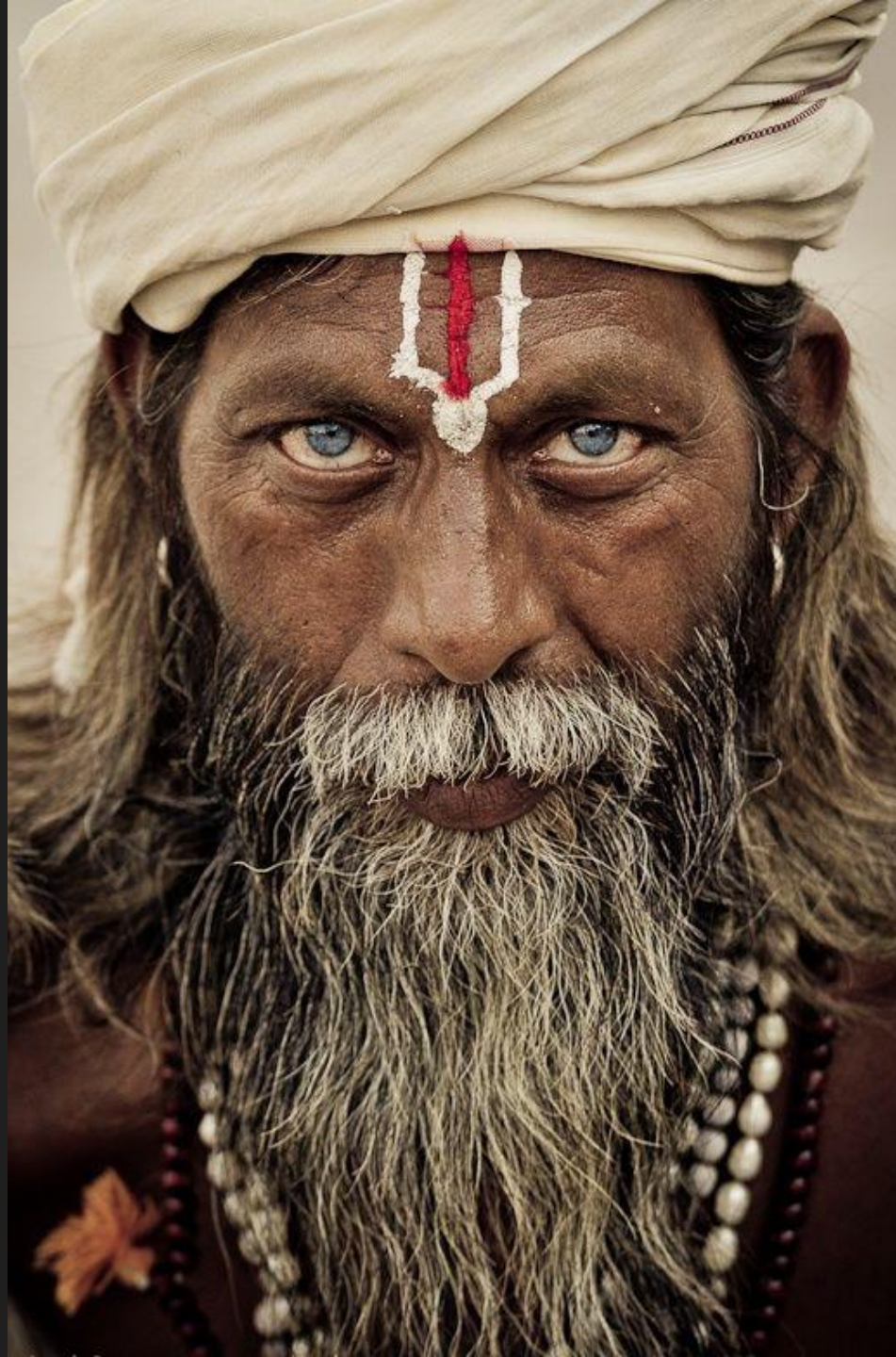


Misi

The Hand-held Devices Distribution Challenge

Using GL Programs and Smart Queue As Basis for on-line Multi-Platform Compilation of OpenGL Formulas

By Ela Shapira and Tamir Fridman - Managing Directors

www.misi-tech.com

Table of content

Introduction **3**

Purpose 3

Role of GPU programs 3

The Handheld Devices Distribution Challenge 3

Scope of Network Queue in filters distribution **4**

What is GL filter? 4

Design of a network-based compiler 4

Misi smart queue 5

Challenges of network-based compilation 5

Using GL program as network-based compilation buffer **6**

Understanding GL programs capabilities 6

The issue of compilation in hand-held devices 6

It’s all about easy distribution of new filters 7

Now let’s compile a network-based filter 7

Code Example 8

Summary **10**

Introduction

Purpose

The purpose of this paper is to describe new ideas regarding GPU programs as a key elements for network-based compilation and distribution of new graphic filters. These ideas are based upon review of the physics of the problem and an analysis of applicable technological approaches.

Role of GPU programs

A GPU program is a software code that represents mathematical manipulation on a matrix (image). The GPU runs the program concurrently for all the pixels in the image (parallel processing). Complexity of the program would affect the GPU performance by slowing down the frame rate.

The Handheld Devices Distribution Challenge

Today's availability of High Resolution video camera in every handheld device combined with the availability of high-speed internet, and the demand for re-compilation of the whole app for app-store distribution projects created the need to develop a mechanism that will allow easy distribution of new graphic filters without requiring extensive development expenses when releasing new graphical updates.

