Light and Water Drops

Peter Barnum

CMU-RI-TR-11-11

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

May 2011

Thesis Committee:
Takeo Kanade, Co-chair
Srinivasa G. Narasimhan, Co-chair
Lee Weiss
Ramesh Raskar, MIT

Abstract

Water drops are present throughout our daily lives. Microscopic droplets create fog and mist, and large drops fall as rain. Because of their shape and refractive properties, water drops exhibit a wide variety of visual effects. If not directly illuminated by a light source, they are difficult to see. But if they are directly illuminated, they can become the brightest objects in the environment.

This thesis has two main components. First, we will show how to create two- and three-dimensional displays using water drops and a projector. Water drops act as tiny spherical lenses, refracting light into a wide angle. To a person viewing an illuminated drop, it will appear that the drop is the same color as the incident light ray. Using a valve assembly, we will fill a volume with non-occluding water drops. At any instant in time, no ray from the projector will intersect with two drops. Using a camera, we will detect the drops locations, then illuminate them with the projector. The final result is a programmable, dynamic, and three-dimensional display.

Second, we will show how to reduce the effect of water drops in videos via spatio-temporal frequency analysis, and in real life, by using a projector to illuminate everything except the drops. To remove rain (and snow) from videos, we will use a streak model in frequency space to find the frequencies corresponding to rain and snow in the video. These frequencies can then be suppressed to reduce the effect of rain and snow. We will also suppress the visual effect of water drops by selectively "missing" them by not illuminating them with a projector. In light rain, this can be performed by tracking individual drops. This kind of drop-avoiding light source could be used for many nighttime applications, such as car headlights.

Contents

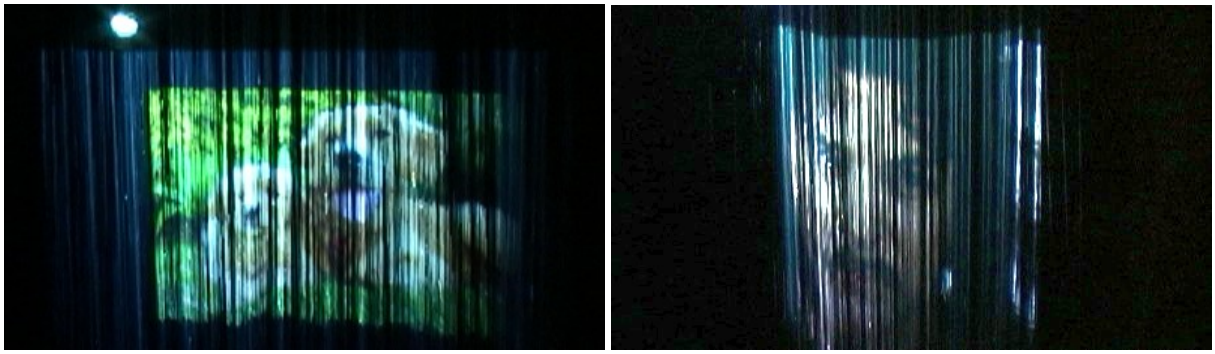
List of Figures

Introduction

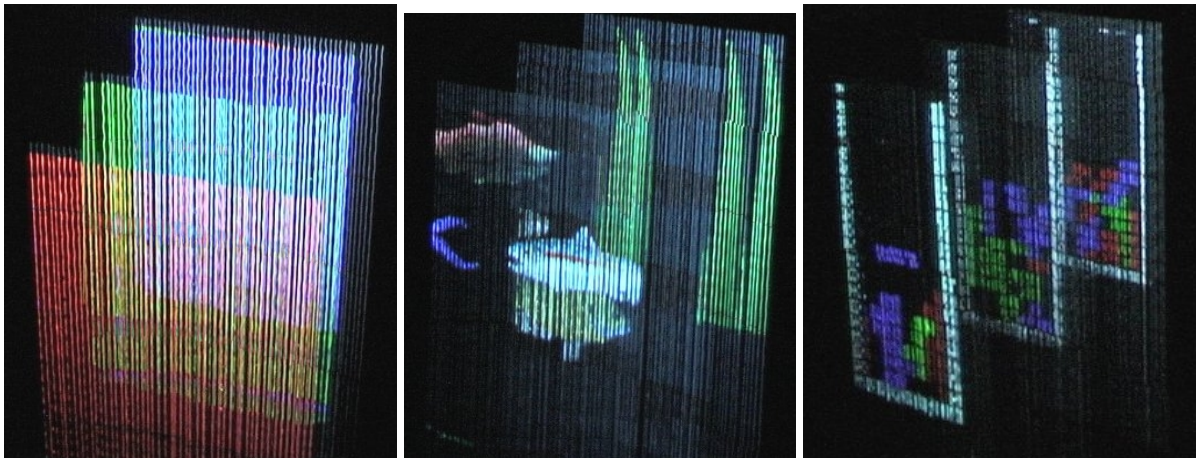
Water drops appear regularly in our daily lives. During rain showers, millimeter sized drops create familiar rain streaks in videos. And when an area is foggy due to numerous microscopic droplets, objects appear blurred and light sources cast halos. In movies, these effects can be used to add ambiance to an outdoor scene, enhancing its emotional content. But for computer vision applications, these effects can prevent accurate recognition, tracking, and color detection. Water drops cause these effects due to their unique nature when illuminated.

Due to the combination of shape and refraction properties, the appearance of water drops varies dramatically under different types of illumination. Water drops are spherical and transparent, meaning they act as tiny fish eye lenses. These little lenses collect light from various environmental light sources, such as the sun or street lamps, and refract it out at a wide angle. This means that if they are illuminated directly, they will be among the brightest objects in the environment. However, the opposite is also true. If a drop is *not* directly illuminated, then it will be nearly invisible. Few objects in the world have such variable appearance.

The visual effect of water drops has been explored in various fields. In computer vision, the work of Narasimhan *et al*. [Nayar and Narasimhan, 1999] [Narasimhan and Nayar, 2002a] examined the effect caused by microscopic drops. [Garg and Nayar, 2004] [Garg and Nayar, 2005] researched the case where drops are large enough to be individually visible. In optics, the low-level physics of drop-light interactions have been discussed by researchers such as Van de Hulst [Van de Hulst, 1957]. In meteorology, the visual effect of rain has been used to detect the



(a) Examples of a 2D water drop display that cannot handle occlusion



(b) Examples of a 3D water drop display that can handle occlusion

Figure 1: (a) Two example 2D displays. The image on the left is a flat sheet of water, used much like a standard projection screen. But the image on the right demonstrates how the surface can form any non-occluding shape, such as a curved surface. Part of our proposed work is to extend this type of 2D display to allow for multiple occluding layers. (b) Two example 3D displays.

properties of individual raindrops [Schönhuber et al., 1994] and [Löffler-Mang and Joss, 2000]. And in entertainment, water drops are used to create art [Pevnick, 1981] [Eitoku et al., 2006b].

We will explore the interaction between water drops, cameras, and lighting. First, we will create a water drop display using a camera, projector, and valve assembly. Second, we will develop algorithms and apparatus to make rain and snow more difficult to see both in videos and in the real world.

Part One: *A water drop display*

If there are water drops falling in a volume and we have control of a projector, then a colorful

2D display can be created. Shining colored light on a drop will cause it to appear to be that color. Because of this effect, the drops can be used as a projection screen. The drops do not need to fall in a plane, so we are not limited to a flat screen. Any shape where drops do not occlude each other can be used, including curves and discontinuities.

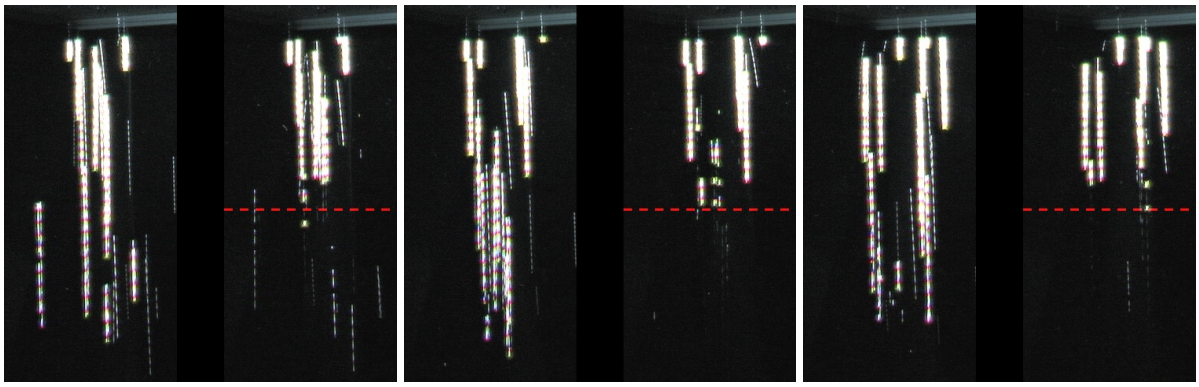
To create a 2D display, we will use a camera, projector, laser plane, and water drop generator. The individual drops are located by the image of their intersection with a laser plane. Based on the geometric calibration between the hardware, a user can "paint" the drops from the point of view of the camera, causing the projector to illuminate them with the correct color at the correct time. Figure 1 (a) shows two examples of this type of 2D water-projector display.

If we have control over both a projector and a valve assembly to emit drops, then a 3D display can be created. Control of the drops with a valve assembly allows a volume to be filled with a large number of drops that do not occlude each other. Specifically, we will construct a 3D display out of a collection of 2D layers. Each 2D layer is created by a linear manifold of emitters. Each layer will emit the same number of drops every second (i.e. they will have the same frequency), but the emission times will be offset (i.e. they will have different phases). Because of this phase difference, no light ray from the projector will ever intersect two drops. Consequently, once we capture the drop locations with the camera, we can direct the projector to illuminate each drop with different pixels. If the drops are emitted at a high enough frequency, then the projector switching will be imperceptible. To a human observer, it will appear that all layers are illuminated with constant light.

Part Two: *Making rain invisible*

If rain or snow is viewed by a camera, then it will appear as motion blurred streaks that are irritating to people and confusing to vision algorithms. Individual streaks are difficult to detect. But by creating a streak model in frequency space, we can detect rain and snow's global spatio-temporal properties. These frequencies can then be suppressed. This will reduce or even eliminate rain and snow in a video. Figure 2 (a) shows how snow streaks can be removed from a video sequence.

(a) Postprocessing to remove snow from a video



(b) Real-time lighting control to make water drops invisible

Figure 2: By performing a spatio-temporal frequency analysis on a video sequence, we can detect and remove the streaks caused by rain and snow. (b) Extending this idea to the real world, and using a projector to illuminate everything in a scene *except* for the drops.

And if we have control over the light source, then we can shine light everywhere except the drops. This means that the drops will be invisible in the real world. By using a stereo camera rig and a fast-switching projector, then individual drops can be tracked, and the projector can be directed to not illuminate them. Figure 2 (b) shows the visual difference when drops are illuminated, and when they are not illuminated.

Part I

Creating displays with water drops

Chapter 1

A two-dimensional water-drop display

In this chapter, we describe how to create a animated display with a camera, projector, laser plane, and an un-actuated water drop generator. When it is created and calibrated, this display is able to show arbitrary visual content on a sheet of water drops. A key of this idea is that although no drop can occlude another from the point of view of the projector, the shape of the projection surface can be arbitrarily curved, complex, or disjointed, allowing for a variety of different piecewise-distinct 2D configurations in the 3D space. With some creativity, it is even possible to give the appearance of a 3D structure, largely because the human visual system does not expect to see the occluded sides of 3D objects, etc.

Figure 1.1 (d) is a photograph of the setup from slightly behind the camera. The photograph shows the primary elements needed to calibrate and demonstrate the display. A simple container of water will release drops (or potentially streams) of water into the display area. Using a combination of the camera and the laser plane (Figure 1.2), the positions of these drops are determined, essentially via stereo triangulation. (The additional light is needed for calibration, as it can be difficult to detect the positions of waterdrops in some cases, due to noise of the camera causing difficulties in correctly detecting and locating small, individual drops). Based on geometric calibration between the camera, projector, and laser plane, we can warp different parts of the image

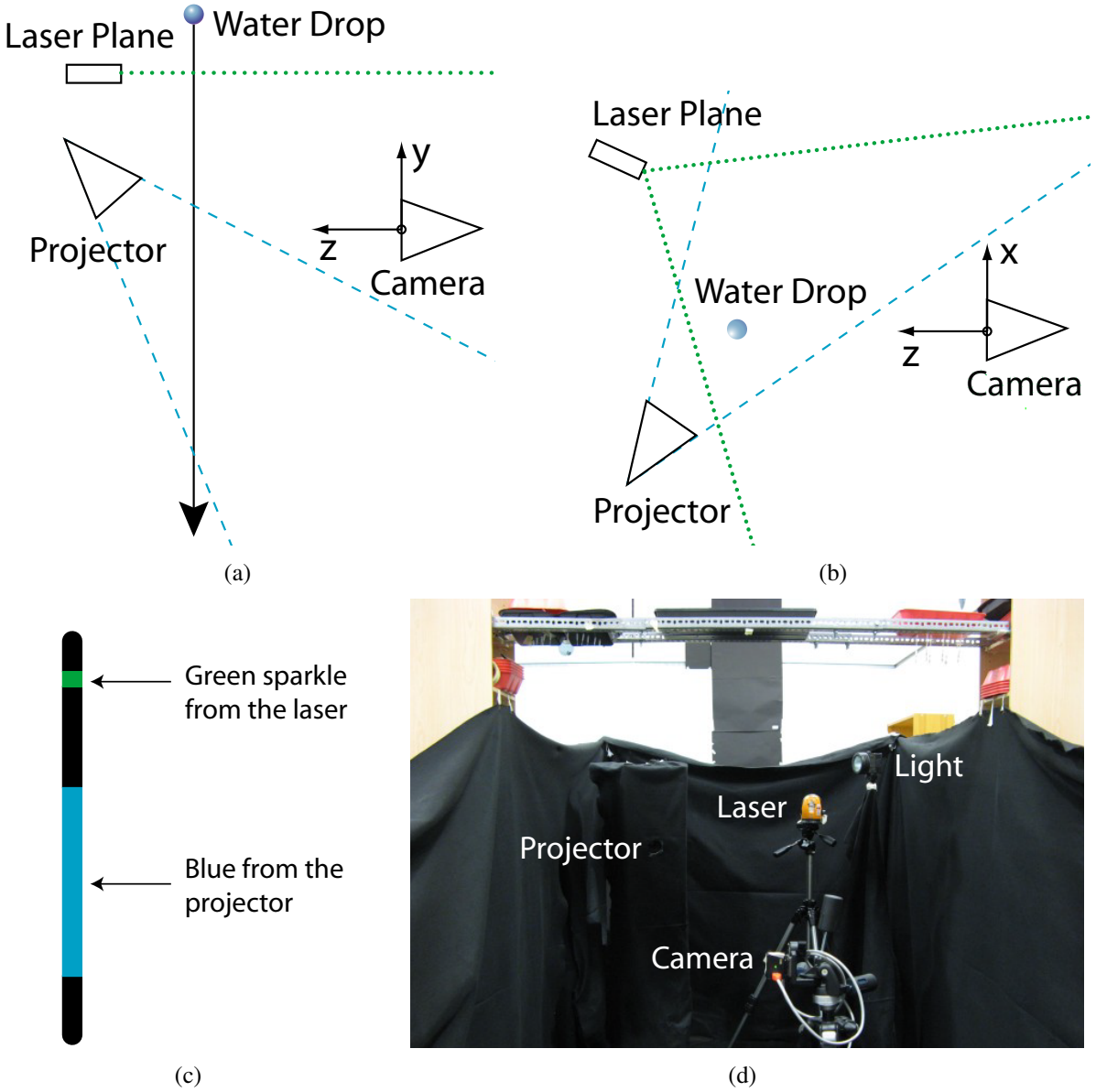


Figure 1.1: A side view (a) and top view (b) of a water drop as it crosses the laser plane and projector field of view. The image generated is shown in (c), and a photograph of the setup in (d). The photograph also shows the drop generator, which is a plastic box with plastic pipette tips or small holes as water emitters.

from the projector differently, projecting the correct image on the falling water drops.

In this chapter, we require that the drop generator be designed so that no drop occludes another from the point of view of the camera or the projector. If two drops occluded each other from the point of view of the projector, it would be necessary to project different colors at different times. Although it is not possible to display the image of an arbitrary 3D object with such constraints, we will show that some interesting effects can still be achieved. The display is animated by computing a movie made of the geometrically calibrated images, then projecting the movie on the drops. Due to the photometric properties of water drops, the light from the projector, reflecting and refracting within the drop, appears fairly constant across a wide viewing angle. The combination of the light from the projector with the falling drops makes it appear that colored lights are floating in midair.

Creating a display involves two steps. The "canvas" that describes the geometry of the object or scene being display is designed and created, then the canvas is painted with static or moving images.

First, the shape of the model must be designed. The exact characteristics or format of the model is not of prime importance, as long as the idea naturally lends itself to being displayed on sheets of water drops. The model could be a polygon mesh or a voxel occupancy grid. In either the polygon mesh or voxel occupancy grid cases, the main design concern is that the density of the model or graphical idea should be proportional to the number of water drops that are going to be used to display it. Creating a million-polygon mesh for display on fifty drops is not liable to be a good use of time, although the reverse might be true. Even if having a large number of drops is not necessary because of a sampling limit, filling in constant areas with more drops could make the result have a more pleasant appearance. And specifically for our setup, the end result of this model and idea creation is that the artist must decide where to punch holes in the drop generator. The locations of the holes determines where the drops fall in the environment, which is the three dimensional shape of the water canvas.

Second, the canvas is painted with a static or moving image. For the current work, we assume that although the canvas could change shape over time (though never in a self-occluding way,) the main observable change should be in the lighting from the projector. (Although in practice, the water drops tend to shift slightly over time, giving the entire display a wavy appearance, which could be used to enhance the perception of motion to a human observer).

Figure 1.1 shows an example of a single falling water drop. The drop is released from a hole in the water container, where it then begins to fall and accelerate. After a short time, it will hit the laser plane, and creates a bright sparkle. This sparkle will be easily visible from the position of the camera, marking the drop's location at that point in time. Then as the drop continues to fall, it is illuminated by the projector (the projector will be displaying bright white, making the drop as easy to detect as possible.) Because of the finite exposure time of the camera, the drop will appear as a motion-blurred streak in every image. One streak will have a bright green dot where the drop crosses the laser. An exact correspondence between the image, projector, and drop can be obtained by a one-time geometric calibration and the image location of the sparkle and the drop. With this correspondence, and the computed calibration, it will then be possible to direct the projector to illuminate future drops falling along the same path. Based on the model and coloring developed by the artist, the final result will be a colorful display.

1.1 Previous work on non-traditional displays

Some non-traditional displays create a picture by controlling the medium. These displays require large banks of valves controlled with high speed and precision. Examples of these use falling drops [Pevnick, 1981] [Rayner], bubbles [Heiner et al., 1999] [Nakamura et al., 2006] [Hirayama and Kakehi, 2010] [Suzuki et al., 2008], and fire [NAO Design] [Black Rock City, llc]. (A good review of other types is presented in [Moere, 2008]). In these cases, a constant light illuminates the apparatus, which emits particles in a controlled fashion. These particles act as voxels. For

example, a large water drop will act as a bright voxel, while collections of small water drops create many dim voxels.

Alternately, a display can be created by controlling the illumination. In order for the displays to be coherent, these methods involve precise design of the drop or fog emitters, to insure that they create predictable and constant streams. Some designs focus on abstract patterns. The water drop display by Eitoku *et al*. [Eitoku et al., 2006b] makes colorful three-dimensional patterns with a projector and falling water drops. Other designs emulate traditional displays, such as flat fog screens ([Araki et al., 1991] [Palovuori and Rakkolainen] [Kataoka and Kasahara, 1993] and others). Fog screens are especially suitable for interaction, as demonstrated in [Rakkolainen and Palovuori, 2004] [Rakkolainen and Palovuori, 2005] [Rakkolainen and Palovuori, 2002].

Sometimes these illumination displays can be based on wearable projectors or lighting. One example of wearable technology is the Sixth Sense project [Mistry and Maes, 2009a] [Mistry and Maes, 2009b], where the user wears a combined camera-projector rig that can recognize gestures and act as a glasses-free augmented reality experience, by using the projector to add pictures and text annotations to the world. Wearable displays can also include clothing, such as [Agnelli et al., 2004]. Other display augment static, architectural environments, such as Digital Kakejiku [Hasegawa], the Water Lamp projects [Dahley et al., 1998], [Ishii et al., 1998], or other types of ambient displays [Wisneski et al., 1998].

1.2 Geometric calibration

The key advantage of our method is that knowing the explicit 3D coordinates of each drop is unnecessary. The 3D coordinates would be necessary to create a full 3D display with arbitrary self-occlusions and the ability to automatically register a 3D model to the display's coordinates. However, in this chapter, we will focus on a certain subset of the set of possible 3D displays.

In our case, because the drops are not occluding each other, the artist will paint the drops

from the point of view of the camera. So all that is required is a homography between the image location of each drop and the projector location that will illuminate it. Each homography will only be used for a small part of the image, because each drop will only occupy a few of the image pixels. But even though drops are small, they do occupy various positions in the 2D image, meaning that we still require an entire 3x3 homography matrix for each drop. Note that this homography is not the same for every drop. We know which homography to use, based on the location of the sparkle where the drop crossed the laser plane. We will calibrate the display with a lookup table between sparkle locations and homographies. (This is in contrast to the idea of doing a full 3D-reconstruction, or even a homology based calibration, as shown in [Barnum et al., 2009b]. The homography lookup table can be created very quickly and automatically, and is sufficiently accurate for our case.)

A rear-projection screen is used for calibration. Any image that is displayed by the projector will be visible on the surface of the rear-projection screen from both the front and back. As a result, if we place the screen at depth z, parallel to the camera's image plane shown in Figure 1.1, then anything projected will be imaged by the camera. During the entire process of the calibration, the screen will be kept parallel to the image plane. Because the screen is always parallel to the image plane, the only different in the images captured by the camera is due to the screen's depth z, relative to the center of projection of the camera.

The laser plane will always be perpendicular to the image plane, as long as the camera is set up correctly. The cylindrical lens that generates the laser plane is set on a pendulum that causes the laser plane to be perpendicular to gravity. And as long as the camera is upright, as measured by a level, then the laser will also be perpendicular to it. (And in fact, even if the camera is not set up correctly, the angle error will at least be consistent, as the camera is stationary). However in practice, the screen can have some deviation from being completely parallel, and the calibration will still be adequately accurate. The computation of the homography fails gently as the screen becomes less and less parallel. And in fact, the variance due to non-parallel screens is liable to

be less that errors caused by varying drop trajectories.

But for now, we will assume that the screen is totally parallel to the camera's image plane and perpendicular to the laser plane. As a result, the laser plane will hit the rear-projection screen, then be imaged by the camera as a horizontal line at some height y. Due to perspective projection, the line has a unique y location for each depth z. For a plane Π_z that creates a laser line at y, there is a homography that warps the image coordinates to the projector coordinates.

This set of homographies that determines the correct plane based on laser height can be semi-automatically computed. First, the projector will be made to display a standard calibration checkerboard pattern. The person calibrating will take a rear projector screen, and carefully keeping it as parallel to the image plane as possible, sweep it through the display volume. The camera will capture a video of the screen as it moves, which will give a video sequence of different checkerboard patterns and laser lines. The height of the laser line can be automatically computed for each video frame. And to determine the homographies, the user first clicks points to manually determine correspondence between one of the images and the known checkerboard pattern that the projector displays. Using a standard KLT [Tomasi and Kanade, 1991] [Shi and Tomasi, 1994] [Lucas and Kanade, 1981] tracking algorithm, the point correspondences between the checkerboard patterns in all images can be easily and accurately computed. The final result is a lookup table of homographies between camera and projector, based on the height of the green line. With this mapping, all tracking and inference will now be done completely in the image coordinates.

1.3 Drop location calibration and display

The next step is to find the drops' locations in the image. Since the setup is indoors and we have total control over the lighting, we do not require sophisticated algorithms to find the locations of drops in the images. Instead, we use standard image processing techniques to background

Figure 1.2: Hardware for generating a red or a green laser plane. A laser is focused on a cylindrical lens, which splits it into a plane. The lens and emitter are positioned on a pendulum, making them perpendicular to gravity.

subtract, Gaussian blur, threshold, and segment with connected components. (This technique is explained in more detail in Chapter 4)

The background subtraction needs to be somewhat robust to camera noise, as there is relatively little light in the environment. It is performed by first computing the per-pixel median over a short sequence of images (approximately 20 images appears to be sufficient in this case.) Any pixel that is brighter than this median background is judged to be a drop. (Pixels that are darker than the model are often due to drop shadows, which should not be detected). In addition, because when bright images are imaged by the cameras we use (Point Grey Firefly MV,) a bright ghost tends to occur for several frames. As a result, we require that the drop images are not only brighter than the background model, but also brighter than the previous frame. This could cause issues if there are too many drops passing the same pixel in a short time period, but this is generally less of an issue than the "ghosts" appearing to be new drops.

We segment the streaks and track them by matching the endpoints of neighboring streaks across consecutive frames. We use the maximum exposure time for each image, meaning that the end of the streak in one image will be in almost exactly the same place as the beginning of the streak in the next image. This means that even if the drops are not falling exactly straight down,

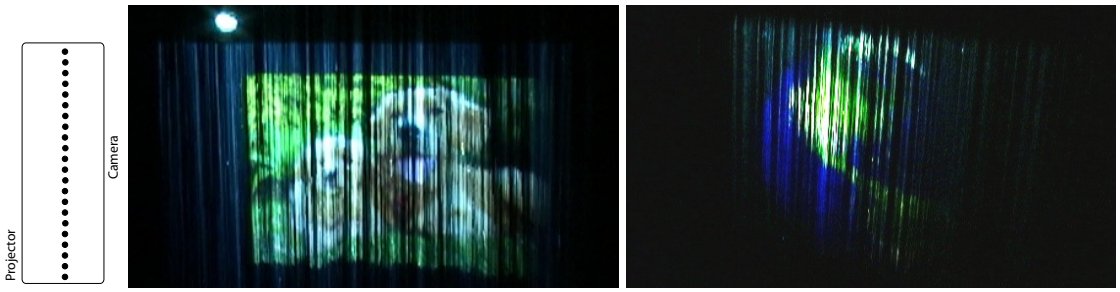
they can still be tracked across frames. For the current display, they do not vary greatly from vertical. But this adds robustness to the technique, and means it could be extended to different drop generation technology. (A lot of the variation that is currently present in the drop generation is likely to be because of momentary turbulence, and the fact that the drop emitters are not always exactly vertical, imparting some additional horizontal velocity to the drops).

The locations of the drop intersections with the laser plane are difficult to detect automatically, as they are very compared to both the drops illuminated by the projector and the level of camera noise. Therefore, the user tags the locations where drops from each generator cross the laser plane, indicating which homography to use. The user only needs to click once per drop emitter, making this a fast process. An experienced user could label about 100 drop locations per minute, and the display we show only uses between 10-50 unique drop locations.

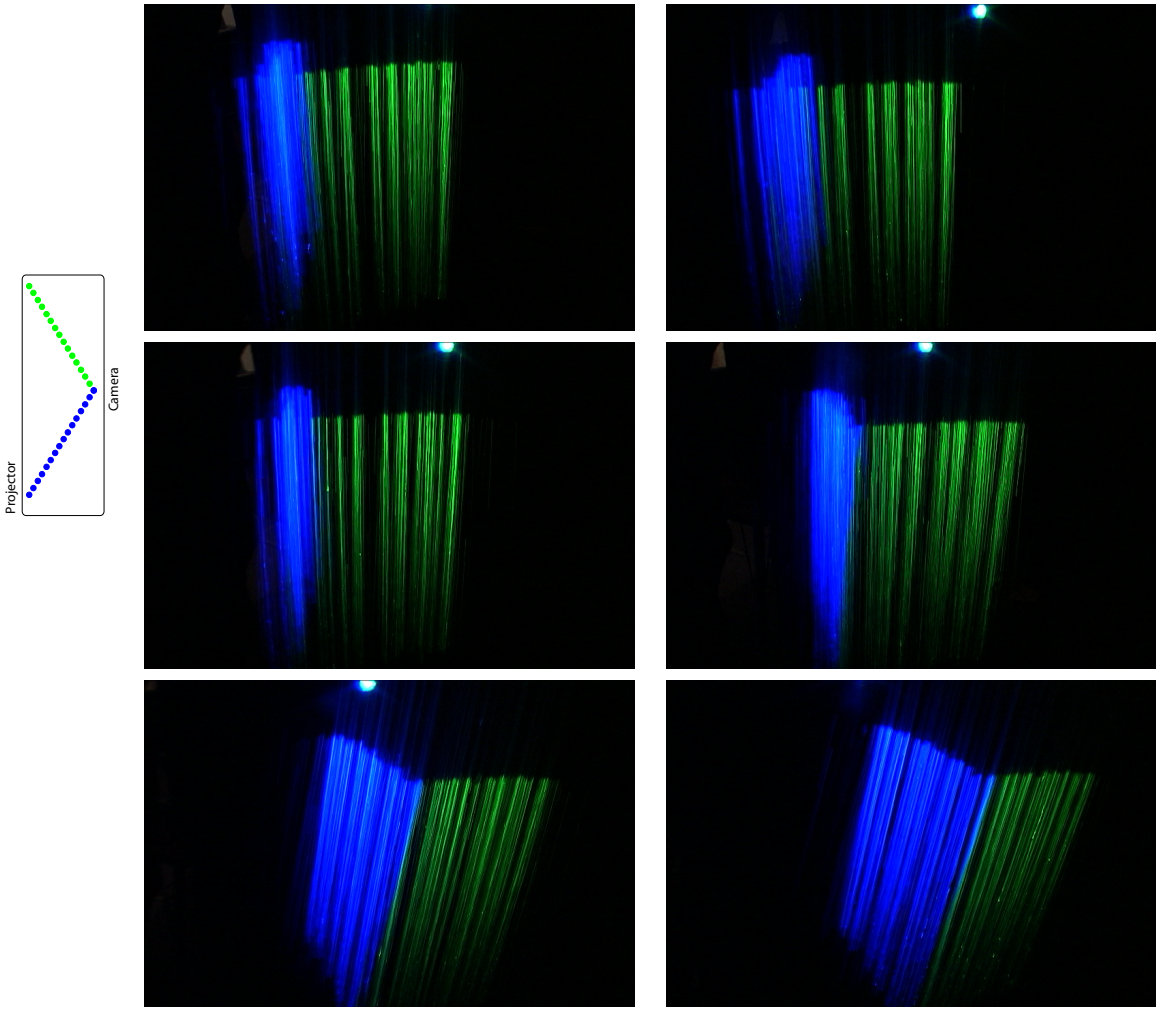
And finally, once drops from each emitter have been tracked, the water drops are "painted". The artist colors the model from the point of view of the camera. We created a simple GUI in Matlab that allows the user to add images, videos, and colors to the model. The results are then automatically warped to the correct locations, and the final composite projector images are saved to a video file, which can be shown on the projector.

