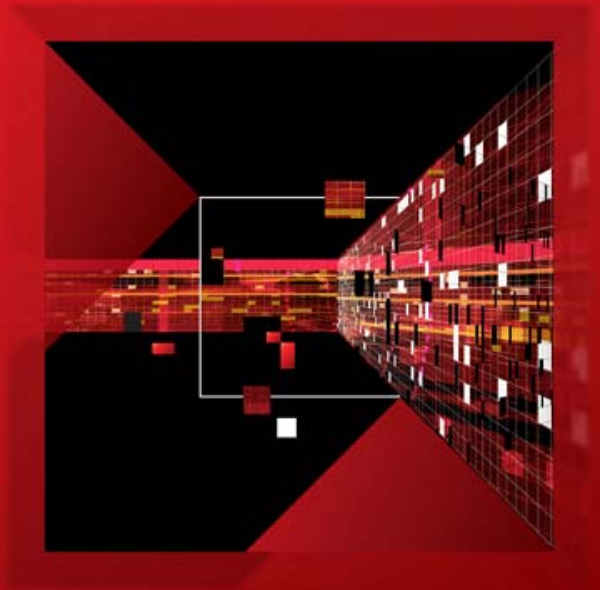

AMD
Fusion 11
DEVELOPER SUMMIT


# *HIGH QUALITY AND EFFICIENT POST PROCESSING ON GPU COMPUTE*

**Philip Swan**
**AMD**
**Principal Member of Technical Staff**

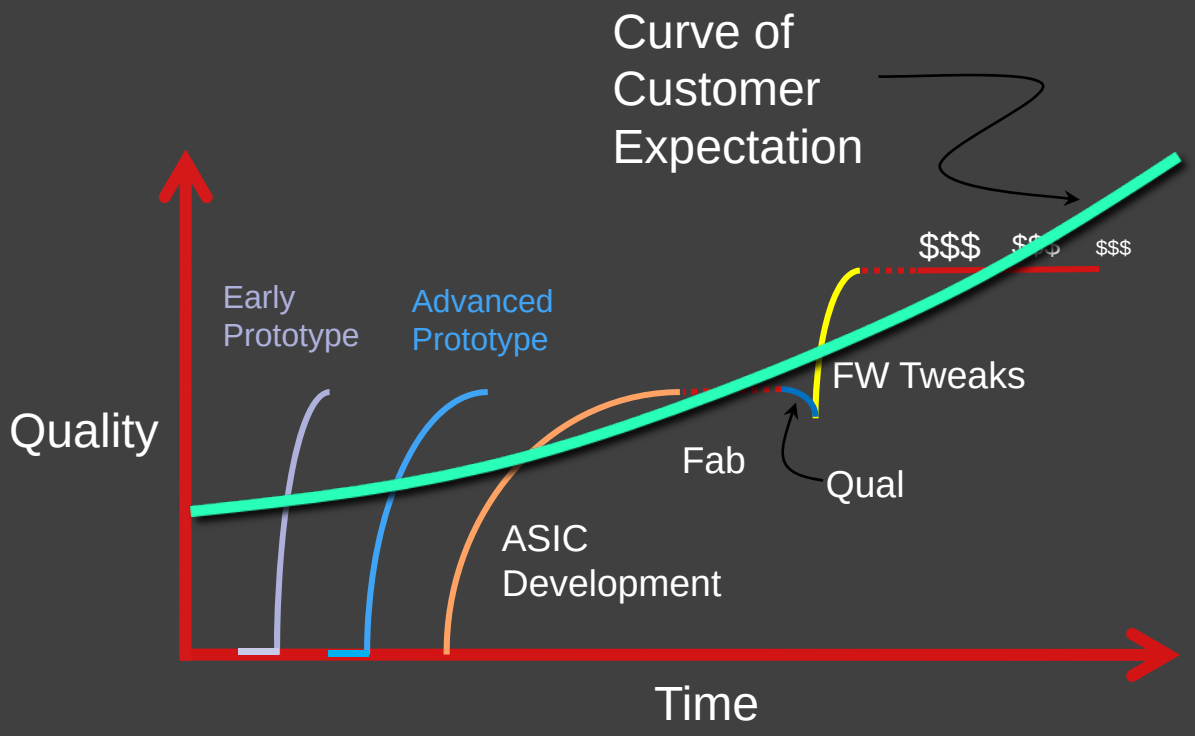
# *HIGH QUALITY AND EFFICIENT POST PROCESSING ON GPU COMPUTE*



# OVERVIEW

- The Video Algorithm Design Problem
  - Traditional Approach
  - A Better Approach, using Fusion APU
- Proof Points
  - Histogram Example
  - Skin-Tone Correction
  - Mosquito Noise and Deblocking
  - Camera Shake
- Summary

