \mathbb{R}^n \) that contains all its \( n \) pixels color values. The camera is considered to have six degrees of freedom for its motion and if it moves through the scene capturing images \( I_t \), it samples the appearance manifold of the scene under the current illumination. Given that the camera has only six degrees of freedom, the appearance manifold can at most have dimensionality six. In addition, it is clear that a mapping \( \left( \cdot \right) \) from the camera parameters \( P \) to the appearance manifold or images \( I \) exists. The mapping \( \left( \cdot \right) \) allows us to obtain the image \( I = f(P) \) from the camera parameters, if the appearance manifold of the scene is known.

It is likely that the appearance manifold of a scene is highly non-linear, requiring a large number of samples in order to obtain the appearance manifold of the scene [48, 4, 25, 3], but learning the full appearance manifold is infeasible for an AR system that is deployed in unconstrained environments. Our approach exploits a locally linear representation of the appearance manifold along with an efficient method to track the camera motion and re-learn the local appearance manifold.

To linearize the local appearance manifold, consider a camera \( P_0 \) at time \( t \), which acquires image \( I_0 \); it can be assumed that \( m \) nearby images \( I_{0...m} \) taken at time \( t \) from perturbed camera positions \( P_{1...m} \) are known. The local appearance manifold \( f(P) \) for \( P_0 \) can be linearized using the camera samples \( k = 1...m \) as follows:

\[
\begin{align*}
I_k &= I_0 + \frac{\partial f(p)}{\partial P} \left( P_k - P_0 \right) \\
\text{d}I_k &= F \text{d}P_k
\end{align*}
\]

(1)

with image \( I \) seen as \( n \)-dimensional vectors containing the pixels of the image and \( \text{d}I_k \) being the difference of image \( I_0 \) to \( I_k \) in vector form. \( P_k \) is the six-dimensional camera pose parameter vectors and \( \text{d}P \) is the difference vector between \( P_0 \) and \( P_k \). \( F \) represents the linearized appearance manifold and is equivalent for all samples. For each particular appearance sampler,
Equation (1) is underdetermined to solve for the change $dP$, but the constraints of all $m$ views can be combined to
\[
[dl_{1}, ..., dl_{m}] = F[dP_{1}, ..., dP_{m}] \quad (2)
\]
If the $m$ images provide six linearly independent constraints, Equation (2) can be solved as a least squares problem. To maximize applicability, in one embodiment the number of images is reduced to a minimum $m = 6$ to recover the camera motion of a new image $I_{0}$ by solving
\[
Fd\bar{P} = d\bar{I}_{0} \quad \text{with} \quad F = [dl_{1}, ..., dl_{m}][dP_{1}, ..., dP_{m}]^{-1} \quad (3)
\]
where $d\bar{I}_{0}$ is the difference image between image $I_{0}$ at time $t$ and the new image $I_{0}$ at time $t + 1$. This provides the incremental camera motion $d\bar{P}$ through a linear solution of Equation (3) if the $m = 6$ views span the local linearized appearance manifold $F$. One approach to obtain those six views will now be described.

In practice it is expected that $\{r\}$ will contain a single row or a few tens of rows at most. Using fractions of the image to track the differential motion enables our presented method to track with a sampling rate $S$ of
\[
S = \frac{\#(\text{rows})(\text{camera frame rate})}{\{r\}|} \quad (4)
\]
Equation (4) reveals that megapixel-sized images would allow $S$ to be in the kHz range.

In preliminary work [69], we used three auxiliary cameras offset in the x-, y- and z-directions to obtain three samples in orthogonal directions. Three rotated views can be generated by virtually rotating each camera's image around the three axes of the coordinate system. For non-degenerate scenes, these images provide the constraints needed to estimate the differential camera motion $d\bar{P}$.

Figure 3 shows an exemplary tracking camera cluster according to an embodiment of the subject matter described herein. Figure 3 shows an example of our preliminary system and the images obtained.

In particular, the center image in Figure 3 shows the tracking camera cluster, and the images that surround the center images show images captured by the camera at various camera rotations and translations.
In some cases of fast camera motion (e.g., head motion), the sub-pixel motion constraint is not upheld. In these extreme motion conditions, we exploit the observation of Forsse and Ringaby's [19] analysis of motion artifacts for rolling shutter cameras: the major source of image motion between frames is camera rotation. When estimating the motion of the camera from the set of rows \{r\}, we know the estimated motion of the previous set of rows. This motion estimate can then be used to compensate for the rotational motion by incorporating the prediction into Equation (3):

\[
F_R^{(r)} d\tilde{P} = d\tilde{I}_{R0}^{(r)}, \text{with}
\]

\[
F_R^{(r)} = \left[ (I_k - H_{R0}I_0)^{[r]}, \ldots, (I_k - H_{Rm-1}I_0)^{[r]} \right] [dP_0, \ldots, dPm - I]^{-1}
\]

Eliminating almost all rotational image motion ensures that a large range of motions fulfill the sub-pixel motion constraint, enabling our system to track in almost all practical scenarios.