1.4 Example displays

We created several display geometries, shown in Figures 1.3 and 1.4. Each was calibrated using only a few drop emitters. Only a few emitters are required, because there are only a few unique positions in each of the display. For example, a display with two planes would only require three emitters. One emitter each would be needed to determine the outer limits of each plane, and one to mark where they intersect. This is demonstrated in the V-shaped example display shown in Figure 1.3 (b). More complex geometries would naturally require additional emitters to describe them. One important point to note is that displaying images on a section requires more drop

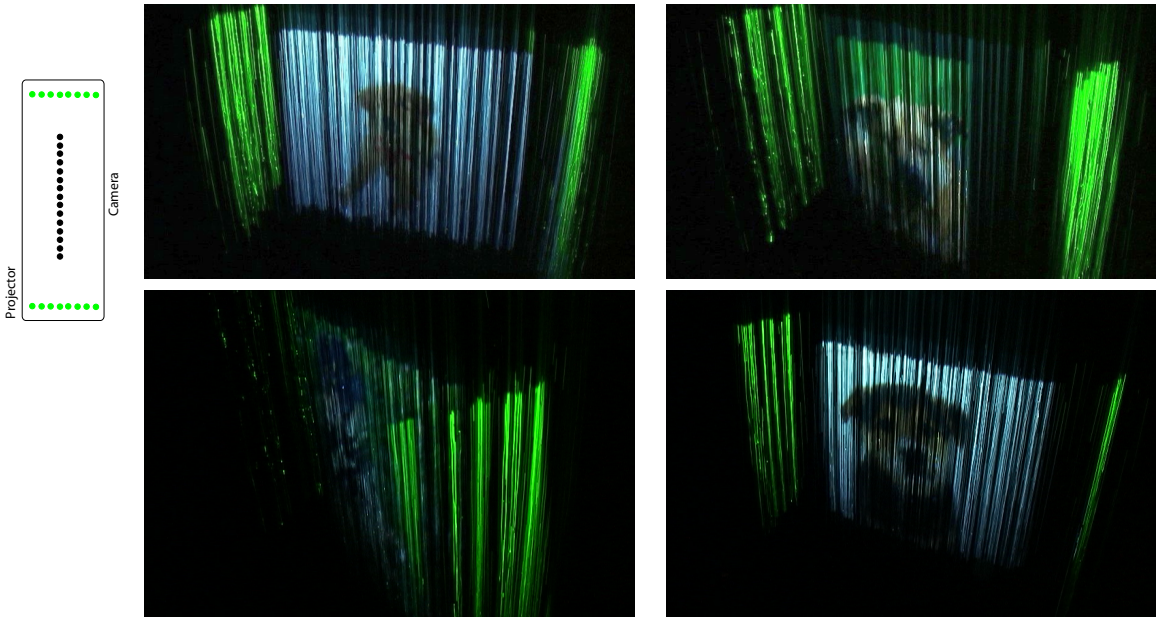


(a) Flat display: The flat display was calibrated with only one emitter at the center of the display.

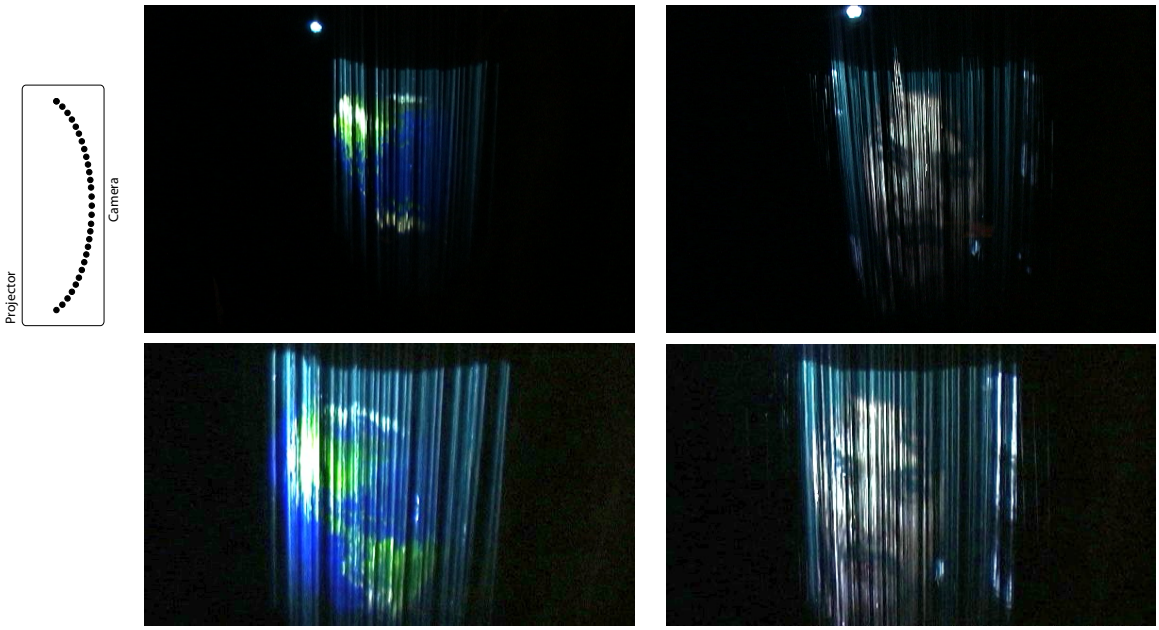


(b) V-shaped display: This display also required only one emitter to calibrate, at the intersection of the two planes.

Figure 1.3: Several example displays with different simple geometries. Each display was calibrated with a few emitters dropping individual water drops. Once calibrated, additional large emitters were added to create streams of water.



(a) H-shaped display: Combining the high resolution display of the first example with the geometry of the second.



(b) Curved display: Four emitters and a thin-plate spline were used to calibrate this display.

Figure 1.4: Several example displays with different complex geometries.

emitters than displaying a solid color. This is because the central portion of a solid color is correct regardless of the perspective warp caused by the angle between itself and the projector. For a user to define and to correctly display an image that is not perpendicular to the camera's image plane, it must first be warped to the appropriate location, then placed in the correct camera location. The automatic algorithm will then determine the additional warping required to display the correct image on the projector. Improving this to allow arbitrary planes at arbitrary angles relative to the parallel image plane is left as an exercise for the reader.

To calibrate the locations of the different drops, we needed to design emitters. To create a flow of approximately 10 drops/sec, we used low binding $10\mu l$ pipette tips for each emitter. The core idea is that a slight water pressure above a small hole will create a steady dripping. Using a short tube instead of a hole tends to act similar to a "buffer", preventing minor turbulence from upsetting the system. Such water systems tend to eventually reach an equilibrium, depending on the water pressure, shape of the holes, number of holes, and the length of the tubes between the main water tank and the final orifice where the drops are released.

We use two different water tank configurations for each display. First, we calibrate with the small holes that let through a small amount of water, creating steady water drops. But once calibrated, we use a second water tank that has 2.38 mm holes. These holes are much larger, and increase the water flow to a stream. Placing additional holes and increasing the amount of water increases the display's brightness and resolution, but does not require additional calibration since the water covers the same calibrated planes. The exact size of these holes are also much less important than in the calibration step. Larger holes will pass larger columns of water, which are simultaneously brighter and more difficult to control. But the main limit is that the larger the holes, the faster the water flows out, requiring either a larger tank, or a fast pump to replenish the water supply. Ideally, the final tank would not use round holes, but would have a thin line cut out in the pattern desired. This would require a great deal of water, so would require a high-speed pump to replenish even a large tank. Currently, we use the relatively small holes, as this allows

enough brightness and resolution to demonstrate the display, but does not require large tanks or sophisticated pumping hardware.

Many arrangements are possible, even with the geometry restrictions needed to prevent occlusion. The simplest is to arrange all emitters in a line, to create the flat water screen shown in Figure 1.3 (a). In this case, the water drop (or streams) act like a standard projection screen. Such a water drops display could be used to show any 2D movie. The movie is most visible while facing the projector compared to standing on the same side as the projector, and the drops refract light more than they reflect it.

Moving emitters forward and backward creates the V-shaped display in Figure 1.3 (b), which has one blue plane and one green plane. This appears to be a true 3D shape, although because of the positions of the drops relative the projector and camera, it can still be displayed

Horizontal resolution is traded for depth in the H-shaped display in Figure 1.4 (a). This example also shows how the geometry of the water surface can have discontinuities. Such discontinuities need to be mapped by having additional drop emitters. But because the light is only visible if there are both drops and light in a position, and errors at the crossover between the green "walls" and the image are invisible.

And finally, the example in Figure 1.4 (b) demonstrates a cylindrical screen generated by drop emitters arranged in a curve. The curve is approximated by a thin-plate spline. Using a thin-plate spline is an approximation, but it sufficiently close to give the appearance of the correct curvature.

1.5 Conclusion and future directions

Most of the obvious future work for a 2D water drop display is covered in the next chapter. Other than extending the 2D display to 3D, the main improvements would be to increase the density of the water drops, and to simplify calibration. More and smaller drops would increase the reso-

lution and appearance of the display. In the limit, this could just be a curved and discontinuous fogscreen. And calibrating such a system could use more traditional projector calibration work, such as [Raskar et al., 1999] [Sukthankar et al., 2001] [Lee et al., 2004].

Chapter 2

A three-dimensional water drop display

In the previous chapter, we showed how it is possible to create a waterdrop display with a camera, projector, and water drops. This technique can be used to create 2D displays, as well as certain classes of 3D displays where no drop occludes each other from the point of view of the projector at any time.

But if we have control over not just the lighting, but the water drops themselves, then it is possible to create an arbitrary three dimensional display. It is still important the no drop occludes each other at the same time, but a single projector ray can share an arbitrarily large number of drops, conditioned on the number of drops per second and the refresh rate of the projector.

2.1 Related work

Water drop displays can be compared by asking two questions. First, how accurately can the drops locations be predicted or controlled? A flat 2D display requires no control, but a 3D display with drops occluding each other at different times needs precisely positioned drops. Second, how quickly and accurately can the light source be controlled? The faster and more accurate the light source, the more tolerance the display has to drops occluding each other.

The simplest drop displays control only the light source. Fog screen displays [Araki et al. 1991; Palovuori and Rakkolainen 2004; Lee et al. 2007] use a sheet of microscopic droplets and a standard projector to create high-resolution, walk-through projector screens. But because they have no control over individual drops, they require a different projector for each screen.

In addition to light-control, some displays selectively turn on and off fountains of water, such as Disneys Fantasmic show [Disney 1992-2010] and the Submerging Technologies [Dietz et al. 2006] pieces. Such limited water control allows for interesting interactions and large-scale performances. The purpose of these displays is to create specific artistic effects, so they should not be evaluated only with quantitative metrics. However, this limited water control does not allow complexity much beyond that of fog screens.

Instead of controlling the light source, some displays, such as [Pevnick 1981], use a constant light source and arrays of individually controlled drop emitters to print out falling images. This has the advantage of spatial scalability. But as the drops are constantly falling, individual images can only be shown temporarily, and this display cannot be used for video.

2.2 Evaluating the options for drop and projection technology

Complete control over the water drops would mean the ability to create drops of any size, in any location, and moving with any velocity. For example, water drops could be created that are micrometers across, or millimeters across. They could be dropped from a height, or shot up or at various angles. Although it would be interested to create a system that could handle any size, location, or velocity of drops, realistically, we must choose a subset of possible sizes, locations, and velocities. Each subset has unique advantages and disadvantages, briefly outlined in Table 2.1. The overall objective we want to optimize is the ability to create as high resolution a display as possible, with as fast a refresh rate as possible. Higher resolution is always better, whether it is regarding 2D resolution, like the resolution of a standard LCD monitor, or depth resolution, to

allow the display to show greater depth and more accurate 3D objects.

2.2.1 Drop size: micrometer or millimeter

Coherent water drops can exist stably with diameters between 1 μm and 5mm [Pruppacher and Klett, 1997]. Different sized drops cause different visual effects. The smallest microscopic drops cause fog and mist. Larger drops are individually visible, and in nature are called raindrops. There are different challenges for different sized drops. Smaller drops allow for higher resolution, because each drop will generally only be able to display one color. If you display two colors on a single drop, then they will tend to mix and flip. For example, projecting blue on the left and red on the right of a drop will cause a viewer to see the opposite effect on the other side of the drop, namely blue on the right and red on the left. With totally accurate calibration and control of all drops, it would be possible to flip parts of the projected image for each drop, but this would become more and more difficult as the number of drops increases, and their size decreases.

Both microscopic drops and macroscopic drops have successfully been used to create displays. Microscopic drops have been used to create high-resolution flat displays with a projector. Fog displays, as shown by [Araki et al., 1991], use tiny droplets that are individually invisible. These drops are emitted either as part of a laminar flow of air, or surrounded by laminar flows. This allows for a fairly flat screen to be created. But unlike a standard translucent back projection screen, fog screens have no substance, and people can touch them or walk through them (this does not cause them to get wet, although it feels slightly cool.) One advantage of such fog screens is the possibilities of interaction, as shown in [Rakkolainen and Palovuori, 2002, 2005, 2004].

Macroscopic drops have been used to create low-resolution scrolling patterns [Pevnick, 1981]. In systems such as this, long rows of solenoid valves are individually controlled, and they print out an image in waterdrops that quickly fall. Such displays have the advantage that they can be of very large size, and present a novel way to advertise or show waterdrop-based artwork. Such

	Advantages	Disadvantages
Drop size		
Micrometer	High XY resolution Well developed technology	Requires one projector per surface Light leakage with multiple surfaces Individual drops difficult to control
Millimeter	Drops are spatially sparse in Y Individual drops easy to control	Low X resolution
Emitter locations		
Top	Simple 1D drop trajectories	Display height limited by emitter height
Side		Complex 3D drop trajectories
Bottom	Simple 1D drop trajectories Unlimited display height	Different drop speeds at different times Each drop passes a 3D location twice
Fall pattern		
Planar layers	Simple projector control	People may "see" the discrete layers
Helix	People may not see helical structure	Complex, fast projector control
Valve-emitter ratio		
High	Emission is predictable and precise	Complex electronics Expensive
Low	Simple electronics Cheap	Emission is unpredictable and imprecise
Liquid		
Clear water	Leaves no residue	Only part of each drop appears bright
Dyed water	Most of the drop appears bright	May leave residue
Control		
Fully adaptive	Can recover from errors No global clock required	Projector and valve control is difficult
Fully synchronized	Projector and valve control is easy	Difficult to synchronize components Cannot recover from errors
Partially adaptive	Control is somewhat difficult Can recover from errors	Complex software AND hardware
Projector type		
DLP	60-180hz refresh rate	10,000hz switching rate
LCD	85hz refresh rate Color	~100hz switching rate
CRT	120hz refresh rate Color	Scanning beam moves at 300,000hz Complex beam and drops interactions Cumbersome
Geometry model		
2D	Easy to calibrate and code Fast code execution speed	Cannot easily show a 3D object
3D	Easily show a 3D object	Hard to calibrate and code Possibly slow code execution speed

Table 2.1: The advantages and disadvantages of different water display technologies. Each option is discussed in detail in Section 2.2.

large drops have also been combined with a projector system, to create low resolution and refresh rate 3D images [Eitoku et al., 2006a,b, 2007, 2009, 2010]. But we are interested in creating a high-resolution, three-dimensional display that can show non-transitory objects.

Projecting at a specific spatial location in a volume of fog is no easier than doing so in air, as in [Kimura et al., 2006]. But although a display with macroscopic water drops will be lower resolution, the ability to control individual drops in space and time makes it the best choice. Using multiple planes of fog could allow for a semblance of 3D, but would require a different projector for each layer, so could not scale well. In addition, because the fog droplets are always present, light from the glow of one layer will also be absorbed and scattered and refracted on other layers. As the number of layers increases, the contrast of the display will go down. This will be especially evident at the user is close to the display, which is where the greatest possibilities of interaction exist. If fog display planes were not stacked directly in front of each other, such a system might scale better, but it would need to be determined if such a system could still display the type of content desired.

Instead, by using millimeter sized drops, we will be able to trade 2D resolution for depth, and create a true 3D display. Larger drops are easy to handle, but if they are too large, they will break apart. Although drops as large as 10mm have been demonstrated in the laboratory, in practice, the larger the drop, the more likely it will shatter. Shattering is less of a problem when the drops are moving slowly, but as they increase in speed, eventually reaching their terminal velocity, the buffeting of the airflow causes them to lose cohesiveness.

Smaller drops keep their shape better when falling, but can adhere to the nozzles and cohere to each other, leading to irregular creation times and satellite drops. Such tiny drops can be created and ejected, as demonstrated in an inkjet printer, but doing so can require extensive engineering. In addition, these ink drops are only designed to go a short distance, to hit the piece of paper. But for a water drop display, the drops need to traverse a long distance, and while partial shattering and multiple drops may be acceptable for inkjet printers, they would could a much

greater problem in a display. (As discussed in [Ambravaneswaran et al., 2002], different combinations of size, ejection velocity, and fluid characteristics have different tendencies to create satellite drops).

We find that drops approximately 2mm in diameter are a good combination ease-of-handling and structural stability. As a general rule, a drop emitter will tend to create drops about twice its diameter, meaning that our emitters should have approximately 1mm diameter openings.

2.2.2 Emitter locations: top, bottom, or side

Although clouds can condense raindrops from damp air [Yau and Rogers, 1998, Rogers, 1974], a practical drop emitter must have a nozzle at some spatial position. (It would be interesting to create a display the uses ambient moisture in the environment to generate drops, but this would likely require an enormous apparatus). The nozzles can either point up, down, or to the side, and artistic displays have been made with each. The most common displays that show arbitrary content use drops that fall under the force of gravity, as this makes them the easiest to control. But decorative fountains and water shows can shoot streams and fountains of water at a variety of angles, to achieve specific artistic effect.

Since we are going to synchronize the drops with a projector, we would like to keep their trajectories simple, so should either point the nozzles up or down. The main advantage of pointing the nozzles up is that they can be shot as high as desired. The display would not require and scaffolding, which would create an excellent sense of space. But even if the drops could be successfully and individually shot high into the air, they would likely arc and not come back at exactly the same path (or worse at an unknown path.) Therefore, we choose the simplest setup where all nozzles point straight toward the ground. The drops are released, then they fall under the influence of gravity, until they reach a collector. The collector will filter and recycle the water, and pump it back to the top, to be used to create new drops.

2.2.3 Fall pattern: layers or helix

To create a good three-dimensional display, water drops should regularly pass through each spatial location in the display volume. The number of drops per second is independent of the size of the drops, or where they are coming from. The closest equivalent to the refresh of a water drop display is a CRT monitor, except the multiple water drops will act as multiple beams, and the slow loss of luminance of the CRT phosphor is similar to the fact that it takes a short time for a water drop to pass a given point in space.

The important part of either the layers or helix fall pattern is that at any instant in time, two water drops should not occlude each other from the point of view of the projector. This occlusion-avoidance is a standard problem with three dimensional displays. Three-dimensional displays that use a moving projection surface usually use either a spinning plane [Parker and Wallis, 1948] or a helix [de Montebello]. It is not possible to make a spinning plane with water drops, but the helix is possible. However unlike solid projection surfaces, the water drops do not need to complete a cycle. This means that it is also possible to set up multiple linear manifolds that alternate in emitting drops, which is the equivalent of having multiple planes stacked together, and illuminated at the same time. A third possibility is to use a random or chaotic assortment of drops. However, drop generation is most predictable when the drops are emitted at regular intervals, which allows the system to stabilize. This become increasingly important as the drops are being generated faster. In addition, the perceptual advantages and disadvantages of using such a random assortment would have to be determined.

The advantage of the helix is that the structure of the drops is likely to be less apparent than the layers. In a multi-layered display, it is reasonable to suspect that even with a large number of closely spaced layers, that people will pay attention to the layers themselves, rather than the images projected on them. People have a natural tendency to be able to see straight lines, and since the drops are not completely dark even when they are not directly illuminated, this could

be problematic. However, the disadvantage of the helix is the it would require a extremely fast projector, and the projector control could be more complex. It seems reasonable to suppose that with a fast enough projector, we could instead have a very large number of planar layers, to minimize the unpleasant perceptual effects. Since the benefits of the helix are questionable, and the difficulties are clear, we chose to use planar layers.

2.2.4 Valve-emitter ratio: high or low

To cause the formation of individual drops, we need to actuate with high speed valves. The two most likely possibilities are to use solenoids or pneumatically actuated valves. Pneumatically actuated valves have the benefit that the source of the energy does not have to be close to the part that actually moving, as is similar in hydraulic systems. But given the additional complexity, and the fact that for most imagined setups will have plenty of room, the best actuators are solenoid coils. But in either case, the main question is whether to use a single valve per emitter, or to control multiple emitters with one valve. The main advantage to using a single valve per emitter is that the system is liable to be more robust to mild failure. But such a system would have much greater cost, and it is possible to engineer a multiple-emitter-per-valve system that has good response.

Let us examine the problem in more detail. If we want to create an NxM grid of drop emitters, the most obvious valve arrangement is to have one valve for each emitter. Each valve could draw from the same water source (or if desired, totally different water or fluid sources with different properties or colors or fluorescence,) but would have a separate actuator. Each valve could be designed and tweaked to emit drops of exactly the right size and dynamics, and any failures would be limited to a single emitter. The primary disadvantage of using a one-valve-per-emitter system is scalability. Even a 10x10 grid would require 100 actuators. And if the system expanded to a 100x100 grid, then 10,000 separate actuators. This would mean that not only would the cost be high, but the mean time to failure of at least one valve would be extremely short, and it is

liable that entire segments of the display would cut out at once. This would necessitate a long and costly replacement process.

The key insight here is that we do not require most of the emitters to operate independently. If we want to create multiple planar layers of non-occluding drops, then we only require that each row emit identically, but we do not require control beyond that. This means that a viable option is to have a single large actuator for each row. The advantage is that the aforementioned 100x100 grid would then only require 100 separate actuators. It is true that any failure in a valve would cause the removal of more drops, but fixing even a catastrophic failure with 100 actuators would require the same replacement or maintenance as only 1% of the valves for the previous case. The disadvantage is that the mechanics of controlling the formation of 100 drops with one actuator is significantly more complex, and can even behave unpredictably. Small variations in the water supply, valve dynamics, or even partial blockage of one emitter could cause odd partial failures across multiple emitters. In addition, it is not possible to tweak individual emitters to fix them; each row of drops would need to be addressed together.

But even with the challenges of using one valve for multiple emitters, it appears that the advantages outweigh the disadvantages. As a result, our current setup uses one actuator for a row of emitters. Using large solenoid valves, each row can emit up to sixty discrete drops per second. The valves are fairly large compared to the small amount of water passing through, and are not rated to repeatedly actuate at even 1hz. However, because we switch on the solenoid for only a short time, the valve only opens a small amount. When the solenoid is switched off, the valve quickly closes. When a solenoid valve is completely opened, it tends to shudder as the diaphragm is stopped by hitting the casing. However, because we do not totally open the valves, they are very quiet and the lack of hard collisions mean that there are not shock waves in the water, simplifying the drop generation. Using such a large valve is not necessarily the best method, although it seems to be an effective technique in this case.

Future work could include a different type of drop generation, such as generation via stream

breakup [Christopher and Anna, 2007]. In this type of method, the fluid is ejected at a constant velocity, but is mechanically perturbed. This perturbation eventually causes regular spacings of droplets, and can operate with extremely small drops and at very high frequencies.

2.2.5 Liquid: pure water or dyed water

Pure water drops act as little spherical lenses, refracting light from the source to the viewer. Because they act as fish eye lenses, the source is visible at an extreme angle (up to about 80 degrees each way [Garg and Nayar, 2004]). Different amount of light will be transmitted due to reflection, refraction, and internal reflection [Van de Hulst, 1957]. In general, a majority of the incident light on a drop will be passed through with standard refraction. A smaller amount will be directly reflected from the rear of the drop. And a minimal (although still visible) amount will experience a single internal reflection within the drop, and lead to the standard chromatic aberration that causes a rainbow. The result of all of this reflection and refraction is that a display using water drops and a projector is visible from various locations and angles relative to the light source. In addition, pure water also has the advantage that it leaving no residue on hands or clothing, is odorless, and completely non-toxic. Because of these advantages, our setup currently uses only pure water.

However, mixing the water with a small amount of white dye could dramatically increase the visibility of the display. A refracted light source will appear as a tiny speck in a drop, but white dye would scatter the light, making the entire drop light up. Even if the light source was close, the scattering of light would increase the visibility a hundred times. The most obvious disadvantage of using dye or any additive is that the residue left from the dye will likely cause major and minor clogs and sticking within the moving parts and the drop emitters. In order to make such a dye system feasible, there would likely have to be a regular, automatic cleaning process to keep the residue from sticking on important components. Our current system with pure water can run for hours, while such a dye-based system might require much more frequent

maintenance. Even if the maintenance is automatic, the system would have to be temporarily shut down. In addition, the drops will likely appear somewhat less bright, as some of the light will be lost within the drop, instead of simply passing through, as with the clear drops.

2.2.6 Projector type: DLP, LCD, CRT, or laser

The choice of light source is essentially a choice between types of projectors. The four main options are Digital Light Processing (DLP), Liquid Crystal Display(LCD), Cathode Ray Tube (CRT), or laser.

DLPs switch tiny micro-mirrors over ten thousand times per second, although almost all commercial models are only controllable between 60-85 for standard models, and 120hz for those made for stereo shutter-glasses. Color is created by a spinning wheel that alternates between red, green, blue, and white, 120 times per second (some newer models use a 6-segment wheel, to allow for greater color-reproduction accuracy.) It is possible that we will be able to run separate images in each of the RGB color channels, increasing the speed to 180hz, but this remains to be seen. And there are options that can project at much higher speeds, such as the Discovery series and the LightCommander projector, although these are somewhat experimental solutions, and tend to have issues.

When used with a water drop display, the DLP color wheel will cause rainbow streaking instead of color, so they can only be used for grayscale. The color wheel can be manually removed, to allow any color projector to function as a grayscale projector. However, projectors normally need to know the exact position of the color wheel, to know which color to display at each point in time. This is often done with a simple photosensor and a black stripe. Simply unplugging the wheel will cause the projector to not function. It seems reasonable that the output of the color wheel position sensor could be emulated with simple electronics, although it is perhaps easier to just fix it to the side of the projector, and let it spin.

LCD-based projectors use three separate LCDs, one for each color. The white light from the

bulb is sent along three separate paths, then the color output from the three LCDs is combined into the final image that the projector displays. This means that a single projector can make a color drop display. Unfortunately, the LCDs take time to switch on and off, limiting the display speed. In addition, it appears that many LCD projectors have issues with displaying high-spatial-frequency patterns, perhaps due to software solutions to overcome limitations in the optics.

CRT projectors use a scanning beam that moves incredibly fast. (Likewise, laser projectors use a scanning beam). Although the fastest of the three technologies, the temporal interactions between the beam and drops are liable to be complex, leading us to prefer either LCD or DLP based illumination.

The optimal solution would be to have three DLP projectors each displaying red, green, or blue (or one 3-DLP projector displaying all three). This would have all the refresh-rate and quick-switching ability of a DLP projector, but would allow us to create a color display. It is likely that there would also be issues with synchronizing and handing the data flow for the three projectors, although there is technology used in the film and television industries made for such cases.

2.2.7 Control: adaptive or synchronized

Having determined our hardware choices, we now need to look at algorithms and software to control it. The hardware must operate fairly consistently, but determining the exact parameters can either be done open-loop via precise temporal calibration, or closed-loop via observation with a camera.

Closed-loop control means adaption via alternately emitting drops, projecting images, and analyzing images captured with a camera. The projector vertical-synchronization signal can be detected by the computer. An internal model of the drop dynamics would then be adjusted to match with the images observed. Since the different components use different clocks, the temporal calibration will drift over time. This would necessitate periodic updates, possibly requiring

the display to be switched off. Closed-loop control has the advantage of requiring no additional synchronization hardware, but the amount of algorithmic control could make it prone to estimation error.

Open-loop control is conceptually easier, but requires hardware clock synchronization of the display's heterogeneous components. The advantage is that once the hardware is using the same clock, there will be no drift. Since open-loop control will be more robust for our case, we chose to use it, and base all timing off of the video card's VGA vertical-synchronization signal.

Using the temporal synchronization signal together with closed-loop control is the final option. It appears currently that the open-loop control is sufficient for running the display for a few minutes, but occasional errors in sending and receiving the clock pulse can require recalibration. A closed-loop system could detect these errors and automatically adapt, while still keeping basic synchronization with the clock pulse.

2.2.8 Geometry model: 2D or 3D

The final question is how to turn an artistic idea into pictures for each layer. Since each layer of drops is planar, the simplest and fastest is to treat each layer as an independent two-dimensional problem. The disadvantage of performing all processing in 2D is that projecting a 3D object would require a great deal of manual effort. Currently, all processing is performed in 2D, but one of our goals is to reconstruct the drops locations in 3D, to allow 3D models to be automatically displayed.