Scope of Network Queue in filters distribution

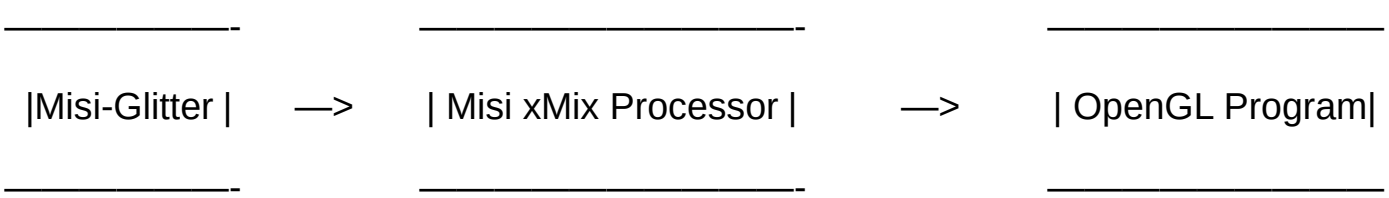
What is a GL filter?

GL filter is a set of GL programs which implements image processing filters like Sobel, Canny, Blur, etc. A single GL filter can run several programs in a pipeline to achieve advanced filtering methods.

Design of a network-based compiler

Design of a network based compiler for GL filters, tailored to project's specific requirements by receiving set of instructions from the network, generating the GL filter by preparing C code for it on the fly and finally, compilation as a GL program.

To achieve this we've predefined a set of GL programs as basic instructions set; with the right parameters it allows to create huge variety of filters with a very small instruction set – Misi Glitter is less than 300 bytes:



Scope of Network Queue in filters distribution

Misi smart queue

Misi Smart Q enables Xmix© processor to get Misi-Glitters over the network in real-time without the need to restart the software, and since Misi-Glitter is very small (approx. 300 bytes) the xMix processor can cache a lot of Misi-Glitters for offline mode. In addition, the small size of Misi-Glitter reduces network traffic and allows real-time update for GL programs to unlimited number of clients.

Challenges of network-based compilation

The main challenge when trying to compile GL programs on-line is to expose enough small units of GL programs that can be used as instructions. The second challenge is to make sure that the instruction payload (parameters) will be very small but yet generic enough to support all the instructions we've defined.

The last challenge is to make sure that the generated compiled code is running efficiently (by checking that the engine frame rate (FPS) is as defined). The efficiency test is important; while there are many GPU vendors with different performance issues, the common symptom for all GPU running inefficient programs is overheating, which can damage the device. Therefore, Xmix© processor performs efficiency test to the running GL program to remove Misi-Glitters from unsupporting devices.

Using OpenGL as network-based compilation buffer

Understanding OpenGL capabilities

Understanding OpenGL programs capabilities is a key element when trying to run a new formula from an on-line server. OpenGL can compile and run program during runtime; the compilation requires resources from the system and can cause delays. Since Misi-Glitter is an instruction set for predefined functions with known compilation time, the Xmix© processor can calculate the total compilation time to avoid delays.

The issue of compilation in cross-platform environment

Generation of new binary formula or even updating existing formulas in modern devices requires new software compilation. The reason for this, is the difference between Internet-based application and client-based application. For example, if we have a cross platform game (iOS/Android/Linux/Windows/Game Console) and we want to update the graphic engine, it would requires a huge team of designers, developers (for each platform), testers and sometimes even new software distribution (update).

Using OpenGL as network-based compilation buffer

It’s all about easy distribution of new formulas

Today, the main cost of every software change is the long process of definitions, development, Q&A, app-store tests etc. Now we can allow the designer to implement changes without the need for an expert GL developer to implement the new formulas, and to distribute the results immediately to the end devices without the need of long development and testing procedures.

Now let’s compile a network-based filter

Now let’s create a cartoonish GL filter and compile it during run-time. For this we need to use 24 bit source, regular color palette and 100% mix; then we’ll create the glitter we need to test it and save it to Misi-Net for distribution.

To run the filter we’ll need to update the Xmix© processor with the glitter.

Enjoy :)

Using GPU surface for data stream parallel processing

Code Example

```
//pseudo code


Create Filter(“maw”)
{
                        create Misi-Glitter
                        set source -> 24 bit
                        set mix -> 100%
                        save Misi-Glitter -> “maw”
                        Misi Smart Q send Misi-Glitter to Misi-Net
}
Test Filter (“maw”)
{
            xMix connect source -> camera
            Misi Smart Q get Misi-Glitter -> “maw”
            xMix load Misi-Glitter -> “maw”
            xMix connect output -> screen
}
```

Using GPU surface for data stream parallel processing

```
on (GL Frame)
{
        get frame from input
        create filtered image for MisiCV
        SYNC
        create filtered output
        SYNC
        Render
}
```

Summary

This paper presented the rationalization for the development of an easy way to create complex graphical filters and distribute them between multi-platform clients externally to standard pipeline of software development.

Thank you

Misi Tech Inc.

MisiCam video-camera and gaming console, download:

Download on the App Store

ANDROID APP ON Google play

[Watch our demo on YouTube]

Misi Tech INC.
340 S LEMON AVE #2396N , WALNUT, CA, 91789
www.misi-tech.com info@misi-tech.com

Misi