Lightstream technical whitepaper (WIP)

Main features of lightstream:

- Transfer data while in flight-mode
- Share files with full anonymity
- Make backups to disconnected devices
- Quickly share your clipboard

How it works:

1. The sender generates a series of QR codes that represents a file (sequences a file from arbitrary pieces of data) (File -> Packets)
2. The receiver scans the QR codes and recombines them into a file (sequences arbitrary pieces of data into a file) (Packets -> File)
3. The Receiver verifies the integrity of the file by checking the file hash (Asserts payload with "truncated-sha256-hash")

HCCQR:

High capacity color quick response code

Problem:

- Splitting "HCCQR-frame-colors" into layers black and white layers
- Reading QR layers

Solution

- Split colors into layers (divide into quadrants, and distribute tasks to multiple cpu-cores)
- Read data from the first layer. Extra the meta-data and inform caller with meta-data (stop further reading of qr-layers if frame has already been read by StreamLib)
- Crop all successive layers after the "quad-meta-data" has been extracted from the first layer
- Return payload when all qr-layers has been read Features:
- Enables you to store more information in a QR image
- HCCQR uses the color spectrum and image analysis to transmit information
- 4 color map equals double capacity. (16 color map equals 4x capacity and so on)

Layers

More colors equals more layers of data:

16-colors: 4 Layers 32-colors: 5 Layers 64-colors: 6 Layers 128-colors: 7 Layers 256-colors: 8 Layers

HSV-color-spectrum:

HSV model is robust to illuminance level changes, especially for the differentiation of red, green and blue colors. In order to carefully differentiate black from other color

Duplex sync protocol (2-way protocol)

Basically the duplex sync protocol enables you frames that was missed. It's more convenient for bigger data transfers but it's also less safe than the simplex sync protocol.

Simplex communication

Simplex communication is a communication channel that sends information in one direction only For example, in TV and radio broadcasting, information flows only from the transmitter site to multiple receivers. A pair of walkie-talkie two-way radios provide a simplex circuit in the ITU sense; only one party at a time can talk, while the other listens until it can hear an opportunity to transmit. The transmission medium (the radio signal over the air) can carry information in only one direction.

Caching missed frames as a video

1-way communication requires that you dont miss frames, if you do, you have to wait for the loop to return to the parts that was missed. It's there for crucial to cache all frames. That way, if the camera records "an-in-between" frame (aka a blurry, unusable frame) you can jump back a few frames and try to recover the frame. We only try to extract data at the original fps of the screen, in order to conserve cpu/gpu capacity. Also blurry frames, or shacky frames can still be useful by combining more frames into one.

Redundancy (error correction):

Lightstream uses 7% reduandancy.

L - [Default] Allows recovery of up to 7% data loss M - Allows recovery of up to 15% data loss Q - Allows recovery of up to 25% data loss H - Allows recovery of up to 30% data loss

Capacity:

Capacity is determined by many factors:

- Each QR layer can store from 21bytes to 2953bytes (0.0021 - 2.9kb)
- The special lightstream QR library can store raw bytes in QR frames (not publicly available)

CPU / GPU

GPU's has 1000s of threads and are efficient at extracting colors/data. But GPU is tricky to program for. Right now everything is done on the CPU. which has only 4-6 cores. AS such a big bottle-neck is programming for the GPU right now.

Screen HZ (FPS / refresh rate)

Current screens flicker at 60fps. You need a camera of at least 120fps to make use of that.

Camera FPS

From iPhone7 cameras has 240fps, But CPU is only capable of extracting data at 10-20fps right now. When GPU is programmed for. We can probably use way higher FPS speeds.

Speed:

1. 500byte-QR code + 30FPS Screen + 60FPS camera + 8-colored-HCCQR + GPU data extraction = 30 * 0.5kb * 3 = 45kbps
2. 1000byte-QR code + 60FPS Screen + 120FPS camera + 32-colored-HCCQR + GPU data extraction = 60 * 1kb * 5 = 300kbps
3. 1000byte-QR code + 120FPS Screen + 240FPS camera + 64-colored-HCCQR + GPU data extraction = 120 * 1kb * 6 = 720kbps

4. 1000byte-QR code + 120FPS Screen + 240FPS camera + 256-colored-HCCQR + GPU data extraction = 120 * 1kb * 8 = 960kbps

- These numbers are based on unpacked binary-byte-data. Using packed binary byte data we can increase the capacity 2-fold ref: https://stackoverflow.com/questions/38010550/what-is-the-packed-binary-data-and-unpacked-binary-data-in-iso-8583-message and https://www.tads.org/t3doc/doc/sysman/pack.htm
- Some file formats can also be compressed. Powerpoints, raw images etc can see 50% reduction in size. JPEG, MP3, And MP4 are already compressed and will not reduce in size.

Public / private transfer encryption scheme

Public-key cryptography, or asymmetric cryptography, is a cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are known only to the owner. Lightstream is primarily a one way transfer technology. so Pub / pri encryptions isn't easily achievable unless you use the duplex transfer option. Since lightstream is a local and requires direct line of sight, encryption isn't essential.

File verification

File verification is the process of using an algorithm for verifying the integrity of a computer file. This can be done by comparing two files bit-by-bit, but requires two copies of the same file, and may miss systematic corruptions which might occur to both files. A more popular approach is to generate a hash of the copied file and comparing that to the hash of the original file. The lightstream

hashing library produces a small hash of files I.e: "173539113" (it needs to be small, because space is scarce but it's still enough to not cause collisions) In contrast sha256 is very long: 235ef3f11f9d0d7e79c64c380f939b6b1c6c1ef7c34fceaa31075114c7280c49 Lightstreams Filehasher is very fast and makes persistent hashes across devices and sessions (unlike apples own)

Partial file compilation

- The data comes in asyncronouse, and blocks of data can finish before other blocks of data.
- The FileStream library ensures that each block or segment of data is put into the correct place. Similar how the bittorrent technology works

Cross-platform:

- Lightstream is built from one modular codebase (native swift) and deploys to mac and iOS simultaneously.

Competitors

Cobra: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.307.7160&rep=rep1&type=pdf

Rainbar: https://www.researchgate.net/publication/283877651_Rain_Bar_Robust_Application-Driven_Visual_Communication_Using_Color_Barcodes