



HOT
C H I P S



Touch-free technology

Itay Katz
Founder and CTO



Overview

eyeSight's groundbreaking **Touch Free technology** provides an **enhanced user experience**, allowing to **easily** and **intuitively** control a variety of devices using simple hand gestures.

eyeSight's **Natural User Interface** solution utilizes the device's **standard 2D camera**, along with advanced **real-time** image processing and machine vision algorithms, to track the user's **hand gestures** and convert them into actions.

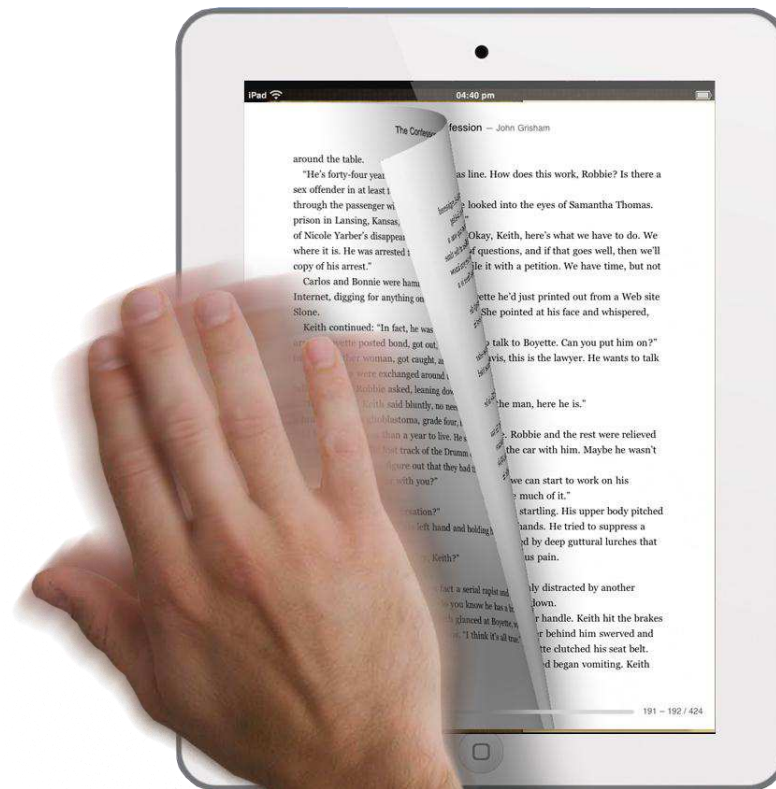
Components of eyeSight's Gesture recognition technology:

- **Directional gestures detection**
- **Real-time hand / fingertip tracking**
- **Face detection and tracking (multiple users)**
- **Hand signs; "OK", "Like" etc.**



Implementation Challenges of such technologies:

- Very high processing requirements
- Memory Throughput
- Real-time performance



Solutions: Managing performance

- **Algorithm- the key tool for minimizing performance**
- **Target specific optimizations**
 - Intrinsic – using special instructions
 - ISA Extensions: x86 SSE, ARM NEON
 - Assembly
- **GPGPU – OpenCL**



Solutions: Coping with memory throughput

- **Image processing algorithms require reading/writing the video many times, placing a lot of load on the memory subsystem. Reading a 720p YUV422 streams just once requires 55 Mbytes/Sec.**
- **DMA – not usable in most systems, since there are no tightly-coupled memories.**
- **It is key to minimize the number of passes on the video data.**
- **Designing for Cache**
 - Minimize data footprint
 - Locality
 - Automatic pre-fetching
 - Software pre-fetching

Solution: Handling real-time performance

- **Most systems are non-deterministic:
Caches, other processes, O/S behavior**
- **Algorithms require more processing when there's "action" in the video**
- **This makes the instantaneous performance requirements vary greatly**
- **As a result, it is not possible to guarantee hard real-time performance**
- **Instead, our solution is design to be soft real-time, and to handle real-time violations**
- **One challenge is the lack of standardized high-precision timers to allow software to monitor execution time**

Platform portability - Challenges:

Challenges:

- **Maintaining a single code base across multiple products, OS types and compilers**
- **Stay up to date with recent OS version releases**
- **Maintain backward compatibility**
- **Maintaining pixel portability**

Platform portability - Solution:

Solution:

- Working with automated tools for each core code change
- Maintain strict coding conventions to avoid using platform specific libraries
- Use cross platforms frameworks such as OpenCL
- Work with a configurable device abstraction layer to keep platform specific code from migrating into different platforms or devices
- Using a framework developed in eyeSight we can convert any video stream, any resolution or format (including IR), to a single representation that serves our algorithms

Thank you.

www.eyesight-tech.com



eyeSight™

