Project Proposal

Panoramic Image Stitcher

To: Professor Mike Boctor

February 11th, 2016

By:

Group 5

Evan Kennedy

Brian Magnusen

Punleuk Oum

Description

Image stitching is currently an application limited to expensive digital cameras and higher-end smartphones, and is lacking for those without such devices. While there are commercial panoramic image solutions in the market, the vast majority of them are applications that run on the client side, which can be a limiting factor on the client's ability to produce a panoramic image.

Our solution is to provide a web-based panoramic image stitching service. The architecture of the proposed cloud panoramic image stitching service is straightforward. Users are possibly able to provide raw images from multiple sources to a locally stored list, and send a request along with the images to our server to be stitched. On the server side, the panoramic image stitching engine will be powered by the OpenCV library (Open Computer Vision). As the server completes a panorama, it is sent back to the user to be viewed in a list either via local storage or a server-side database. Because panoramic images vary in terms of projection, viewing them is generally not considered a trivial task for the end-user. To this end, we provide a browser-based viewing application that will render the variety of panoramas that we support.

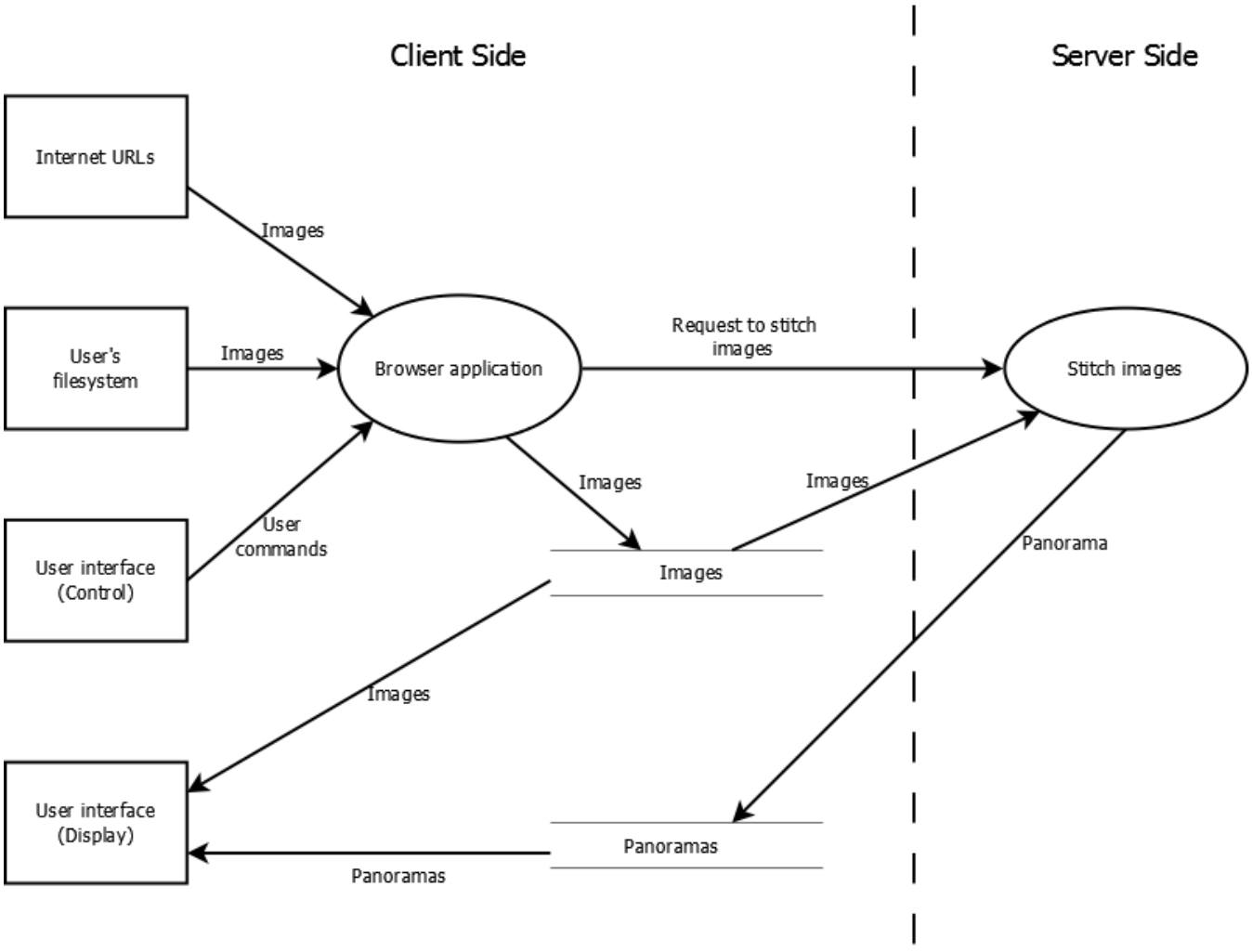

Figure 1 - Data Flow Diagram

Several optional user stories have been defined and will be implemented if time allows.