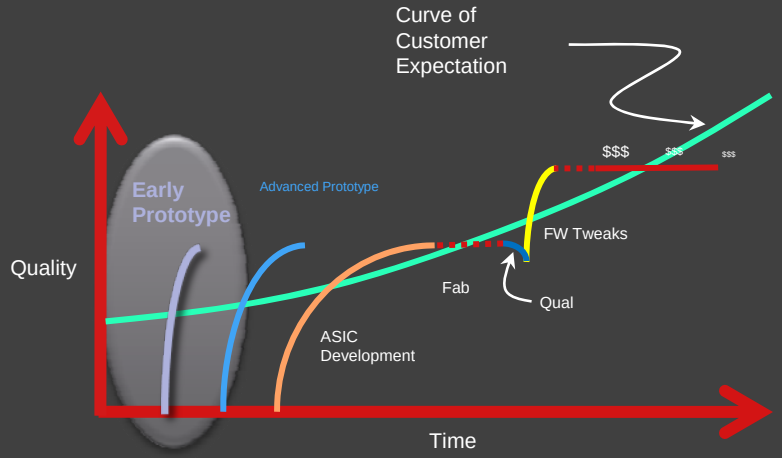
# THE SCIENCE OF MEETING CUSTOMER EXPECTATIONS




Customer Expectations Change – Time-To-Market is Key

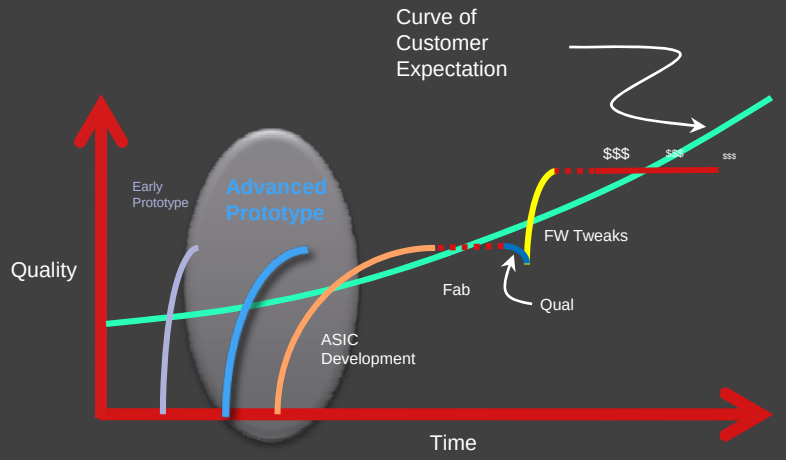
# EARLY PROTOTYPE

- Generally a C/C++ Model
- Calls for a researcher's skill set
- Well instrumented
- Easy to iterate rapidly
- Generates impressive and "demo-able" results on select clips
- May be too slow for thorough "field testing"



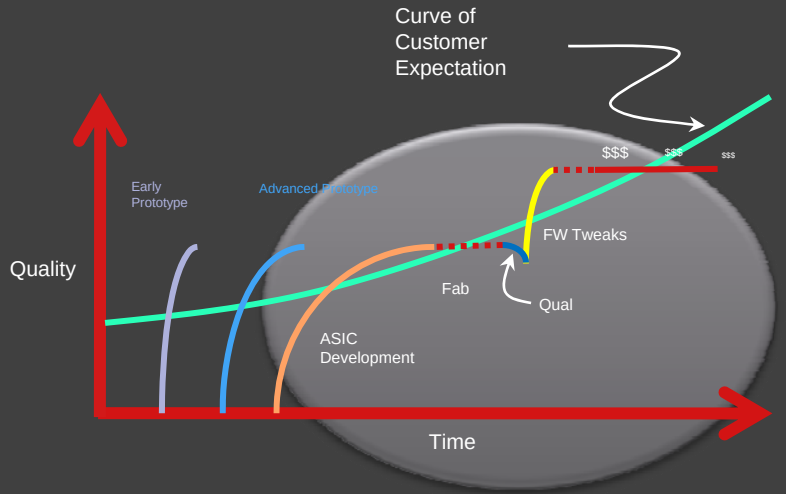
# ADVANCED PROTOTYPE

- Traditionally
  - FPGA based or…
  - Advanced C Model (supported by server farm)
  - Faster Throughput -> Allows Stress Testing of the Algorithm
  - More likely to secure customer commitment
  - Calls for an implementer's skill set to build and maintain
  - More difficult to instrument
  - Iteration overhead increases
  - Possibly finicky in the customer's hands