2.3 Overview of the display system

Figure 2.1 shows our setup in action. The left subfigures illustrate the positions of the projector, the camera, and the drop-generation manifolds. Each manifold contains fifty drop emitters arranged linearly, and is controlled by a solenoid valve. The valve cycles at up to 60 Hz, releasing

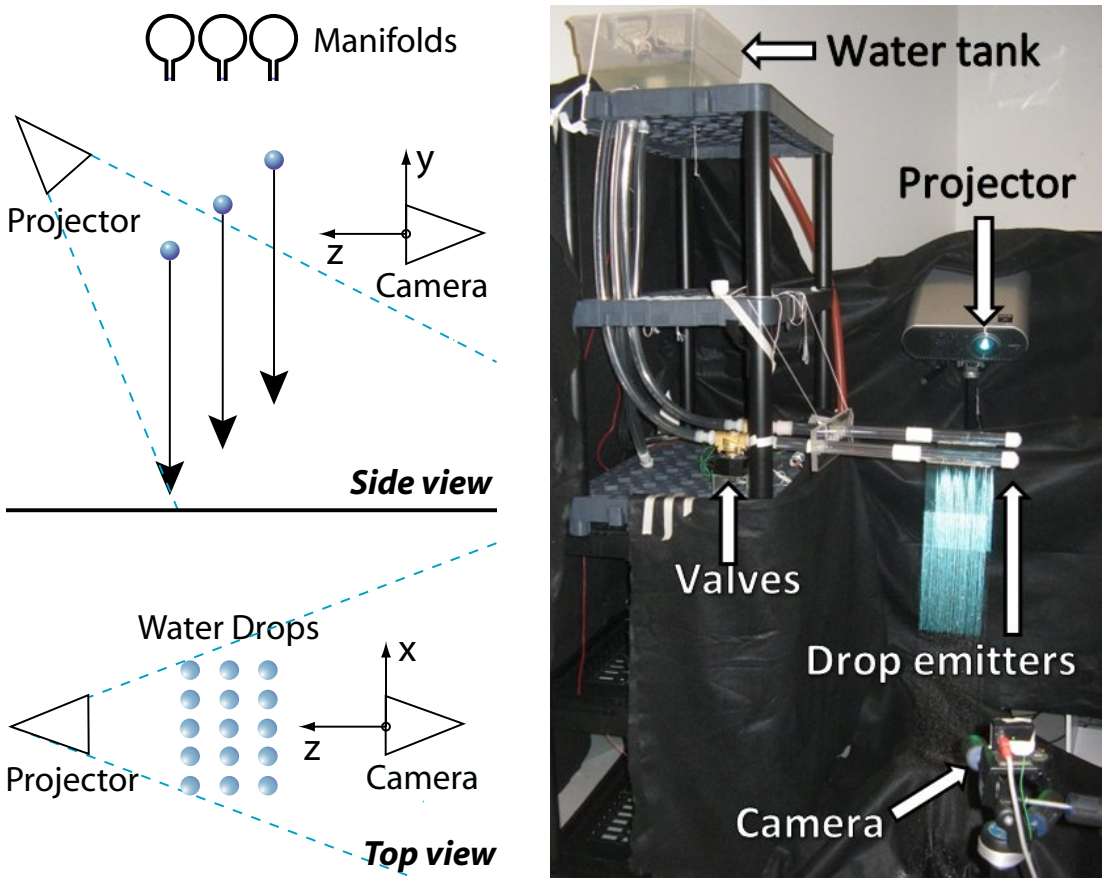


Figure 2.1: Diagrams and a photo of the drop display setup. By emitting drops and illuminating them with a Sony VPL-HS51A 3LCD projector, we can create multi-layer water drop displays. (The left subfigures show the setup from the side and from the top. The photo is taken from slightly behind the Point Grey Firefly MV camera.) The water head is about 0.8m above the valves, and is regulated by a Rule 360 Bilge Pump and a float switch in the water tank.

up to 3000 drops every second for each layer. The drops fall fast and are quickly replaced, and to the naked eye, they appear as a row of streaks along a vertical plane. But because each manifold releases drops at slightly different times, no drop ever occludes another from the point of view of the projector.

Consider the order that the drops enter the projector's field of view (the top dotted line in the side view of Figure 2.1). The leftmost drop has already crossed it, next the middle one will, followed by the rightmost. The projector starts displaying an image when the first drop crosses, then switches images as the second drop crosses, and so on. For example, if the drops are offset

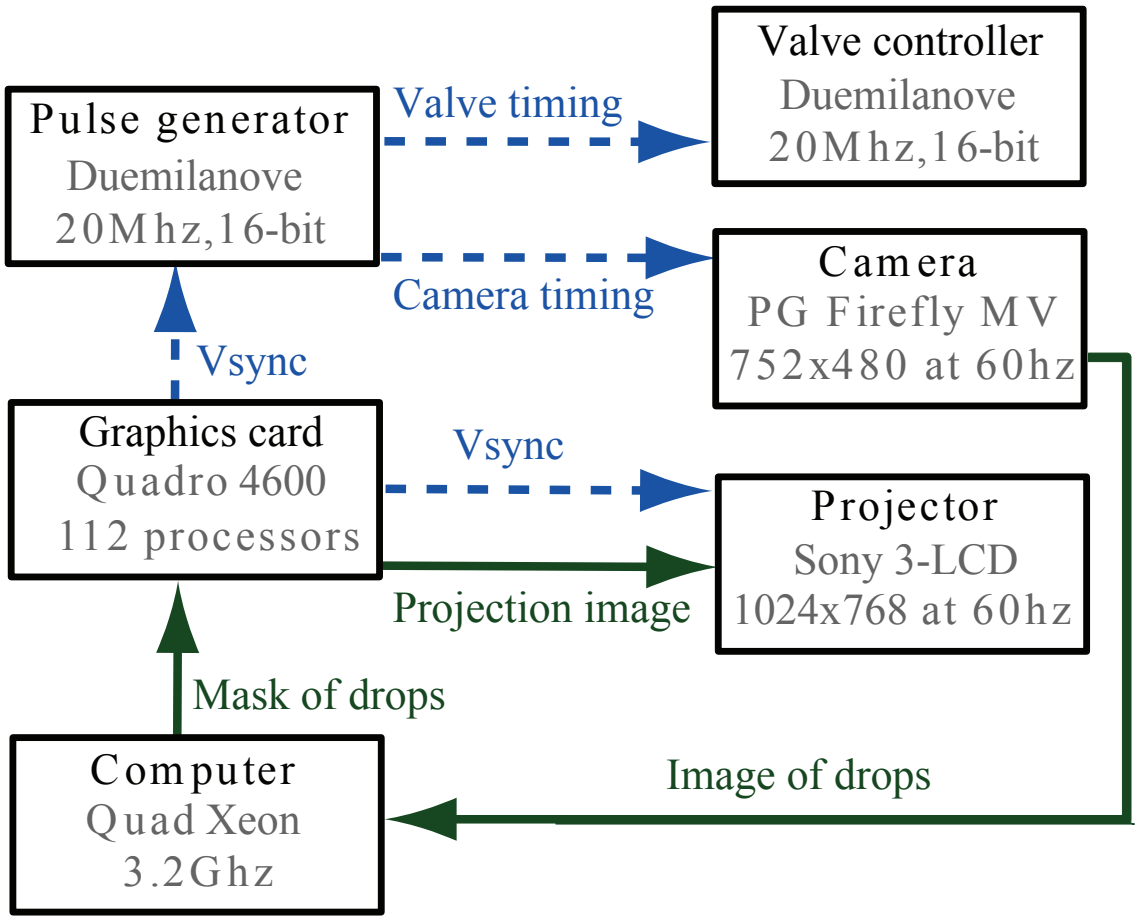


Figure 2.2: A chart of the data and synchronization flow. The dotted lines represent the synchronization signal, which starts from the graphics card, and travels to the projector, camera, and valve controller. The solid lines represent data flow. During the calibration step, the camera detects the locations of the drops and sends the images to the computer. The computer detects and stores the drop locations. After calibration is complete, the graphics card rapidly sends images to the projector, which illuminates the drops. (The pulse generator and the valve controller are built around separate Arduino Duemilanove microcontrollers.)

by $1/10$ seconds and we do not consider perspective effects, then the projector has to switch frames at 10 Hz. The optimal timing offsets has drops spaced as far apart from each other as possible. For N manifolds operating at S Hz, the drops should be separated by $1/(SN)$ seconds. A simple bound on the necessary projector refresh P is therefore:

$$P \geq SN \tag{2.1}$$

The equation predicts that a single 60 Hz projector could illuminate 5 layers at 12 drops per second, 2 layers at 30 drops per second, or trivially, one layer at 60 drops per second. This equation is a reasonable first approximation, although we have included a tighter upper bound in Appendix A, that accounts for uncertainties in drop emission times and finite projector switching time.

Determining the locations of the drops at any time instant and projecting to hit them requires precise design, synchronization and control. The flowchart in Figure 2.2 shows the passage of timing signals and data (images) between different components of our system. To start the process, the graphics card sends a V-sync timing pulse to the projector and to a custom pulse generator. At the same time as the projector begins displaying an image, the pulse generator can signal the valve controller to begin releasing the drops from the first manifold, and can signal the camera to begin acquiring an image. Because the projector, camera and valves are all synchronized, there are only a small set of drop-locations for all projector refresh cycles. From Equation 2.1, we see that for a projector running at P Hz and drops falling at S Hz, there are only P/S possible states. This means that capturing P/S images from the camera and warping these images to the projector reference frame tells us which projector pixels correspond to which drops at which times. And because these locations are precisely known, it is straightforward to direct the projector to illuminate them. The individual components for creating this precise setup are detailed in the following sections.

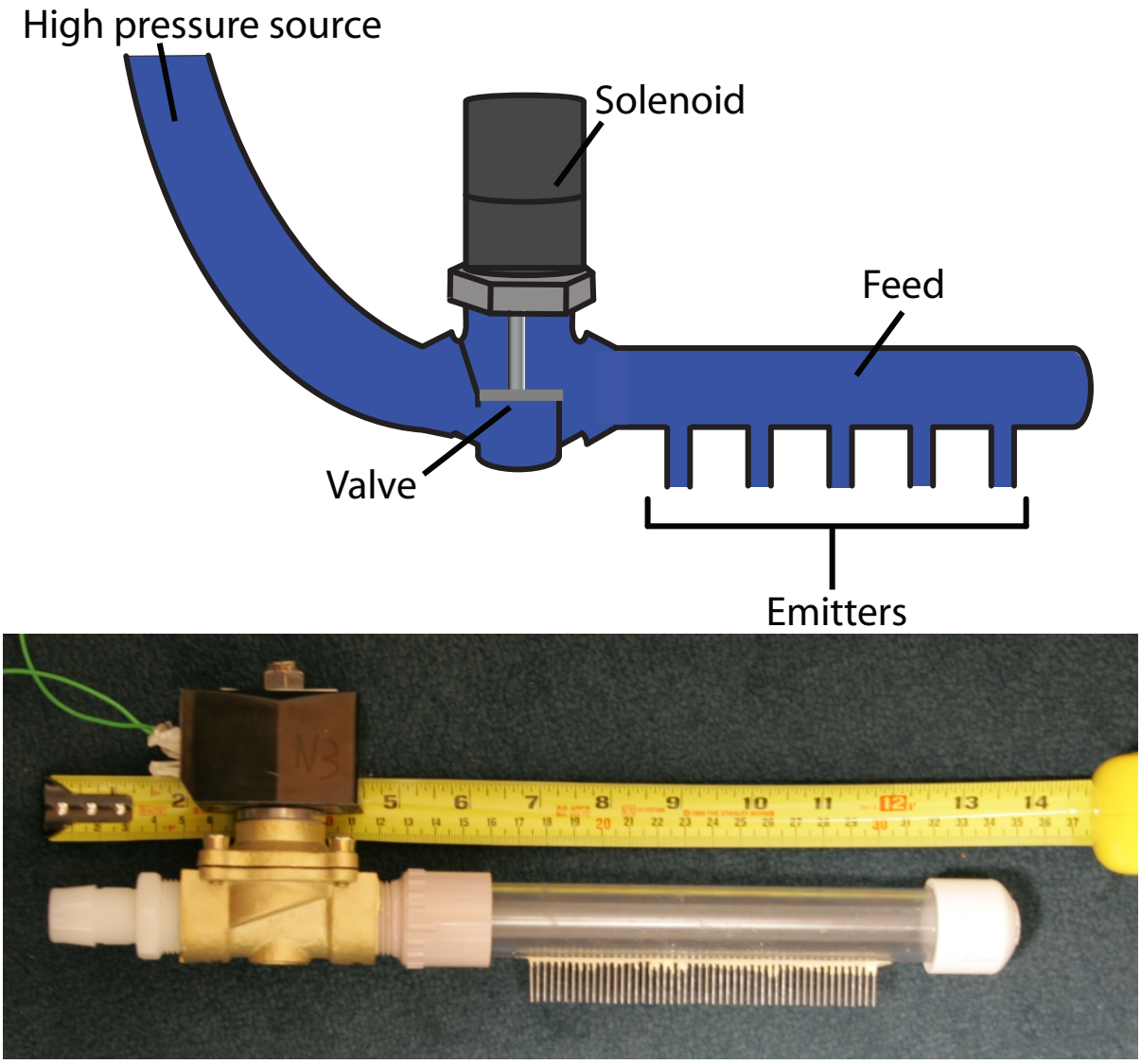


Figure 2.3: An illustration and photo of a water drop emitter manifold. When charged, the solenoid pulls up the valve, allowing water to pass from the high-pressure source to the feed. This will pressurize the feed, causing drops to be formed at each emitter. The valve is a generic 3/4 inch, 12VDC, 2 Amps, normally-closed diaphragm solenoid valve. Its Cv flow coefficient is about 5, although we only turn on the solenoid for about 10ms, so the valve never completely opens and the achieved flow rate is much less.

2.4 Accurate generation of drops

Accurately generating well-defined and stable drops at precise time instants is crucial to our display. For this, we make the following design choices based on basic principles of fluid me-

chanics.

2.4.1 Drop sizes

Coherent water drops can exist stably with diameters between about $1\mu m$ and $5mm$ [Pruppacher and Klett, 1997]. Different sized drops cause different visual effects. The smallest, microscopic drops cause fog and mist. Larger drops are individually visible, and in nature, are raindrops. Both microscopic drops and macroscopic drops have successfully been used to create displays. Microscopic drops (mist or fog) have been used to create high-resolution flat displays with a projector [Araki et al., 1991]. Using multiple planes of mist could allow for a semblance of 3D but would result in a loss of contrast due to light scattering as the light passes through each plane.

Although a display with macroscopic water drops will have fewer voxels, the ability to control drops in space and time makes it the best choice for our system. By using millimeter-sized drops, we trade 2D resolution for depth and create a multi-layered display. Such drops are less effected by air currents and are hence easy to handle. Even so, we must be careful about choosing their exact size — if they are too large, they will break apart or cause satellite drops. If they are too small, they can adhere to the nozzle and coalesce with each other, leading to irregular timing. As discussed in [Ambravaneswaran et al., 2002], different combinations of size, ejection velocity, and fluid characteristics effect the ability to create usable drops. From their work and our experimentation, we find that drops approximately 2mm in diameter provide a good combination of ease-of-handling and structural stability.

2.4.2 Fluid mechanics of drop generation

One option for our display is to control every emitter independently using a dedicated valve [Pevnick, 1981]. Though providing accurate drop generation, it would be costly for large numbers of emitters. Instead, we choose to control a linear manifold of emitters with one valve

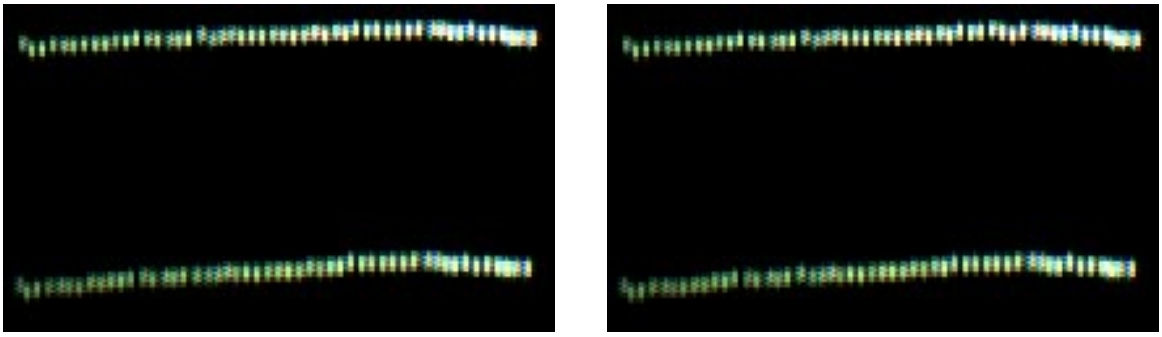


Figure 2.4: These photos show two 4ms exposures, taken 1/30th of a second apart. The drops were emitted from a linear manifold, as shown in Figure 2.3. The valve was actuated at 30hz, so the rows are spaced by 1/30th of a second. Although the drops are not in completely straight horizontal lines, each line is nearly identical. If a manifold is emptied and refilled with water, it will have a different line pattern. But as demonstrated in these images, there is almost no change over the short term.

(Figures 2.3 and 2.4). The disadvantage is that the mechanics of controlling the formation of a row of drops with one actuator is more complex, and the drops may behave less predictably.

The emitters: First, we must design the individual drop emitters. As discussed in [Ambravaneswaran et al., 2002], important factors in the accurate emission of drops are the *Ohnesorge number* (which relates the viscosity to inertia and surface tension), the *Weber number* (which relates inertia with surface tension), and the thickness and hydrophobicity of the emitter walls.

Since we are using pure water, the main factors we can vary are the materials, and the wall thickness and diameters of the emitters. We choose Type 304 stainless steel, as it is fairly hydrophobic and its strength allows us to minimize the wall thickness. Hydrophobicity is increased by applying a Rain-X coating to the emitters' exteriors. Based on the results of [Ambravaneswaran et al., 2002] and our own experimentation, creating emitters with an inner diameter of 1.27mm allows predictable and fast drop generation, with few or no satellite droplets (Figure 2.4).

The feed: For drops to be released in the same way by all emitters, the feed needs to have similar pressure at all emitters. The fluid flow in a closed cylindrical pipe can be predicted using well-known principles of fluid mechanics [Munson, 2009]. Accordingly, assuming laminar flow and

no vertical pressure variation, the water pressure at each emitter of a pipe manifold depends on the distance between the emitters l, the flow rate Q, and the diameter of the pipe D. The pressure difference between two emitters is given by:

$$\frac{128\mu l Q_e}{\pi D^4} \tag{2.2}$$

where, Q_e is the fluid flow out of the emitter further from the source, and μ is the dynamic viscosity of water. To minimize the pressure difference between emitters, we can either reduce the distance l between them, or increase the diameter D of the pipe. Therefore, to create our emitter manifolds, we use a large diameter tube (19mm) and place the emitters as close as possible (2.5 mm).

Electronics for controlling the valves: Creating 10 drops per second from 50 emitters requires a flow rate of $10 \times 50\times$ [volume of a drop] $\approx 1.7 * 10^{-5}\frac{m^3}{s}$. Even slight variations in the flow rate can prevent correct drop emission. This means that the opening and closing of the solenoid valves must be accurate to within a few microseconds. Using a dedicated microcontroller, each solenoid is controlled by a MOSFET. The microcontroller is synchronized by a timing signal from the pulse generator (Figure 2.2).

2.5 Space-time division of illumination

The ability to easily design visual content is key to a display's usefulness. We have described how to densely fill planes with non-occluding water drops. Similar to [Barnum et al., 2009a], we will allow the user to "paint" the drops from the point of view of the camera. The user will select images or movies to display on each layer, and the system will automatically determine how to control the projector to display them, with no further user input. This involves three steps. First, the correspondence between camera rays and projector rays is computed. Second, the camera

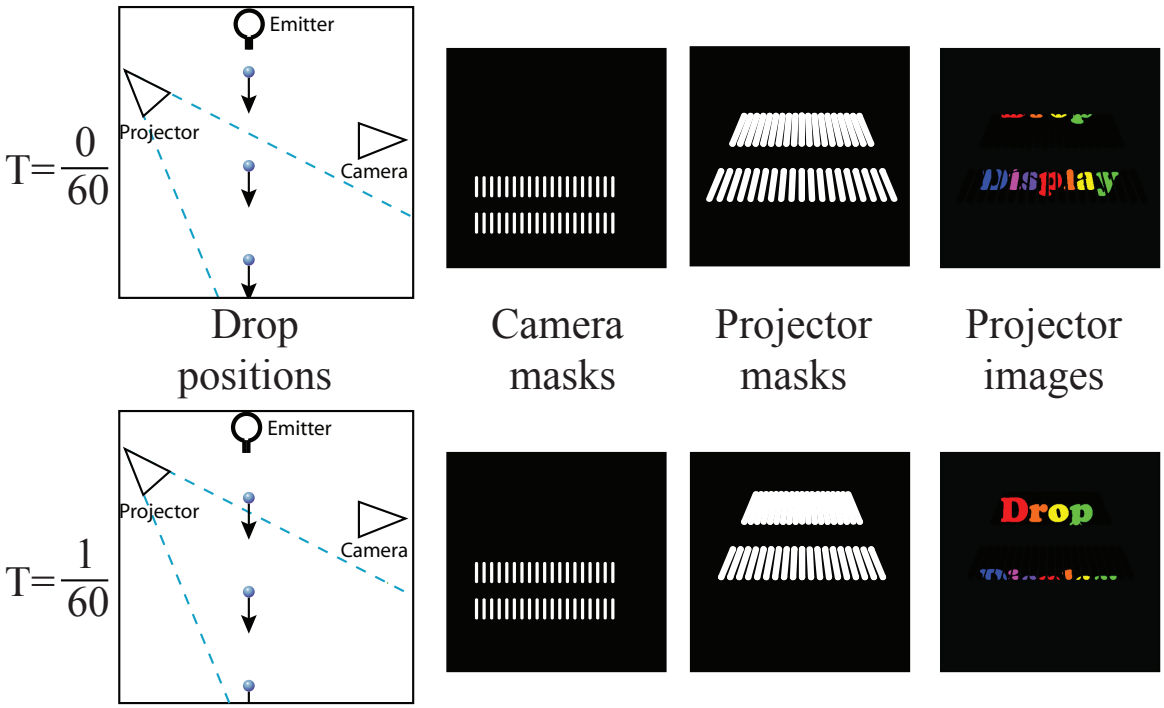


Figure 2.5: A summary of the drop display process, using drops being emitted at 30hz and a projector refreshing at 60hz. The water drops fall a short distance in each 1/60th of a second. The camera captures images as they fall, which are thresholded to create binary masks. The camera mask creation is performed before any content is displayed. During runtime, the camera masks are warped with a homography to the projector coordinate frame. The warped masks are then used to create the final projection image.

is used to determine where the drops of each layer are at every time instant. Finally, using the detected image locations, the projector is controlled in real-time to display images on all layers.

Projector-camera calibration: To compute the correspondence between projector and camera rays, we first display and capture a checkerboard pattern at each layer. The user then selects a few point correspondences between the captured image and the original pattern, and a homography is computed for each layer, which maps image coordinates to projector coordinates.

Drop location estimation: As the valves, projector and camera are synchronized (Figure 2.2), the display is created as illustrated in Figure 2.5. For example, let us consider a single manifold emitting drops at 30 Hz, illuminated by a projector operating at 60 Hz. The state of the drops

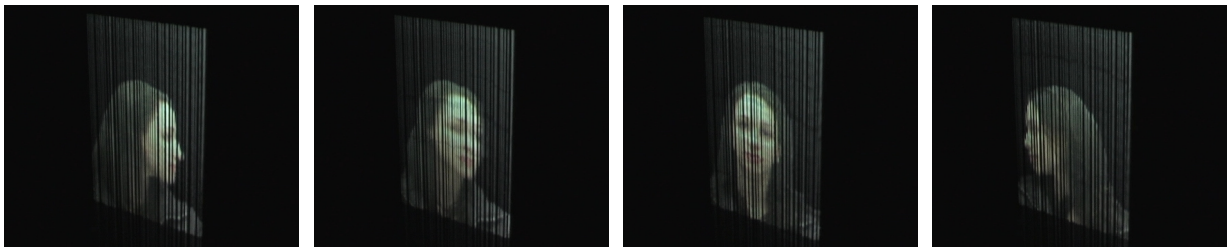
will repeat every 1/30 seconds, during which time two images can be projected. Recall that for a projector operating at P Hz and drops falling at S Hz, there are P/S unique projector states. To determine where the drops are in each projector state, a camera acquires images at 60 Hz, shown in the second column of Figure 2.5. These binary images contain the locations of the drops during the time the projector can display its images.

Real-time synthesis of display content: Once the projector-camera calibration and drop location estimation are done, the display is ready to run. Using an NVIDIA Quadro 4600 GPU and the homographies computed in Section 2.5, the binary drop masks are transformed to the projector coordinate frame (third column of Figure 2.5). The user selected image (Drop Display, in this example) is also warped to the projector coordinate frame, and is masked. When the projector displays the warped and masked image, the drops will be illuminated, making the text appear. Because the projection images are computed and displayed in real-time, the user can modify them as the display operates, and any of the layers of drops could even serve as an interactive touch screen.

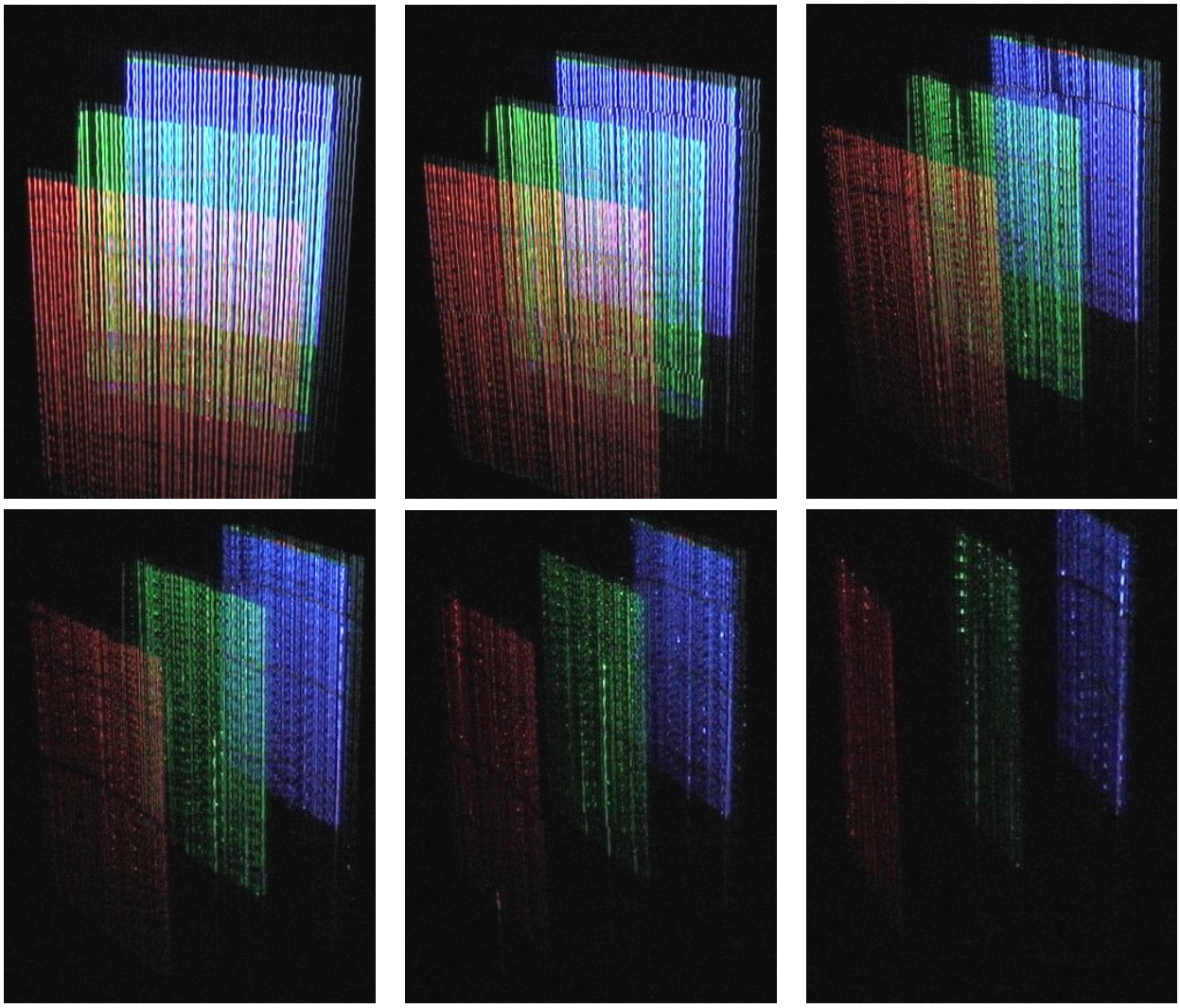
2.6 The display in action

Figure 4.1, Figures 2.6, 2.7, and 2.8, and our video show several example displays. We have demonstrated displays with up to four layers. Each layer is made of rows of 50 drops, and the projector resolution is 1024x768.

The drops are spatially sparse, but the layers appear solid to the human eye, with some flicker. As a camera integrates temporally differently than an eye, we have been unable to reproduce the visual effect in our videos, but no one at our demos has found the flickering disturbing. Although it is possible to run individual layers at up to 60 drops per second, we found that the decrease in perceived flicker is marginal above 15 drops per second. The effect is most prominent during eye or camera motion. When the camera moves (as shown in our video), different pieces of a layer

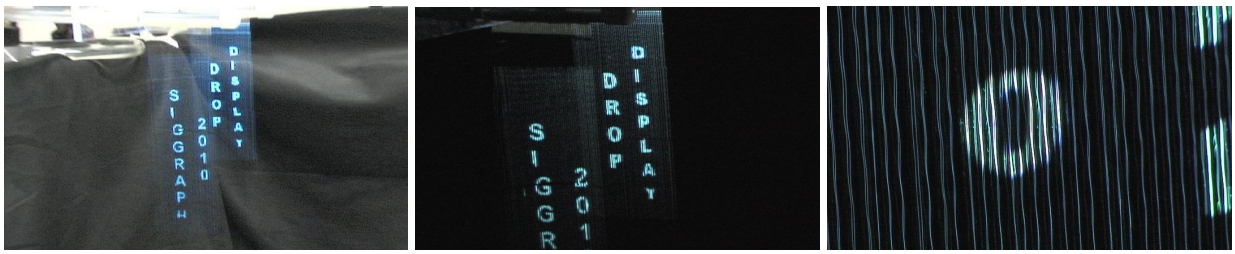


(a) A single layer with a video of a woman from the VidTIMIT dataset (60 drops per second).