**Tracking drift.** Given the incremental approach of the presented appearance-based tracking method, integration of accumulated errors will lead to drift over time. Our method only accumulates frame-to-frame error and does not accumulate the tracking errors over the multiple sets of rows \{r\} in the image. In one embodiment, the effects of frame-to-frame drift are mitigated by coupling our appearance-based tracker with a global tracker that uses traditional structure-from-motion methods to serve as an absolute reference. In one embodiment, the presented tracker may be combined with one of our previously described methods [6] at frame rate. This will suppress the frame-to-frame drift by using salient environment features to determine a global camera pose.

**Accuracy of rolling shutter tracking.** Performing the presented rolling shutter tracking technique on synthetic data yields encouraging results, with one caveat: the quality of the results is dependent upon the number of rows used to form each subimage, as illustrated in Figure 4.

Figure 4 illustrates graphs of the performance of a fast tracking algorithm according to an embodiment of the subject matter described herein. In particular, Figure 4 illustrates graphs comparing preliminary
results between the ground truth motion, tracking by taking samples per frame (using half of the rows per motion estimate), and tracking with 25 samples per frame (using one twenty-fifth of the frame's rows per motion estimate). For brevity, only the rotation estimates are shown. The translational estimates have similar performance characteristics. Figure 4 suggests that the rolling shutter tracker produces ringing at speeds of 25 samples per image. In one embodiment, a Kalman filter or similar may be implemented to smooth the tracker's estimates. Regardless, the ringing in the first several hundred measurements after correction is minimal, suggesting that sufficient accuracy at high sampling rates will only be possible if registration-based drift correction is performed at frame rate.

Degenerate cases. For the sampling and camera motion computation using full images [69], degenerate cases include far away scenes as they do not provide translational constraints; hence the solution of Equation (2) becomes under-constrained, leaving a three-dimensional family of solutions. Insufficient parallax within a scene will signal our system to adjust the camera motion estimation leveraging Equation (5) to only include 3 rotational degrees of freedom. In one embodiment, a technique to estimate the scene's parallax involves high-frequency optical flow analysis between the system's center camera and each of the translational cameras. An additional degenerate case of only our presented method will be homogenous image rows, which generally do not provide enough constraints except in special spatial configurations.

In summary, in one embodiment, the minimal-latency tracking technique includes implementation of the presented novel single-row or set-of-rows tracking method. A performance evaluation may be performed to determine practically feasible tracking rates. Such an evaluation may be performed on indoor and outdoor datasets. Drift compensation may be achieved through global mapping and integration of the correction into the AR system such as to avoid visual artifacts. Such systems may implement rotational motion compensation in the tracking, as presented in Equation (5). A mechanism which identifies parallax deficiencies within a scene and
adjusts the camera motion estimation leveraging Equation (5) may be included.

3 Realization of Computer Vision Hardware

Systems and methods according to the subject matter described herein utilize each individual row of a frame immediately after it is exposed by the rolling shutter, in order to provide real-time pose estimation by solving Equation (5).

3.1 High-level Architecture

In order to bring a high-frequency, row-wise rolling shutter motion tracker to fruition, we must stray from today's typical CMOS image sensor controller design. Related previous work ([20] - [49]) assumes that all image sensor addressing, row-wise readout, and exposure control is driven by the image sensor control board. Systems and methods described herein may use instead a lightweight control board which acts as a thin glue layer between the CMOS sensor and the host device. This control board will be responsible for exposing and retrieving rows from the CMOS sensor as rapidly as possible. To minimize the sensor-to-host latency, no image processing is performed by the camera hardware.

By taking such a simplistic approach, the system can remain compact, low-cost, and low-power, fitting the performance profile necessary for a mobile augmented reality system. In one embodiment, the system may use off-the-shelf CMOS image sensors and controllers implemented on FPGA boards. In one embodiment, a custom driver and API for the camera may be developed to enable user-defined programs to grab 'incomplete' frames from the camera. In one embodiment, an integrated controller may be developed for controlling all four cameras in a cluster to enable tracking within an FPGA.
4 Rendering and Display

4.1 Introduction

In addition to tracking latency, rendering and display are the remaining major sources of significant latency in existing AR systems. Even in the absence of tracking and rendering latency, the remaining latency induced by the display results in misregistration that is very noticeable at reasonable object movement. As we have shown in the previous section, the presented scanline-based tracking methods promise update rates of several KHz, perhaps up to 30 kHz. Rendering and display at rates comparable to those of the presented tracking methods are not possible with traditional displays or graphics processors.