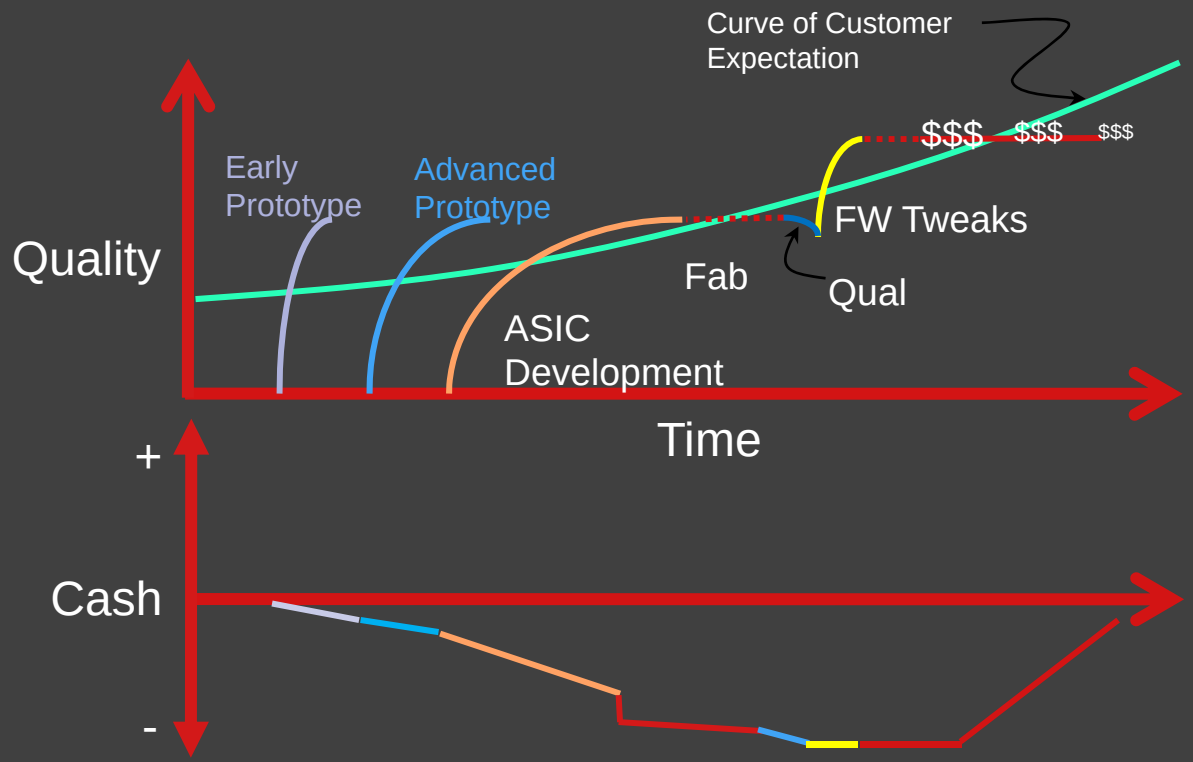


# ASIC DEVELOPMENT, QUAL, AND FW TWEAKS

- Generally a Full Custom ASIC design
- Can meet performance, cost, and power requirements
- High NRE (Non-Recurring Expense/Engineering)
- Requires several disciplines (HW, QA, FW)
- Inflexible - changes impact schedule
- Supports minimal instrumentation – debugging effort is high
- Often incorporates a small CPU and lots of registers
- “FW Tweaks” often save the day