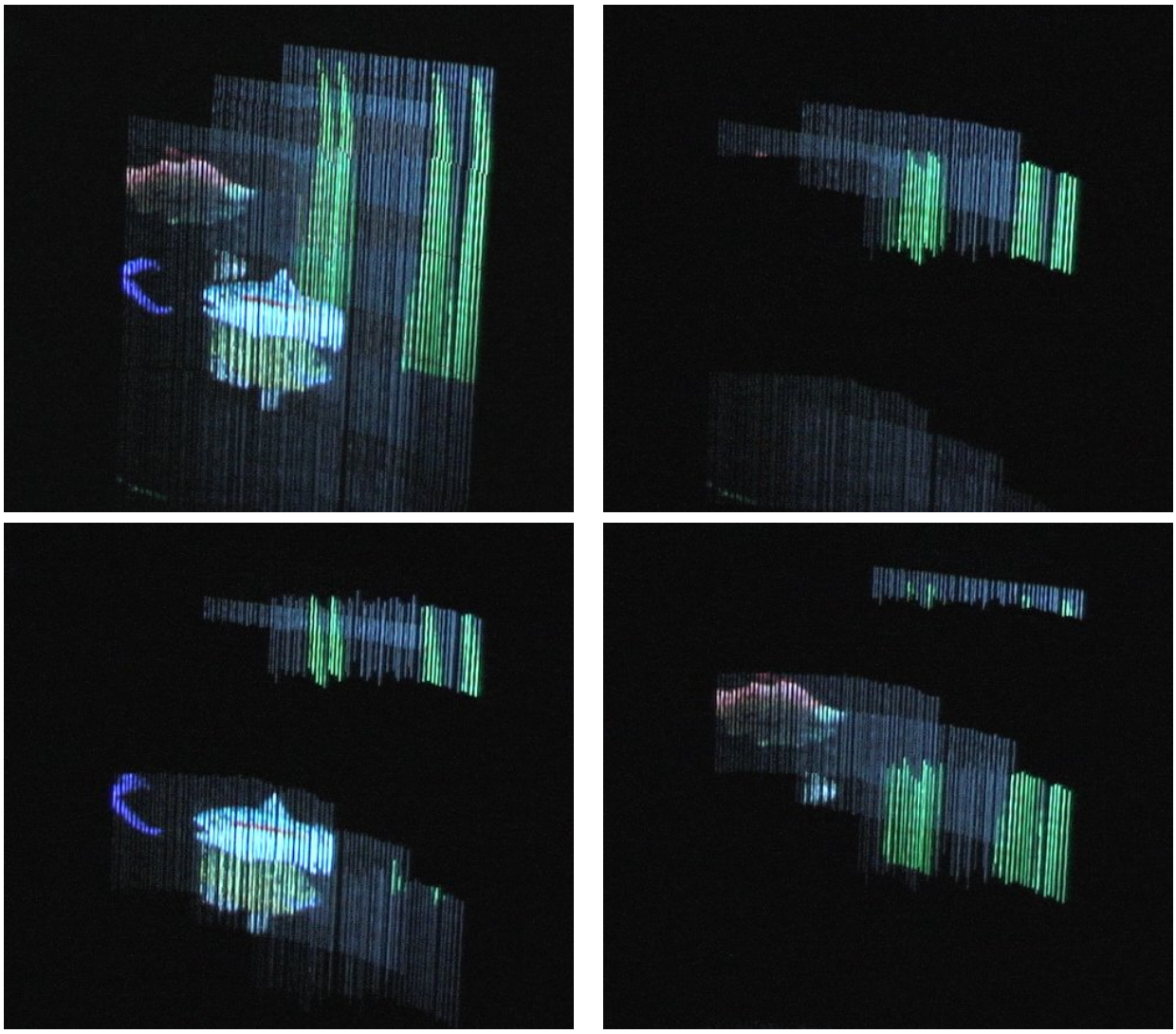


(b) Red-Green-Blue (10 drops per second): The display is brightest when viewed from directly opposite the projector (leftmost image). But the colors are still clearly visible at an extreme angle. These images were taken with fixed camera settings, and the intensity change is much less evident to the adaptive human eye.

Figure 2.6: Several examples of different images and movies displayed on layers of water drops.



(a) Zooming in on text (15 drops per second): Because the drops oscillate as they fall, they appear as wiggly lines when viewed close up. But because they are generated with the same initial oscillation, each drop distorts in the same way as it falls, and the wiggles appear stationary.



(b) Aquarium simulator (10 drops per second): The projector images are created in real time and we can generate content on-the-fly. This aquarium simulator has swimming fish and waving plants. To create the leftmost image, we took the pixelwise max for the right three images (30FPS, 1/30 second exposure). The human eye temporally integrates the falling drops, so the display appears like the left image. But the other images demonstrate that the drops are actually spatially sparse.

Figure 2.7: Two examples demonstrating subtle and hidden character of the display.

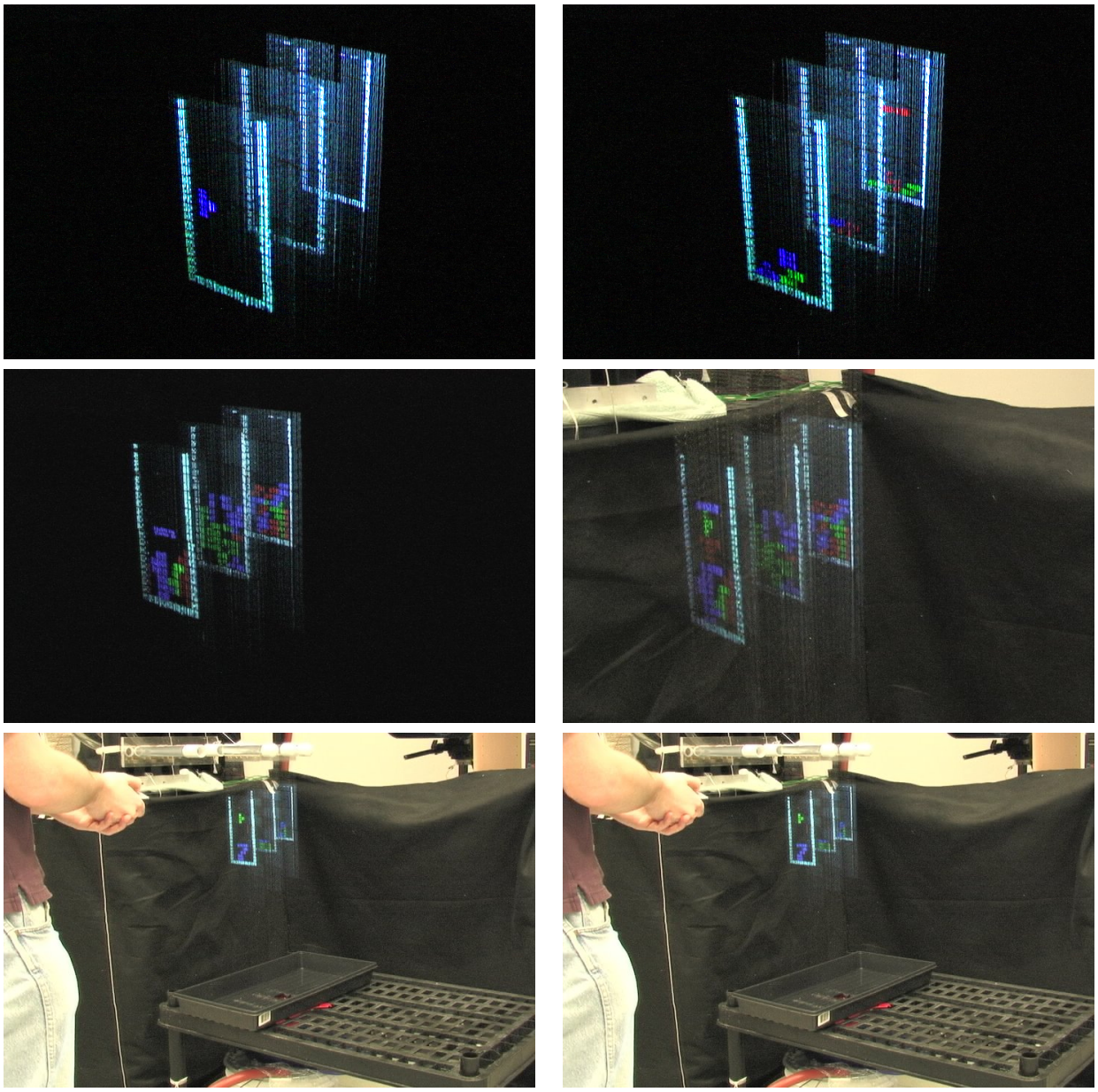


Figure 2.8: 2.5D Tetris (10 drops per second): This twist on the classic game demonstrates interaction with the drop display. Only one piece is dropped at a time, but the player can choose which layer to use it in. The rightmost image shows a player using a remote presenter mouse to move and rotate the pieces.

appear to be in slightly different positions, causing breaks in the images. Interestingly, due to the skill of the human visual system, these position shifts are not seen in real life.

The masking and warping takes approximately 5ms, leaving about 10ms free to load video

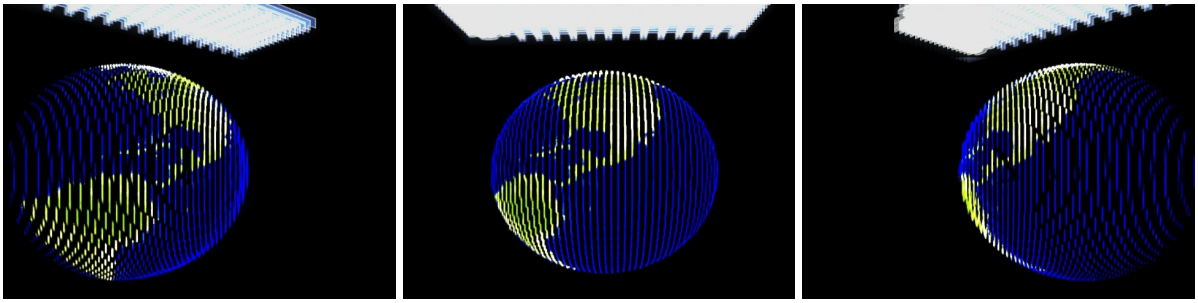


Figure 2.9: A simulated example of a 3D globe composed of 17 layers. With a projector running at 1440hz, it would be possible to make such a display.

frames and render scenes. The pre-masked-and-warped image for each layer is stored as a DirectX Surface, so any application that can render to a surface can be seamlessly used.

The first example in Figure 2.6 (a) shows one layer at 60 drops per second, with a video of a woman from the VidTIMIT dataset [Sanderson and Paliwal, 2002]. Figure 2.6 (b) has two layers with text. The leftmost image demonstrates that the text is still bright enough to be seen with the overhead lights turned on.

Although the display is brightest when the viewer faces the projector, Figure 2.7 demonstrates that the display has a wide viewing angle. It also demonstrates the display's accuracy. Every projector pixel has to switch colors at exactly the right time, making this example particularly difficult. Any errors in projector-camera calibration or drop emission timing will be instantly visible. Figure 4.1 (d) shows a similar four-layer example at 5 drops per second.

Figure 2.8 (a) and (b) demonstrate how we can generate content on-the-fly. Figure 2.8 (a) is an aquarium simulator, with swimming fish and waving plants. Figure 2.8 (b) is a version of Tetris played on multiple layers. The player uses a remote presenter mouse to move and rotate pieces, and to move pieces between layers.

2.7 Tighter upper bound on scalability

By Equation 2.1, the required projector refresh rate scales linearly with the number of layers and the number of drops emitted per second. But this assumes that the drops are emitted with identical initial velocities, and the projector can switch frames instantaneously. A more accurate upper bound considers drops with different initial velocities and finite frame switching time.

Let us assume that the projector is orthographic, and the drops are points. For an initial velocity v, the vertical position y of a drop at time t is given by the free-fall equation:

$$y(t, v) = vt - 4.9t^2 \tag{2.3}$$

We need to determine if the beginning location B of a drop on one layer intersects with the end location E of a drop on the next layer. Let the projector refresh rate be denoted by P, the current projector refresh cycle by $\tau \in \mathbb{N}$, the LCD switching time by ρ, the initial velocity of drops from the i-th emitter on the j-th layer by $v_{i,j}$, and the time offset for the j-th layer by o_j (for example, if $o_1 = 0$ and $o_2 = 0.1$, then the second manifold releases drops 0.1 seconds after the first). Typical values for these quantities are $P = 60hz$, $\rho = 0.01$ for an LCD projector, and $v_{i,j} \approx 0.4m/s \pm 0.05m/s$. Therefore, for the k-th drop emission, the beginning location B of a drop on one layer and the end location E of a drop on the next layer can be computed by substituting into the free-fall equation $y(t, v)$:

$$B = y(\frac{\tau}{P} - o_j - \frac{k}{s}, v_{i,j}) \tag{2.4}$$

$$E = y(\frac{\tau + 1}{P} - o_{j+1} - \frac{k}{s} + \rho, v_{i,j+1}) \tag{2.5}$$

To check if any drop on any layer intersects the same projector pixel at any time, for all j

such that $o_j < o_{j+1}$, we minimize the following expression:

$$\underset{o_1,\ldots,o_N}{\text{argmin}}\{\sum_{j=1}^{N-1}\sum_{k=0}^{\lfloor S(\frac{\tau}{p}-o_{j+1})\rfloor} max(0, B-E)\} \tag{2.6}$$

If the result is greater than zero, then there are drops that cannot be illuminated different colors.

An LCD projector switches at $\rho = 0.01$ and has a projector refresh rate of $P = 60$. Therefore, according to the model, the display scales to 3 layers at 10 drops per second, or 2 layers at 15 drops per second. We have successfully implemented 3 layers at 10 drops per second and 2 layers at 15 drops per second, which validates this theoretical estimate.

If instead of the LCD, we used one 60hz DLP for each color channel, then the frame switching time ρ would be on the order of microseconds. With three $P = 60hz$ DLPs, we could have 4 layers at 10 drops per second, or 3 layers at 15 drops per second.

There are also projectors, such as the Pico 2, that are user controllable at up to 1440hz. These high-speed projectors could allow for displays with additional layers. For $P = 1440hz$ switching projectors, we could have 17 layers at 10 drops per second, or 11 layers at 15 drops per second. In this case, the main bottleneck is the variance in velocities and not the projector refresh rate.

2.8 Conclusion and future directions

By using a camera, a projector, and several water drop emitter manifolds, we have created a 2.5D water drop display that can be used to show pictures and video on-the-fly. The drops are generated in sync with the camera and projector, therefore the camera can be used to automatically discover the drop locations. This allows users to easily create content without requiring extensive manual calibration.

The main limitation of this work is that it only has a few layers. As discussed in Section 2.7, a faster projector could allow for a display with many more layers. But creating a true 3D drop

display would also require more precise drop control. The simplest way to increase accuracy would be to better flush air bubbles from the valves and manifolds. (The bubbles are caused by air dissolved in the water and by leaking at the emitters' water/air interfaces). Alternately, drops could be created on demand via other techniques [Le, 1998], although many drop-on-demand applications may only be usable for micrometer-scale droplets. Finally, drop streams could be created by disturbing a jet, as summarized in [Frohn and Roth, 2000]. Using these alternate techniques could greatly increase the number of drops in the environment, allowing for dense 3D, or perhaps even immersive water drop displays.

Part II

Reducing water drops visual effect

Chapter 3

Removing rain and snow from videos

Rain and snow are often imaged as bright streaks. Not only can these streaks annoy or confuse a human viewer, but they degrade the effectiveness of any computer vision algorithm that depends on small features. For example, feature point trackers can fail if even small parts of an image are occluded. If these streaks are removed, then the tracker can work with greater accuracy. Alternately, rain may need to be added to a scene. For example, after a shot is taken, a movie director may decide that there should be more rain. The scene could be filmed again, but this would be costly and time consuming. Rather than requiring people to wait for the weather to be perfect, we develop techniques to digitally control the amount of rain and snow in a video.

Rain and snow are specific examples of bad weather. Although one good day is much like another, the properties of bad weather vary depending on the size of the constituent particles. Static bad weather, such as fog and mist, are caused by microscopic particles. Due to the small particle size, fog and mist are usually spatially and temporally consistent. Since their effect does not vary significantly over space and time, it is sufficient to only analyze their effect locally, on individual pixels [Nayar and Narasimhan, 1999] [Narasimhan and Nayar, 2002a] [Cozman and Krotkov, 1997] [Narasimhan and Nayar, 2000] [Narasimhan and Nayar, 2001] [Schechner et al., 2001] [Narasimhan and Nayar, 2002b] [Narasimhan and Nayar, 2003]. For large particles such

Figure 3.1: The snow in the left image has been detected by finding its global spatio-temporal frequencies. For the right image, the brightness and amount of snow is then manipulated to increase from its left to right.

as raindrops and snowflakes, analysis is more difficult. Spatially and temporally neighboring areas are affected by rain and snow differently, so must be handled differently.

Several methods have been developed to remove rain and snow from videos. The earliest use a temporal median filter for each pixel [Hase et al., 1999] [Starik and Werman, 2003]. Temporal median filtering exploits the fact that in all but the heaviest storms, each pixel is clear more often than corrupted. The problem is that anything that moves will become blurred. Zhang et al. [Zhang et al., 2006] extended the idea of per-pixel removal by correcting for camera motion via planar image alignment and detecting rain with k-means clustering. This method is an improvement over simple median filtering in cases where the scene is static and the video frames can be accurately aligned. Other methods include [Liu et al., 2009] [Padole and Vaidya, 2008].

Garg and Nayar [Garg and Nayar, 2004] suggested that streaks can be segmented by finding pixels in individual streaks that change over space and time in the same way as rain. False matches can then be reduced via a photometric constraint that models the appearances of streaks. Searching for individual streaks this way can theoretically work for dynamic scenes with a moving camera. But this method is most effective when the streaks are against a relatively textureless background. Other methods that remove the effect of water drops that use local neighborhoods

include [He et al., 2009] [Yamashita et al., 2005] [Yamashita et al., 2008] [Brewer and Liu, 2008] [Tripathi and Mukhopadhyay, 2011] [Roser and Geiger, 2009].

Garg and Nayar [Garg and Nayar, 2005] also demonstrated how to prevent rain from being imaged in the first place, by modifying camera parameters during acquisition. They suggest using temporal and spatial blurring, either by increasing the exposure time or reducing the depth of field. This removes rain for the same reasons as the per-pixel median filtering, and will not cause blurring when all objects are at the same depth or the scene is static. Other examples of changing the properties of the image include [Hara et al., 2009] [Nayar et al., 2004].

In this work, we combine realistic streak modeling with the knowledge of the statistics of dynamic weather. Unlike previous works that detect rain by only looking at individual pixels or patches, we treat rain and snow as image-global phenomena. In order to determine the influence of rain and snow on a video, we develop a global model in frequency space. In image space, single rain and snow streaks appear similar to any type of vertical stripe. Likewise, in frequency space, a single streak is difficult to distinguish in the clutter. But as the number of streaks increases, the pattern they cause in frequency space becomes distinct. Although spotting an individual tree might be hard, finding the forest is easy.

We begin with a physical model of a single raindrop or snowflake. The dynamics of falling particles are well understood [Foote and duToit, 1969] [Spilhaus, 1948] [Magono and Nakamura, 1965] [Böhm, 1989], and it is simple to determine the general shape of the streak that a given raindrop or snowflake will create. Based on the shape, the streak's appearance is then approximated as a motion-blurred Gaussian. The statistical characteristics of rain and snow have been studied in the atmospheric sciences [Marshall and Palmer, 1948] [Ulbrich, 1983] [Feingold and Levin, 1986] [Gunn and Marshall, 1958] [Ohtake, 1965] . It is possible to predict the expected number and sizes of the streaks as well. The information of how one streak appears, combined with a prediction of the range of streak sizes, allows us to predict the appearance of rain and snow in an image.

The problem with the image space model is that it is difficult to apply to real scenes. However, even such complex and chaotic phenomena as rain and snow can be well behaved in frequency space [Heeger, 1987] [Langer and Mann, 2003]. Therefore, rather than trying to find every rain and snow pixel in an image, we instead model their effect in frequency space. Although it is not possible to predict the exact streak sizes and locations in a video a priori, we can create a frequency-space model by sampling in particle size, depth from the camera, and streak orientation. The frequency model is then fit to an image sequence by matching streak orientation and rain/snow intensity. The inverse Fourier transform of the ratio between the model's predictions and the actual frequencies highlights the rain and snow in image space.

We perform a comprehensive comparison of this work with other methods of rain and snow detection and removal. We compare each algorithm both on the amount of rain/snow removed versus background corrupted and on how much the removal increases the accuracy of a feature point tracker. Six sequences are tested, half from real storms and half with realistically rendered rain added. The results demonstrate the advantage of image-global rain and snow analysis

Once detected, we are then able to either decrease the amount of rain and snow by subtraction, or increase it by sampling and cloning. Other researchers have also developed methods for rain and snow synthesis. Rain can be generated via various approximate methods [Langer and Zhang, 2003] [Langer et al., 2004] [Reeves, 1983] [Starik and Werman, 2003] [Tatarchuk and Isidoro, 2006] [Puig-Centelles et al., 2009] [Changbo et al., 2008] [Feng et al., 2006], but physically accurate rain synthesis [Garg and Nayar, 2006] involves accurately modeling how a raindrops deforms and refracts light as it falls. Well-designed rain textures combined with a particle system can be used to create realistic scenes [Tariq, 2007].

The advantage of combining detection, removal, and synthesis is that when the scene is uniformly illuminated, no additional scene analysis is required; the streaks are already correctly formed and illuminated. Instead of using separate tools to remove and to render rain and snow, we present a framework that does both.

3.1 Image-space analysis

A frame from a movie m acquired during a storm can be decomposed into two components: a clear image c and a rain/snow image r. Generally, a background scene point is occluded by raindrops or snowflakes for only a short time, therefore we can approximate their effect as being purely additive. For location (x, y) at time t, we have:

$$m(x, y, t) \approx c(x, y, t) + r(x, y, t) \tag{3.1}$$

In this paper, we develop an algorithm to find r, based on the overall appearance and statistical properties of rain and snow. Although it is sometimes possible to create clear videos by increasing the camera aperture and exposure time [Garg and Nayar, 2005], this paper focuses on cases where this is not possible, such as when the entire scene needs to be in focus or when there are fast-moving objects that should not be blurred. When in focus and not blurred by a long exposure time, rain and snow appear in images as bright streaks. We begin the analysis in image-space by creating an appearance model of a streak.

3.1.1 The shape of a rain or snow streak

Raindrops and snowflakes can have complex shapes. However in a typical video sequence, their shapes are not prominently visible, therefore we ignore any variation in their shape and consider them to be symmetric particles. At a given instant in time, a camera with focal length f images an in-focus particle of diameter a at a distance from the camera z as an image with breadth b:

$$b(a, z) = a\frac{f}{z} \tag{3.2}$$

If a particle is not in focus, then its image will be broader. For the purposes of image analysis, out of focus raindrops or snowflakes are less important than in-focus ones, because they have a

milder effect on images. (The appearance model in Section 3.1.2 implicitly handles slightly out-of-focus particles).

The lengths of streaks depends on how fast the particles are falling and how far they are from the camera. Because they are so small, wind resistance is a major factor, and their terminal velocities depend on their sizes. For common altitudes and temperatures, a raindrop's speed s can be approximated by a polynomial in its diameter a [Foote and duToit, 1969]:

$$s(a) = -0.2 + 5.0a - 0.9a^2 + 0.1a^3 \tag{3.3}$$

Finding the speed of snowflakes is more difficult [Magono and Nakamura, 1965, Böhm, 1989], because they have more complex shapes. But since our detection algorithm uses a range of streak sizes, it is not necessary to obtain exact bounds on individual snowflakes. As a result, snowflakes can be assumed to fall half as fast as raindrops of similar size. If the ratio between size and speed is approximately correct, then the streaks can still be detected.

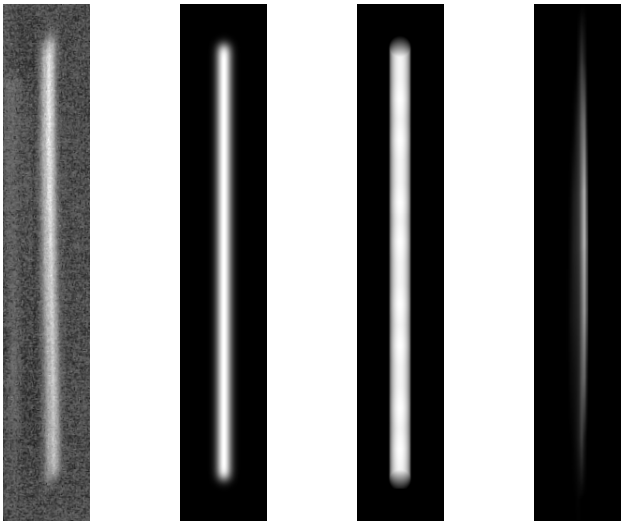
If the camera and particle are moving at constant velocities and the particle stays at a uniform distance, then it will be imaged as a straight streak. A falling particle imaged over a camera's exposure time e creates a streak of length l:

$$l(a, z) = (a + s(a)e)\frac{f}{z} \tag{3.4}$$

3.1.2 The appearance of a rain or snow streak

The appearance of a raindrop or snowflake depends on the particle's shape and reflectance, and the lighting in the environment. As shown in Figure 3.2 (a), under common lighting conditions, a falling drop will produce a horizontally symmetric streak.

A completely accurate prediction of a streak's coloring would require extensive physical modeling, as was done to render rain in [Garg and Nayar, 2006], and is shown in Figure 3.2 (c)



(a) (b) (c) (d)

Figure 3.2: Raindrops and snowflakes create streaks of different appearances, depending on factors such as the environmental illumination, their depth from the camera, and how much they are in focus. (a) is a streak from a real water drop under illumination from a broad source. The streak's appearance can be modeled by a blurred Gaussian (b) (Equation 3.5). (c) is a rendered streak from [Garg and Nayar, 2006] with broad environmental lighting. If the lighting is from a point source, then the streak would appear as in the point lighting example (d), which is also from [Garg and Nayar, 2006]. In this paper, we use the blurred Gaussian, because it has approximately the correct appearance and is efficient to compute.

and (d). But in most cases, the breadth is only a few pixels, and it is not necessary to form an exact model of light reflecting off a snowflake or determine the exact distorted image that will appear in a tiny drop [Garg and Nayar, 2004, Van de Hulst, 1957]. Instead, we use a simple, analytical model that is fast to compute and well-behaved in frequency space.

To begin, the image of a raindrop or snowflake is approximated as a Gaussian, which appears similar to a slightly out-of-focus sphere. As the particle moves in space, the image it creates is a linear motion blurred version of the original Gaussian. If the sphere is larger or closer to the camera, the Gaussian will have a higher variance. If it is falling faster, then it will be blurred into a longer streak. The equation of a blurred Gaussian g, centered at image location $\boldsymbol{\mu} = [\mu_x, \mu_y]$, with orientation θ, variance given by the breadth b of the streak, and motion blurred over the

length l of the streak, is given by:

$$g(x, y; a, z, \theta, \boldsymbol{\mu}) = \int_0^{l(a,z)} exp(-\frac{((x - cos(\theta)\gamma - \mu_x)^2 + (y - sin(\theta)\gamma - \mu_y)^2)}{b(a, z)^2})d\gamma \quad (3.5)$$

The values for diameter a and depth z are combined with Equations 3.2, 3.3, and 3.4 to compute the correct values of breadth b and length l. In the notation, a semicolon is used to differentiate between the parameters of image location versus all others. For example, $(x, y; a, z, \theta, \boldsymbol{\mu})$ means at location (x, y), with parameters $a, z, \theta, \boldsymbol{\mu}$.

An example of this appearance model is shown in Figure 3.2 (b). With broad environmental lighting from the sky, the variations due to the drop oscillations discussed in [Garg and Nayar, 2006, Tokay and Beard, 1996, Kubesh and Beard, 1993] are subtle, so Figure 3.2 (a), (b), and (c) appear almost identical. Even though blurred Gaussians are an inaccurate approximation of raindrops illuminated with a point light source, most outdoor scenes are not lit with point sources during the day, so the effects of oscillation can be ignored.

3.1.3 The appearance of multiple streaks

The pixel intensity due to rain or snow in one movie frame at a given location (x, y) should be the sum of the streaks created by all N visible drops:

$$\sum_{d=1}^{N} g(x, y; a_d, z_d, \theta_d, \boldsymbol{\mu}_d) \quad (3.6)$$

For a given time t, each of a_d, z_d, θ_d, and $\boldsymbol{\mu}_d$ are drawn from different distributions. Drops are equally likely to appear at any location in space, so the x and y positions in $\boldsymbol{\mu}_d$ are drawn from uniform distributions. Because a greater volume is imaged further from the camera, more drops are liable to be imaged at greater depths, so z_d is drawn from a simple quadratic distribution. Streak orientation has a mean orientation θ_d, with a slight variance. The most problematic

parameter is the drop size a_d. Fortunately, many researchers in the atmospheric sciences have studied the expected number of each size of raindrop or snowflake, and we draw upon their conclusions.

It is well known that in a single storm, there will be particles of various sizes. Size distributions are commonly used for raindrops [Marshall and Palmer, 1948, Ulbrich, 1983, Feingold and Levin, 1986], snowflakes [Gunn and Marshall, 1958] [Ohtake, 1965], and various other hydrometeors, such as graupel and hail [Auer Jr. and Veal, 1970] [Auer Jr., 1972]. Previous works on rain removal [Garg and Nayar, 2004] [Garg and Nayar, 2005] have used the Marshall-Palmer [Marshall and Palmer, 1948] distribution. For more information, Microphysics of Clouds and Precipitation by Pruppacher and Klett [Pruppacher and Klett, 1997] is a good general resource for the physics of precipitation. Unfortunately, as discussed by several authors [Jameson and Kostinski, 2001] [Jameson and Kostinski, 2002] [Desaulniers-Soucy et al., 2001] [Desaulniers-Soucy, 1999], size distributions can be inaccurate. Nevertheless, they give useful general bounds.

Drops as large as 4.5mm in diameter (and 10mm in base diameter) have been produced in laminar air streams in the lab [Pruppacher and Pitter, 1971]. As drops become larger and larger, their shape changes from a sphere to oblate spheroids, with a flat base and a rounded top [Pruppacher and Klett, 1997] Beard [1976] [Gorgucci et al., 2006].

As discussed in many meteorology papers, both size distributions and observational studies show that in all but the most intense rain, drops rarely grow larger than $3mm$ [Pruppacher and Klett, 1997] [Mason and Andrews, 1960] [Blanchard, 1953] [Willis and Hallett, 1991] [Hobbs and Rangno, 1998] [Uijlenhoet, 1999] [Fleishauer et al., 2002]. In addition, drops smaller than $.1mm$ cannot be seen individually. Although not accurate for every storm, we find that using a uniform distribution between $.1mm$ and $3mm$ is sufficiently accurate.

3.2 Image-space versus Frequency-space analysis

Having developed an image-space model of rain and snow streaks, it is conceivable that we could search for streaks directly in the images. An exhaustive approach would be to search across all pixel locations (x, y), with each possible number of streaks N, drop location $\boldsymbol{\mu}_d$, drop size a_d, and drop depth z_d. Although it would be prohibitive to do this exhaustive search, computing statistics over long sequences might allow the streaks to be gradually found and localized.

The advantage of using frequency space is that the separation of the phase and amplitude components is well suited to the structure of rain and snow. As a first approximation, we want to find the parts of the images that roughly correspond to rain. As the amplitude is shift invariant, we can easily detect these overall properties, before we find exactly where they are appearing.

Perhaps the best solution would be to alternate between a frequency space and an image space representation. The rough locations of streaks could be found in image space, the image could be inverse-transformed, then the locations could be refined in image space (perhaps using gradient descent, or a standard convex optimization.) Alternately, a different representation might be appropriate, such as a similarity map between different pixels or spatio-temporal regions.