Displays combine electronics, mechanics, and optics to generate, modulate, and steer light to viewers' eyes. The number of combinations of available mechanisms is staggering and continues to grow as new technologies emerge. Abstracting these mechanisms as traditional 1930s CRT televisions, i.e. serializing pixel access with regularly spaced raster updates, provides a common interface, but imposes latencies that may not be present in the underlying technologies. Ironically, random access to this serial array is provided through another abstraction, frame buffer memory, which imposes yet another layer of latency. Similarly, today's hardware (GPUs) and software (drivers and graphics APIs) are not designed for minimal latency but rather for highest possible throughput, which they achieve through a combination of parallelism and pipelining. The latter has been a source of increased latency even as it increased frame rates. Often enough the systems buffer an entire frame's worth of data at various pipeline stages. For a 60-Hz display, frame buffering alone induces minimum latency penalties of 16 ms and 33 ms at the top and bottom of the display respectively. As shown in Figure 2, delays of this (minimal) magnitude can greatly degrade the performance of AR applications. Thus, high frame rates alone are insufficient for achieving dynamic registration in AR applications; they must be accompanied by minimal total display latency, by which we mean the amount of time spent by new data passing through the rendering
and display engines until the corresponding updated photons are emitted by the display device.

We present systems and methods to de-abstract the display and its interface to expose low-level mechanics, which permit us to obtain the fastest possible control of photon production by the display. Our graphics processing architecture has a pipelined cascade of successively simpler and faster stages, with the characteristic that each stage may be updated with the most current tracking parameters. This is in contrast to the conventional pipeline, in which scene and viewing parameters are fed to the front end and trickle through the pipeline, accumulating latency at each stage.

In one embodiment, processors have parallel and/or random access to individual pixels. In one embodiment, the rendering process could access independent components of the display in order to, for instance, distribute modulation across both illumination sources and mirrors in a digital micromirror device (DMD) to simultaneously optimize for latency, power, and quality.

4.2 Minimal-latency Display Elements

Previous work [39] demonstrated high-frequency display updates (over 4,000 low-bit-depth images/sec) with projectors based on the (widely used) DMD technology. We have therefore chosen DMDs as the initial display platform.

The unique features of this readily available DMD technology inspire us to rethink the display of color information. Traditionally, DMD systems have simply used frame-sequential time-multiplexed accumulation of color during the frame time. But since in a low-latency system it is no longer acceptable to let the frame time elapse in order to allow the color data to "build up," we will investigate an approach that we consider suitable for low-latency presentation. The systems and methods according to the subject matter described herein take advantage of the color LED illumination and implement a high-frequency three-LED unit that can display any color (by controlling the intensity of the simultaneously activated LEDs) and can thus change illumination color with each single-scanline update, i.e. at a rate of
25 MHz. For each single-scanline display update, the system will access the one scanline that requires the most significant update in terms of color difference in order to show the user the imagery needed at that particular time. The system will be able to deliver such a "scanline color moment" through both exact illumination color and mirror state changes on the selected target scanline. This concept replaces the traditional "one frame at a time" notion that all current displays are based on.

While alternative technologies such as LCDs or OLEDs feature their own unique properties and require their own control mechanisms, a DMD prototype provides a level of access not readily available in most other displays. However, the general approach presented here can be extended to a wide variety of display technologies.

A rendering process for the above approach maintains an estimate of the image currently presented to the user and continually calculate and assess the difference between that image and the desired current imagery based on the latest tracker readings. In one embodiment, the method includes generation of synthetic images extended past the current field of view of the display, as far as the user can possibly move, in the time until the next complete conventional full-frame rendering becomes available (30 msec). In one embodiment, the method includes tracking the user's gaze and attempting to maintain an image of the display contribution that accumulates on the user's retina. Predicting what exactly our users will perceive and how effective or deleterious any particular display strategy will prove is a daunting task, and to our knowledge, no current or recent immersive display eyewear has explored the issues addressed by the subject matter described herein. Commercial pioneers such as the Google Glass team have likely recognized the difficulty of these issues and have refrained from promising or attempting to deliver registered AR in their first commercial offerings.

4.3 End-to-end Augmented Reality System

Integrating these ideas into a novel end-to-end framework, we present a low-total-latency optical see-through AR system, consisting of a
visual tracker as described in Section 2, as well as rendering, post-processing and display as detailed in this Section.

Figure 5 is a block diagram illustrating an exemplary system for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays according to an embodiment of the subject matter described herein. Figure 5 shows a high-level overview. A tracking module 500, at the topmost layer, delivers viewing transformations (i.e. tracking pose estimates) at up to 30 KHz. In the second layer, a rendering and processing module 502 combines traditional real-time image synthesis generating high-quality color+depth maps (~30 Hz) with a post-rendering 3D warp [46] which interpolates between two color+depth maps, for synthetic image generation at much higher rates (~300Hz). A third rendering step will perform an even smaller correction, in the form of a single-map warp or a simple image reprojection. The fourth and final rendering step will merely execute a 2D image shift to approximate an orientation-only view correction. Thus, rendering rates comparable with the tracking rates may be achieved. The most recently generated image will be compared with the system's continually maintained estimate of the image currently presented to and perceived by the user. A post-processing unit will produce a sequence of high-frequency 1-bit arbitrary-single-color images using spatio-temporal dithering. In the third layer, illustrated by display control module 504, the scanlines which need updating most are identified and displayed at extreme speed (25 MHz for a single scanline). In the following two subsections, we will discuss rendering, post-processing and display control in details.

