

360도 영상 변환을 위한 HTML5기반 클라우드 포토스티칭 어플리케이션 구현

*유성근 *정서경 *박상일

서울과학기술대학교

*orcogre@gmail.com, *tjrud1438@naver.com, *sangilparkmaikl@gmail.com

Development of Cloud Photo-stitching Application for 360 degree video converting based HTML5

*Yoo, Sunggeun *Jung, Seo-kyung *Park, Sangil

Seoul National University of Science and Technology

요약

최근 기존의 설치형 어플리케이션으로 제공되던 워드 프로세서나 스프레드 시트 등이 RIA(Rich Internet Application)와 Ajax(Asynchronous JavaScript and XML)같은 기술의 발달로 구글 독스(Google Docs)와 같이 웹브라우저(Web Browser)에서 동작하는 형태의 클라우드 기반 웹 어플리케이션으로 제작되어 널리 사용되고 있다. 또한 기존의 웹 기술로는 구현하기 어려웠던 동영상 포맷 변환이나 사진이나 동영상에 필터를 적용하는 것과 같은 영상처리가 가능한 클라우드 기반 웹 어플리케이션이 등장하고 있는 실정이다.

이에 본 논문은 컴퓨팅 자원을 많이 사용하고, 장시간의 변환이 필요한 360도 영상의 변환과정에 꼭 필요한 기술인 포토스티칭(Photo-stitching)을 클라우드에서 가능하게 하는 프론트엔드 및 백엔드를 구현하고, 웹소켓(Websocket)기술을 활용하여 실시간으로 변환결과를 전달 받을 수 있도록 하였다.

1. 서론

클라우드 컴퓨팅은 컴퓨팅분야에서 인터넷을 구름으로 표현하던 것을 비유하여 마치 구름처럼 컴퓨팅 자원을 언제든지 가져와서 이용하고, 사용한 만큼만 비용을 지불하는 컴퓨팅 서비스를 가리키는 말이다[1]. 2006년 구글의 엔지니어인 크리스토프 비시글리(Christophe Bisciglia)가 유휴 컴퓨팅 자원에 대한 활용을 위하여 만든 개념으로 현재 구글 독스(Google Docs)와 같은 생산성 어플리케이션을 포함하여 수많은 서비스들이 클라우드 컴퓨팅을 기반으로 구현되어 있다[1].

HTML5와 Javascript 기술의 성숙으로 인하여 OS(Operating System)의 API(Application Programming Interface)로 구현하여야만 하였던 네이티브 어플리케이션(Native Application)이 웹 기반으로 구현되기 시작하였다. 2005년 2월 18일에 미국 Adaptivepath라는 회사에 올라온 Jesse James Garrett가 작성한 'Ajax : A New Approach to Web Applications'이라는 에세이에서 시작한 Ajax(Asynchronous JavaScript and XML)은 웹브라우저 내의 DOM(Document Object Model)을 새로고침 없이 조작할 수 있고, 한 페이지 내에서 비동기로 웹서버에 여러번의 응답을 요청 할 수 있어 기존 네이티브 어플리케이션과 같은 수준의 사용성과 효율적인 서비스를 제공 할 수 있게 되었다[2].

또한 서버 쪽에서 클라이언트에게 바로 메시지를 전달 할 수 없는 HTTP의 한계를 극복하고 실시간 통신을 가능하게 하기 위한 기술로 Ajax Long Polling이나 Streaming등의 방법이 고안되어 실시간 처리

나 서버쪽 푸시 메시지(Server-side Push Message)를 전달하는 방법이 고안되었다. 그러나 HTML5의 일부분으로 웹소켓(Websocket)이 도입되어 웹에서도 서버와 클라이언트간에 실시간 양방향 통신이 가능해지면서 채팅과 같이 즉답성을 요구하는 서비스부터 파일변환과 같은 시간이 오래걸리는 서비스까지 실시간으로 결과를 확인할 수 있다.

본 논문에서는 컴퓨팅 자원을 많이 사용하고, 장시간의 변환이 필요한 360도 영상의 변환과정에 꼭 필요한 기술인 포토스티칭(Photo-stitching)을 클라우드에서 가능하게 하는 HTML5기반의 프론트엔드(Front-end)와 NodeJS 기반의 백엔드(Back-end)를 구현하고, 웹소켓(Websocket)기술을 활용하여 실시간으로 변환결과를 전달 받을 수 있도록 하였다.

2. 구현된 백엔드 어플리케이션

포토스티칭 백엔드를 구현하기 위하여 기존의 파노라마 포토스티칭 오픈소스 소프트웨어인 OpenPano를 사용하였다. 리눅스 기반(Ubuntu 16.04.1)의 Amazon AWS 클라우드 서버에 Cmake와 Eigen, libjpeg-dev등의 의존성 패키지를 설치하고 소스코드를 빌드하였다. 빌드 후 리눅스 터미널에서 실행 파일명과 붙여야 할 사진의 파일명을 입력하고 출력 파일로 그림 1.과 같이 포토스티칭이 완료된 사진이 나오는 것을 확인하였다. 백엔드 서버로는 Node.js v4.2.6을 사용하였다. Node.js는 JavaScript 서버/런타임으로비동기 Non Blocking기반 웹서버로 사용가능하며 NPM(Node Package Manager)으로 다양한 패

키지를 설치하여 기능을 확장할 수 있다. 사용한 Package 목록은 다음과 같다.