But although there are pros and cons to using a frequency space model, it allows for a simple representation and can be fitted to a video sequence relatively fast. It appears to be fairly effective, therefore we will begin with this approach.

3.3 Frequency-space analysis

Since rain and snow streaks create repeated patterns, it is natural to examine them in frequency space. Rather than attempting to find each pixel of each streak, we can instead find their general effect on the Fourier transforms of the images. But applying the Fourier transform to Equation 3.6 does not make it easier to analyze images. For this, we make three key observations of the magnitude of the Fourier transform of rain and snow.

Observation 1: The shape of the magnitude does not depend strongly on streaks' locations in an image. Figure 3.3 shows an example of the Fourier transform of a sequence with real rain. The middle column is a sequence of three consecutive frames. They were generated from a sequence of heavy rain with a stationary scene and with an almost stationary camera, by finding the difference of each pixel with the median of itself and its two temporal neighbors:

$$|M(x,y,t) - median(M(x,y,t-1), M(x,y,t), M(x,y,t+1))| \tag{3.7}$$

The left column is three separate two-dimensional Fourier transforms, one for each image. Notice that even though the streaks are in different locations in different frames, the magnitudes appear similar. Appendix B contains a derivation and simulation that shows the magnitude is only weakly dependent on the number and positions of streaks. We find that although the expansion of the magnitude of the Fourier transform of rain and snow can be arbitrarily complex, it can still be well behaved, which explains the phenomenon seen in the left column.

Observation 2: The shape of the magnitude is similar for different numbers of streaks. Although the exact Fourier transforms of images with different numbers of streaks are different from each other, changing the number of streaks has a similar effect to multiplying all frequencies by a scalar. This pattern is shown in 3.4. Appendix B also contains a validation of this observation, for the special case where there are the same number of streaks of each length at each location.

Observation 3: The magnitude is approximately constant across the temporal frequencies. This rightmost column of Figure 3.3 shows the three-dimensional Fourier transform of all three frames. Interestingly, apart from a few artifacts, the magnitude appears similar across temporal frequency w. This observation allows us to predict that the three-dimensional Fourier transform will be constant in temporal frequency w.

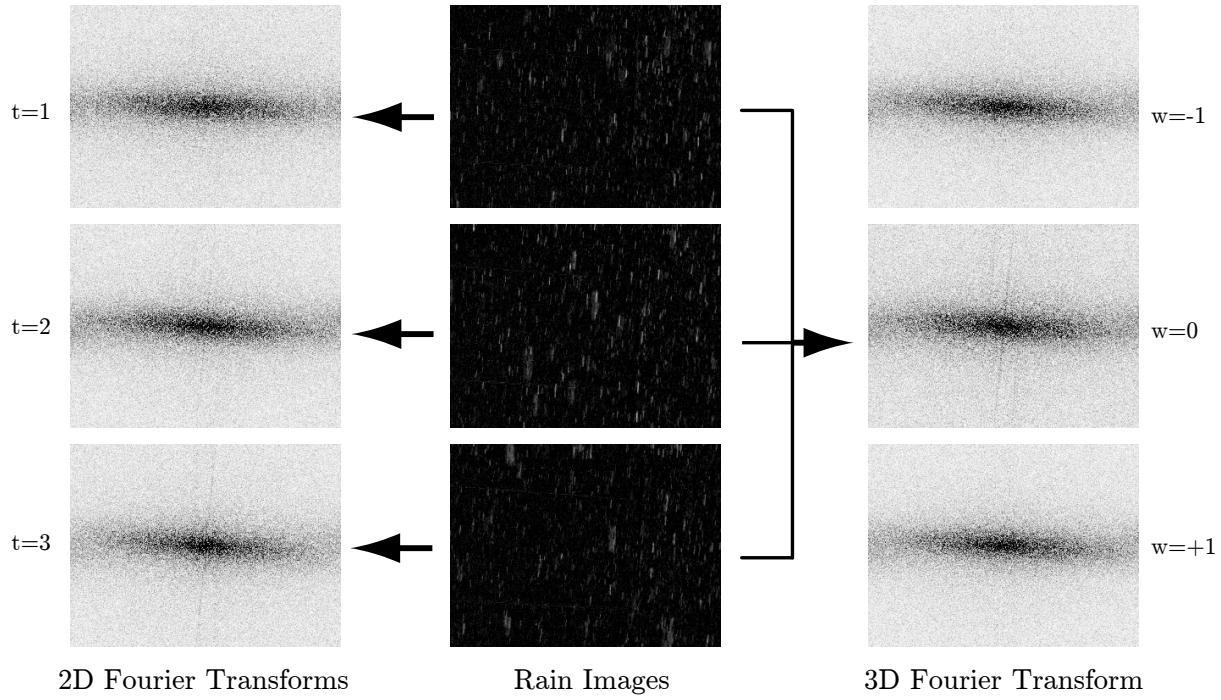
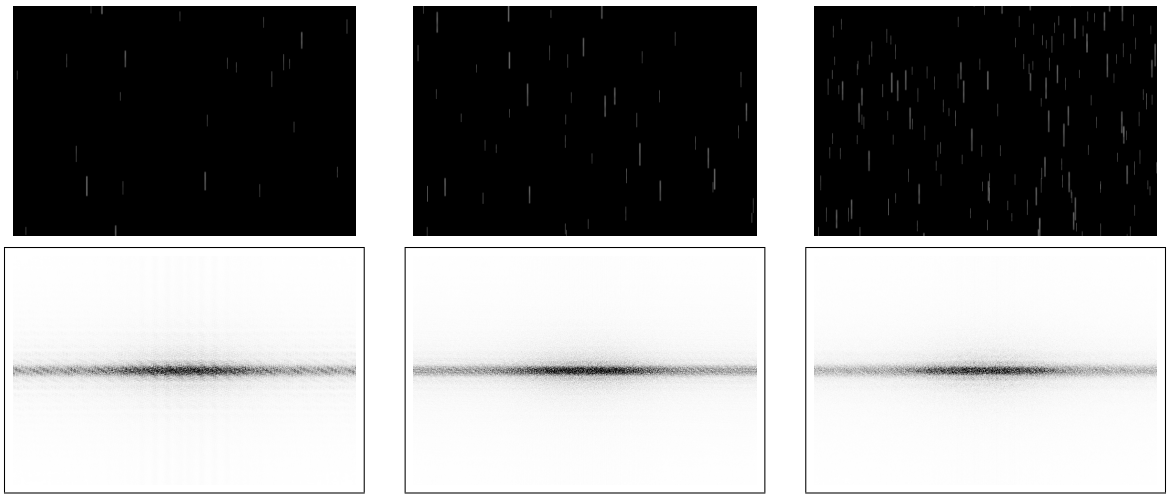


Figure 3.3: The center column is three consecutive frames of rain acquired at times $t = 1, 2, 3$. The left column is three two-dimensional Fourier transforms, one for each of the images. The right column is a single three-dimensional transform of all three frames, with temporal frequency $w = -1, 0, 1$. As expected, the $w = 1$ and $w = -1$ frequencies are mirror images. But what is interesting is that *all* of the Fourier transform images appear similar, due to the statistical properties of rain.

These observations will allow us to create a simplified model of the frequencies of rain and snow. Instead of finding every streak individually, we can fit this model by only estimating a few parameters.

3.3.1 A frequency-space model of rain and snow

As shown in Equation 3.6, an image full of rain or snow is the sum effect of a group of streaks. The same is true in frequency space, where the magnitude of the Fourier transform of Equation



(a) (b) (c)

Figure 3.4: Three examples of images with streaks rendered by [Garg and Nayar, 2006] and their corresponding two dimensional Fourier transforms. From left to right, the images have approximately 50, 100, and 300 streaks. To make them appear similar, each Fourier transform is multiplied by a scalar. Apart from being scaled differently, their magnitude appear similar.

3.6 is:

$$||\mathcal{F}\{\sum_{d=1}^{N} g(x, y; a_d, z_d, \theta_d, \boldsymbol{\mu}_d)\}|| \tag{3.8}$$

which is equivalent to the sum of the Fourier transforms of each streak g:

$$|| \sum_{d=1}^{N} G(u, v; a_d, z_d, \theta_d, \boldsymbol{\mu}_d)|| \tag{3.9}$$

(Note that in this work, we use only the main lobe of the blurred Gaussian G, which has a similar appearance to a standard oriented Gaussian).

Based on the Observations 1 and 2 in the previous section, Equation 3.9 can be simplified as:

$$\sum_{d=1}^{N} ||G(u, v; a_d, z_d, \theta_d)|| \tag{3.10}$$

Equation 3.10 is simpler, but still depends on the number of streaks N in the image. This is

where the statistical properties of rain and snow discussed in Section 3.1.3 become helpful. Since determining the exact value of each frequency is not vital, we can simplify Equation 3.10 further, based on three assumptions. First, each spatial location $[z_{min}, z_{max}]$ is equally likely to have a raindrop or snowflake. In a perspective camera, the volume imaged at a given depth is relative to the depth squared. This means that in a perspective camera, the number of drops imaged at a given depth will also be relative to the depth squared. Second, the resulting streaks are equally likely to have any orientation within the range $[\theta_{min}, \theta_{max}]$. Third, a given particle is equally likely to be any size between $a_{min} = .1mm$ and $a_{max} = 3mm$.

Instead of trying to determine the properties of each of the N streaks, we use a model R^* that has frequencies proportional to the mean streak and scaled by overall brightness Λ:

$$R^*(u, v; \Lambda, \theta_{max}, \theta_{min}) = \Lambda \int_{\theta_{min}}^{\theta_{max}} \int_{a_{min}}^{a_{max}} \int_{z_{min}}^{z_{max}} z^2 ||G(u, v; a, z, \theta)|| dz\, da\, d\theta \tag{3.11}$$

These integrals can be approximated by sampling across θ, a, and z, yielding an estimate of the frequencies of rain and snow.

From Observation 3, we can predict that the magnitude will be constant in temporal frequency w:

$$R^*(u, v, w; \Lambda, \theta_{max}, \theta_{min}) = R^*(u, v; \Lambda, \theta_{max}, \theta_{min}) \tag{3.12}$$

The scalars for rotation θ and brightness Λ are based on the specific movie. In the next section, we show how to fit θ and Λ.

3.3.2 Fitting the frequency-space model to a video

Only a single intensity Λ needs to be estimated per frame, and often only one orientation θ per sequence. To estimate these parameters, we can use the fact that rain and snow cover a broad

part of the frequency space. Most objects are clustered around the lowest frequencies, while rain and snow are spread out much more evenly. This means that even if the total energy of the rain or snow is low, a frequency chosen at random is fairly likely to contain a strong rain or snow component. This is especially true if we only examine the non-zero temporal frequencies, which are those that correspond to changes between frames.

The model parameters can be estimated with two heuristics. The scalar multiplier Λ should be such that the rain/snow model is approximately the same magnitude as the rain or snow in the movie. For the Fourier transform of a small block of frames, Λ can be estimated by taking a ratio of the median of all frequencies, except for the constant temporal frequencies $w = 0$:

$$\Lambda \approx \frac{median(||M(u,v,w)||)}{median(R^*(u,v,w;\Lambda = 1,\theta_{max},\theta_{min}))} \tag{3.13}$$

Taking the median is effective, because as discussed in Observation 3 in the beginning of the section, rain and snow are strong in non-zero temporal frequencies, while most of the scene is concentrated in the zero temporal frequencies.

The streak orientation can be automatically computed if there is a short subsequence where only rain and snow are moving. Again using Observation 3, we expect that individual rain and snow frequencies will change greatly, even though their overall effect stays the same. To find orientation, we do not need to find the correct values for each frequency, we only need to determine which are due to rain and snow. Therefore, rather than using the median of the frequencies as in Equation 3.13, we use the standard deviation across time as a more robust estimator. An estimate $\tilde{R}$ of the important frequencies can be obtained by computing the standard deviation over time for each spatial frequency, for T frames:

$$\tilde{R}(u,v) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(||M(u,v,t)|| - ||\overline{M}(u,v)||)^2} \tag{3.14}$$

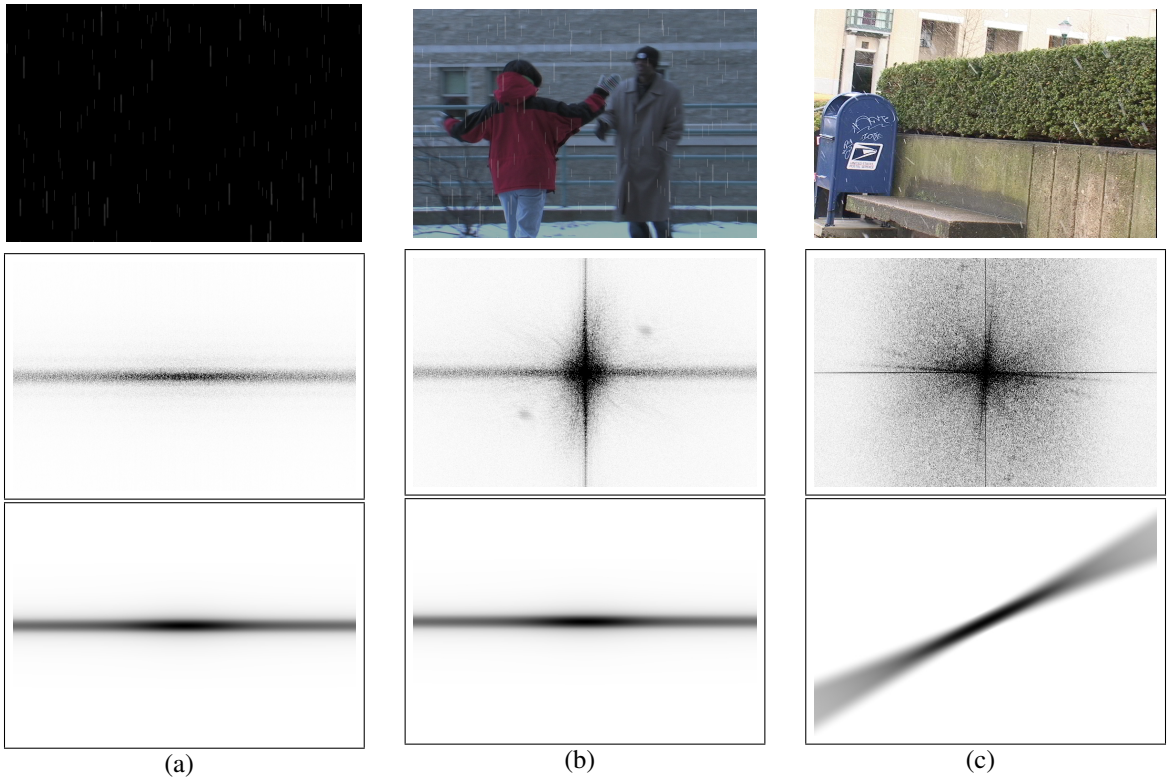


Figure 3.5: Examples of the model for three video sequences. From top to bottom, we have the original image, its two dimensional Fourier transform, and the corresponding rain/snow model. (a) A sequence of a black background, plus 300 rendered streaks per image. (b) The same 300 streaks, but now against a moving background with a moving camera. (c) Real snow and a moving camera.

The correct θ is found by minimizing the difference between the model and the estimate:

$$\underset{\theta}{\operatorname{argmin}} \iint (||R^*(u,v;\Lambda,\theta_{max},\theta_{min})|| - \tilde{R}(u,v))^2 dv\, du \tag{3.15}$$

Because the search space is one dimensional and bounded, an exhaustive search can be performed. Different raindrops generally fall in almost the same direction, so $\theta_{min} = \theta_{max}$ for rain. But since snow has a less consistent pattern, a range of orientations is needed. Using $\theta_{min} = \theta_{max} - .2$ radians is effective for most videos with snow.

Figure 3.5 shows the models obtained by fitting to three videos. The frequencies corresponding to the rain are easy to see in (a) and (b), but it easier to see the snow frequencies in videos.

3.4 Applications

The model that we developed in the previous sections can be used to either decrease or increase the amount of rain and snow. For both cases, the first step is detection, which requires an analysis across entire images, and is performed in frequency space. Once detected, the rain or snow can either be directly removed by subtraction, or else the detected pixels can be blended with their temporal neighbors. Alternately, to increase the amount of rain or snow, individual streaks can be found by matching with blurred Gaussian in image space and then copied onto another image.

3.4.1 Detecting rain and snow using frequency space analysis

The frequency model can be used to detect rain and snow in a similar way to notch filtering [Gonzalez and Woods, 2002]. Intuitively, we want to highlight those frequencies corresponding to rain and snow while ignoring those corresponding to objects in the scene. This can be done with a simple ratio. For example, suppose that the model predicts a low value for a given frequency, but the actual value is high. Something besides rain or snow is likely causing the high value. The frequencies that are mostly due to rain and snow should be found first, as estimated by the ratio of the predicted value to the true value.

Detecting streaks in a single frame is done by taking the inverse transform of the estimate of the proportion of energy due to rain or snow. Where $M(u, v)$ is the two dimensional Fourier transform of one movie frame and ϕ is the phase of $M(u, v)$, p_2 is the estimate based on a single image at time t:

$$p_2(x, y, t) = \mathcal{F}^{-1}\{\frac{R^*(u, v; \Lambda, \theta_{max}, \theta_{min})}{||M(u, v)||} exp(i\phi\{M(u, v)\})\} \tag{3.16}$$

The output is an image that is bright only where rain or snow is detected. When $R^*(u, v)$ is less than $M(u, v)$, then the ratio is the estimated percentage of rain and snow at that frequency.

For example, for a given (u, v), if $R^*(u, v) = 3$ and $M(u, v) = 10$, then $R/M = .3$. This means that we believe that thirty percent of the energy at (u, v) is due to rain and snow. If $R^*(u, v) > M(u, v)$, then the ratio is greater than one, which is not meaningful. The ratio of R^* over M is therefore capped at one. This capping is both semantically valid as well as practical, in that it prevents frequencies with a very high value for R^*/M from dominating the result.

Figure 3.6 (b) shows the one-frame estimation. For visual comparison, Figure 3.6 (c) shows the result if 50% of the frequencies in the model are set to zero before using Equation 3.16. Figure 3.6 (d) shows the result if the ground truth of the rain magnitude is used in place of the rain model.

By performing a three-dimensional transform of the images and using (u, v, w) instead of (u, v), Equation 3.16 can be used for a three dimensional transform of multiple consecutive frames, shown in Figure 3.6 (e). Using multiple frames improves the accuracy, but not significantly.

We found through experimentation that the best approach is to perform a three dimensional analysis on a series of consecutive two dimensional estimates. A three dimensional Fourier transform is applied to $p_2(x, y, t)$ to obtain $P_2(u, v, w)$, and the resulting rain/snow estimation is then:

$$p_3(x, y, t) = \mathcal{F}^{-1}\{\frac{R^*(u, v, w; \Lambda, \theta_{max}, \theta_{min})}{||P_2(u, v, w)||} exp(i\phi\{M(u, v, w)\})\} \tag{3.17}$$

Figure 3.6 (f) shows the results from this method. At first glance, it appears even better than the single frame ground truth in Figure 3.6 (d). But the ground truth magnitude actually correctly identifies streaks more precisely, even if it has more false detections. But since the ground truth is not generally known, we use p_3 as our final estimate of the location of the rain and snow.

3.4.2 Reducing rain and snow using the frequency space model

Once detected, the rain or snow pixels can be removed by replacing them with their temporal neighbors. The detected rain and snow $p_3(x, y, t)$ is used as a mixing weight between the original image m and an initial estimate $\tilde{c}$ of the clear image c. We find that a per-pixel temporal median filter works well for $\tilde{c}$, although it could be the output of any rain/snow removal algorithm. The detection $p_3(x, y, t)$ is multiplied by the removal rate α, where the product of $\alpha p_3(x, y, t)$ is capped at one:

$$c(x, y, t) = (1 - \alpha p_3(x, y, t))m(x, y, t) + \alpha p_3(x, y, t)\tilde{c}(x, y, t) \tag{3.18}$$

Since rain and snow are brighter than their background, $c(x, y, t)$ is required to be less than or equal to $m(x, y, t)$. For a large α, $\alpha p_3(x, y, t)$ equals 1 for all (x, y, t), therefore c will approach $\tilde{c}$.

Images created with this equation will be temporally blurred only where the rain and snow is present. But the disadvantage is that it can never remove more rain and snow than the initial estimate $\tilde{c}$. If removal is more important than smoothness, then we can iterate the detection and removal.

The first iteration c^1 is the result from Equation 3.18 on the original sequence. Subsequent iterations are based on the last clear estimate c^{n-1} and the last initial estimate $\tilde{c}^{n-1}$. In this case, $\tilde{c}^{n-1}$ is the per-pixel temporal median of c^{n-1}. Where p_3^n is the detection from Equation 3.17 as applied to c^{n-1}, the next iteration is:

$$c^n(x, y, t) = (1 - \alpha p_3^n(x, y, t))\ c^{n-1}(x, y, t) + \alpha p_3^n(x, y, t)\ \tilde{c}^{n-1}(x, y, t) \tag{3.19}$$

Selecting a good value for α is not difficult. We use a fixed $\alpha = 3$ for all the results in this paper.

And as with Equation 3.18, each iteration is required to be less than or equal to the original.

Although the result from each subsequent iteration is more clear than the previous, the amount of rain and snow removed decreases per iteration. This means that it may be necessary to iterate many times to remove most of the streaks. Since this is time consuming, the process can be iterated only a few times, and subsequent c^ns can be linearly extrapolated from the final two iterations. Figure 3.7 shows the results from the iterative removal method on four example sequences with moving cameras and scenes.

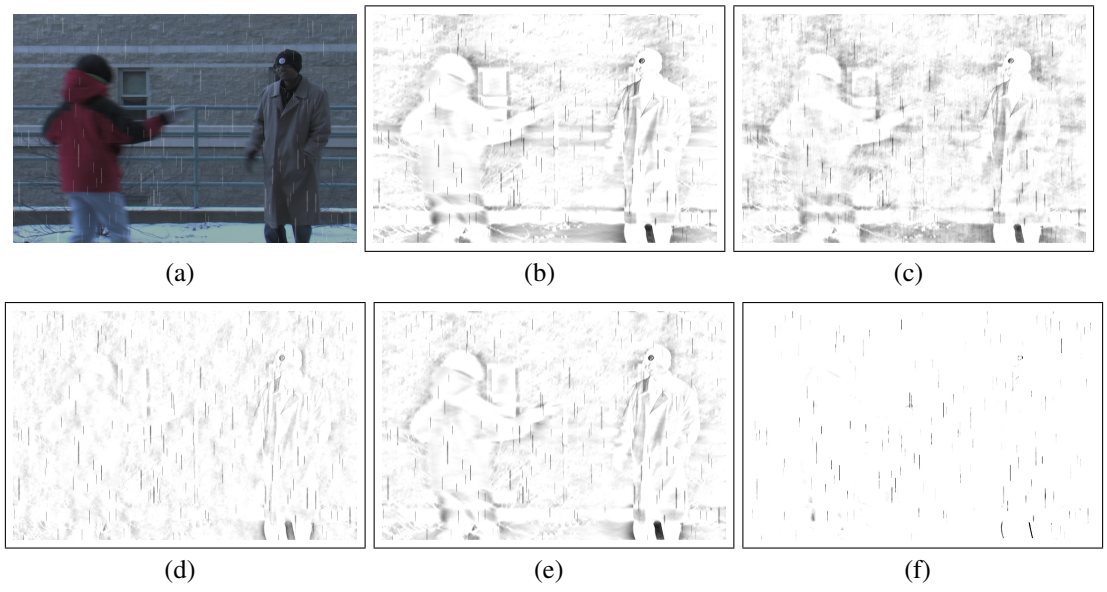


Figure 3.6: Rain can be detected in several ways, with the same frequency model. Subfigure (a) shows a frame from the original sequence, which has rendered rain streaks. (b) With detection based on a single frame, the rain is segmented fairly accurately, but there are many false detections. Even the fairly textureless ground is mistakenly detected, because it shares many of the low frequencies of the model. In (c), detection still uses a single frame, but a random 50% of the model's frequencies are set to zero. (The effect of setting some frequencies to zero is more evident in the videos on the website). The true magnitude of the rain is used instead of our model in (d). The rain is still detected accurately, although there are fewer erroneous detections. The reason the ground truth magnitude has any errors is because our method of computing the rain/snow component does not generally allow a complete separation of rain/snow and the clear image. This example shows the theoretical limit of using a ratio of magnitudes for a single frame. (e) shows detection based on three consecutive frames, with similar accuracy to one frame. The best results are in (f) when detection is performed on a single frame, then refined over three frames. The exact frequencies of rain and snow change from frame to frame, and using the two step estimation finds only those frequencies that are both rain-like and rapidly changing.

(a) The mailbox sequence: There are objects at various ranges, between approximately 1 to 30 meters from the camera. The writing on the mailbox looks similar to snow. Most of the snow can be removed, although there are some errors on the edges of the mailbox and on the bushes.

(b) Walkers in the snow: This is a very difficult sequence with a lot of high frequency textures, very heavy snow, and multiple moving objects. Much of the snow is removed, but the edges of the umbrella and parts of the people's legs are misclassified.

(c) Sitting man sequence: This scene is from the movie Forrest Gump. The rain streaks are fairly large, as is common in films. The rain can be completely removed, although the letters and windows in the upper portion of the images are misclassified.

(d) A windowed building: The rain is not very heavy, but this sequence is difficult, because there are a large number of straight, bright lines from the window frames and the branches. Almost all of the rain is removed, but parts of the window frames and the bushes are erroneously detected.

Figure 3.7: Several examples of rain and snow removal based on spatio-temporal frequency detection. Some of the sequences have several moving objects, others have a cluttered foreground with high frequency textures, and all of them are taken with a moving camera.

3.4.3 Increasing rain and snow in image space

Figure 3.8: An image from the original mailbox sequence and one with added snow. The snow is removed and then sampled, yielding a streak database. The streaks are then added to the new image, increasing the amount of snow.

Although the rain and snow can be detected using only the frequency magnitude, creating new streaks requires manipulation of phase as well. The main advantage of working in frequency space was that the locations of the streaks could be ignored. But since we need them for rendering, it is simpler to work in image space. Our approach is to use the blurred Gaussian model to sample real rain and snow.

We start with the rain/snow estimate $m(x,t,y) - c^n(x,y,t)$. Large streaks are detected by filtering the rain/snow estimate with a bank of size derivatives of blurred Gaussians, similar to scale detection in [Mikolajczyk and Schmid, 2001]. For an average a and z with orientation θ, the size derivative is given by:

$$f(x,y;\gamma_1,\gamma_2,a,z,\theta) = g(\gamma_1 x,\gamma_1 y;a,z,\theta) - g(\gamma_2 x,\gamma_2 y;a,z,\theta) \tag{3.20}$$

where γ is a scalar, $\gamma_1 < \gamma_2$, and both of the blurred Gaussian terms are normalized to sum to one. Each image is filtered with a set of different γs. The filter with the maximum response corresponds to the size of the streak at that location.

This detection method will find many strong streaks, but will have a few false matches as well. Since we do not need to find every streak, several steps are taken to cull the selection. First, locations that appear to have very large and very small scales are eliminated, and non-maxima suppression is performed in both location and scale. Next, to ensure that only bright streaks are used, only the streak candidates with the most total energy are kept. (The total energy is the sum of the values of neighboring pixels within a small window). Finally, to prevent multiple off-center copies of the same streak, only the streak candidates that have the greatest percentage of their energy near their centers are kept. For a window of size (s_x, s_y), the energy near the center of a streak at (x, y, t) is given by:

$$\frac{\sum_{d_x=-s_x/2}^{s_x/2} \sum_{d_y=-s_y/2}^{s_y/2} (\sqrt{(\frac{s_x}{2})^2 + (\frac{s_y}{2})^2} - \sqrt{d_x^2 + d_y^2})\ m(x + d_x, y + d_y, t)}{s_x s_y \sum_{d_x=-s_x/2}^{s_x/2} \sum_{d_y=-s_y/2}^{s_y/2} m(x + d_x, y + d_y, t)} \tag{3.21}$$

The images of streaks of various sizes are then combined into a database. Optionally, artifacts can be reduced by projecting the magnitude of the sampled streak onto the magnitude of the blurred Gaussian streak model.

Once the database is created, it can be used in the same way as the database of streaks rendered with area-source environmental lighting from [Garg and Nayar, 2006]. The advantage of our method is that the sampled streaks already have natural variation in size and defocus blur. The disadvantage is that our method has no concept of lighting direction, so will not be accurate for night scenes where drop oscillations create complex specular effects. For scenes with area-source illumination from the sky, streaks from both our work and [Garg and Nayar, 2006] can be added in the same way.

Since the main focus of this work is on rain and snow detection during the day, we show only examples where the camera exposure is short and the scene is well-illuminated by the sky. Given an approximate depth map of the scene, the streaks can be rendered with the appropriate sizes

and densities. However, most of the visible streaks are within the depth of field of the camera, we sample uniformly in this volume to create both the spatially varying example in Figure 3.1 and the full-frame example in Figure 3.8.

3.5 Comparison of rain and snow removal methods

Various methods have been proposed to remove rain and snow from images [Hase et al., 1999] [Starik and Werman, 2003] [Garg and Nayar, 2004] [Zhang et al., 2006]. Ideally, a removal algorithm should output images of the scene as it would appear with no bad weather effects. No algorithm to date can completely clear an image without corrupting the background, but some are more effective than others. In this section, we quantitatively compare and qualitatively discuss the accuracy of each method on several sequences with real and rendered rain and snow.

3.5.1 Evaluation methodology

Each algorithm is compared quantitatively in two ways. First, we compare the amount of rain and snow removed versus the amount of the images incorrectly modified. Second, we run a feature point tracker on both the original sequence and the output of each algorithm, and compare the number of feature points correctly tracked.

We test each algorithm on three sequences with real snow (Figure 3.12) and three with rendered rain (Figure 3.11). Each sequence is either 720x480 or 640x480 pixels, and all are 60 frames long. Since the various algorithms require between three to thirty frames to initialize, only the accuracy on frames 30-59 is evaluated.

In the real sequences, the camera had a small aperture and a short exposure time, which causes bright, well-defined streaks. The streaks in the rendered sequences are generated with the photorealistic process of Garg and Nayar [Garg and Nayar, 2006]. We use the streaks rendered with large area source environmental-lighting instead of point-lighting, because our scenes are

illuminated by the sun.

Quantifying removal accuracy

The first metric is a per-pixel comparison of the amount of rain/snow removed compared to the amount of the background erroneously changed. (All equations are given for grayscale, although algorithms are evaluated by separately computing the error for each channel and averaging).