4.4 Rendering

For optical see-through AR applications in fields such as medicine and telepresence [21], images can be synthesized using a variety of methods that generate a complete color+depth image. At much faster rates, we will generate through a 3D warp [46] a new image matching an estimated future position. To perform this process efficiently, we will warp only regions of the color-plus-depth map that are within the bounded region representing
all possible changes in the viewing transformations across scanlines. Furthermore, we will generate each scanline just before it is sent to the display, by keeping track of the scanline currently being scanned out by the graphics hardware.

Modern desktop GPUs are designed for general-purpose parallel floating point computation, and can thus be adapted to a variety of rendering algorithms; mobile GPUs are increasingly following this trend as well. We also note that we can generally trade off rendering performance for latency by adjusting the number of scanlines that are rendered together with full spatial coherency.

4.5 Rendering Post-processing and Display Control

Figure 6 is a flow chart illustrating an exemplary process for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays according to an embodiment of the subject matter described herein.

At step 600, a user-perceived image is calculated based on a projected image and a perception model. The user-perceived image may be calculated based on models of human perception and may take into account not only physiological models of human retinas but also models of brain function that consider human abilities of edge detection and pattern matching, relative importance given to subjects in the visual field based on perceived threat or desirability, etc., as well as models of characteristics of the display hardware or other aspects of the physical system.

At step 602, the desired image is calculated. For example, the desired image may be calculated or based on changes in pose of the wearer of a head mounted display, changes in augmented reality content, or both.

At step 604, the user perceived image is compared to the desired image to calculate the error image. In one embodiment, the user perceived image may be calculated based on persistence of images on the human retina or other physiological factors. Because DMD embodiments display the virtual image on glass, there is no image decay such as is normally associated with CRTs phosphors, LED or OLED segments, and the like.
These aspects may be taken into account as part of the calculation of the user perceived image.

At step 606, a portion of the error image ("the error portion") is identified. In one embodiment, the error portion may be identified as the portion of the image having the largest error, but other metrics may be used. Largest error may be defined as largest instantaneous error, largest accumulated error, and so on. In one embodiment, the minimum portion of the image considered may be a row, in which case "the error portion" may be whichever row of the image has the largest collective error. This resolution is suited for display devices which update values on a row-by-row basis, such as a DMD, but other portions (e.g., column, area, portion of row, distributed bit-field, pixel sub-address, etc.) may be used as appropriate for the display technology.

At step 608, a portion of the projected image that corresponds to the error portion is updated. In the case of DMDs, for example, an entire row may be updated. In this example, updating the row may change some mirrors from ON (e.g., reflecting light towards the screen or user's eye) to OFF (e.g., reflecting light away from the screen or user's eye), from OFF to ON, or neither.

The process the returns to step 600, where the user perceived image is (re)calculated based on the updated projection image and the perception model. The process may repeat continually.

The steps illustrated in Figure 6 may be performed by the components illustrated in Figure 5. More particularly, display control module 504 may control display of a projected image that includes real world and synthetic elements to a user. Rendering and processing module 502 may calculate a desired image and a user perceived image based on the projected image and a perception model for a user. Rendering and processing module 502 may calculate an error image as the difference between the desired image and the user perceived image, identify as an error, a portion of the error image. Display control module 504 may update a portion of the projected image that corresponds to the error portion.
Examples of the projected image, the user perceived image, and the error image are presented in Figures 7A - 8F.

Tracking module 500, rendering and processing module 502, and display control module 504 may be implemented or executed by one or more processors 506. Processors 506, display control module 504, rendering and processing module 502, and tracking module 500 may be components of a head wearable display 508, such as an eyeglasses mounted optical see-through display. Head wearable display 508 may include a digital micro mirror device (DMD) 510, an illumination source 512 for illuminating the DMD, and one or more projection surfaces 514 for displaying images to the user. In the Google Glass display, projection surface 514 may be one face of a prism that is located optically in front of a wearer’s eye for displaying projected synthetic images that are overlaid on the wearer's view of the real world. Illumination source 512 may include color filters for modifying the hue of illumination. In an alternate embodiment, illumination source 512 may be a multicolor illumination source. In one embodiment, illumination source 512 may include light emitting diodes.

As will be explained using examples below, DMDs have binary states, i.e., the mirror is either fully ON or fully OFF, which means that intensity of a perceived image may be adjusted by manipulation of duty cycle of the ON and OFF states. For this reason, even for a static image rows will be continually updated as necessary to produce the desired perceived intensity.