Sprints

Sprint 1

Sprint 1 focuses on core stitching implementation, image importing, and image container development.

User Stories	Tasks
As a user I would like to add images from my local filesystem	• Configure server with web server • Add browser-based image container • Develop image container display capability • Add UI elements for specifying and adding an image from filesystem • Develop "file-picker" type interface for image file selection • Develop image container file input interface
As a user I would like to stitch images to create a planar panorama (create only, no viewing at this point)	• Develop basic version of server-side OpenCV-based image stitching application with support for planar panorama stitching • Develop client-side interface to host for sending image stitching requests • Develop server-side interface to host for initiating image stitching requests • Add UI elements for stitching controls and status/progress reporting
As a user I would like to load sample images provided by the server	• Add UI elements for accessing the server's sample images • Develop interface for pulling images from server storage into client's image container

Sprint 2

Sprint 2 focuses on panorama viewing, cylindrical panorama support, image deletion, and URL image importing.

User Stories	Tasks
As a user I would like to view a panorama	• Develop client-server interface for retrieval of stitched images • Add browser-based panorama container based on existing image container • Add UI elements for panorama viewing
As a user I would like to stitch a cylindrical panorama	• Add cylindrical panorama support to image stitching application • Add UI elements for specifying panorama type • Add support to client-server request interface for panorama type parameters
As a user I would like to import images from a URL	• Develop URL retrieval interface • Add UI elements for specifying and importing a URL image • Integrate URL retrieval mechanism with image container interface

As a user I would like to delete an image	• Add UI elements for specifying and deleting an image • Develop image container interface for image deletion
As a user I would like to delete a panorama	• Add UI elements for specifying and deleting a panorama • Develop image container interface for panorama deletion

Sprint 3

Sprint 3 focuses on panorama viewing flexibility, built-in user help, and saving panoramas to local filesystem.

User Stories	Tasks
As a user I would like to cancel in-progress stitching requests	• Develop client-server interface for request cancellation • Add UI elements for request cancellation
As a user I would like to view help documentation	• Generate help content on image stitching basics and website functionality • Add UI elements for accessing help content
As a user I would like to be able to zoom and pan around a panorama	• Develop pan and zoom functionality in image container display component • Add UI elements for panning, zooming, and related controls
As a user I would like to save a panorama to my local filesystem in PNG format	• Develop image export interface for image container • Develop client-side file saving functionality • Add UI elements for panorama file saving

Sprint 4

Sprint 4 focuses on image stitching and saving options, expands built-in help, and adds sharing via social media.

User Stories	Tasks
As a user I would like to stitch a spherical panorama	• Add spherical panorama support to image stitching application • Add UI elements for specifying spherical panorama type • Add support to client-server request interface for spherical panorama type parameters
As a user I would like to view tooltips for all user interface elements	• Generate tooltip description content for UI elements • Add UI elements for tooltip support
As a user I would like to share a panorama via social media	• Develop interface to social media site • Add UI elements for sharing via social media
As a user I would like to convert a panorama to JPEG when saving to local filesystem	• Develop file conversion support • Add UI elements for image file type specification

As a user I would like to crop a panorama	• Add UI elements for cropping controls • Add client-side interface for image cropping to panorama container

Technology Stack

Client-Side

- AngularJS
- HTML5 Canvas
- HTML5 Local Storage
- Pre-processing / compiling
    - Babel (JS)
    - LESS (CSS)

Server-Side

The server side stack could be improved further by using DynamoDB for things like sessions, user credentials, and any indexing of objects in S3 as it provides basic query / sorting.

Simple (option 1)

- AWS EC2 Container Service (would be www subdomain)
    - Load balancer (built in)
    - Docker
    - OpenCV
    - Node.js
- AWS S3 bucket for public files such as images or scripts (would be cdn subdomain)
- AWS S3 bucket for private files such as user data and stored images

"Server-less" (option 2)

- AWS S3 bucket open to public for website hosting (would be www subdomain)
- AWS S3 bucket for private files such as user data and stored images
- AWS API Gateway to accept request (would be api subdomain)
- AWS Lambda functions to perform tasks from API Gateway