**VideoPhotoTourism – Using Video Input to Extend Phototourism**

Scott Rick

University of Wisconsin-Madison

CS 534 - Computational Photography

Spring 2011

**Abstract**

VideoPhototourism is a project which aims to extend phototourism applications to support video input. Phototourism applications such as Microsoft Photosynth and PhotoCity Game are able to create interesting 3D representations of popular landmarks and tourist locations by uploading, analyzing, and combing photos. My project is an application which makes incorporating videos into phototourism applications as simple as possible. Videophototourism is as simple as preprocessing videos using my MATLAB script to create video still images which are used as input to Photosynth. From one or more videos of tourist landmarks, a beautiful photosynth recreation of the tourist location is recreated online to share and enjoy.

**Introduction**

In this paper, I discuss the motivation for my project, explain the problem solved by my project, describe related work, give details of my videophototourism theory, describe my entire method from video input to photosynth output, offer experimental results, and give concluding remarks. I will refer to the created 3D landmark locations as photosynths and the saved images captured from video input as video stills.

**Motivation**

I found that when I visited tourist landmarks, in addition to taking photos, I would often also capture videos of the landmark. When playing around with Microsoft’s PhotoSynth program and the PhotoCity Game, I came up with the idea of creating a program that could extend these phototourist software applications to allow video input.

**Problem Statement**

A user of phototourism applications should be offered the option of using video as input to recreate tourist landmarks. Extending phototourism applications to use video input is a matter of preprocessing videos to create useable video stills. Video stills captured from video input should be able to create a phototourist landmark (photosynth) on their own or with supplementary photographic images.

**Related Work**

Microsoft Photosynth and PhotoCity Game are the two most important related works. The research required to create these applications was conducted at the University of Washington, Cornell University, and Microsoft Research. I choose to interface with the Photosynth application because I found it to be much more reliable and the results to be easier to share.

**Theory**

Applications such as Photosynth and PhotoCity Game use photos from tourist landmarks as input images which can be used to recreate interesting tourist locations. My idea was to use videos to create these images instead. Since tourists already take videos of tourist landmarks, it is a reasonable application to extend phototourism applications to allow video input. Instead of forcing users to capture video stills manually, my project expedites the process, creating hundreds of video stills from multiple video clips in just seconds.

**Method**

For each PhotoSynth I created, I followed this methodology. It evolved as I created more experimental results, but here is the final method used to create photosynths from video inputs.

1. *Convert Video Format:*

MATLAB function mmReader and VideoReader do not support all video formats (such as .mov (QuickTime movies)), and it is essential to use as input only supported types of video files. For my project work, I started with QuickTime movie files (.mov) and discovered that Windows versions of MATLAB’s builtin video reading toolbox didn’t support these types of files. So, I needed to convert my files to a supported format, such as AVI (.avi), MPEG-1 (.mpg), or Windows Media Video (.wmv, .asf, .asx). I choose to use AVI video format and converted my files using Media Converter, a free web application which allows conversion of audio and video files between formats.

1. *Select* *Videos:*

Then, I selected the videos that I wanted to use as input for the vidoephototourism application. Then I renamed their videos as v1.avi to vX.avi, where X is the number of video inputs. In addition to renaming the input videos, I placed my videos in the folder with final.m, my MATLAB script. Also, I specified a folder name in my script (variable outputFolder) where I wanted the video still output files to be saved. The video stills I used for each photosynth were saved in a different directory to keep them organized and to enable me to compare the source video, video stills, and the created photosynth. I need to manually update the variables in my script related to number of videos I am inputting and frame options.

1. *Run final.m MATLAB Script:*

I then run my script on each video in the directory. It reads in each .avi video, captures a single frame, and then writes the images as a JPEG file to the specified directory. I initially tried to save video stills as .PNG bitmaps to avoid loss of color data, but Photosynth doesn’t recognize .PNG files. The script runs quite quickly, taking less than a minute regardless of the number of input videos.

1. *Rotate Images*

From taking videos in landmark as well as portrait orientations, I found that PhotoSynth had a hard time realigning images that were rotated 90° relative to other images. So, as an additional step, I needed to review the video stills and manually rotate any images that were rotated from my desired alignment.

1. *Remove Images*

I found that in some of my videos, I had focused on myself or other people unrelated to the tourist landmark, and these images were not useful in creating photosynths. They were simply unused by Photosynth, so I decided to manually remove them before running the Photosynth application to lower the number of video stills which the application had to process.