In this manner, after each scanline update is transmitted to the display as quickly as possible, the system will immediately emit "updated" photons. This will be ensured by:

(1) A novel post-processing method which will keep track of the state of the display elements such that the rendering process can determine which parts of the image require updating most urgently—likely by means of a virtual panorama surrounding the user and to which spatio-temporal dithering will be applied in order to calculate said display updates;

(2) An experimental high-speed micromirror based display [1], which provides low-level raw chip access and is capable of updating scanlines (1024 pixels), or fragments thereof, at 25MHz.
Each pixel's micromirror is either "on" or "off"—that is, angled to reflect the global light source either to the viewer, or into a light-absorbing baffle. In conventional designs, the micromirrors are used in this way, typically with color-sequential illumination and time-multiplexed to accumulate the desired intensity through numerous on-off cycles executed at each pixel and during each 1/60th second frame. In one embodiment, illumination is provided by three-color (red-green-blue) LED illuminators that can be activated simultaneously and with smoothly modulated intensities, thus capable of illuminating the micromirrors at any moment with an arbitrary, controlled color that can change at update rates. Thus, at each display update (up to a full scanline at a time), a new optimal color will be calculated such that the currently active micromirrors or pixels, together with the current update, deliver the most urgently needed incremental contribution. This process is repeated at very high rates, delivering the fastest updates DMDs are capable of.

As mentioned, these high-frequency scanline (or scanline fragment) updates will be calculated by comparing the difference between the image the system wants the user to see and the image it estimates the user is actually seeing. Finally, blank scanlines will of course not be updated at all.

As mentioned, the approach described here is very uniquely suited for the rapidly switching, selectively addressable (in small groups such as scanlines) DMDs. Other display technologies, such as Liquid Crystal Displays (LCDs) have higher settling characteristics after state changes. Since DMDs operate many orders of magnitude faster than all other technologies, we focus the presented work on them as the first target, although the principles described herein are not limited to DMDs but may be applied to other technologies as well as appropriate.

Figures 7A through 7E illustrate the operation of systems and methods according to an embodiment of the subject matter described herein using a simple graphic image. In each of Figures 7A through 7E, the timestamp is displayed at the top of the figure, below which are four images. These four images are, left to right, the desired image, the error image, the projector image, and the user perceived image. Below the four images are
two graphs. The graph on the left shows the mean absolute error of the error image. As the user perceived image more closely approximates the desired image, the absolute error should be reduced. If the user perceived image exactly matches the desired image, the mean absolute error should approach zero. The graph on the right displays a row selection histogram, i.e., how often a particular row is updated. In the example illustrated in Figures 7A through 7E, the desired image is static, i.e., it does not change or move, but because it is an image having different intensities, it is expected that pixels on the bottom row of the triangular figure will be ON more often than OFF, while the pixels at the top point of the triangular figure will be OFF more often than ON, reflecting the duty ratios required by the micromirrors to approximate varying intensities.

Referring now to Figure 7A, time = 0 ms, meaning that the desired image has just appeared, the user perceived image is blank, and thus the error image is essentially identical to the desired image. As a result of the calculation of error, it can be seen that the row of pixels located at the bottom of the error image (row 87 in the row selection histogram) has been identified as having the largest error value. The pixels in that row were formerly all OFF, so the row update will turn some pixels in that row to ON, as shown in the projector image. Mean absolute error has not yet been calculated.

In Figure 7B, time = 1 ms. As can be seen from the row selection histogram, there have been approximately 25,000 row updates within the first millisecond operation, which brought the user perceived image very close to the desired image, as can be seen by the error image, which shows slight errors near the top of the triangular image. The projector image is showing that at this instant in time, several of the lowest rows have pixels that are ON and the remainder of the rows have pixels that are all OFF. The row selection histogram shows that rows near the top of the image, i.e., rows 1-10, are selected least often. The mean absolute error has dropped to below 0.025 within just a few microseconds of operation.

In Figure 7C, time = 2 ms. The user perceived image is almost exactly identical to the desired image, with only very slight error values near
the top of the error images. The projector image at this instant of time again
happens to show only rows near the bottom of the projector image still
having pixels that are ON.

In Figure 7D, time = 3 ms, and the error between the desired image
and the perceived image is approaching zero.

In Figure 7E, time = 4 ms, and the error image is blank, indicating no
difference between the desired image and the perceived image. This is also
shown by the mean absolute error graph, which shows a value of zero. In
other words, the systems and methods described herein can provide a user
perceived image having acceptably low error in only 1 millisecond and
having essentially no error in only 4 milliseconds. In contrast, conventional
systems have a latency of 50 milliseconds or more.

To show how well this method works for complex images, we now
refer to another example using a complex, grayscale image, although the
same principles may be applied to render color images.

Figures 8A through 8K illustrate the operation of systems and
methods according to an embodiment of the subject matter described herein
using a complex image.

Referring now to Figure 8A, time = 0 ms, the desired image is on the
left, the user perceived image is on the right, and the error image is the
difference between the two and thus looks like the desired image. The row
selection histogram indicates that the algorithm identified row 246 of the
image as the row having the largest error, and a close look at that row in the
desired image reveals a high incidence of very bright pixels. The algorithm
determined that all pixels in that row should be set to ON, as can be seen by
the projector image.