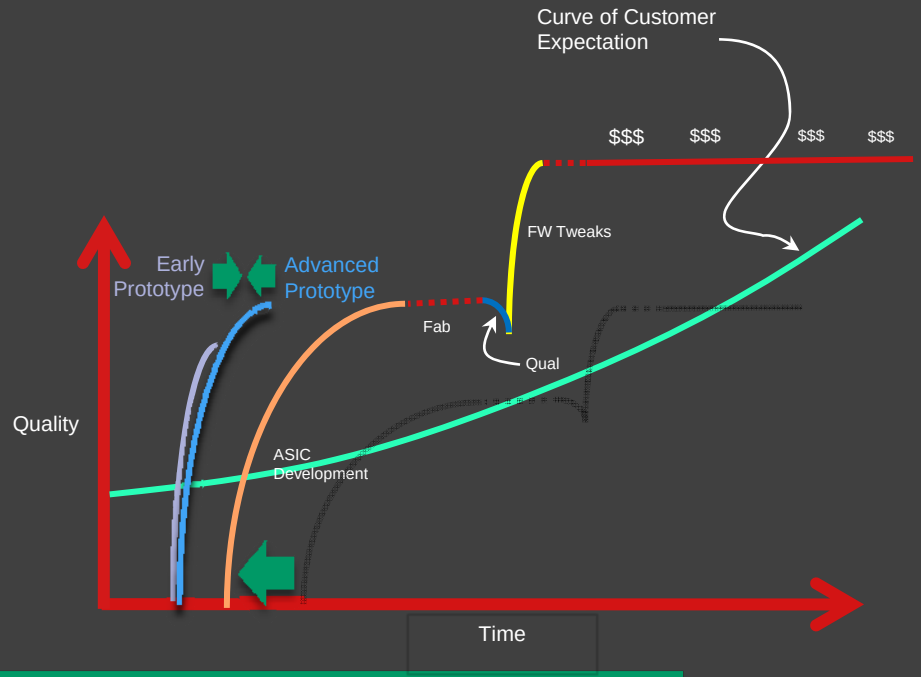
# EACH PHASE OF THE PROJECT HAS AN ASSOCIATED COST




Traditional Approach = High Financial Risk

# *A BETTER APPROACH - PART 1*
# *USE GPU TO ACCELERATED R&D*

- Load the R&D systems up with one or more high-end discrete GPUs
- Accelerate semi-stable C code…
  - Pixel Shaders or Compute Shaders
  - (you're already using Heterogeneous Compute!)
- ***Increase R&D Rate***
  - Real-time execution rates
  - Still well instrumented and can iterate very fast
  - Stabile compared to FPGA/Farm
- Improve Cost, Power, Quality, Features, etc:
  - Algorithm Improvement ROI > Architecture ROI > Implementation ROI > Layout ROI

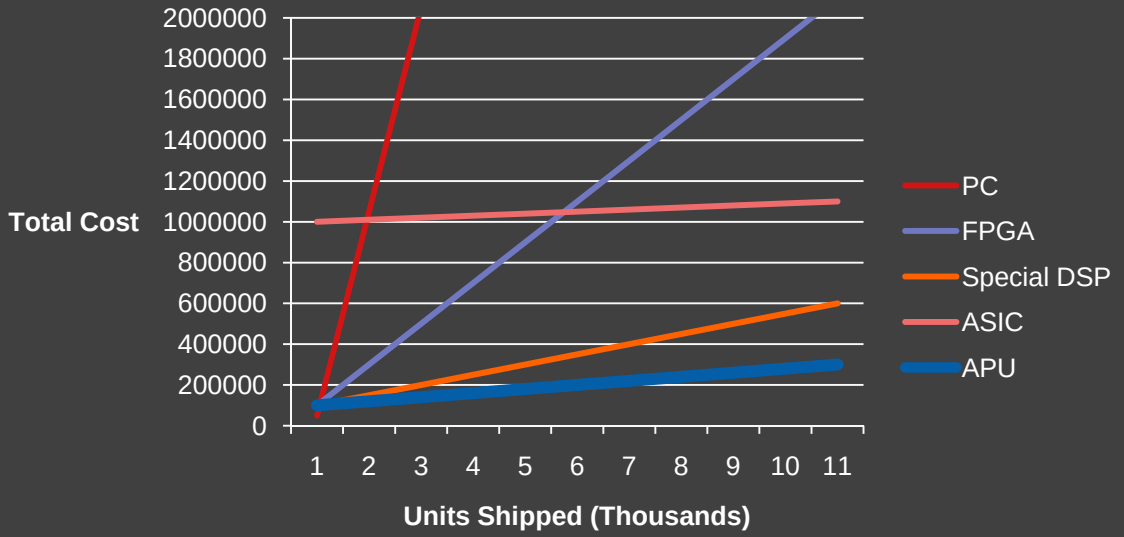



Downstream Effect is *Huge*. Its Like Owning a Time Machine.

# *A BETTER APPROACH - PART 2*
# *USE FUSION APU TO IMPLEMENT*

Choices:

- Embed a PC Motherboard
- FPGA
- Specialized DSP
- Full Custom ASIC
- **(New) Fusion APU**