Each algorithm outputs an estimate of the true brightness of the rain at each pixel $\tilde{r}$:

$$\tilde{r}(x, y, t) = m(x, y, t) - c(x, y, t) \tag{3.22}$$

Since adding rendered rain to an image only increases the image brightness, a removal algorithm should ideally either decrease the image brightness or leave it constant. If the image is darkened, then the difference D between the true rain component r and the estimate $\tilde{r}$ is the arithmetic difference. And since any increase is an error, if the image is brightened, the difference D is how much the removal algorithm increased the image brightness:

$$D(x, y, t) = \begin{cases} r(x, y, t) - \tilde{r}(x, y, t) & \tilde{r}(x, y, t) > 0 \\ \tilde{r}(x, y, t) & \tilde{r}(x, y, t) < 0 \end{cases} \tag{3.23}$$

Once the difference D has been computed, each algorithm's accuracy is determined. H is the ratio between the amount of rain not removed and the total rain present. E is the ratio between the amount of the background incorrectly changed and the background's total energy:

$$H = \frac{\sum_{x,y,t} \{D(x, y, t) : D(x, y, t) > 0\}}{\sum_{x,y,t} r(x, y, t)} \tag{3.24}$$

$$E = -\frac{\sum_{x,y,t} \{D(x, y, t) : D(x, y, t) < 0\}}{\sum_{x,y,t} c(x, y, t)} \tag{3.25}$$

Figure 3.11 shows the removal accuracy for each method in the following sections. Methods that use a threshold can have varying values for H and E, and are plotted as lines. Specifics of the results are discussed in Section 3.5.2.

Quantifying feature-point tracking accuracy

The second metric is the increase or decrease in the number of feature points that can be tracked. Feature point tracking accuracy is selected as a comparison for several reasons. First, it does not require ground truth like the per-pixel accuracy evaluation, so sequences with real rain and snow can be used. Second, quantitative evaluation is simple; accuracy is the number of points correctly tracked. Third, tracking accuracy should be correlated with an algorithm's accuracy in preserving and revealing high frequencies in the scene.

We compare the results of tracking feature points in all six of the sequences mentioned in Section 3.5.1 and their corresponding de-weathered versions. The features are selected using the method of Shi and Tomasi [Shi and Tomasi, 1994], and tracked using a Lucas-Kanade tracker [Bouguet, 2000]. The strongest features are selected independently for the original sequences and the outputs of the removal algorithms.

We use the same evaluation method as [Sand and Teller, 2006], which is to track points while the sequence is played forward then backwards. Since the sequence starts and ends on the same frame, each point should be in the same location at the beginning and the end. Tracking accuracy is defined as the distance between each point at the beginning and end of the loop. For algorithms that use a threshold, results are reported at the threshold where the highest number of points are tracked within one pixel of accuracy.

Results are reported for the number of points tracked to within one pixel and five pixels of accuracy. We report both numbers, because rain and snow more often cause point tracks to be slightly dislodged than completely lost. The necessary accuracy depends on the application. Structure from motion requires points tracked within one pixel of accuracy, but five pixels of

accuracy is sufficient for object tracking. Figure 3.13 has point tracking results for all methods, with details discussed in the next section.

3.5.2 Explanation and evaluation of algorithms used in the comparison

Each detection and removal algorithm has three steps. First, using some combination of image processing, machine learning, and physical models, pixels are clustered into two categories: rain and non-rain. Second, an initial estimate of the true background is obtained as a temporal average or median. Third, pixels detected as rain are either partially or completely replaced by a pixel from the initial estimate. The first step is the main difference between methods, but each computes the initial estimate in a slightly different way. We use the temporal median filter with image alignment as the initial estimate for all algorithms, which allows for an indirect quantitative evaluation of detection accuracy.

No explicit detection

In some cases, it is not necessary to explicitly detect rain and snow in order to remove them. Temporal median filtering is the simplest method for cleaning videos [Hase et al., 1999] [Starik and Werman, 2003]. For this method, each pixel is replaced with the median of its values over time. If the scene and camera are completely static, then this is often the most accurate and visually pleasing way of removing rain and snow.

The main advantage of this method is that it is extremely fast, but if the camera is not stationary or there are moving objects, then median filtering performs poorly. The problems are most noticeable when tracking feature points.

Image alignment can increase the accuracy significantly. To align frames, we perform RANSAC [Fishler and Bolles, 1981] on SIFT [Lowe, 2004] features, computing either the image translation or a full homography between images. Interestingly, in some difficult cases, translational alignment can give more accurate results. This is likely because correcting for translation only

requires the values of the x and y offsets to be found, which is less prone to errors than computing a full homography. In both cases, the aligned image is made the same size as the original. Pixels with unknown values are set to their corresponding value in the unaligned image.

The point tracking results in Figure 3.13 are evidence of the usefulness of this simple image alignment. But unless the alignment is accurate, errors are visible in areas with strong gradients. These errors result in low scores in comparisons between the rain-removed and background-corrupted comparisons. However, the strongest features are kept and enhanced, allowing for superior tracking results. It is interesting that aligning can increase performance even in stationary scenes. This appears to be because strong, repeated features are aligned first and then median filtered, enhancing them rather than blending them with noise. On simple stationary scenes, translation-only alignment has better 5-pixel accuracy, while homography alignment has better 1-pixel accuracy. But in general, translation-only alignment is more accurate, and should be used for applications requiring either high accuracy or large numbers of point tracks.

It would be interesting to test other types of image alignment. Computing layers of motion with methods such as [Torr et al., 1999, Ke and Kanade, 2002, Zelnik-Manor et al., 2006], then aligning and filtering each layer separately could increase performance. In addition to layer extraction, techniques such as Gaussian mixture models [Stauffer and Grimson, 1998], kernel density estimation [Elgammal et al., 2000], or robust PCA [de la Torre and Black, 2001] could model the variation caused by rain and snow more effectively.

Per-pixel detection

Instead of applying the same approach on all pixels, most techniques first determine which pixels are rain. This is similar to the idea of background subtraction, except the foreground layer is just the rain and snow.

If the only difference between aligned frames is rain or snow, then each pixel should have one of two values. In Zhang et al. [Zhang et al., 2006], k-means with two clusters is used on

(a) (b)

Figure 3.9: Two example frames and detection results for [Zhang et al., 2006]. (a) K-means can often correctly segment streaks, although there are errors around strong gradients and very bright parts of the images. (b) In this other frame, the grayvalue intensity of the mailbox increased by approximately 10, which causes additional false positives. Only the false detections around strong gradients cause noticeable errors in the snow-removed images, because areas with weak gradients do not change when median filtered.

the grayscale intensity of each pixel over all frames. Since rain and snow are bright, the pixels corresponding to the cluster with the higher grayvalue intensity are tagged for removal.

Zhang et al. [Zhang et al., 2006] also discuss reducing false matches by using the fact that rain is normally has a neutral hue. Colorful pixels are unlikely to be rain. In practice, this generally requires hand tuning for each sequence, so it is not included in the comparison.

Two examples of applying this technique are shown in Figure 3.9. In such scenes, the rain and non-rain clusters are not always well separated, and some pixels are misclassified. Although many pixels are correctly labeled, individual pixels sometimes flicker, causing the low numbers of tracked feature points shown in Figure 3.13. The flickering causes points to be lost more often than slightly mis-tracked, causing low scores for both 1-pixel and 5-pixel accuracies. A tracker that uses more than two frames might not have as much difficulty with single frame impulses, but this would be true of unmodified rain and snow videos as well.

Proper application of morphological operators and blurring have potential to improve results, although we did not find an effective combination for the sequences tested. It might be possible

(a) (b)

Figure 3.10: One frame each from two sequences, and its corresponding correlation magnitude [Garg and Nayar, 2004] (scaled linearly for display). (a) For the stationary sequence of the reflective ball, all of the current streaks are visible in the magnitude image, plus ghosts from streaks in the previous frames. (b) The general characteristics of the scene can be seen in the magnitude image, but because the camera is moving, the scene appears ghosted.

to extend this method, so that instead of doing a hard assignment between two clusters, a distance metric is used to determine how far a pixel is from the rain cluster.

Patch-based detection

Instead of looking only at individual pixels, Garg and Nayar [Garg and Nayar, 2004] suggest that detection should involve examining small patches over a long sequence of frames. The algorithm has three steps. First, all pixels that flicker from dark to light then back to dark are labeled as "candidate" pixels. Second, these candidate images are thresholded and segmented with connected components. Components that are not linearly related within a threshold are eliminated, resulting in a binary image where each pixel is either 0 for non-rain or 1 for rain. The correlation of individual pixels within small patches is computed to find the magnitude and angle of the rainfall.

We found that for the second step, it is difficult to set a threshold that allows individual streaks to be segmented. Therefore, we skip the second step, and set all candidate pixels to 1 to create the binary images. Figure 3.10 shows examples of the magnitude of the correlation.

In the original paper, a single threshold was set to differentiate rain from non-rain. Rather than trying to find the optimal threshold by hand, an ROC curve is computed for the magnitude of the correlation. If the magnitude is above a given threshold, the pixel is replaced by the three-

frame temporal median. As more and more pixels are classified as rain, the rain-removed images become increasingly closer to the median images.

There is a clear trend on the removal accuracy curve, but the point tracking accuracy is hard to quantify for this algorithm. As increasingly lower thresholds are chosen, this method converges to the median filter result, which usually has the best tracking accuracy.

As with other removal methods, converting from hard to soft constraints would likely improve accuracy. In addition, no image alignment is advocated in this method, but aligning via one of the methods discussed earlier could also improve the results.

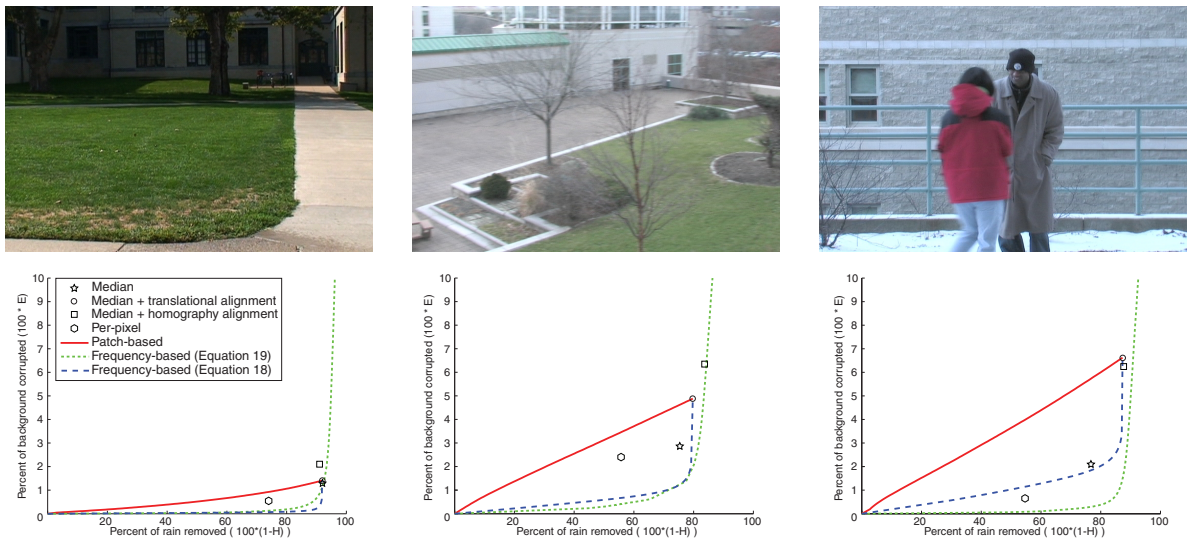
Frequency-based detection

Two versions of removal for the spatio-temporal frequency method presented in this work are compared. For the method of Equation 3.19, we use a fixed removal rate $\alpha = 3$ with four iterations, and we linearly interpolate to predict the pixel values at different levels of removal. As the level of removal is increased, all pixels are forced to decrease monotonically. For the method of Equation 3.18 the rain/snow detection is only computed once, and the value of α is changed to generate the ROC curves.

Both of these frequency-based methods are usually accurate in terms of amount of rain removed versus background corrupted, but do not increase the number of tracked points more than the aligned median, shown in Figure 3.13. This is because this method is able to reduce the brightness of most streaks with few errors, but it rarely completely eliminates all streaks. This means that features still become occluded by flickering streaks.

3.6 Conclusion and future directions

We have demonstrated a method for globally detecting rain and snow, by using a physical and statistical model to highlight their spatio-temporal frequencies. Previous works have shown that



(a) (b) (c)

Figure 3.11: Three sequences are used for tests with rendered rain. The origin corresponds to unmodified images, and the lower right corresponds to perfect removal accuracy. As the thresholds for the patch-based and frequency-based (Equation 3.18) are changed, they become more and more similar to the median. The per-pixel accuracy will always lie somewhere between the original image and the aligned median, depending on how many pixels are detected as rain. (a) Grass lawn: A stationary camera views a grass lawn and a few buildings. All algorithms are able to remove most of the rain with little background corruption. (b) Park and patio: A rotating camera was used to acquire this sequence of a small park and patio. The scene is mostly stationary, except for the trees waving slightly in the breeze. Results are similar to the stationary grass lawn, except all algorithms have larger error. (c) Two friends: This video of two people greeting each other was acquired with a moving camera. Because the foreground motion causes errors in the automatic alignment, the median actually performs better without alignment. But even if the background was fully aligned, the foreground motion would cause low scores for the median filter methods. And because there are large blocks of uniform color, per-pixel detection is able to correctly classify most of the pixels, yielding a low corruption score. For all sequences, the frequency-based removal performs the best.

examining only pixels or patches can be used to enhance videos in some cases, but the best results come from treating rain and snow as global phenomena.

Even a human observer can have difficulty in finding individual streaks in an image, although groups are easy to see. By generating a rain and snow model based on the expected properties of groups of streaks, we are able to achieve accuracy beyond what can be expected from local image analysis. On several challenging sequences, we show that rain and snow can be reduced

(a) (b) (c)

Figure 3.12: Real sequences of rain and snow. (a) Reflective sphere: This scene is of a reflective sphere, acquired with a stationary camera. The entire scene changes very little, except for a light amount of snow. (b) Snowy mailbox: This sequence of a mailbox, bushes, and building is acquired with a moving camera. The camera motion is mostly rotational, and the images can be aligned reasonably well with a planar homography. (c) Pedestrians in the snow: With multiple people walking, heavy snow, and a moving camera, this sequence is the most complex.

	Maximum tracking error	
Method	1 pixel	5 pixels
With no rain	838	953
With rain added	241	600
Median	750	927
Median+trans	766	924
Median+homog	817	920
Per-pixel	623	906

(a) Grass lawn

	Maximum tracking error	
Method	1 pixel	5 pixels
With no rain	729	814
With rain added	95	424
Median	202	698
Median+trans	618	789
Median+homog	597	802
Per-pixel	292	684

(b) Park and patio

	Maximum tracking error	
Method	1 pixel	5 pixels
With no rain	304	343
With rain added	151	273
Median	192	346
Median+trans	249	367
Median+homog	215	363
Per-pixel	70	281

(c) Driving car

	Maximum tracking error	
Method	1 pixel	5 pixels
With rain	267	680
Median	684	980
Median+trans	676	982
Median+homog	720	977
Per-pixel	528	914

(d) Reflective sphere

	Maximum tracking error	
Method	1 pixel	5 pixels
With rain	349	659
Median	129	687
Median+trans	700	793
Median+homog	690	808
Per-pixel	636	795

(e) Snowy mailbox

	Maximum tracking error	
Method	1 pixel	5 pixels
With rain	131	394
Median	14	240
Median+trans	438	642
Median+homog	289	515
Per-pixel	183	453

(f) Pedestrians in the snow

Figure 3.13: Results for feature point tracking. For each method, the columns signify the number of points that are tracked within 1 and 5 pixels of accuracy. Points tracked within 1 pixel are completely correct, while those within 5 pixels have drifted a small amount. Rain and snow tend to slightly disrupt point tracks more often than causing them to be completely lost, therefore we show both 1-pixel and 5-pixel accuracy. Results are reported for median filtering with no alignment, with translational alignment, and with homography alignment. For sequences where ground truth is available, accuracies with no rain are also displayed. In the limit, both the patch-based and frequency-based techniques are identical to the median and have identical tracking accuracy, so are not included in the charts. Results for each method are explained in more detail in their respective sections, from 3.5.2 to 3.5.2.

or enhanced by studying their global properties.

Two possible future work directions are to improve the detection of the rain and snow, and to improve their removal.

Improved rain and snow detection

Most of the work in this chapter is based around developing a good model of rain and snow in frequency space. Although the algorithm is able to detect small, thin rain and snow streaks, it cannot detect large, defocused streaks. Correcting this would likely require modelling or learning the non-rain elements of the videos. This would make the problem more of a standard discrimination problem, where each pixel would have some membership in a rain/snow class as well as in one or more background classes.

Improved rain and snow removal

As we demonstrated in Section 3.4.2, once the pixels that contain rain or snow have been detected, using simple per-pixel temporal blurring can replace the corrupted pixels with a cleaner version. Another option is to do this replacement with texture synthesis [Efros and Leung, 1999] [Wei and Levoy, 2000] [Drori et al., 2003] [Liang et al., 2001]. This texture synthesis could be done either in image space or frequency space, although in general, texture synthesis methods appear to be most effective in image space. Simple temporal averaging has two specific problems, that could be overcome with texture synthesis.

First, the rain or snow will not be totally removed in the first iteration, and perhaps not even after many iterations. This is due to a soft probability on whether there is rain or snow at a given pixel. Texture synthesis could be used more discretely, as it could pick a reasonable guess, even if we assume that any corrupted pixel has no useful information. Doing this texture synthesis on individual images causes flickering artifact. But using temporal textures (or video textures)

[Szummer and Picard, 1996] [Schödl et al., 2000] [Kwatra et al., 2003] could effectively replace the bad pixels without causing additional flickering. Other options include a global smoothness constraint or a weighted average between texture synthesis and temporal smoothing. (Other options are discussed briefly in Section 3.2).

Second, errors may be less evident with texture synthesis. Fast moving edges are sometimes mis-detected as rain, as in the umbrellas in Figure 3.7 (d). If a texture synthesis method properly detects the properties of the textures in the images, it could even "replace" the mis-detections with their original values.

Chapter 4

Seeing through drops

Imagine you are driving at night and it is raining hard. The rain, the splashes, the drops on the windshield and the shiny wet road all cause significant degradation of scene visibility, making driving an unpleasant experience at best and a dangerous endeavor at worst.

Many systems based on visual, lidar or radar sensing on vehicles provide assistance to drivers, warning them of lane departure [Bertozzi and Broggi, 1998] [Dickmanns, 2002], impending rear or side collision with other vehicles [Mertz et al., 2000], road-signs [Lauzière et al., 2001], pedestrian proximity [Gavrila, 2001], &c. All these systems depend on good visibility and their performance degrades in bad weather conditions, precisely when the driver most needs help.

Based on data from the World Meteorological Organization, around the world, there are a surprisingly large number of rainy days. Table 4.1 shows some examples for different cities. Although in some months, in some cities, almost no rain falls, in many populous cities, half or more days can be rainy.

Recent work in computer vision proposes to digitally process images captured in bad weather and remove the effects of fog, haze, rain and snow, via physical modeling [Nayar and Narasimhan, 1999] [Garg and Nayar, 2004] [Barnum et al., 2010a], statistical processing [He et al., 2009] [Bossu et al., 2011], or modifying camera parameters [Garg and Nayar, 2005]. These methods

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
Asunción, Paraguay	8	7	7	8	7	7	4	5	6	8	8	8	83
Bangkok, Thailand	1	2	2	4	13	12	13	15	18	14	5	1	100
Berlin, Germany	10	8	9	8	9	10	8	8	8	8	10	11	107
Bratislava, Slovakia	14	12	13	11	11	12	11	11	10	10	14	15	144
Brussels, Belgium	13	10	13	11	11	11	10	9	10	11	13	13	135
Bucharest, Romania	6	6	6	7	6	6	7	6	5	5	6	6	72
Cape Town, South Africa	6	5	5	8	11	13	12	14	10	9	5	6	104
Caracas, Venezuela	6	4	3	7	13	19	19	18	15	15	13	10	142
Chicago, USA	7	6	9	10	9	8	8	8	8	7	8	8	96
Doha, Qatar	2	2	2	1	0	0	0	0	0	0	0	1	8
Kuala Lumpur, Malaysia	11	12	14	16	13	9	10	11	13	16	18	15	158
Mobile, USA	9	8	8	6	7	9	13	12	9	4	7	8	100
Paris, France	10	9	10	9	10	9	8	7	9	10	10	11	112
Pittsburgh, USA	10	10	12	11	10	10	9	8	8	8	10	11	117
Seattle, USA	16	13	14	11	8	6	3	5	7	10	15	16	124
San Juan, Puerto Rico	13	9	8	9	13	12	14	15	13	14	15	14	149
Tehran, Iran	6	5	6	5	4	1	1	0	0	3	3	6	40
Tokyo, Japan	5	6	10	10	10	12	10	8	11	9	6	4	101
Yuma, USA	1	1	1	1	0	0	1	2	1	1	1	2	12
Zanzibar, Tanzania	7	4	12	17	15	5	5	6	5	7	13	13	109
Zürich, Switzerland	11	10	11	12	12	12	11	12	8	8	10	10	127

Table 4.1: Mean number of days with at least 1mm of precipitation. The years being averaged vary, but most of these countries use about a recent 30 year period. (Data is from the World Meteorological Organization's World Weather Information Service).

Figure 4.1: At night, illuminated rain creates bright streaks. But as this paper shows, adaptive lighting controlled by high-speed drop detection and prediction can reduce the drops' visual effect.

can be useful as a pre-processing tool before other algorithms for driver-assistance are applied. The result of removing rain from images/videos can be shown on a display in the vehicle to the driver. But it would still split the attention of the driver between the display and the front windshield.

Instead of a display, can we somehow illuminate the scene, such that the visibility of rain is reduced for a human observer? Note that the rain becomes visible as it is illuminated by the various light sources, like the vehicle headlights, street lamps, billboard neon signs, etc. Thus, such a system would need to first identify where the rain drops are, then adaptively "turn off" rays of light that hit the drops.

There are significant hurdles to realizing such a system. For instance, the trivial solution would be to not illuminate at all. In this case, the drops are not illuminated, but neither is the scene. Therefore, the system should instead maximize the "light throughput" while minimizing

the number of light rays hitting the drops. Second, depending on the velocity and density of rain (a drizzle versus a thunderstorm), the system must be fast enough to be able to adaptively turn rays on and off so the light throughput remains high and there is no flicker visible to the human eye.

In this chapter, we seek a theoretical and practical understanding of how to design adaptive illumination for rain. To this end, this chapter is divided into two components.

First, we will show how to determine the maximum light throughput of generic adaptive projection systems. We want to emit as much light as possible into the scene, but we also want to avoid illuminating any drops. Using work from the meteorology literature, we can simulate rain with different characteristics, and examine the theoretical bounds on adaptive projection systems with different refresh rates and latencies.

Second, we will present a drop-generation testbed and prototype adaptive projection system. Although not fast enough to function effectively in real rain, we can use this system to determine how an adaptive projection system should interact with falling water drops. We show pictures and analysis of the successes and failures.

4.1 Overview of Active Sensing/Illumination Systems for Cars

There are a variety of systems that provide active sensing and/or illumination for cars. These include radar, lidar, and infrared lights, and cameras.

Automotive radar is most often used for adaptive cruise control, and has the possibility of being used for emergency braking as well. As discussed in [Wenger, 2005], short range radar could help prevent 88 percent of rear end collisions. [Grosch, 1995] discusses details on how radar systems can be helpful, even when used only to give early warning of collision. Using short and long range automotive radar can give even greater benefits [Wenger and Hahn, 2007] [Wenger, 2007] [Metz et al., 2001]. In this case, the different radar types use different wave-

lengths and fields of view, optimizing for applications ranging from cruise control to emergency braking. (Some technical details of such radars are discussed in [Li et al., 2004]).

Radar is especially useful for autonomous vehicles, and may be combined with lidar and visible-light cameras as well [Thrun et al., 2006] [Urmson et al., 2008]. Certain sensors are more able to detect and understand different types of obstacles, ranging from people, to other cars, to the curbs on the road.

Another sensing option that has possibilities for cars is lidar. As discussed by [Widmann et al., 2000] [Rasshofer and Gresser, 2005], there are advantages and disadvantages of each. Especially related to our work is that of [Shearman et al., 1998], which discusses how well such active sensing systems work with sprays of droplets.

Infrared is another common automotive safety sensing system that is beginning to appear on high-end cars. [Jones, March 2006] and [Austen, October 31, 2005] discuss how modern sensing systems are now being used on cars at night. Usually these systems are used merely to project the alternate spectrum on a screen or HUD, although some companies, such as BMW [Grundhoff, 2008], have begun to incorporate intelligent detection as well.

Other manufacturers such as Honda [Honda, September, 2004] and Oerlikon [Optics, May 23, 2007] have also developed such IR systems. Some such as Cadillac, have built such systems in the past [Vale, 2001], but may not continue to develop them. (Also discussed in [Vale, 2001] is how heavy rain and fog can reduce the effectiveness of such systems).

Water drops can also bead on the lenses of cameras and the windshield. Ways to reduce or negate these effects include comparing images across a stereo camera pair [Yamashita et al., 2005] or blocking parts of the lens with a programmable shutter [Hara et al., 2009].

4.2 Light throughput in rain

Our goal is to create a system that senses the current state of rain, then directs a projector to illuminate only rays that do not intersect a drop. To determine the resolution and speed requirements for this system, we first need to understand how natural water drops occur during rain.

The more raindrops in a scene, the more projector rays will intersect drops. If all of the raindrops are illuminated, they will all be visible. However, if we selectively turn off some rays, so the drops are not illuminated, they will be much less visible. The downside of this selective illumination is that as we are projecting fewer light rays, there is less overall light in the scene. This is compounded due to the finite speed and accuracy of any adaptive projection system. For example, if 1% of the projected rays intersected raindrops, any real system would have to turn off more than 1% of the rays.

For a finite camera or eye integration time T_i, the quality Q of a projection is defined as a temporal integral of the ratio between the total number of rays k and number of rays that are occluded o,

$$Q(t;T_i) = \int_{\tau=0}^{T_i} \frac{o(t+\tau)}{k} d\tau \tag{4.1}$$

The light throughput H of an active projection system depends on its accuracy a, for each refresh interval T_r,

$$H(t;T_r) = a\, Q(t;T_r) \tag{4.2}$$

To not illuminate any drop, the accuracy a would have to be greater than or equal to one. Also, one would expect that any system would be more accurate for some drop sizes and positions than others, although this should even out, due to the large numbers of drops in any rainstorm.

The two questions to answer now are, "What percent of rays does real rain occlude?" and

given that answer, "How does accuracy and speed effect light throughput?" We will answer these first by simulation.

4.3 A simulator for adaptive lighting in rain

Based on the discussion in the previous section and work in the meteorology literature, we can use a simulation with various parameters to compute the light throughput H, from Equation 4.2. We want to test how different types of systems perform under different levels of rainfall. A reasonable range for an adaptive projection system would be the ability to refresh the projected image between 60hz to 960hz. Most standard projectors can run at 60hz, and many of the faster ones, such as DLPs, switch between white and black in a few microseconds. Regarding the upper limit, although there are projectors that can switch images faster than 1khz, and there are signal processors that can process data at much higher rates, even fast moving raindrops will not change dramatically within a millisecond. (Although it would probably be useful to have an adaptive projection system running at 10khz, the benefit is liable to level off far before then).

As for the simulation of raindrops, we will test rainfall rates between 1-25 mm/hr. In nature, rain can vary between 0 mm/hr (no rain), to around 1000 mm/hr during the peak of a hurricane. But in a majority of the rainfalls in the temperate part of the world, the rain does not get significantly heavier than 25 mm/hr. In addition, we will see that trends related to increasing the rainfall become evident even with this range.

For the purposes of analyzing the results of the simulation, we will assume that every drop can be detected. But since real systems will have inaccurate localization, we also test different levels of error, by simulating blacking out additional pixels around those that contain raindrops. For example, if a rain streak is 2 pixels wide, and the system has 4 pixels of error, the simulation will black out a 6 pixel wide stripe. This means that the raindrop will still be dark, but the system will not emit as much light as it could.

(a) A distant scene (b) A close scene

Figure 4.2: Images taken at night, with a standard point-and-shoot camera (Canon Powershot S710). The flash will only illuminate objects that are close to it. The tree in (b) is close, and the waterdrops in both images are very close. It would be challenging to remove the images of the drops with any postprocessing, as so much information has been lost. But if the drops were never illuminated, then the images would appear as if the flash magically passed through the drops unhindered.

4.3.1 Meteorology theory

Appearance of a water drop: As discussed in [Garg and Nayar, 2005] [Van de Hulst, 1957], water drops act as wide-angle lenses, reflecting and refracting incident light in all directions. As discussed in some of the previous chapters and sections, a majority of the light shined on a drop will be refracted in a fairly straight path, although a smaller amount will be directly reflected back to the source, and an even smaller amount will experience one or more internal reflections. As a result, they are among the brightest objects in the environment. This is true during the day, when they are illuminated by the sun, and is doubly true at night, when light sources make nearby drops shine brightly. A primary reason that water drops are not dominantly visible at all times, is that their high speed relative to their size results in extreme motion blur. Raindrops are often not visible in images or video, even when the rain is falling very heavily. As shown by [Garg and Nayar, 2005], the spatial and temporal blurring, caused by a small depth of field or a long exposure time, can make water drops appear blurred into invisibility. Even normal camera

and video settings with standard depths of field and exposure times tend to make drops difficult to see. But with short shutter times, especially if a flash is used, then the drops become instantly visible. This is demonstrated in the flash images in Figure 4.2.

Dynamics of a water drop: Due to the distance from the clouds where they formed, rain drops fall at their terminal velocity. Because of their small size, wind resistance is a major factor, making different sized drops fall at noticeably different speeds. The terminal velocity of a waterdrop depends on the air around it and the diameter of the drop, although there are many factors that make a closed-form analysis difficult, such as how the drops shed vortices [Beard, 1976] [Best, 1950]. (The determination of terminal velocity becomes even more challenging with more difficult types of hydrometeors, such as snowflakes, graupel, and hail [Mitchell, 1996] [Langleben, 1954]).

But for our simulation, we do not require extremely high accuracy of the speed of each individual drop, therefore as with [Barnum et al., 2010a], we can use the size-dependent terminal velocity equation of [Foote and duToit, 1969], which states that a water drop of diameter d mm has a terminal velocity of:

$$-0.2 + 5.0d - 0.9d^2 + 0.1d^3 \tag{4.3}$$

For example, a 1 mm diameter drop falls at about 4 m/s. This is quite fast relative to the size of the drop. For comparison, a car would have to move at about 60,000 km/h to achieve the same size-relative motion blur. This means that even if waterdrop were not spherical, their exact shape would be blurred so much it would be close to impossible to determine how they would appear stationary.