In Figure 8B, at time 1 ms, after approximately 25,000 row updates,
the algorithm has chosen to light almost all pixels, resulting in the user
perceived image showing almost all white except for a few pixels scattered
throughout the image that have remained unlit. The error image shows the
calculated difference between the desired image and the user perceived
image, and the projector image is now mostly dark to compensate for the
overly bright user-perceived image.
In Figures 8C through 8K, the projector image remains mostly dark except for a few portions of the image which should continue to be perceived as bright while the rest of the user-perceived image darkens. As can be seen in Figure 81, the user perceived image approaches acceptable error levels in as early as 8 millisecond and approaches zero error only 1 millisecond after that. Thus, even for complex user perceived images, the systems and methods described herein show a latency that is less than one-fifth of the time of conventional systems.

Although the examples shown in Figures 7A through 8K are grayscale images, the same principles of calculating an error image, identifying a row or other portion of the image as having the largest error, updating the projector image, and recalculating user perceived image may be applied to color images as well. For example, error may be calculated as a difference in perceived color or color intensity.

For display systems that use a color wheel or other time-multiplexed color system such as RGB LEDs, error may be calculated for each color independently and the projector image updated to compensate for error of a particular color during that color's time slot, e.g., when the DMD is being illuminated with that color. In this approach, when the illumination color changes, all pixels that do not include that color will have high error values and are thus likely to be identified by the algorithm as needing to be changed, e.g., to turn those pixels OFF.

Alternatively, error may be calculated for all colors at the same time and the projector image updated continually, where the user perceived image is updated while considering the color currently being used to illuminate the DMD. Here also, pixels that do not include as a color component the current color being used to illuminate the micromirror array will have a higher color error than those pixels that do include that color component.

The availability of RGB LEDs makes possible two very interesting alternative embodiments: rather than time-multiplexing illumination of the colors pure red, pure green, and pure blue, one embodiment considers the overall color error of one of the colors and adjusts the intensity of the
illumination source. To use a simple example to illustrate the point, if the
desired image is a solid field of dim blue, for example, by adjusting the
intensity of the blue LED, it may be possible to simply leave all of the pixels
in the ON position and dimly lit, rather than having to continually perform a
50% duty cycle for a blue LED at maximum brightness.

In yet another embodiment, the fact that RGB LEDs may be adjusted
to illuminate the DMD with any color (e.g., by adjusting the intensity of each
of the red, green, and blue LEDs individually) allows us to discard the time-
multiplexed RGB mode entirely, and instead calculate a global color error
and use that color error to dictate what hue and intensity should next be
provided by the RGB LED illumination source. In this embodiment, by
adjusting the hue and intensity of the illumination source, more subtle
nuances of color may be displayed with few row updates and thus even
lower latency. To again use a simple example to illustrate the point, if the
desired image is a solid field that changes from bright red to dim green, for
example, it is possible to achieve this change in user perceived image by
simply changing the hue and intensity of the illumination source, potentially
without changing the state of any of the micromirrors.

The principles described herein apply not only to temporal dithering
but also to spatial dithering as well, making them suitable for displaying
dynamic images. The error calculation would be the same as above. In one
embodiment, the desired image is calculated not only for the current field of
view of the user but also extended to accommodate the range of possible
motion and/or change of pose that the user of the head mounted display
may perform. By calculating a larger image field in advance, the pose
tracker need only identify the portion of that larger field to be rendered as the
desired image, and the methods and systems described herein apply without
modification.

Although the examples described above operate on rows of a
displayed image, the subject matter described herein is not so limited, but
may be applied to portions of rows, areas, or other subsections of an image,
as is appropriate for the display technology. Although conventional DMDs
have row and column addressable pixels, current DMDs are designed such
that an entire row must be updated at the same time, as befits a display that
is designed to accommodate the scanline-based approach used by
conventional television and broadcast technology. However, by modification
of DMD control hardware to change the minimum portion of an image that
can be updated, the principles applied herein may be adapted to calculate
error for some portion other than a single row, i.e., whatever the minimum
portion of the image happens to be.

Although the examples described above calculate error based on the
difference between the desired image and the user perceived image, these
error calculations are not limited to a simple calculation of difference, but
may also consider other factors. For example, error minimization may
consider aspects of human physiology such as the importance of a particular
object in the visual field, human ability to perform edge detection in the
visual field, and the like.

4.6 HMD Integration

In order to test our low-latency tracking and rendering solution, we will
integrate our modified miniature DMD projectors [1] into an optical see-
through HMD with a cluster of miniature tracking cameras, as described in
3.1. In recent years, we have constructed see-through HMDs, as well as
modified commercial ones in order to adapt them for use in AR systems.
One of our designs introduced an "orthoscopic" video see-through unit which
featured direct unobstructed peripheral imagery around the borders of the
display, as seen by the wearer. Across that border, we were able to achieve
registration of the direct peripheral imagery with the merged video/synthetic
imagery within the display [63].