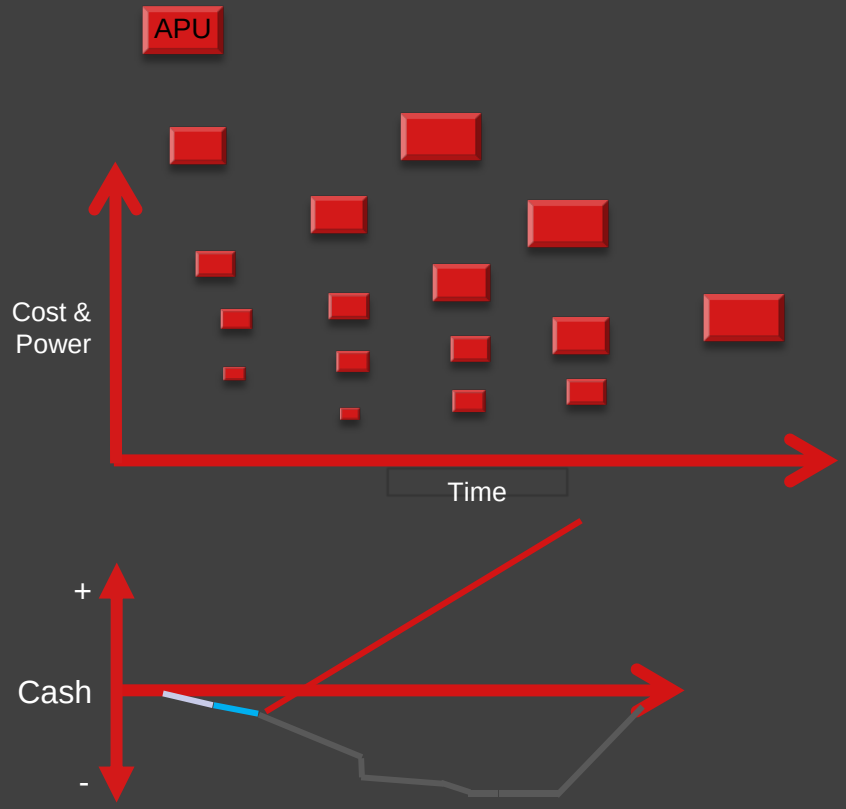



*Data for illustration only

The APU Cost Curve May Look Promising *to You*

# *ADVANTAGES OF APUs*

- Affordable NRE
- Low Unit Cost
  - Leverages our economies of scale
- Many performance/cost options
- Leverages x86 CPU
- Highly optimized ALU logic
- Easy to implement on
- Power efficient
  - Designed for mobile applications
- ***Flexible***




The APU is Game Changer

# *I HEARBY RENOUNCE MY OLD WAYS AND WHOLE-HEARTEDLY DEDICATE MY LIFE TO YOUR NEW RELIGION…*

- Questions of a skeptic…
  - Is the language hard to learn?
  - Is it proprietary?
  - Where can I find people who can program it?
  - Are the tools rugged?
  - How can I forecast things like cost and time-to-market?
  - Will APUs achieve acceptable computational density with *my* algorithm?

# *QUESTIONS (AND ANSWERS)*

- Is the language hard to learn?
  - In my experience no. It's a subset of C. It works surprisingly well. Many others have similar experience.
- Is it proprietary?
  - No.
  - MS supports shaders via DirectX
  - OpenGL/CL are open standards.
- Where can I find people who can program it?
  - Everywhere. Keywords generate lots of hits on resume searches.
- Are the tools rugged?
  - They work great. Way better than other "great tools" I've experienced.
- How can I forecast things like cost and time-to-market?
- Will APUs achieve acceptable computational density with *my* algorithm?

# *TRUE-TO-LIFE EXAMPLES*

- Histogram Based Dynamic Contrast Algorithm
- 2D LUT Skin Tone Correction
- Mosquito Noise and Block Artifact Reduction
- Camera Shake Removal

# HISTOGRAM (32 BIN) – PART 1: PARTITION THE WORK

- A histogram doesn't *seem* like a good algorithm for a parallel architecture…
- … but it is!
- Partition the image into strips of 4x60 pels.
  - 1920x1080 *becomes* 480x18 x (4x60)
- Make a "mini histogram" for each strip…
  - *unsigned char* hist[18][480][NUM_BINS];
- Assign one thread to each strip to generate histograms for the strips in parallel…
- 8640 work items - uses all of the GPUs threads
- Summation of mini-bins also parallelized using reduction tree

```
// C algorithm
unsigned char p[1080][1920];

int x, y, w = 1920, h = 1080, hist[32];
memset(hist, 0, sizeof(hist));
for (y=0; y<h; y++) for (x=0; x<w; x++) {
    hist[p[y][x]>>3]++;
}
```

```
// C algorithm partitioned into 4x60 strips
unsigned char p[1080][1920];

int x, y, dx, dy, w = 1920, h = 1080;
unsigned char hist[18][480][32];
for (y=0; y<h/60; y++) for (x=0; x<w/4; x++) {
    // Part done by each thread…
    memset(hist[y][x], 0, sizeof(hist[y][x]));
    for (dy=0; dy<60; dy++) for (dx=0; dx<4; dx++) {
        hist[y][x][p[y*60+dy][x*4+dx]>>3]++;
    }
}
// Sum mini-bins to produce final histogram
(not shown)
```