Statistics of water drops: As discussed previously, rain intensity is usually measured in mm/hour, with common values between about 1-25 mm/hour. In rain of any intensity, there are drops of various sizes. There has been a great deal of discussion, debates, and experiments to determine

exactly how many drops of each size a given rainstorm should have. Over the long term, the number of drops of each size tends to follow an exponential, gamma, or log-normal distribution [Pruppacher and Klett, 1997]. Over the short term, the validity of using a drop size distribution at all is questionable. Most rain events seem to differ from standard Poisson processes, although there has been some effort to develop a more accurate model.

Even though we will be looking at short time periods, we need to have some understanding of how many drops of each size will appear. Since we are simply interested in overall statistics of the effectiveness of different adaptive projector systems, to determine reasonable numbers and sizes of drops for a given storm, we use an exponential (Marshall-Palmer) drop size distribution [Pruppacher and Klett, 1997], conditioned on rain intensity. An exponential curve is commonly used as a drop-size distribution [Marshall and Palmer, 1948], where for a rain intensity of r, we expect to find the following number of drops of diameter d:

$$N(d) = n_0 exp(-\Lambda d) \tag{4.4}$$

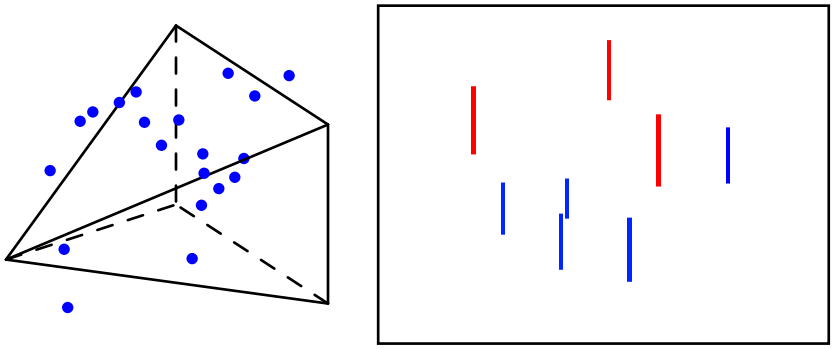
where the constant $n_0 = 8000$, and Λ is related to the rainfall rate r by

$$\Lambda = 4.1 r^{-0.21} \tag{4.5}$$

These are common quantities used in meteorology research, and are quite similar to the same quantities derived directly from the properties of a Poisson process. (Note that one must use the area under the curve to find the true number of each size).

4.3.2 Simulator implementation

A summary of the simulation is shown in Figure 4.3, where a projector emits a frustum of light. The system has three steps. The system will create new drops and simulate their falling, then



(a) (b)

Figure 4.3: Summary of the simulation. Drops are generated according to the distribution in Equation 4.4, and fall according to their terminal velocities (Equation 4.3). Subfigure (a) shows the drops in space, and subfigure (b) shows them in the projector coordinate frame. Streaks from drops that have been illuminated for more than 100ms are shown in red, and the others are shown in blue.

estimate which drops could be detected with a given latency, then determine the image that could be projected and project it.

Drop generation is performed via the Marshall-Palmer distribution (Equation 4.4) discussed earlier. The simulation will be given a rainfall rate (mm/hr) as a parameter. This rainfall rate effects the process in two ways. First, increasing the rainfall rate increase the proportion of drops that are large. (As an intuition, drizzles tend to have very small and few drops, while thunderstorms tend to have larger and a greater quantity of drops). The Marshall-Palmer distribution is used to determine the percentage of drops of each size that should appear. Next, for each time step the system continues to add drops in a greedy manner, until the amount of rain generated during the refresh time is equal to the rainfall rate. This is liable to cause rain that is more constant than in real life, never going greatly above or below the average rainfall rate. As discussed before, there is some question of whether this is appropriate. But since we are interested in the long term performance of our system, we believe it is fine in this case.

The next step of the simulation is to simulate the dynamics of each water drop, via Equation 4.3. This is straightforward. The state and velocity of each drop is known, and their new position

(a) (b)

Figure 4.4: (a) Real rain illuminated by a car headlight. (b) Simulated rain generated with similar parameters. The purpose of this example is not to demonstrate photorealism like [Garg and Nayar, 2006], but to show that our simulation parameters are a reasonable approximation of the statistics of real rain.

is updated.

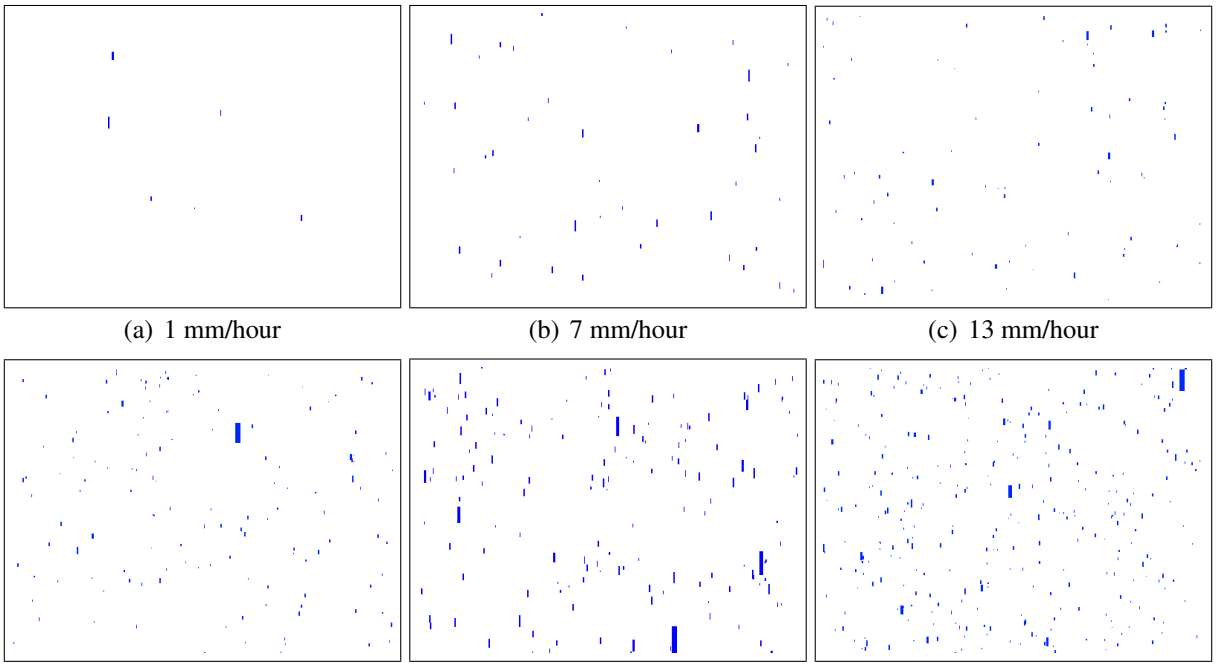
The interesting part is checking whether a drop has fallen into the projector's viewing frustum. We approximate the blurred shape of each drop in 3D as a plane parallel to the plane of the projector. In reality, the drop will carve out a volume similar to a cylinder with spherical caps. However, the computation complexity of doing a full 3D analysis does not appear to be neces-

sary, especially as a majority of the drops occupy less than a single pixel. Another variation from this simple model would be to simulate drop oscillations [Garg and Nayar, 2006]. However, as their effect on light throughput will be small, we simply assume that drops are exactly spherical. Checking if a drop has crossed into the projector field of view then becomes simple. We simply use the projector's projection matrix, and see if any or all of the drop's path during that refresh is illuminated.

If the drop is partially or totally illuminated, the simulation will begin "tracking" it, noting how long the drop has been track-able. (This process is important to determine how systems with different amounts of latencies will perform).

We simulate a latency between the time the camera captures an image, and the time we can project based on that information. For all drops that have been tracked longer than the simulated latency, the projector will project black for any projector rays that intersect the drop at any time during that refresh cycle. The locations to project black is computed by mapping the outer corners of the drop at the beginning and end of the refresh cycle, via the projector's projection matrix. Any projector pixel that intersects any drop will be black. In addition, we simulate uncertainty by expanding the blacked out areas by simple dilate on the final projected image.

We can create a camera image at any location, although this is not directly related to the projector throughput, though is useful to get an intuition of how the drops will appear at different angles. Computing the camera image has two steps. First, the system computes where the four corner points (in 3D) of the drop streak will appear in both the projector image and the camera image. Second, for each streak, the system will compute the homography between the projector and camera, based on the drop location, and copy the image information from the segment from the projector to the camera. Unlike a generic ray-based or Monte-Carlo technique, this method will be guaranteed to find the correct pixel brightness for all streaks, even the smallest. And it will do so without requiring an extremely large number of rays or samples. However, the process takes as long as the rest of the simulation put together, therefore it can often be skipped.



(a) 1 mm/hour (b) 7 mm/hour (c) 13 mm/hour

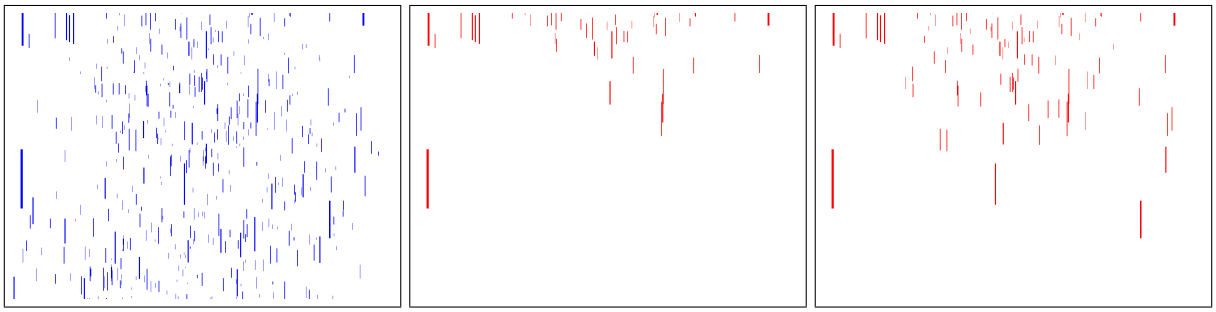
(d) 19 mm/hour (e) 25 mm/hour (f) 50 mm/hour

Figure 4.5: Simulated rain with different densities, with a 1/480 second exposure. Higher rainfall rate increases the relative number of large drops, as well as increasing the likelihood that some drops will be close to the projector. This means that images of higher rainfall rate will have more large streaks.

Some examples with different rainfall rates are shown in Figure 4.5, and light throughput statistics are shown in Figure 4.7. Figure 4.6 shows examples of how camera-projector latency affects detection. A more complete demonstration of which streaks might be missed by systems of different latencies is shown in Figure 4.15.

4.3.3 Discussion on simulation

Based on the simulated results from Figure 4.7, we see that there is significant light loss at lower refresh rates. The light throughput does not start to level off in heavy rain until about 400Hz. This appears to be true across different levels of localization error. Running at 400hz would be well above a human's ability to visually sense the flickering caused by the illumination. (We do not explicitly look at the amount of flicker that humans could tolerate and/or find comfortable,



(a) All drops (b) Drops not detected within 50ms (c) Drops not detected within 100ms

Figure 4.6: Simulated rain with different latencies, at 25 mm/hour and with a 1/60 second exposure. A majority of the drops are in the projector field-of-view for longer than 100ms, but many of the large (and therefore fast) drops are only illuminated momentarily.

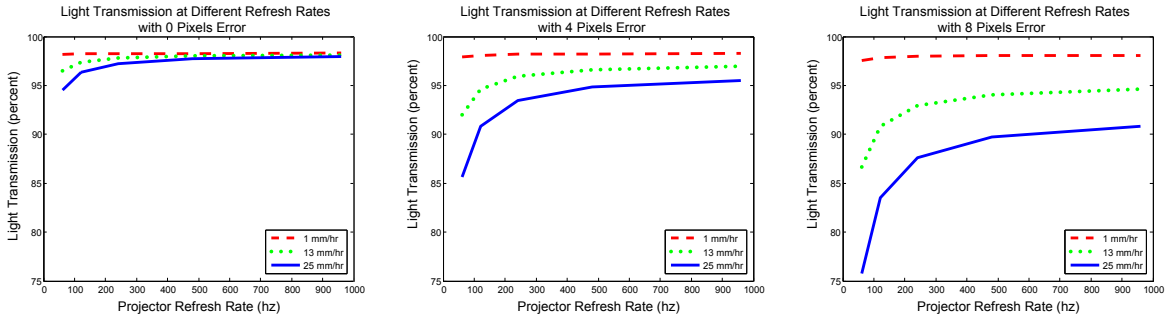


Figure 4.7: The goal of an adaptive projection system for rain is to maximize the amount of light emitted, while minimizing the amount of rain illuminated. When we turn off projector pixels or if the projector pixels illuminate rain, we can consider that light to be wasted. The three plots show light throughput with 0, 4, and 8 pixels of error. Light throughput increases as the projector speed increases, the amount of rain decreases, or the amount of error decreases.

although the light should switch at over 120hz, and perhaps faster than 200hz would be best).

In addition, from Figure 4.6 we see that even severe 100ms latency between the camera and projector does not affect most of the drops. However, the drops that are not detected in time tend to be the largest and closest ones. Human testing would be required to determine which latencies are acceptable, although from our rough analysis, it appears that 10ms or less is required.

An alternate to using a low-latency system is using a light source that people can't see, but the camera can. For example, if we user infrared light, then the drops could be detected and tracked well before they are illuminated by the projector. The main disadvantages of this is that a) the

infrared light would have to be dim enough that it would be safe to shine in the environment, and b) the further we track the drops from where they will be illuminated by the projector, the more error we are liable to get.

But if it were possible to create a system with 1ms latency, then we could use a completely different technique. Instead of tracking and predicting the locations of the drops in the future, we could regularly flash incredibly quick light pulses, to illuminate the drops for the camera to see. Once seen and quickly processed, the car headlights could be directed to miss them in the next 1ms, and so on. Only a very small amount of each streak would be visible, and the system would not have problems with accumulating error over time.

In either case, the next step is to begin validating these theoretical and simulated results with a prototype system. We have tested two different types of adaptive projection testbeds, discussed in the next section. The first uses a single camera and laser plane to determine the locations of the drops. This system is a first step in determining how adaptive projection systems could be feasible, although it requires a great deal of user interaction to function correctly. The second uses a stereo-camera and projector rig, which requires very little user intervention (only for the initial calibration step.) We have created an adaptive projection testbed, discussed in the next section.

4.4 What can we test with real hardware?

As discussed in Section 4.3.3 a useful adaptation projection system would require a projector running at about 400 Hz, with detection-projection latency under 10 ms. Creating such a system would require extensive specialized hardware. However, we can still collect useful data from a slower projector-camera system. The main interesting questions that we can answer are:

1. **How difficult is it to detect and localize drops?** There could be problems if the drops are too dark to see, are moving too fast, or cannot be matched correctly either with a single

camera or a stereo camera. An important first step is to determine what combinations of light and cameras are necessary and useful. Also, we can begin to answer questions about camera parameters and how changing things like exposure time effect the ability of the system to find drops.

2. **What are the best strategies for detecting and localizing drops?** Similar to the question of the difficulty of detecting and localizing drops, we also need to examine different possibilities for detection. Is it better to use a technique such a single camera optically co-located with a projector? Or is a stereo camera rig better or required?

3. **How do drops appear when the projector displays black versus white?** This is a fundamental question, as the key is to make the water drops appear less bright. If any adaptive projection system fails to make the drops significantly less bright compared to the illuminated background, then it is not adequate. This is largely a question of contrast. Certain light projection systems have better contrast ratios than others. For example, LCDs tend to be bad, while DLPs tend to be better. Systems like lasers could be the best of all, although such systems are generally quite expensive (although there is a push for consumer-priced versions, that may appear over the next few years.)

We will explore two methods to solve this problem. First, we will test using a single camera and a laser plane. This will require manual intervention to label some drop positions, although has the advantage of simplicity and speed. But for a more realistic test, we will also perform experiments with a stereo-camera and project rig, that is able to detect and predict the locations of drop autonomously.

4.5 Adaptive projection with a camera and laser plane

In this section, we focus on algorithms and apparatus design appropriate to adaptive projection in an outdoor environment, with some user intervention. To display the correct image with the

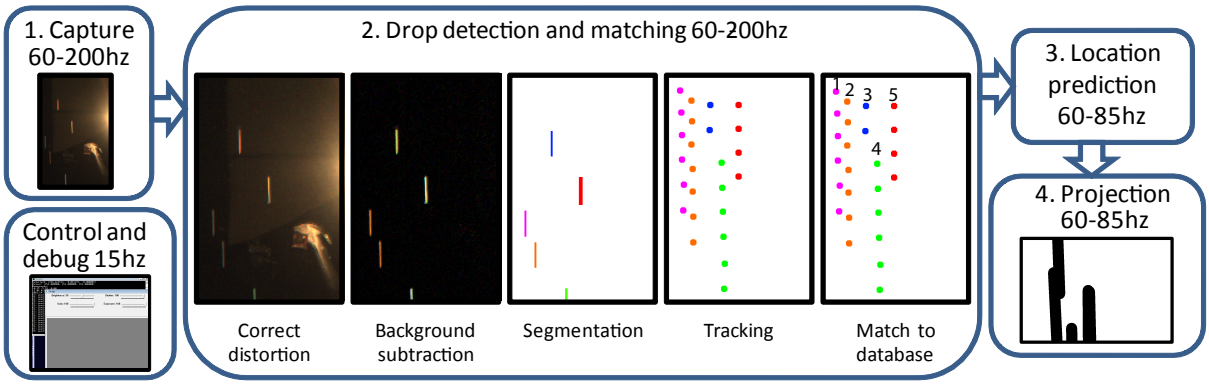


Figure 4.8: Control loop of the real-time tracking, prediction, and projection system. Each of the five bubbles represents an asynchronous loop, running at the given speed. Training ends after the fourth step of "Drop detection and tracking", while display uses all steps. For training, frames are captured at 60-200hz, then drops are segmented and tracked, and a database is created offline. Testing takes the tracks and matches them to the database to determine their future locations, then directs the projector to illuminate them. Control and debug displays recent image captures and sliders for background subtraction thresholds.

projector, we must detect and compute the trajectory of each drop before it is illuminated by the projector. We demonstrate an algorithm that tracks and predicts the future location of hundreds of water drops per second, using one of the laser planes shown in Figure 1.2. Our depth and velocity measuring technique bears similarity to optical distrometers [Schönhuber et al., 1994] [Löffler-Mang and Joss, 2000] [Schönhuber et al., 2007] [Wang et al., 1980] [Wang et al., 1977] [Thurai et al., 2009], but can handle dynamics greater than first-order polynomial. In practice, raindrops in a natural environment can be described fairly well be a first-order equation. However, in cases of strong wind (or for drops being generated indoors, as in this case,) a second-order polynomial model is more appropriate and correct.

Streak detection and tracking We will first try to make the system run as fast as possible, using human interaction when necessary. We will use a Point Grey Firefly MV camera, that can run at up to 200hz with reduced resolution and without synchronization.

Since the camera captures image at up to 200FPS, the entire process of segmentation and

tracking must be completed within .005 seconds, or images will be skipped and tracks lost. The main bottleneck on modern hardware is memory, so each stage outlined in Step 2 of Figure 4.8 is designed to read each image pixel in order and as few times as possible.

The first step is to correct for radial and tangential distortion. A single pixel of image error will lead to several pixels of the projector error, and could cause the drop to be missed entirely. Based on a four-parameter distortion model [Fryer and Brown, 1986], we determine the correction with the Camera Calibration Toolbox for Matlab and OpenCV, then compute an integer-precision lookup table between distorted and undistorted pixels. This nearest-neighbor correction is very quick, although it does have issue with aliasing. The end result of this is that if a drop is aliased, the system will detect it as being slightly larger than it actually is. In practice, this is not a great issue, although it does effect the accuracy of the prediction somewhat.

The background subtraction model is a per-pixel median image that is trained over several hundred frames. Streaks candidates are detected as pixel values above an image global threshold. Pixel noise is suppressed by Gaussian blurring and thresholding. This technique is liable to not be sufficiently accurate for unconstrained outdoor environments, but has the advantage of speed. As a result, we can examine how well a high speed adaptive projection system will function, without worrying about robustness. With a combination of smarter machine learning-based algorithms, and embedded hardware, much more sophisticated background subtraction could be performed.

We perform approximate connected components [Muerle and Allen, 1968] with a simplified version of raster-line based region growing [Bruce et al., 2000] [Bailey, 1991]. Groups of consecutive pixels in each row are added found. One linked list for each row holds the start and end points of each raster-line component. The lists are scanned in order, and each single-row group is assigned the label of its first neighbor on the previous row. The final output is a rectangular bounding box for each component. An image of the letter U would be split into two vertical lines with this method, but since streaks are mostly convex, no additional processing is required. For a common 752x480 image of streaks, connected components are found within .001 seconds on

a single 3.2Ghz Xeon core.

Once the images are segmented, tracking involves matching streak endpoints across frames. The camera shutter is left open, so the end of a streak in one frame will be in the same location as its beginning in the next. If a streak has no neighbor in the previous frame, a new track is started. A majority of streaks can be tracked for most of the image, but tracks can fail for several reasons. The most common is that the background subtraction mis-labels too much of the streak as background. The connected component segmentations of the streak will not overlap in both frames, causing the drop to be split in two tracks. A less common error happens if two streaks are so close that they are segmented as a single component. Only one of them will be tracked through the merging. The other will appear as two distinct tracks, one above and one below the merging. Having split tracks is not generally a problem, since later steps in the algorithm will simply treat them as two drops in approximately the same space-time position.

Drop dynamics calibration In order to determine which projector pixels to illuminate, we must predict each drop's image position (x, y) at a future time t. Real raindrops have reached a terminal velocity between 1-9 m/s [Pruppacher and Klett, 1997], so position can be predicted by first order model. The drops from a drop generator fall slower, but they have more complex dynamics.

Drops are released from each emitter at slightly different initial velocities, then fall at a rate determined by gravity, air density, and their diameters. The drops fall mostly straight down, from high to low y, and with a small change in x. Since the initial velocity varies randomly, we cannot predict x position for a new drop based on a previous drop. The x position is therefore fit for each new drop, as discussed in the next section. However, the y position is determined by consistent factors, therefore it can be predicted with a model of drop dynamics.

Standard mechanics state that the position of a body in free-fall in vacuum is governed by a second-order polynomial in time. Air of a given density decreases the acceleration relative to the

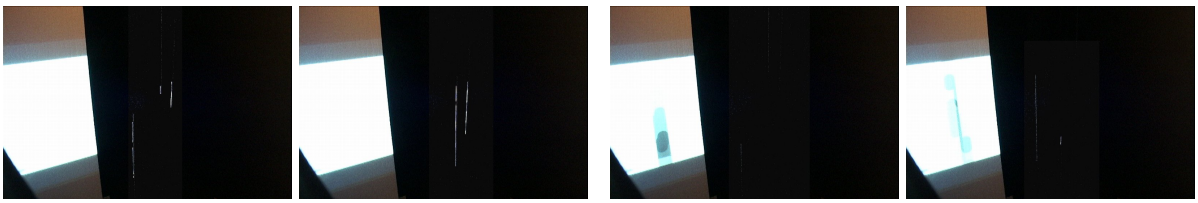
body's shape and relative movement. Because of their complex, fluid shape, the drag on a drop is complex [Beard, 1976] [Pruppacher and Klett, 1997].

But although the exact properties of the drag are complex, it is normally consistent between different drops from the same emitter. All drops are close to the same size and follow close to the same path, therefore they have close to the same acceleration and velocity both in space and in the image. Drop-dynamics calibration therefore involves observing the drops for a few seconds, then fitting second-order polynomial models between image locations and times. Since drops at different depths will have different dynamics in the image, a separate curve is fit for each drop generator. A second order model would be inaccurate in extrapolating the position of a drop. Because the acceleration is decreasing, it would overestimate the translation. However, a second-order model is accurate for interpolating between known locations, and is robust to noisy data.

We still require drops emitters to have unique coordinates in the camera image. As with Section 1.3, the user labels the intersection of each emitter with the laser plane. The dynamics model is then trained for each emitter, based on streak locations. The top of each streak is the location of the drop at the beginning of the camera's exposure, and likewise for the bottom. (Since the bottom of one streak becomes the top in the next frame, each streak only gives one data point). Since the first and last track will begin or end outside the image, a drop tracked for $N+1$ frames will have N triplets, $(x_0, y_0, t_0), (x_1, y_1, t_1), \ldots, (x_{N-1}, y_{N-1}, t_{N-1})$, where $\frac{1}{t_i - t_{i-1}}$ equals the camera capture frequency.

We train on the longest track passing through each laser-intersection sparkle. We subtract the starting time from all pairs, obtaining a sequence where t is in the range $[0, t_{N-1} - t_0]$. Using the N scaled (y, t) pairs, we use robust linear least squares to learn a model of y at time t as a polynomial with three coefficients a:

$$y(t) = a_0 + a_1 t + a_2 t^2 \tag{4.6}$$



(a) (b)

Figure 4.9: Selectively missing streaks. A screen is placed behind the drops, so the image projected can be viewed on the left and the effect on the drops on the right. Due to the low dynamic range of the camera, the streaks in the right half of each image have been brightened. (a) When the projector displays a white image, the streaks are visible. (b) But by tracking and selectively displaying black, the drops are not illuminated, so streaks are not visible.

Adaptive projector control Matching a new query drop to the model is performed in a similar way to Section 1.3. A query drop is matched to its nearest neighbor in the database by finding the closest laser crossing. The future x position is predicted with a linear fit. The future y location is determined by fitting to the trained Equation 4.6.

First, the inverse of Equation 4.6 is used to find the value of t that corresponds to each of the query drop's y values, giving a set of (y_i, t'_i) pairs. If the query drop is falling at exactly the same rate as the trained model and estimation is perfect, then for a given camera frame rate r

$$\forall i(t'_0 = t'_i - ir) \tag{4.7}$$

In practice, the estimates vary, so with t^*_0 being the mean of all t'_0 estimates, the predicted location of the drop is

$$y(t) = a_0 + a_1(t - t_0 + t^*_0) + a_2(t - t_0 + t^*_0)^2 \tag{4.8}$$

Once the query drop is fit to the correct database drop, projector location prediction only requires the calibration from Section 1.2. For a given projector refresh frequency f, the drop will trace a line of image pixels over time $[t, t + 1/f]$. The line in the image is then warped to the projector reference frame. The end result is that the projector will illuminate the drop when it

passes in front.

Results for real-time tracking and prediction Figure 4.9 shows an example demonstrating real time detection and projector control. We placed a screen behind the drops, so it is possible to see both the streaks and the image that was projected at that time. When the projector outputs a white image as in Figure 4.9 (a), the streaks can be seen. But since we know where the drop will be, we can project a black line, effectively "missing" the drop (Figure 4.9 (b)). Because we track and predict drops in real time, we can create a display with less restriction on drop generator geometry, but in practice, it is limited to only a few drops at a time.

4.6 Adaptive projection with stereo cameras

4.6.1 Detection

As discussed in [Garg and Nayar, 2005], a water drop's appearance varies dramatically, based on the lighting in its environment and the camera settings. Naturally, the drops will appear brightest when there is a lot of light in the environment. However, drops will also appear bright in other, more surprising cases. For example, if the camera has a short exposure time or a large depth of field, the drops will be very easy to see. In addition, if there is light shining on the drops from any angle relative to the observer, then the drops will appear bright. (This is in contrast to standard Lambertian surfaces, where there is little light visible when the light shines at an extreme angle, and without subsurface scattering, most light behind an object will not noticeably affect its brightness, except regarding global illumination).

Most of the research has only considered the cases of water drops being imaged by cameras. When cameras are being considered, a lot of the properties of how well the drops can be seen depends on the properties of the camera itself. But in this section, we do not just want to consider how to reduce (or increase) the effect of water drops on camera images. In fact, to do the

best projector control, we actually want to have the cameras be able to see the drops as well as possible. But we will also arrange the system in such a way that when we have correctly controlled the light, people will be unable to see the drops. In short, we want to minimize the visual effect of rain for a human observer, but we want the detecting cameras (or generic imaging sensors) to see them as well as possible. To allow the cameras to be able to see the drops most easily in arbitrary lighting environments, the two most important parameters to modify are the camera's exposure time, and its angle relative to the light source.

Since longer exposure times lead to increased motion blur and comparatively darker drops, we want the camera to have a very short exposure time. The exposure time should be short, although not as short as possible. There are several things to consider when selecting the best shutter speed for the camera:

1. **The brightness of the drop relative to the background:** In scenes with very fast moving objects, such as those with falling rain, drops have unusual brightness characteristics for different shutter speeds. In a static scene, reducing the shutter time by half will reduce the brightness of all objects by half. Likewise, and increase in shutter time will cause a linear increase in the amount of light incident on each pixel of the sensor. However, fast moving objects cause a different effect. Although the total amount of light on the sensor will vary approximately linearly, the amount on each pixel will not. Consider an object that moves three pixels during the exposure. It gives one unit of brightness to each of the three pixels. If we double the exposure, it will cross six pixels, but still only give one unit of brightness to each pixel. The fast motion increases the number of pixels, but does not increase the brightness in any individual pixel. However, every other object in the environment will contribute linearly based on the exposure time. So in the short exposure case, each pixel gets one unit of brightness from the drop, and three from the background, or a ratio of 3:1. In the longer exposure case, the ratio becomes 6:1. We see that as we increase the exposure, it will become more and more difficult to distinguish the drops from

the background. Therefore, to best distinguish individual rain pixels from the background, we want the ratio to be as small as possible, 1:1, which means a very short exposure time.

2. **The total brightness of the drop:** As the exposure time becomes shorter, the drops will have less brightness relative to the background. But at some point, some of the slower moving drops will only occupy one pixel, and then will shift to the standard linear relationship between shutter time and brightness. Given camera noise, the ability to detect them will again decrease. Therefore, we need to pick an exposure time that works best for the most drops, which will be conditioned on the speed of the drops and the camera parameters.

3. **The speed of the drops:** Larger drops will move faster, and thus will benefit from a shorter shutter speed. Also, larger drops will be more visible, as they can have more light incident on them, and therefore refract and reflect more. It may be a valid strategy to pick an exposure time that best detects the larger drops, and does not detect the smaller ones as well, as they are less important. To verify this either way, it would be necessary to do human perceptual testing.

4. **The distance of the drops from the camera, relative to the cameras field-of-view:** Similar to the last point, drops that are further away will appear to move slower. But also, they will receive less light, and probably be less important to detect. Selecting the correct shutter speed should likely be biased to the drops closer to the light source (and presumably, the viewer.)