The systems and methods described herein include: a low-latency
rendering method which can periodically generate high-quality color+depth
maps and just-in-time image updates using most recent viewing transform
from the tracker; a spatio-temporal dithering based image postprocessing
approach that considers human visual perception to approximate a rendered
color image with a sequence of 1-bit arbitrary color images which can be
efficiently displayed. Implement a most suitable approach using an FPGA for
color sequence computation and determination of scanlines which need to be updated, and also for low-level control over the micromirror-based display; and a low-latency interface to the micromirror display units with direct scan line access.

5 Benefits for Society, Education, and Research

Societal. The presented AR solution, combined with emerging comfortable, eyeglass-style headworn displays should enable a wide range of applications to benefit from visual augmentation. Many applications that today use stationary stereo TVs for augmented visualization will now be able to use head-worn displays and reap the benefits of more natural hand-eye coordination and augmented imagery anywhere the user looks.

Educational. The presented AR solution will be integrated into at least 3 graduate classes, classes in which students pursue individual term projects: Computer Vision (Comp 776) Exploring Virtual Worlds (Comp 872), and Telepresence (Comp 790). In each of these classes, the investigators expect the presented system to stimulate student exploration of new directions in rendering, tracking, image acquisition and reconstruction, augmented reality applications and telepresence solutions.

Research. The investigators expect the presented latency reduction framework to be adopted by other AR researchers, allowing more effective visual interaction. Beyond head-mounted displays, the presented approach is expected to be useful for spatial augmented reality and handheld AR devices that rely on low-latency visual feedback in response to moving objects and gestures.

Motivated by the realization that decades after the introduction of Augmented Reality, its exciting promise continues to be hampered by the latency present in all of today's AR systems. The subjects and methods described herein make use of the fundamental insight that imagery from vision-based tracking cameras can be processed in scanline fragments instead of frame-by-frame, and that an AR HMD using DMD technology can update scanlines at MHz rates. This strategy enables detection of HMD motion at faster rates and with lower latency; it also enables presentation of
registered imagery with analogous speedups. Thus it holds the potential to minimize lag-induced registration errors, which are the main effect of the unacceptably high latency in today's systems.

Each of the following references is incorporated herein by reference in its entirety:


-29-


It will be understood that various details of the subject matter described herein may be changed without departing from the scope of the subject matter described herein. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation.
CLAIMS

What is claimed is:

1. A system for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays, the system comprising:
   5 at least one processor;
   10 a display control module for controlling display of a projected image to a user; and
   a rendering and processing module for calculating a desired image and a user-perceived image based on the projected image and a perception model and for: calculating an error image as the difference between the desired image and the user-perceived image; and identifying an error portion of the error image; and wherein the display control module updates a portion of the projected image that corresponds to the error portion.

2. The system of claim 1 wherein the rendering and processing module recalculates the user-perceived image based on the updated projection image and the perception model.

3. The system of claim 1 comprising a wearable display, wherein the display control module is a component of a head-wearable display.

4. The system of claim 3 wherein the wearable display comprises an optical see-through display.

5. The system of claim 3 wherein the wearable display comprises a digital micromirror device (DMD).

6. The system of claim 5 wherein the wearable display module includes an illumination source for illuminating the DMD.

7. The system of claim 6 wherein the illumination source includes multiple color filters for modifying the hue of the illumination.

8. The system of claim 6 wherein the illumination source includes a multicolor illumination source.

9. The system of claim 8 wherein the multicolor illumination source comprises a light emitting diode (LED).
10. The system of claim 6 wherein updating a portion of the projected image that corresponds to the error portion includes changing the hue and/or intensity of the light being provided by the illumination source.

11. The system of claim 1 wherein the identified error portion is the portion of the error image having the largest error.

12. The system of claim 1 wherein the error portion comprises a row of pixels.

13. The system of claim 1 wherein the rendering and processing module calculates the desired image based on pose information determined by the tracking module.

14. The system of claim 13 wherein the tracking module calculates pose based on row data received from a rolling shutter camera.

15. A method for minimal-latency tracking and display for matching real and virtual worlds in head-worn displays, the method comprising:
   - calculating a user-perceived image based on a projected image and a perception model;
   - calculating a desired image;
   - calculating an error image as the difference between the desired image and the user-perceived image;
   - identifying as an error portion a portion of the error image; and updating a portion of a projected image that corresponds to the error portion.

16. The method of claim 15 comprising recalculating the user-perceived image based on the updated projected image and the perception model.

17. The method of claim 15 wherein identifying the error comprises identifying a portion of the error image having the largest error.

18. The method of claim 15 wherein the portion of the error image comprises a row, a column, or an area of the error image.

19. The method of claim 15 wherein the projected image is an augmented reality image presented to a user via a head-mounted display.

20. The method of claim 15 wherein the augmented reality image is presented to the user via an optical see-through display.
21. The method of claim 15 wherein calculating a desired image comprises receiving the image from a rendering module.

22. The method of claim 15 wherein calculating an error image includes edge processing.

23. The method of claim 15 wherein calculating an error image includes determining an importance value for features of the image and using the determined values to calculate error values.