# HISTOGRAM (32 BIN) – PART2: PARALLELISM USING WIDE REGISTERS

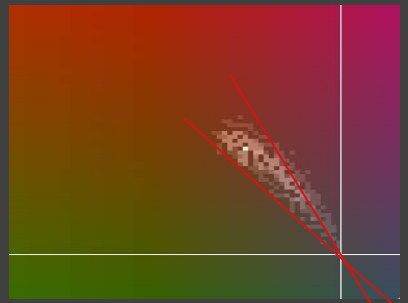
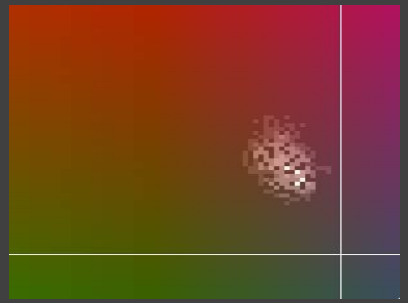
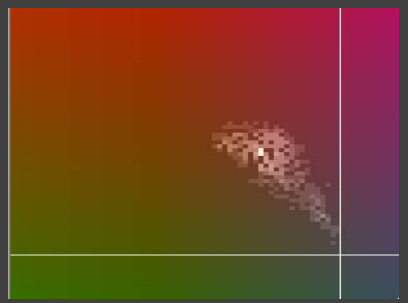
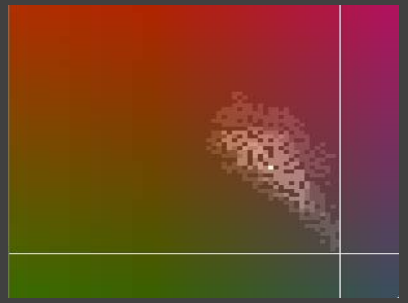
- Each thread can use “by-four” pixel processing…
- Every cycle *four* pixels are processed by the thread.
- Collect 12 pels into 32 four bit bins
- Then dump this into 32 eight bit bins
- 6*60+3*20 = 420 cycles/block
- 420/(60*4) = 1.75 alu cycles/pel

```
memset(hist[y][x], 0, sizeof(hist[y][x]));
for (dy=0; dy<60; dy++) for (dx=0; dx<4; dx++) {
    hist[y][x][p[y*60+dy][x*4+dx]>>3]++;
}
```

```
uint4 srcData, index4
uint4 fourbitbins31to0 = 0, lower16bins = 0, upper16bins = 0;
mask = (0x0f0f0f0f0, 0x0f0f0f0f0, 0x0f0f0f0f0, 0x0f0f0f0f0);
for (uint dy = 0; dy < 60; dy++) {
    srcData = s_2D.Gather(samPoint, runCoord); // dy
    index4 = srcData>>3;
    *
    *
    *
}
Out.Color0 = lower16bins;
Out.Color1 = upper16bins;
```

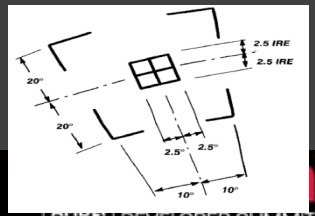
# 2D CHROMATIC HISTOGRAMS (INTENSITY REPRESENTS HITS ON A COLOR)



Skin-tones fall inside a wide range of different (but valid) hue and saturation values.

2D “UV LUTs” are used to efficiently implement the chroma corrections



# *MOSQUITO NOISE AND DEBLOCKING*

- Complex spatial algorithms
- Require large pixel support area
- Technology was implemented faster than real time
  - *in months*

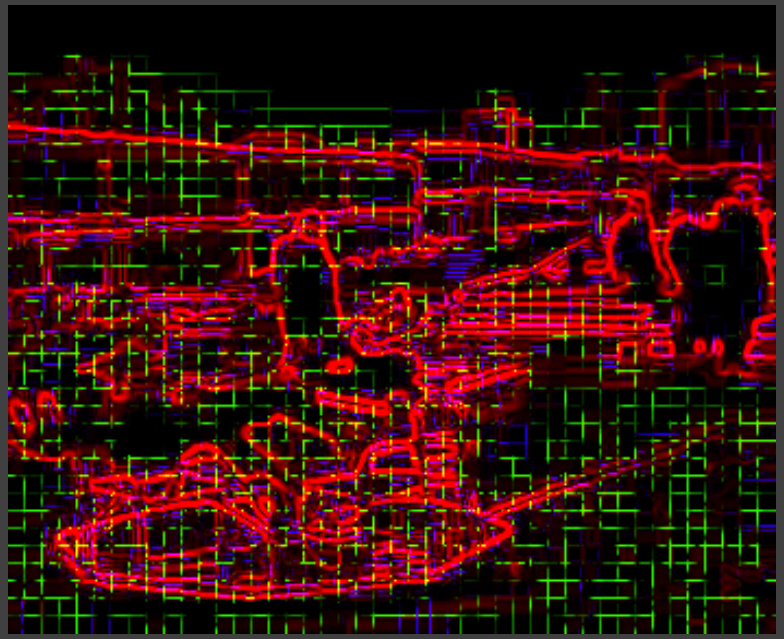
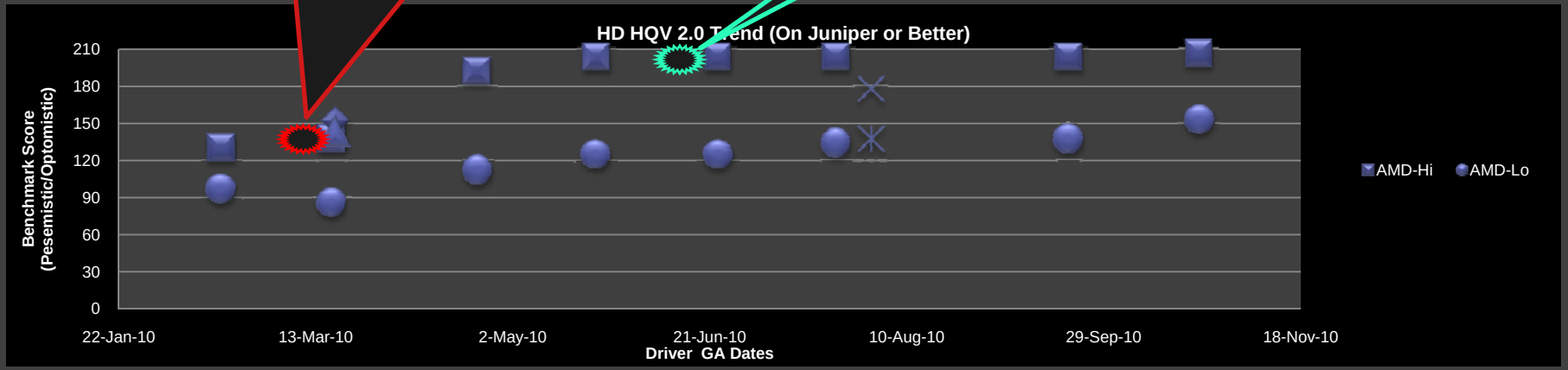