- child_process(이미지 스티칭 프로세스 실행)
- Express(API 구현)
- multer(여러 파일 업로드)
- Bluebird(비동기 프로세스에 promisify 위해 적용)
- Socket.io(웹기반 실시간 통신 위해 사용)

Input: overlap, brightness & contrast & color mismatch, position mismatch



Output: position matching, blending, cropped



그림 1. 포토스티칭을 위한 원본 파일과 결과물의 모습

3. 구현된 프론트엔드 어플리케이션

구현된 프론트엔드 어플리케이션은 HTML <input>태그를 이용하여 사용자는 포토 스티칭할 이미지를 선택한다. Multiple 을 이용하여 여러 개의 파일을 선택 가능하며 사용자가 선택한 파일은 FileReader API를 통하여 웹 화면에 나타나게 된다. FileReader 객체는 파일이나 블롭 오브젝트(Blob Object)를 통하여 웹 어플리케이션이 사용자의 컴퓨터에 있는 파일을 읽을 수 있다. 현재 File API는 그림 2와 같이 IE, FireFox, Chrome, Safari, Opera 브라우저를 지원하고 있으며 모바일에서는 지원 불가능하다는 것이 특징이다.

포토 스티칭 프론트엔드를 구현하기 위해 jQuery라이브러리를 사용했다. jQuery는 모든 브라우저에서 동작하는 클라이언트 자바스크립트 라이브러리로 문서 객체 모델(DOM)과 관련된 처리를 쉽게 구현할 수 있다. 또한 시각적 효과를 쉽게 구현하여 웹 표준만으로도 플래시와 실버라이트로 구현한 것과 비슷한 수준의 시각적 효과를 제공할 수 있는 특징이 있다. 본 프론트엔드에는 jQuery 3.2.1버전을 사용했다.

Feature	Desktop		Mobile			
	Chrome	Edge	Firefox (Gecko)	Internet Explorer	Opera	Safari (WebKit)
Basic support	13	(Yes)	3.0 (1.9.1) 7 (7)	10.0	11.5	6.0
name	?	(Yes)	3.6 (1.9.2)	11.0	?	?
lastModifiedDate	?	No support	15 (15)	?	?	?
webkitRelativePath	(Yes)	(Yes)	49 (49)	?	(Yes)	(Yes)
size	?	?	?	11.0	?	?

그림 2. 각 브라우저 버전별 File API 지원현황

3. 구현된 전체 어플리케이션

구현된 전체 어플리케이션의 모습은 그림 3과 같다. 구현된 웹 어

플리케이션은 스티칭 할 사진을 선택하면 순서대로 웹브라우저에 표시되게 된다. 그리고 업로드 버튼을 누르면 콘솔창에서 일어나는 스티칭 과정의 메시지가 웹소켓을 이용하여 실시간으로 브라우저에 나타나게 되고 완료되면 웹브라우저에 결과물이 바로 출력되어 다운받을 수 있게 하였다.

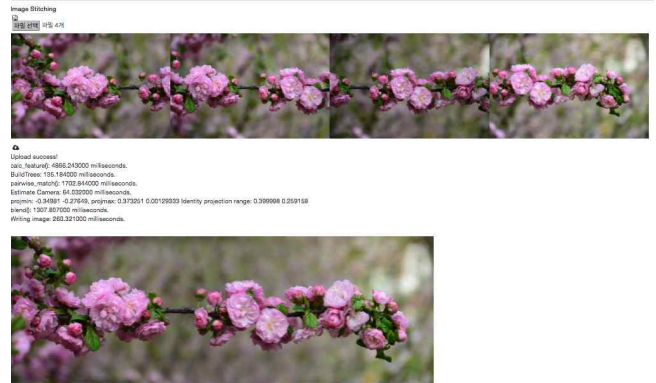


그림 3. 실제로 구현된 포토스티칭 웹어플리케이션의 모습

4. 결론

본 논문은 클라우드 컴퓨팅과 HTML5, AJAX, 웹소켓등의 기술을 이용하여 포토스티칭이 가능한 웹 어플리케이션을 구현하였다. 본 어플리케이션은 많은 시간이 걸리는 포토스티칭과 같은 영상처리 작업에서 웹기반의 어플리케이션으로도 충분히 기존 네이티브 어플리케이션을 대체할 만한 사용성과 효율을 낼 수 있다는 것을 보여주었다. 이 기술은 더 많은 시간과 자원을 사용하는 360도 영상을 클라우드 컴퓨팅으로 처리할 수 있다는 단초를 제공하였고, 차후 360도 영상변환을 클라우드에서 처리하는 어플리케이션을 구현할 예정이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2016-0-00144, 시청자 이동형 자유시점 360VR 실감미디어 제공을 위한 시스템 설계 및 기반기술 연구]

참고문헌

- [1] 김남규, 박성욱, 고미현, 서신원. “클라우드 컴퓨팅 기술의 연구동향 분석,” 한국기술혁신학회 학술대회, 597-607., 2015
- [2] 김영보, “웹 어플리케이션의 새로운 패러다임, Ajax (상)” 한국오라클, 2006
- [3] https://en.wikipedia.org/wiki/Push_technology
- [4] 강윤희, “IoT 환경에서 WebSocket을 활용한 솔기없는 메시징 응용 설계,” Proceedings of KIIT Summer Conference, 245-247, 2015