24. The method of claim 15 wherein calculating an error image includes calculating a difference of hue and/or intensity between the desired image and the image currently being perceived by the user.

25. The method of claim 15 wherein updating a portion of the projected image that corresponds to the error portion includes updating a row of a multi-row display.

26. The method of claim 15 wherein recalculating the image currently being perceived by a user based on the updated projected image includes calculating the image using models of human perception.

27. The method of claim 15 wherein calculating the desired image comprises calculating the desired image based on pose information.

28. The method of claim 27 wherein the pose information comprises row data received from a rolling shutter camera.

29. A non-transitory computer readable medium having stored thereon executable instructions that when executed by the processor of a computer control the computer to perform steps comprising:

   calculating a user-perceived image based on a projected image and a perception model;
   calculating a desired image;
   calculating an error image as the difference between the desired image and the user-perceived image;
   identifying as an error portion a portion of the error image; and
   updating a portion of a projected image that corresponds to the error portion.
(RE)CALCULATE A USER-PERCEIVED IMAGE BASED ON A PROJECTED IMAGE AND A PERCEPTION MODEL

CALCULATE THE DESIRED IMAGE

CALCULATE THE ERROR IMAGE AS THE DIFFERENCE BETWEEN THE USER PERCEIVED IMAGE AND THE DESIRED IMAGE


UPDATE A PORTION OF THE PROJECTED IMAGE THAT CORRESPONDS TO THE ERROR PORTION

FIG. 6
FIG. 7C
TIME $t = 3$ msec

**FIG. 7D**
TIME $t = 4$ msec

**FIG. 7E**
FIG. 8A
TIME $t = 1$ msec

DESIRED IMAGE (300x300)  ERROR IMAGE  PROJECTOR IMAGE  USER PERCEIVED IMAGE

FIG. 8B
TIME $t = 2 \text{ msec}$

**FIG. 8C**
TIME $t = 3 \text{ msec}$

**DESIRED IMAGE (300x300)**  
**ERROR IMAGE**  
**PROJECTOR IMAGE**  
**USER PERCEIVED IMAGE**

**FIG. 8D**
FIG. 8E
TIME $t = 5$ msec

**FIG. 8F**
TIME $t = 8$ msec

**FIG. 8G**

- **Desired Image (300x300)**
- **Error Image**
- **Projector Image**
- **User Perceived Image**

**Mean Absolute Error**

![Mean Absolute Error Graph]

**Row Selection Histogram**

![Row Selection Histogram]
TIME $t = 7$ msec

**desired image** (300x300)  **error image**  **projector image**  **user perceived image**

**FIG. 8H**
TIME $t = 8$ msec

**Desired Image (300x300)**  **Error Image**  **Projector Image**  **User Perceived Image**

**Mean Absolute Error**

**Row Selection Histogram**

*FIG. 81*
TIME $t = 9 \text{ msec}$

**Figure 8J**

**Mean Absolute Error**

**Row Selection Histogram**
TIME $t = 10$ msec

**Desired Image (300x300)**

**Error Image**

**Projector Image**

**User Perceived Image**

**Fig. 8K**
INTERNATIONAL SEARCH REPORT

International application No. PCT/US 2015/058959

A. CLASSIFICATION OF SUBJECT MATTER

G09G 5/377 (2006.01)
G06T 5/50 (2006.01)
G02B 27/01 (2006.01)
H04N 21/236 (2011.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 1/00-17/50, G06T 1/00-17/30, G06K 9/00-9/82, G02B 27/00-27/64, G09G 5/00-5/42, H04N 5/00-21/858,
G02F 1/00-1/39

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 2009/0213037 A1 (SAAB AB) 27.08.2009, abstract, paragraphs [0014], [0025H0027]</td>
<td>4-10, 14, 20, 22, 24, 28</td>
</tr>
<tr>
<td>Y</td>
<td>US 2012/01 19978 A1 (OSTERHOUT GROUP, INC.) 17.05.2012, abstract, paragraphs [0035], [0045], [0052]</td>
<td>7-10, 22, 24</td>
</tr>
<tr>
<td>Y</td>
<td>US 201 1/021 1082 A1 (PER-ERIK FORSSEN et al.) 01.09.2001 1, abstract</td>
<td>14, 28</td>
</tr>
<tr>
<td>A</td>
<td>US 2005/0168403 A1 (JOHN FRANKLIN EBERSOLE et al.) 04.08.2005</td>
<td>1-29</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:
A+ document defining the general state of the art which is not considered to be of particular relevance
E+ earlier document but published on or after the international filing date
L+ document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
O+ document referring to an oral disclosure, use, exhibition or other means
P+ document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search: 16 February 2016 (16.02.2016)
Date of mailing of the international search report: 03 March 2016 (03.03.2016)

Name and mailing address of the ISA/RU:
Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37

Authorized officer: V. Zhakovich
Telephone No. (499) 240-25-91

Form PCT/ISA/210 (second sheet) (January 2015)