Figure (right): Map produced by the deblocking algorithm that shows how algorithm distinguishes block artifact edges (green) versus image content edges (red).

# RAPID REACTION TO EVENTS

"Unfortunately, HQV Benchmark 2.0 suggests that none of modern GPUs can deliver ideal image quality which you can expect from special-purpose video chips installed into players and TV-sets. Hopefully, AMD will be able to improve the video playback quality of its solutions by polishing off their drivers (xbitlabs)."

"KitGuru can state clearly that ATI's Radeon HD 5000 series cards offer the best HD video reproduction quality available in the market… Catalyst 10.6 is without a doubt the best ATI driver yet, specifically in regards to the quality of video rendering."




Allowed us to rapidly respond to the newly published HQV2.0 benchmark

# CAMERA SHAKE – A GOOD EXAMPLE OF HETEROGENEOUS COMPUTE

- Overview of technology
- Discuss use of GPU
- Special SAD instructions
- Some challenges

# *HOW TO "UNDO" THE SHAKE?*

- First determine motion between two pictures.
- Powerful motion search instructions execute in parallel to compare pixels until close matches are found…

# HOW DOES IT WORK…

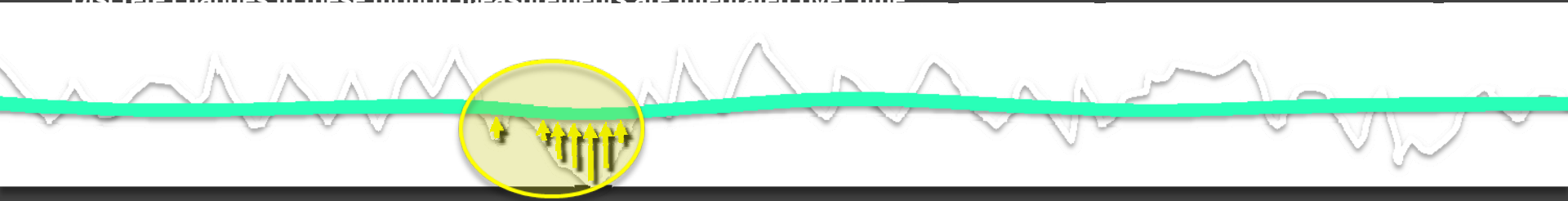
**Statistics are applied to derive models of the camera's motion**

Translation

Rotation around the optical axis

Zoom

**Discrete changes in these motion measurements are integrated over time**



**And a smoother motion path is synthesized for each…**

**Each picture is then warped (translated, rotated, scaled) from it's original position to a position on the smoother motion path**

# HOW DOES IT WORK…

- This can leave missing pixels at the edges. These are filled in using pixels from past pictures.