1. *Create Photosynth:*

I then ran Microsoft’s Photosynth application and select all the output video stills and photographic images I wanted to use to create the photosynth. This was the slowest part of the process, taking 15 minutes or more on some input. I found that the time increases with the more photos used as input, and I suspect that larger, higher quality photos also slowed the process. Restricting the total number of images to a maximum of 300 allowed me to keep the runtime of Photosynth at a reasonable length.

**Experimental Results**

The measure of quality of created photosynths is subjective, but I will be using the following criteria.

1. *Percent “Synthy”*

Microsoft Photosynth uses the term *synthy* to describe how well the photos in a photosynth match together. From the FAQ on Photosynth’s webpage [[1](http://photosynth.net/faq.aspx)] :

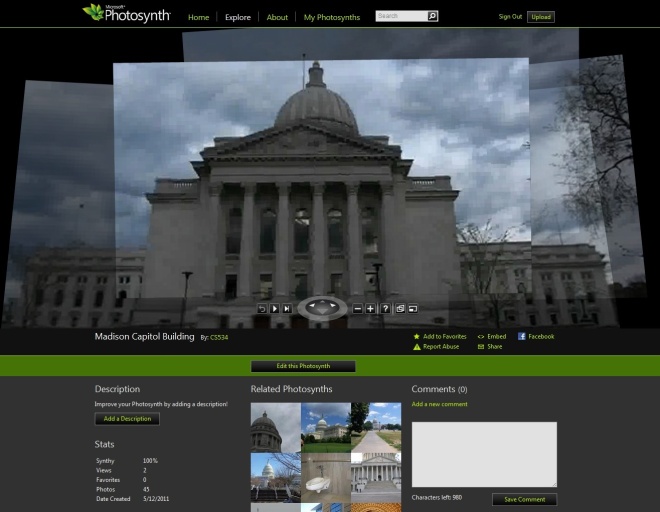
Synthy is how we describe really well matched synths. If you create a synth with 100 photos and 100 photos match together, that would be 100% synthy. The synthy % isn’t a perfect quality measurement on its own though. A 4 photo synth that is 100% synthy doesn’t make the best synth. In geek speak: it’s the probability that one can navigate between two photos selected at random, without replacement.

So, for each experimental result, I will report the synthy percentage given by Microsoft’s Photosynth application.

1. *Subjective Quality*

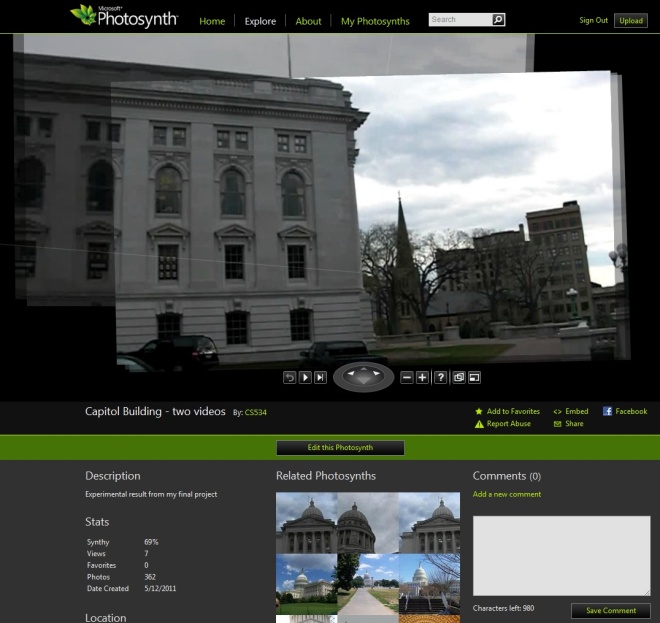
The other criteria I will use is a simple opinion of how well I feel that the results achieved the goal of recreating the tourist landmark as a 3D phototourist location. I will be careful to mention the various advantages of using video input as well as the drawbacks that I observe.

My first experimental results come from using only a single video as input. I used only the first 45 photos from 182 available and created a photosynth from these images. This resulted in very similar images overall being added to the photosynth because they were from the same few seconds of the same video.



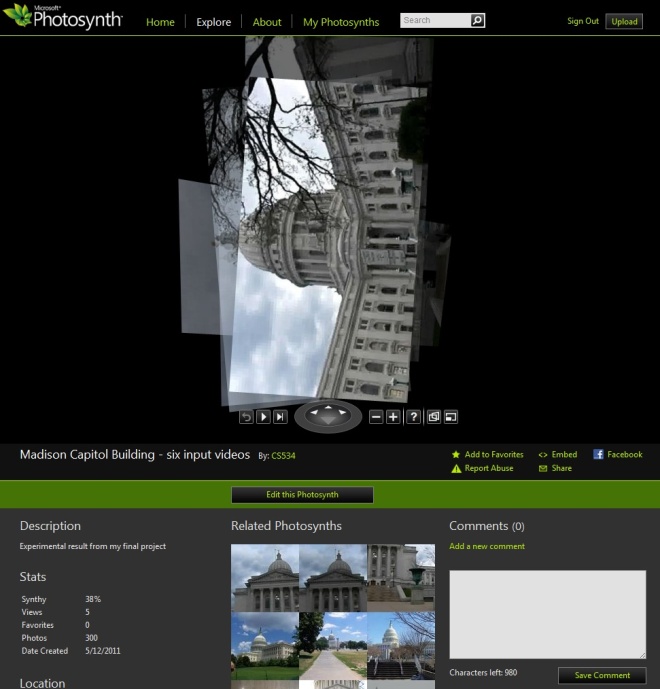
First photosynth parsing a single video as input. 100% Synthy [[2](http://photosynth.net/view.aspx?cid=48ed8852-aa54-4d1d-9aef-0ff0b30fec69)]

My next experiment was to try to use multiple videos to create a photosynth. I took a small step forward and created a photosynth from 382 video stills from 2 videos. The results were satisfactory, but Photosynth was slow in creating the photosynth. After reading the warning that more than 300 photos might take a long time, I tweaked my script to make the max video stills from all videos to be 300. The images which were not included in the final photosynth were from the parts of the video which used a very different perspective. I had expected Photosynth to include these images and was a little disappointed in the lackluster results.



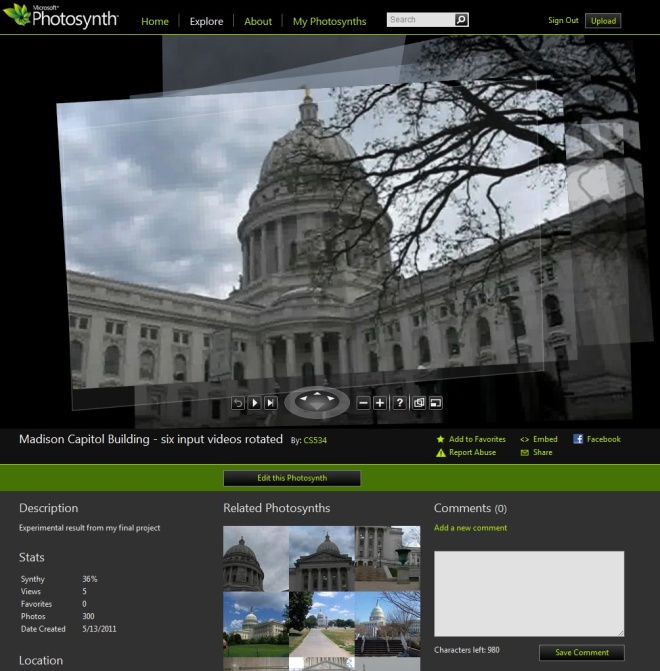
First photosynth parsing two videos as input. 69% Synthy [[3](http://photosynth.net/view.aspx?cid=2977848b-bd92-47bb-b88d-985eed1b4107)]

My next experiment was to use six videos to create a photosynth using many input videos, but I had problems getting Photosynth to match images from many different videos. The most obvious problem was that some of the images from the videos were rotated because I took a video with portrait orientation. Photosynth matched up all of the portrait images and ignored all other input.



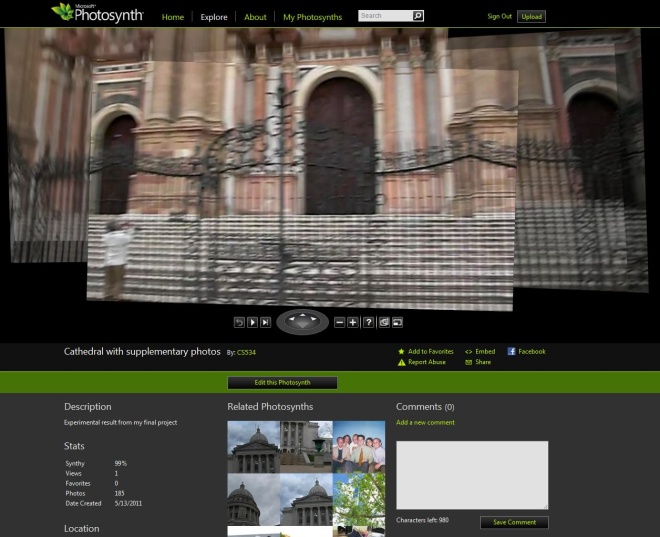
Initial results from using six videos as input. 38% Synthy [[4](http://photosynth.net/view.aspx?cid=d4cf38cc-a192-4479-a7c2-045a559b4ffa)]

Next, I rotated the problem images, expecting to receive better results. The results weren’t much better. All of the images from one video were matched, but all other videos were ignored. I believe that the problem was with using videos which used too different of perspectives. My project should work for videos taken from more similar locations, or with one long video taken while walking around. The rotation fixed the problem of the photosynth showing up sideways in the browser, but a lot of images are still missing.