5. **The ability to predict velocity:** A more subtle point is the ability to predict velocity. With a longer shutter speed, individual drops will appear as longer streaks, and it becomes easier to tell which direction they are heading, and how fast. Reducing the shutter speed makes the drops brighter relative to their backgrounds, but also makes the velocity estimation more difficult. In addition, matching individual drops across frames becomes more chal-

lenging, as the beginnings and ends of streaks are father apart, and it becomes easier to mis-match streaks from frame to frame.

Regarding camera angle relative to the light, there is a complex relationship between the angle of the light source and the camera. The short explanation is the drop is brightest when viewed directly opposite the light source. Although the drop is brightest here, this is not a good position for the camera, as shining a light directly at a camera will cause vision bleach (which has the potential to cause problems with even the most advanced robots). As the angle increases, the drop become less bright. But on the other side, the reflected light is brightest exactly at the light source. Naturally, the camera cannot be placed directly in the path of the light, but in general, the less the angle between the two, the brighter the drop. This means that the cameras should be placed as close to the optical axis of the projector as possible. Ideally, a beam splitter could be used to co-locate the camera and projector, which would also make localization of the drop in the projector coordinate frame trivial. But in practice, beam splitters reflect and scatter more light than water drops, making co-location infeasible. It may be possible to create overlapping optical elements, with small emitting components next to small sensing components. An example of this would be an LED array where every space between LEDs is a photosensor.

As a final note, we cannot reliably see drops beyond a certain distance. Drops are relatively bright, but they are also often quite small relative to a CCD pixel. This means that with a 1100 lumen projector, camera noise becomes an issue beyond 50cm. A real-world system would require low-noise imaging sensors. But for our experiments in this paper, we will simply place the drops close enough that they are easy to detect.

4.6.2 Localization

Since we cannot co-locate the camera and projector, we use two cameras to perform a 3D reconstruction of the drop positions. Although there can be thousands of drops per cubic meter, they are small and sparse enough that stereo matching across epipolar lines is possible. The ability

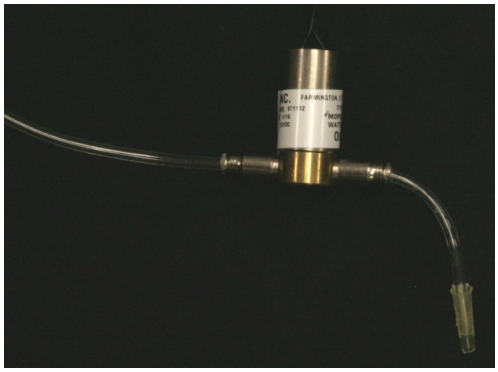
to match is additionally simplified by running the cameras as fast as possible. Many ambiguities will only occur momentarily, and taking more samples increases the probability of being able to find the correct location.

4.7 Drop generation hardware

To test our adaptive projection system, we have created a drop-generation testbed. Real rain is chaotic, making it difficult to obtain repeatable cases to test where our camera-projector system succeeds and fails. Therefore for this paper, we created a drop-on-demand setup, that lets us repeatedly test known configurations of drops. We use a similar setup to [Eitoku et al., 2006b] [Barnum et al., 2010b], where solenoid valves control drop formation. As shown in Figure 4.10 (a), each valve is connected to a single emitter. Each valve receives slightly pressurized water from an elevated tank, with 0.7 m of head. We use KIP 07420904 direct-acting miniature solenoids, with a maximum flow rate of 0.03 cubic meters per minute. This system is simpler than real rain in that the drops are larger and not falling as fast. However, it is more difficult in that the drops have not reached terminal velocity, making prediction of their future position more problematic.

4.8 Camera and projector hardware

Figure 4.10 (b) shows a photo of the stereo-camera and projector rig. The cameras are color Point Grey Firefly MVs at 300x752 resolution (this is not full resolution, which allows the cameras to run synchronized at 60Hz). The projector is an InFocus LP120 DLP, which has a native resolution of 1024x768 and 1100 lumens. This projector has its color wheel removed, making it grayscale. All three are temporally synchronized at 60Hz, and geometrically calibrated.



(a) Drop generation hardware

(b) Stereo cameras and projector

Figure 4.10: Photos of the hardware. (a) We use multiple KIP 07420904 direct-acting miniature solenoids, one for each drop. These valves predictable and repeatably create drops for testing our camera-projector hardware and algorithm. (b) The cameras are Point Grey Firefly MVs, and the projector is an InFocus LP120. All three are geometrically calibrated and temporally synchronized.

4.9 Algorithm implementation

The drops are detected and tracked in a similar way to the technique discussed in Section 4.6. The primary improvements are:

1. We use stereo cameras, instead of a single camera and laser plane, which allows for unsupervised detection of drops.
2. Our tracking algorithm handles missing data. Due to their shape oscillating, drops sometimes momentarily do not refract light toward a camera, and seem to disappear. Therefore, when a drop is not seen for a frame in one camera, we use the nearest neighbor in the other camera, and guess the missing position based on the predicted 3D location.
3. Calibration is more principled. We use a full geometric calibration of the cameras and projector, instead of requiring images of multiple planes that are approximately parallel to the image plane.
4. We do not require additional, external lighting to illuminate the drops for detection.
5. All cameras and projectors are synchronized, reducing temporal uncertainty.

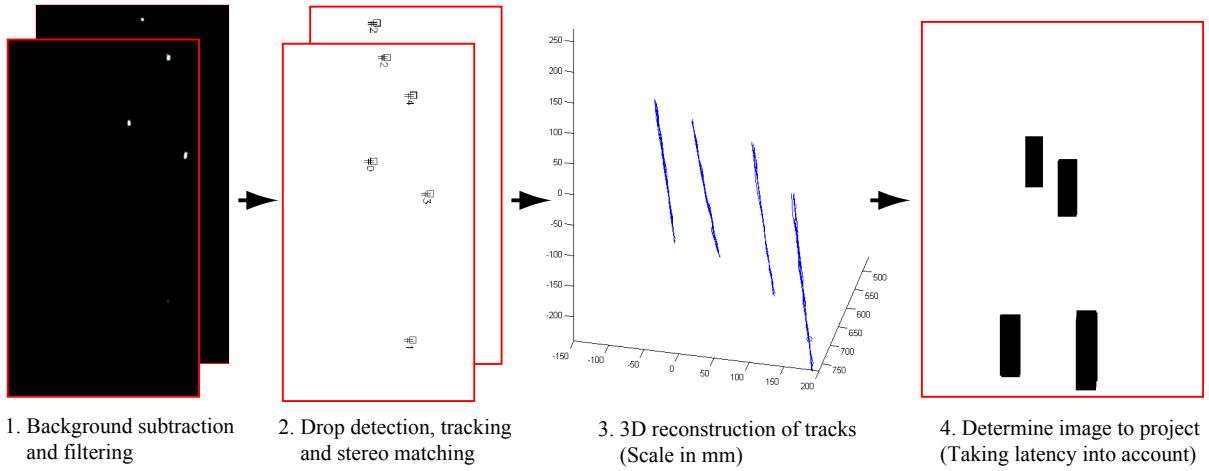


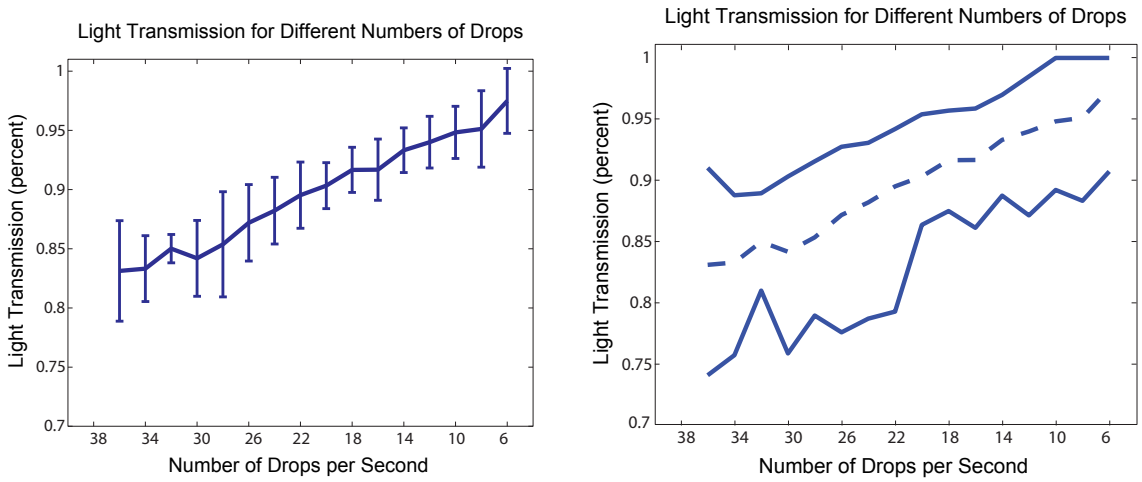
Figure 4.11: Intermediate steps for the drop detection algorithm. We used four drop emitters, each emitting at 2Hz. (Subfigures (a) and (b) are composited from two images each).

6. We use a DLP projector instead of an LCD projector, which has a faster switching speed.

We are interested in detecting and predicting the future locations of drops as fast as possible. Therefore, we have developed a software method that handles data asynchronously, is well-parallelized, and accesses memory in-order as much as possible.

Drop detection: For each of the two stereo cameras, we background subtract, blur, and threshold to find the bright water spots. These spots are segmented with a fast connected components method based on [Barnum et al., 2009a]. Spots in each camera are matched across epipolar lines between the two cameras, and the 3D position of each drop is computed. Figure 4.11 illustrates this process.

Drop prediction: Unlike real rain, the drops generated from our testbed are still accelerating, therefore we cannot predict their future location without many samples. To make the detection and prediction algorithm act more like it should in real rain, we first track several drops as they fall, and fit a second-order model to their position over time. The drop positions over time are clustered with mean-shift, and new drops are then matched to their nearest-neighbor cluster. This means that as with real rain, we only require one detection with known streak length, or two



(a) Mean and variance

(b) Min, mean, and max

Figure 4.12: Quantitative adaptive projection results. (a) The mean and inter-frame variance for light transmission percentage. (b) The min, mean, and max light transmission percentage.

detections with only the starting position of the drop.

4.10 Results

The effect of missing drops is shown in Figures 4.14. Each of the nine image pairs shows how the drops will look both when the light is on, and when the adaptive projection system is running. (We are able to have the same state twice, as we can direct our adaptive projection testbed to emit the same sequence of drops multiple times).

As can be seen from these images, the visual effect of water drops is reduced when the corresponding projector pixels are turned off. The drops are still dimly visible, due the global illumination caused by the projector, and by the finite contrast of the projector. Our system finds most of the drops with this simple testbed, except for a few cases where the drop emitters create multiple small drops, or there are mismatches due to the incorrect stereo correspondence.

Quantitative results are shown in Figure 4.12, for between 6 to 36 drops per second (two drops per second from 3 to 18 emitters.) Figure 4.12 (a) is the mean transmission, and error bars

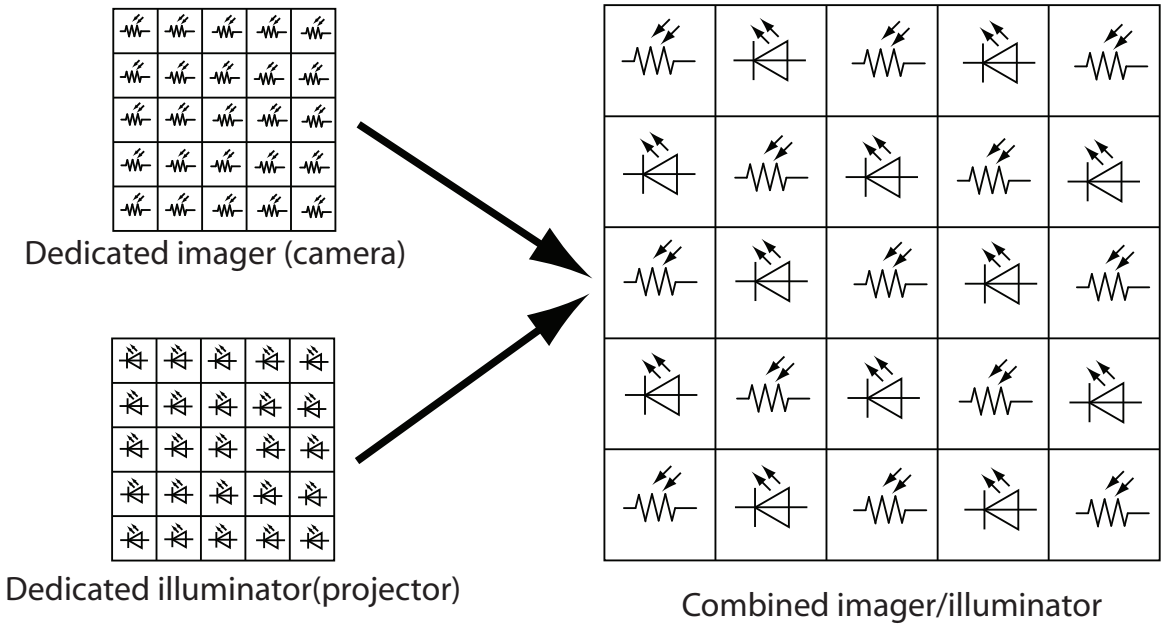


Figure 4.13: Illustration showing the concept of a combined camera projector. Depending on how easy it is to detect drops, there might be a different ratio between the number and positions of the sensors relative to the lights.

for one standard deviation. (The standard deviation is computed by computing the transmission percentage for each frame, and finding the variance between frames). Figure 4.12 (b) shows the min, mean, and max transmission percentages.

4.11 Conclusion and future directions

Faster hardware and software

Our adaptive projection system functions indoors, correctly blacking out drops at 60hz. However, real raindrops would have to be detected and predicted much faster and more accurately, for the system to be useful on a real car. There are two possible directions to take. First, the light source could be controlled much faster. A 1mm drop will fall about 4mm in 1ms. If we can detect and adjust the lighting within 1ms, we could use a simpler model to simply detect, rather than to

predict. This simple control could be little more than a photosensor and an LED, hooked up to a few transistors.

If the sensors and illumination components could be small enough, placed closely enough, and operate quickly enough, a nearest neighbor approach might be sufficient to determine which lights to turn on. But for larger and slower components, it would probably be best to use a compressive sensing/compressive illumination setup, as shown in Figure 4.13. The pattern shown is similar to a Bayer array for a camera, which creates an NxMx3 picture out of NxM samples. It might be possible to use an even fewer number of sensors relative to illuminators, to increase the brightness. (Some interesting related works are [Wyatt Jr. et al., 1992], [Romberg, 2008], [Takhar et al., 2006], [Tropp et al., 2006], and [Gupta et al., 2010]). This simple circuit grid could be attached to a DSP or FPGA, to allow for the data to be processed with low latency and high data throughput.

Alternately, it may be possible to use off-the-shelf hardware to good effect. We are currently using a 60hz projector and two 60hz cameras. Cameras such as the Point Grey Grasshopper can capture images in real time at 200hz full frame, and even faster for partial frames. Many projectors now run at 120hz (to allow for 3D viewing). In addition, there are faster projectors, such as the Pico 2 and the LightCommander that can display binary patterns at 1440hz. Even faster projectors are based around the DLP Discovery series, and allow for control of DMD chips faster than 30kHz. The main issues with such hardware would be acquiring, processing, and transmitting information fast enough and with sufficiently small latency.

Statistical modelling

Second, although individual rain streaks can be missed by tracking them and predicting their future location, this may not scale to heavy storms. It might be possible to develop a statistical theory to reduce the effect of lighting on water drops. Traditional thinking was that over short time periods, rain is a completely random process. However, researchers such as Jameson and

Kostinski [Jameson and Kostinski, 2001] [Jameson and Kostinski, 2002] have proposed that rain is not completely a Poisson process. As they put it, there are three requirements for rain to be a Poisson process:

- The probability of detecting more than one drop in a given volume δV is vanishingly small for sufficiently small δV,
- Drop counts in non-overlapping volumes are statistically independent random variables (at any length scale)
- The process is statistically homogeneous.

According to Jameson and Kostinski, the first requirement is usually satisfied in real rain, but the second is not always satisfied.

Other potentially useful work includes that of Desaulniers-Soucy *et al*. [Desaulniers-Soucy, 1999] [Desaulniers-Soucy et al., 2001], who developed an optical distrometer, and discuss how rain can be characterized by a multifractal. There is also work reported in [Onof et al., 2000] [Uijenhoet et al., 1999], where rain can be successfully modeled as a Poisson cluster process, although the generalizability of such approaches is not fully agreed upon [Desaulniers-Soucy, 1999]. Others measurements and models, such as [Hosking, 1987] [Lovejoy et al., 2003] [Smith and Veaux, 1994] [Kostinski and Jameson, 2000] [Lavergnat and Golé, 1998] may also prove useful.

If the future positions of raindrops can be predicted by a statistical model, then it may be possible to reduce the effect of lighting on rain without modeling individual drops. This could allow for a headlight that would improve visibility in even the most heavy rain, allowing for drivers to view the scene effortlessly.

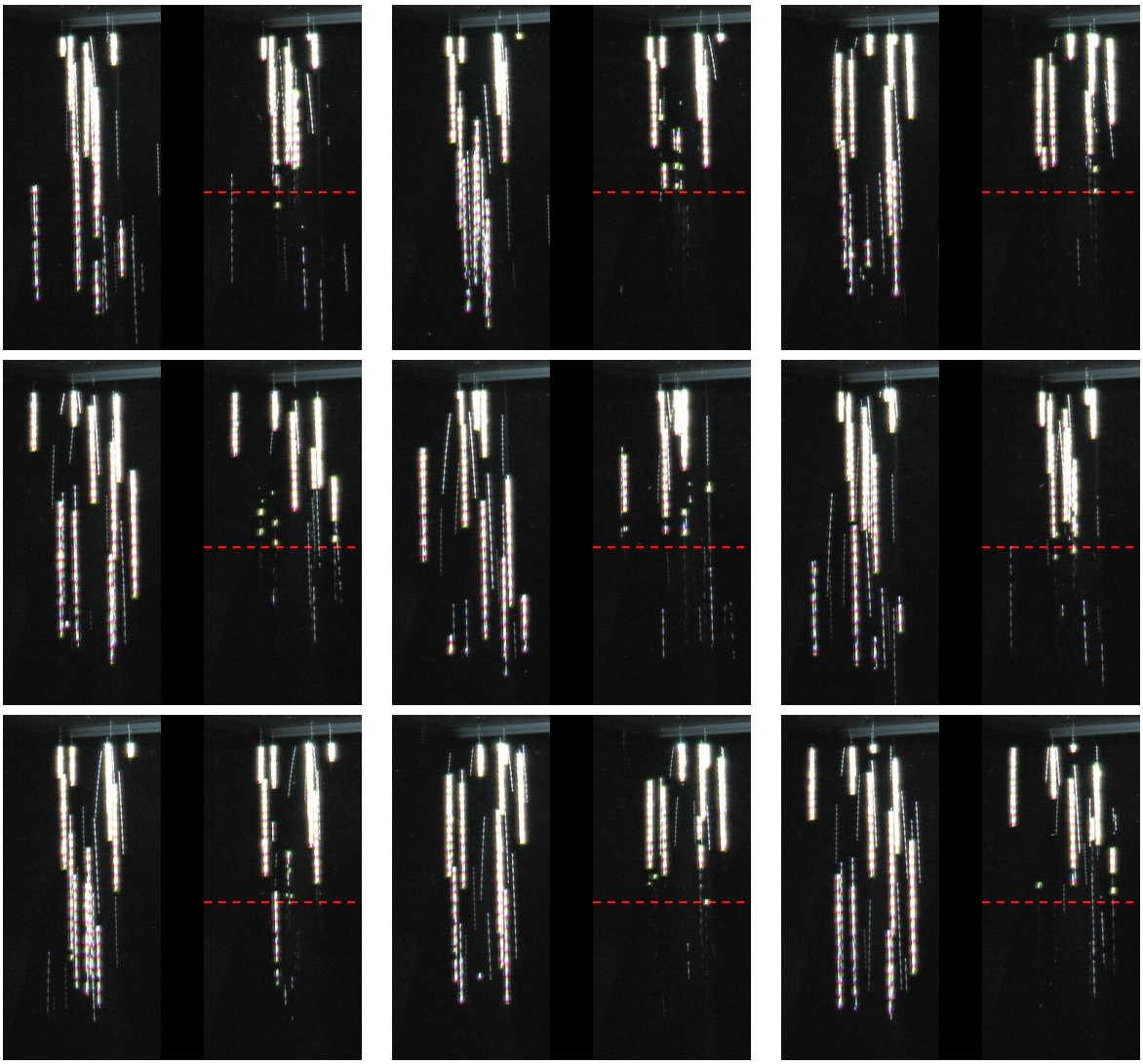


Figure 4.14: These images are taken at two different times, first with the projector displaying white, then with the projector detecting, tracking, and blacking-out the drops. (To show long streaks, both images are composites of multiple images, and the speckled appearance of the eight streaks is due to the DLP projector's dithering). For (b), the tops of the streaks are white, as the system has about 70ms latency between the camera and projector. But once detected and tracked, and the projector displays black, the drops are much dimmer. They are still visible, as the projector causes global illumination effects, which the drops reflect and refract.

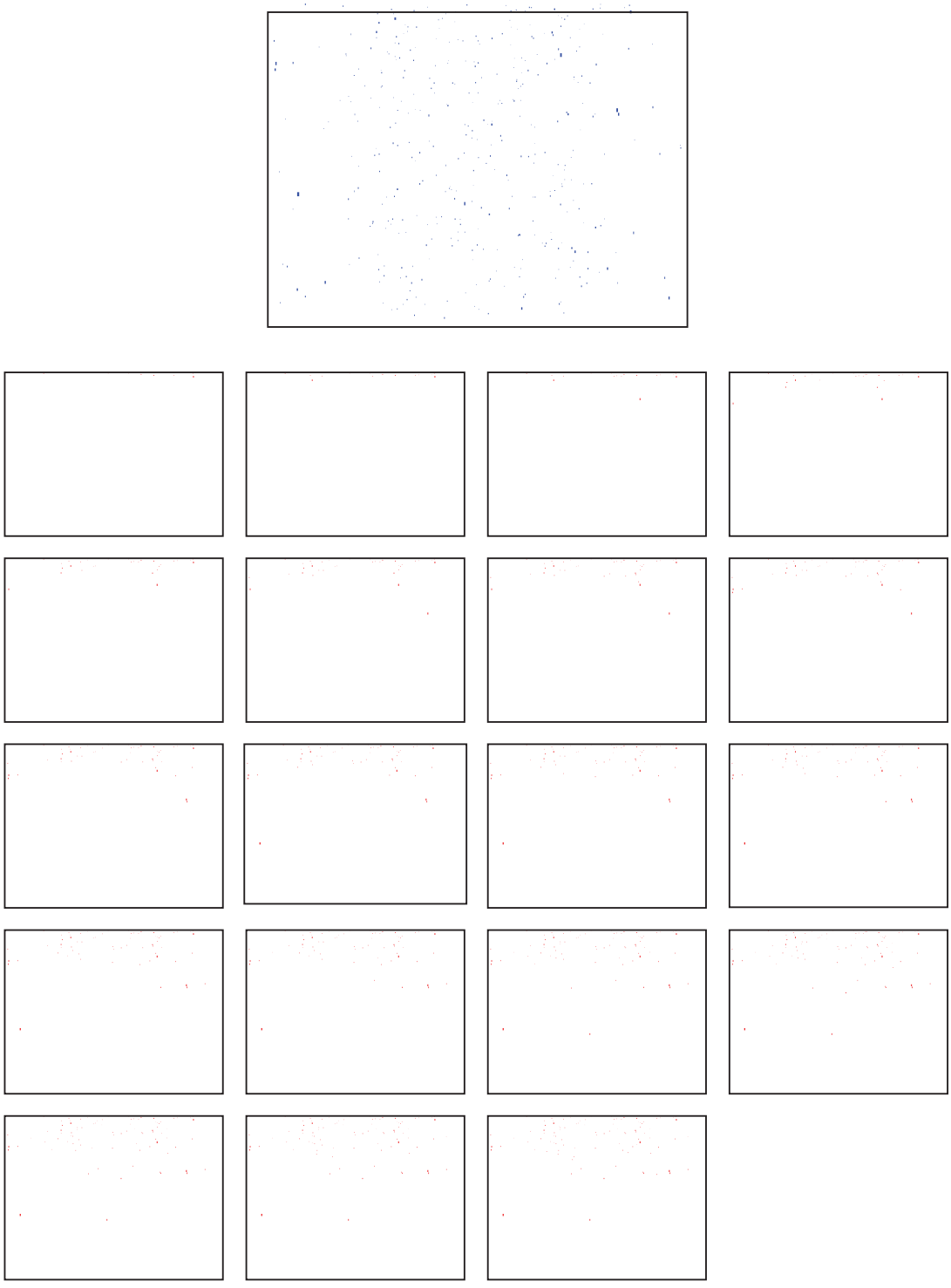


Figure 4.15: A more complete set of examples of the amount of drops missed by the system for different latencies (this image is best viewed zoomed in on a computer screen). The top image shows the drops in the frame. Starting from the top left, the going right, then down (similar to reading text), the latencies increase in steps of about 5ms, up to 100ms at the bottom right.

Conclusion

We have shown that the interactions between light and water drops have implications for fields as diverse as display technology and car safety. For entertainment, we demonstrated how to use a camera, projector, and water valve system to create 2D and 2.5D water drop displays. These displays could be used for static exhibits, like at museums or car shows, or for interactive games. For car safety, we showed how studying the statistics and dynamics of rain and snow can allow us to reduce (or increase) their effect both in videos via post-processing, and in the real world via adaptive projector control. Because of their unique reflectance and refractance properties, water drops can be used for an incredible diversity of purposes.

Appendix A

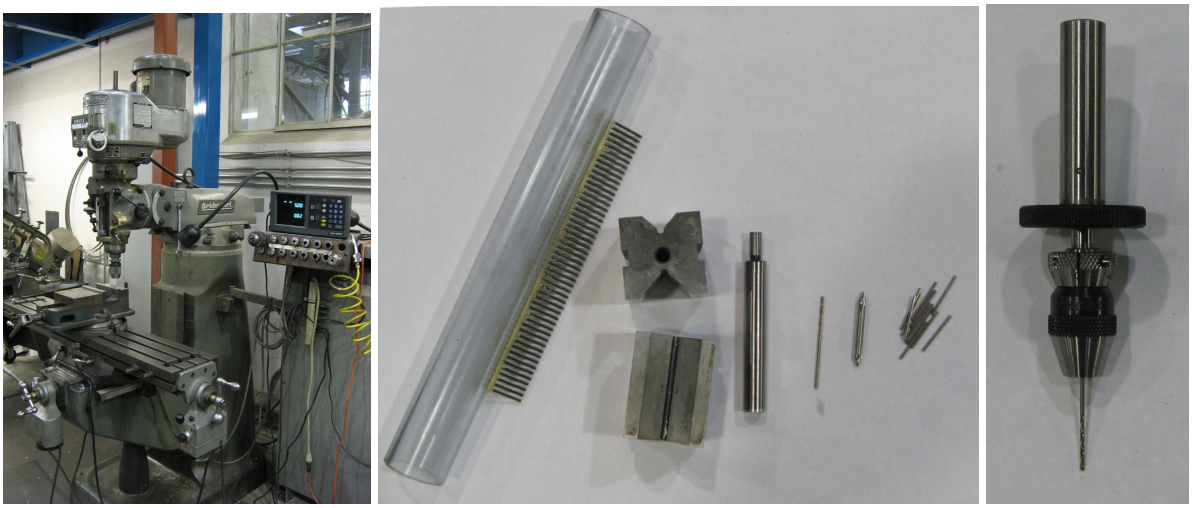
Details on the drop generation hardware

Creating the water drop display of Chapter 2 and the testbed for the adaptive projection system in Chapter 4 required custom designed and fabricated hardware for the water drop emitters. This appendix covers some of the specific requirements and details for fabricating and running these emitters. Due to the need to purchase and use tools and materials inch-based measurements, this section has a mix of inches and metric. Whenever possible, we will use metric, except when discussing specific standards on tools and materials that are based on inch measurements.

A.1 Machining the multi-layer display drop emitters

A.1.1 Basic parts and machining technique

Machining is a complex art and science. In this section, we will cover the basics for machining in our particular case. The best way to learn machining is from an experienced machinist, supplemented by books (or websites) such as [Marlow, 2008] [Harvey, 2005] [Hall, 2004] [Oberg, 2008]. High-precision machining has been around a long time, and there are a wealth of resources. Regarding manual machining, perhaps the biggest difference in the last fifty years is the properties of the materials in the tools. And many of these manual techniques are equally



(a) Milling machine

(b) Tools, parts, and an example final product

(c) Miniature drill chuck and drill bit

Figure A.1: These are the basic parts and tools needed to create a drop generator manifold for the multi-layer water drop display. (a) A standard Bridgeport manual milling machine. (b) The basic tools needed to fabricate a manifold. From left to right, a finished manifold, two v-blocks to hold the round pipe, edge finder, drill bit, center drill, and pre-cut steel tubing. (c) A miniature drill bit, held within a miniature drill chuck, which will be held within the large drill chuck.

applicable to computer-based CNC machining as well.

The most important tool to create the multi-layer drop display manifolds is the milling machine. We used the standard Bridgeport manual milling machine shown in Figure A.1 (a). This milling machine has a spindle that can rotate between about 50 Rotations-Per-Minute (RPM) to about 2000 RPM. This spindle can be attached to a variety of tools, such as a drill bit for making holes or an endmill for making straight cuts. The vise is bolted to the table, and the table can be moved with high precision in three perpendicular axes. The digital readout has a precision of .01 mm. Although in practice, due to flexing of materials, it is difficult to be this precise, and even .1mm precision requires care. For all stages of the milling process, we use an adjustable gear tooth chuck, which is liable to be less accurate than fixed collets, but is faster to set up.

The other basic tools are shown in Figure A.1 (b). From left to right, they are a finished manifold, two v-blocks to hold the round pipe, edge finder, drill bit, center drill, and pre-cut steel tubing. The finished manifold has 50 stainless steel needles inserted into holes in the plastic pipe,

and glued.

We used simple clear “3/4 inch”, schedule 40 PVC plastic pipe. This “3/4 inch”, schedule 40 is a standard size. It has about a 27mm (1.0 inch) outer diameter and a 20mm (0.8 inch) inner diameter (it should be noted that neither of these measurements are 3/4 inch.) This pipe has the advantage of being common, making it easy to get pre-made connectors. Better solutions would a better plastic, like acrylic (although this has a tendency to yellow with age,) or metal. In addition, this pipe has fairly thin walls, which adds to the imprecision.

The v-blocks are needed to keep the round pipe from rotating when in the vise. