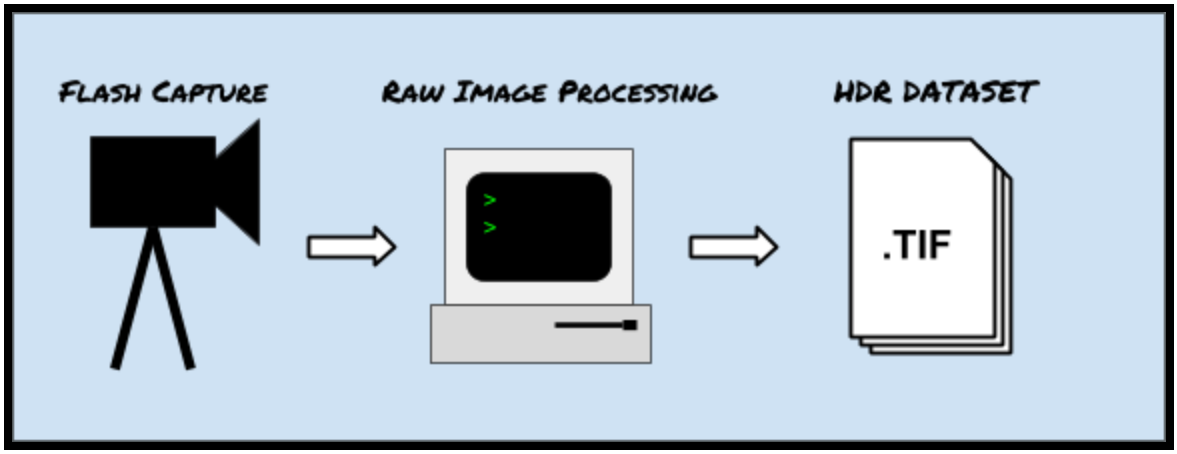
Computational Periscopy Dataset

OVERVIEW

Project Summary:
Given an input photograph, possibly augmented with extra data such as depth, we wish to recover an image of the scene that lies outside the capture frame. Real-world images often contain this information in a highly degraded form. For example, reflections on shiny objects in a photograph form a tiny, blurry and distorted image of the scene behind the photographer. Similarly, the tonality variations along shadow boundaries are manifestations of the light distribution on the other side of an occluder. We want to leverage (deep) supervised learning to extract this hidden information from images. This includes an algorithm to reconstruct an approximation of the scene outside of the capture frame. An important component of this work is to assemble a rich training dataset of images. For this, we will use a custom rig to record HDR pictures of multiple indoor scenes under several lighting conditions.

Dataset and Capture Pipeline Summary:
The dataset and capture pipeline were designed so they could be used for learning something about the materials and lighting in the scene. The pipeline has two major steps: capturing the images, and processing them. Capturing images is done with an automated camera setup that directs a flash at numerous angles and photographs the scene at each angle with multiple exposures. Processing is completed with a suite of python programs that use the raw sensor information from the camera to generate a HDR image for each flash angle. The programs also help extract environment maps from the light / diffuse probes that are placed in the scene.



MATERIALS

Camera Rig

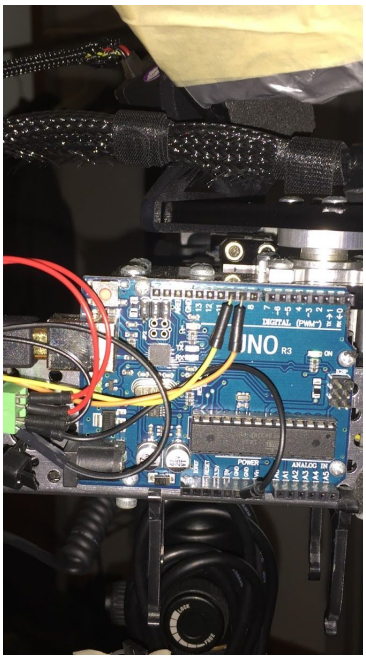
- Sony a6500 Camera Body
- Carl Zeiss Sonna E 1,8/24 ZA 45390321 Lense
- Wallmex Pro APTARIS Camera Cage
- C1350 RoadTrip MeFoto Carbon Fiber Tripod
- Rotating Flash
    - HS-645MG Ultra Torque HITEC Servos (2)
    - XINY 6V 6A AC/DC Switching Power Adapter
    - Sony HVL-F43M External Flash
    - Arduino Uno R3
    - Wall Power Extension Cord
    - USB Extension Cord

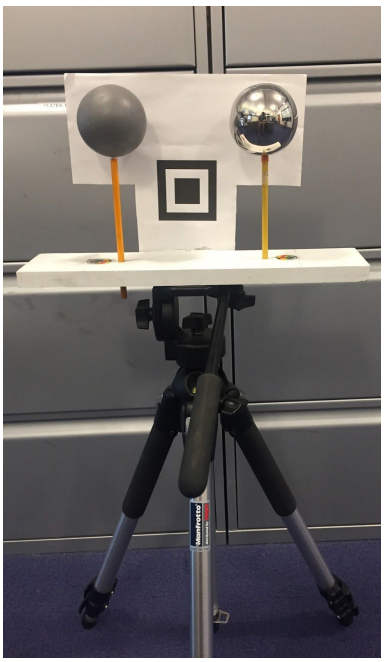
Light / Diffuse Probe

- 3in Painted Wooden Sphere - SEM Color Coat 15393 Medium Gray
- 3in Mirror Sphere “Garden Gazing Globe”
- Manfrotto 3126 and Manfrotto 3021PRO Tripod

Computer

- Windows Laptop (Dell Precision M3800)
- Smartflash Software





CAPTURING

Scene Capture Steps (~15min)

1. Turn on Camera and Position the Camera Rig and Probes
2. Connect the Power Supply to outlet
3. Connect the Laptop to USB
4. Turn on Camera's Wifi Remote Control (Menu -> Applications List -> Smart Remote)
5. Connect the Laptop to the Camera's Wifi
6. Make sure Flash is OFF
7. Run flashcapture.exe with a descriptive metadata file name
    - `$ ./flashcapture.exe --mode "all" --output "capturemeta/aug31_baloons.txt"`
8. Once 3 photos have been taken, turn Flash ON within 5 seconds to finish capture

Detailed Setup Descriptions

- Electrical Connections
    - Laptop is connected to the USB hub (which has 2 USB connections) via extension
    - Power Supply is connected to wall power (with extension cord if needed)
    - Servos are connected to Arduino
    - Flash is connected to Camera
- Positioning
    - Camera is ~1m away from probes
    - Probes are centered in the frame
    - Fiducial marker is just above bottom of the frame
    - Camera tripod is fully extended out (width) and has 3 segments extended (height)
        - Measurement from floor to middle of camera is 43.5in
    - Probe tripod is fully extended out (width) with no segments extended (height)
        - Measurement from floor to middle of spheres is 34in
- Flash / Camera Configuration
    - Flash on ¼ power
    - Flash on M (manual) mode
    - Flash on 105mm zoom
    - Camera on Manual Focus (turn autofocusing off in settings)
    - Camera on M (manual) shooting mode

Design Decisions

- Minimized probe area in frame
- Lower power flash and delays between photographing to prevent flash overheating
- Tried ~8 different diffuse probes spray painting different materials, wood was the best
- Improved wire management so the rig's connections are robust and neat

Flash Capture Program

- To run the program with default parameters (easiest way) you only need a name of the output metadata file, but custom amount of flash angles, delay, and exposures can be controlled. (--help option for more)
  - delay = 5000 (milliseconds)
  - capture_settings = 16,200 5.6,1000 5.6, 6400
    - High Exposure: *ƒ*/16 and ISO 200
    - Middle Exposure: *ƒ*/5.6 and ISO 1000
    - Low Exposure: *ƒ*/5.6 and ISO 6400
  - subdiv = 2
    - 25 flash angles
- The program is in smartflash/Build/Debug
- Uses HTTP requests to control the camera and USB to control the flash servos