Rotated results from using six videos as input. 36% Synthy [[5](http://photosynth.net/view.aspx?cid=d4cf38cc-a192-4479-a7c2-045a559b4ffa)]

Lastly, I attempted to use both video and photographs as input for videophototourism. I thought that it would be interesting to combine the many images of a video input with the high quality images of photographs. The most interesting part of this photosynth is that the high quality results blend in quite nicely with the lower resolution results from the video stills.



Results from using one input video and several images as input. 99% Synthy [[6](http://photosynth.net/view.aspx?cid=8ebef3f3-2998-4d5c-af68-85bfb8c09040)]

All of my Photosynths are located at the CS 534 Photosynth collection page. [[7](http://photosynth.net/userprofilepage.aspx?user=CS534&content=Synths)] I have tagged them all with my name and the word VideoPhotoTourism. I also mentioned in their descriptions that they are the experimental results from my final project.

A common problem with all of the Photosynths was that they were lower quality that using higher resolution photographs to create the Photosynth. However, I found the overall quality to be satisfactory, and it was interesting to be able to mix in photos with video stills.

**Concluding Remarks**

I would like to extend my experimentation to include HD cameras in order to get high quality video input. In some cases, creating Photosynths from only video stills results in much lower quality images in the Photosynth. I expect that HD video recorders would solve this issue by providing higher quality image stills after separating the video into frames. It is an unsatisfying fix to simply use better hardware to improve Photosynth quality, but I think this would be the best way to improve results.

Motion blur is another improvement which could be added to supplement my application. While my video inputs had low motion blur, most videos of landmarks captured by tourists can be expected to have a somewhat higher degree of motion blur present in them. I was very careful while moving around and capturing the images and was able to avoid noticeable motion blur in the video stills.

I would like to incorporate my script as an extension to my panorama project from homework four. It would be interesting and impressive to be able to create panoramas from one or more videos.

Another interesting application for my script would be to use videos to create many video stills to use as input for the PhotoCity Game. All of the images would be easy to capture using video recording, so it would be much easier to score points in the game. I would like to see how effective using videophototourism would be in getting high scores in PhotoCity Game.

I would like to test my project on a very long video which simulates a walking tour of a landmark. I think that the resulting photosynth would provide many interesting viewpoints of a landmark, without undergoing the trouble of matching many different videos.

**References**

Photo Tourism  
<http://phototour.cs.washington.edu/>

Noah Snavely, Steven M. Seitz, Richard Szeliski, “[*Photo tourism: Exploring photo collections in 3D*](http://phototour.cs.washington.edu/Photo_Tourism.pdf),” ACM Transactions on Graphics (SIGGRAPH Proceedings), 25(3), 2006, 835-846.

Microsoft Photosynth   
<http://photosynth.net/>

PhotoCity Game  
<http://photocitygame.com/>  
<http://grail.cs.washington.edu/projects/photocity/>

Kathleen Tuite, Noah Snavely, Dun-Yu Hsiao, Adam Smith and Zoran Popovic. “[*Reconstructing the World in 3D: Bringing Games with a Purpose Outdoors*](http://www.cs.washington.edu/homes/ktuite/fdg2010/photocity-fdg2010.pdf).” (Foundations of Digital Games 2010).

MATLAB Documentation  
<http://www.mathworks.com/help/techdoc/>

Media Converter  
<http://www.mediaconverter.org/>

**Links**

[1]  
<http://photosynth.net/faq.aspx>

[2]  
<http://photosynth.net/view.aspx?cid=48ed8852-aa54-4d1d-9aef-0ff0b30fec69>

[3]  
<http://photosynth.net/view.aspx?cid=2977848b-bd92-47bb-b88d-985eed1b4107>

[4]  
<http://photosynth.net/view.aspx?cid=d4cf38cc-a192-4479-a7c2-045a559b4ffa>

[5]

<http://photosynth.net/view.aspx?cid=733475e5-00d4-4192-b0ad-1a13afb3bae3>

[6]  
<http://photosynth.net/view.aspx?cid=8ebef3f3-2998-4d5c-af68-85bfb8c09040>

[7]  
<http://photosynth.net/userprofilepage.aspx?user=CS534&content=Synths>