# *HOW DOES IT WORK…*

- These are back filled in using pixels from past and/or future pictures.
- These pictures are also warped into the right positions…

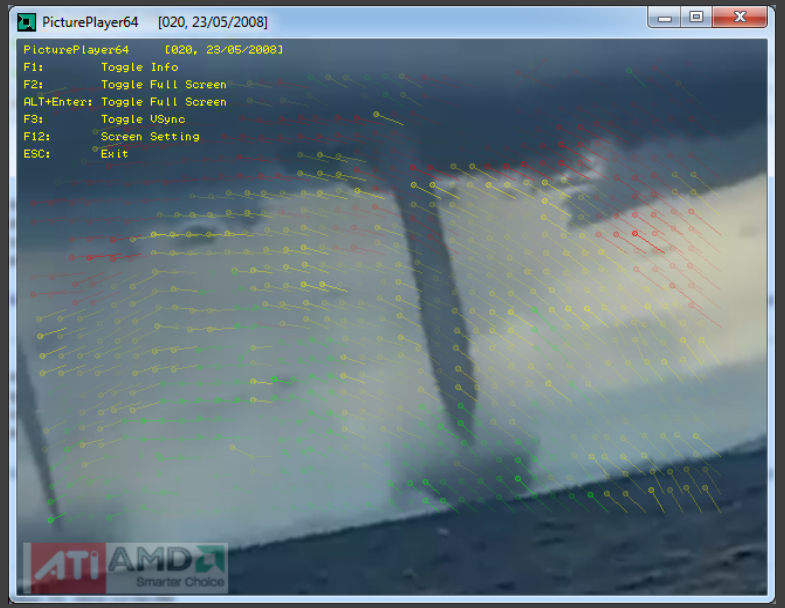
# HOW DOES IT WORK

- … and seamlessly fused together to form a new image.

# PERFORMANCE – ME REQUIRES LOTS OF SADS

- Random shifts +/-32 pels!
- Number of SADs/sec for 1920x1088@60Hz = **513,382,809,600!**
- Today
  - amd_sad16 instruction achieves 16 pels per cycle.
  - extra amd_byte_align instructions feed amd_sad.
- Near Future
  - Motion searches will be faster and more precise:
  - amd_qmsad will integrate amd_byte_align
  - masking capability will accelerate object segmentation

$$\sum_{j=0}^{j=m} \sum_{k=0}^{k=n} | I_0(i, j) - I_1(i, j) |$$



## *HOW THE COMPUTE IS DISTRIBUTED*

- Decode (Dedicated HW Block)
- Motion Search (GPU + CPU working together)
  - Brute force of GPU Parallelism combined with CPU's "intelligence"
  - Uses GPU's Multimedia Instructions (e.g. AMD_SAD)
- Statistics (CPU)
  - More Complex Code
  - Faster Time to Market
- Warp and Fuse (GPU)
  - Manipulating Pixels
  - Need the Speed of the Pixel Processing HW

Camera Shake Algorithm Uses Heterogeneous Compute

# *CHALLENGES*

Several GPU to CPU to GPU hand-offs

- Allows brute force of GPU to be mixed with “intelligence” of the CPU algorithm

But OS is not real time…

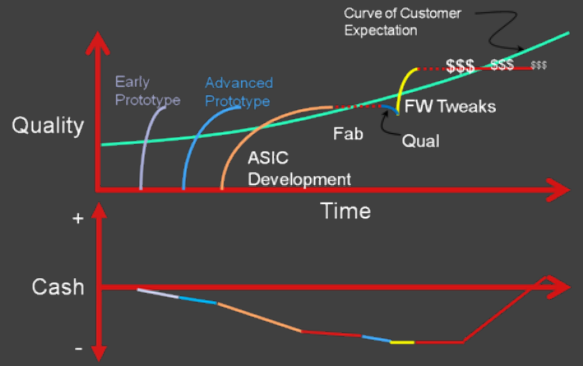
- Do frequent hand-offs make us susceptible to “performance crippling” interruptions? No.

Solutions:

- Overlapped CPU/GPU processing
- Built in latency tolerance and margin
- Assume multiple OS “ADD* events” occur infrequently
- Assume OS ADD events are short in duration
- Consider using gamer exclusive mode for some applications
- Consider shipping with a real time OS

* ADD == Attention Deficit Disorder

# SUMMARY




- Fusion parts can be viewed as an implementation medium
  - Adopt GPU accelerated *prototyping* first
  - When analyzing suitability for product factor in:
    - Time-to-market
    - Customer acceptance criteria increase over time
- Heterogeneous Compute
  - This is really about complementary feature sets
  - It's "The Best of Both Worlds"
- Expect to be *Blown Away*
  - Developers can make these parts to do very advanced signal processing
  - Very high computational density is achievable
  - Algorithm intelligence often trumps theoretical computational density of the implementation medium
  - Reaction time at adapting to change will literally blow you away

***QUESTIONS***

# Disclaimer & Attribution

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. There is no obligation to update or otherwise correct or revise this information. However, we reserve the right to revise this information and to make changes from time to time to the content hereof without obligation to notify any person of such revisions or changes.

NO REPRESENTATIONS OR WARRANTIES ARE MADE WITH RESPECT TO THE CONTENTS HEREOF AND NO RESPONSIBILITY IS ASSUMED FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

ALL IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT WILL ANY LIABILITY TO ANY PERSON BE INCURRED FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. All other names used in this presentation are for informational purposes only and may be trademarks of their respective owners.