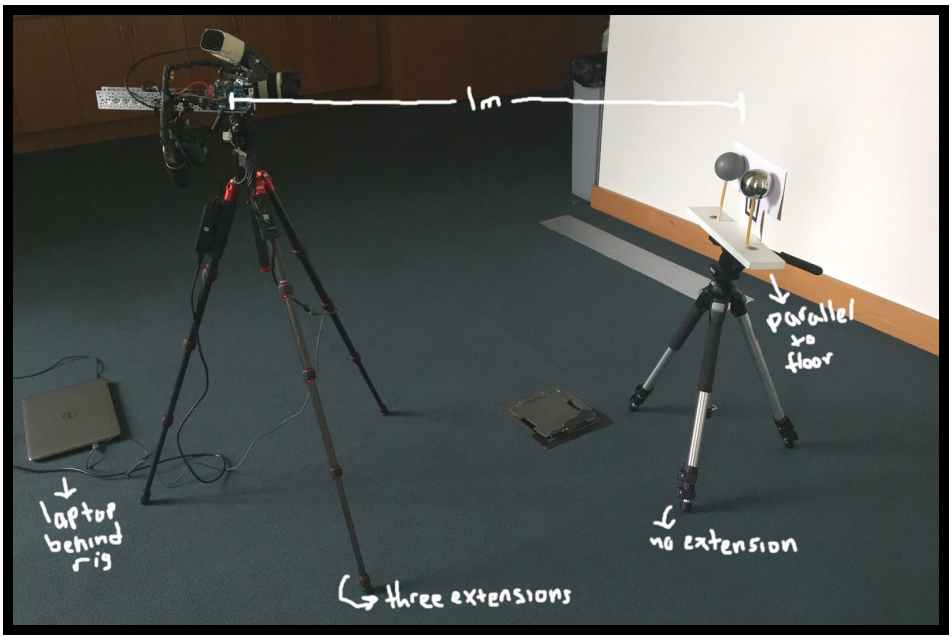
Tips

- Deciding on a scene
  - Many different materials
  - Probe as close to subject materials as possible
  - Lights off is better but not necessary
  - Small enough room for light to bounce
  - Not busy so people don't walk through the capture
  - Permission to use the space
- Managing batteries' charge (laptop, flash, camera)
  - The Camera battery will die first but when it's low there is a yellow battery symbol on the screen. When the camera battery is changed, the flash batteries should be too, for ease and to help mitigate overheating. If two sets of camera and flash batteries are used, one set can be charging while the other is in use. This allows for the cycle of changing batteries to be repeated for indefinite back-to-back captures.
  - The laptop battery, if charged overnight, should last a whole day's worth of capture. Plugging in the laptop charger during capture is cumbersome, so it's easier to have it fully charged before. Lowering brightness also helps.
- Storage
  - Each capture is 20gb and the card holds 256gb so there is space for 12 captures before you have to offload the SD card
- Transporting
  - Short Distance
    - 3 trips back and forth for camera, light probe, and computer + cables
    - Keep everything set up
  - Long Distance
    - 1 trip with everything using a hiking backpack / car

        - Camera Rig: Remove the power cable from the DC supply and remove the USB extension cord. Compress all of the legs on the tripod fully. This is easy to carry
        - Probes: Remove the probes from the block and remove the pencils from the probes. The cardboard backboard is attached by velcro and can also be removed. Safely pack these parts so they aren't ruined in transport. The tripod can be strapped on to the backpack
        - Remember to pack the chargers, batteries, and bring along any troubleshooting tools like a multimeter just in case.
- Efficiency
    - If you are efficient, the only bottleneck is SD card space. A day's worth of capture is 12 back-to-back captures (~4 hours)
    - Captures take about 15min, so you can work on something else / read a paper, but the laptop will be in use, so you might have to bring two.
    - If you are going don't have to pack everything up and unpack. This way, you don't have to pack up everything and unpack it.

Future Improvements

- Test without honeycomb cover on flash
- Design a better way to transport the rig and probes
- Make sure all automatic focusing is off during a capture
    - Right now, some images are blurry because the AF adjusts during a severely underexposed capture and then carries over into subsequent photos
- Power supply / converter to flash so batteries don't need to be used
- Fan / cooling on flash so delay can be shortened (decreasing capture time)



PROCESSING

Materials

- Linux computer
- SD card reader
- Python programs

Steps

1. Connect the USB SD card reader to the linux computer and insert the camera's SD card
2. Move photos from the SD card (under DCIM) to AFS
    - `$ mv -v  /media/rmarten/disk/DCIM/1137081 /media/hdd3tb/rmarten/rawimages`
    - This takes ~10min so continue on to the next step
3. Upload laptop's metadata files to the same folder as the photos
    - One option is to upload the .txt files to google drive on the laptop and download them on the linux computer
4. Run script to separate image dump into directories for each scene
    - `$ python organize_raw_images.py --meta ../rawimages/1137081/*.txt --dest ../rawimages`
5. Run script to convert exposure sets to HDR and extract light probes
    - `$ python raw_processing_pipeline.py -i ../rawimages/table -m ../rawimages/table/*.txt`
    - Needs to be repeated for each scene (~6min for each one)

Metadata

- Name the file in the format "<date>_<unique-scene-description>.txt"
- Some of the early captures don't have the ambient HDR capture so the NOFLASH rows are not present

```
id phi theta fname
NOFLASH -- -- _DSC5802.ARW
NOFLASH -- -- _DSC5803.ARW
NOFLASH -- -- _DSC5804.ARW
0 -2.12437 1.5708 _DSC5805.ARW
0 -2.12437 1.5708 _DSC5806.ARW
0 -2.12437 1.5708 _DSC5807.ARW
```

Image Files

- Currently stored in /media/hdd3tb/rmarten/rawimages on the linux computer
- rawimages/scenes is the current dataset with consistent setups
- rawimages/testing contains tests of different iterations of the pipeline

Programs

- organize_raw_images.py
    - Separates image dump from SD card into directories defined by capture metadata
- raw_processing_pipeline.py

    - Main processing program: Deals with file conversion and HDR merging
    - NOTE: The part dealing with sphere detection / extraction is commented out
- filenaming.py
    - Used by raw_processing_pipeline to generate useful names for new files
- generate_spheres.py
    - Used by raw_processing_pipeline to detect light / diffuse probes
- winedngconvert.py
    - Incomplete program that uses the wine windows emulator to batch convert .ARW files into DNG files. If this part eventually works, then it can be tagged onto the beginning of raw_processing_pipeline.py
- test programs
    - Various individual elements of the raw_processing_pipeline used for isolation and testing. The name of the file is the feature that it is testing.
- Use --help on any python program to get an in depth description and usage

Dependencies

- Python, cv2, numpy, matplotlib
- rawtherapee
- hdrmerge
    - Qt
    - LibRaw
    - Exiv2
    - ZLib

Future Improvements

- unwrapping perspective instead of orthogonal for BOTH probes
- improved circle detection
- proper .ARW support
    - Using hacky Windows DNG converter….
- Visualizer (can use this software)
- COMMIT smartflash changes to LUKAS GITHUB
- Backup data to network storage
    - Currently in /media/hdd3tb/rmarten/rawimages

NOTES

Ryan Marten (9/01/17)

So far I have collected 20 scene captures. The capture rig and processing pipeline have been optimized so it is extremly easy to acquire more scenes. Everything has been automated except for finding locations (this is probably the hard part) and moving all the equipment to the location.

The scenes include locations around CSAIL that contain interesting materials. Some data may be unusable if there was a problem with the autofocus or a person walking into the capture.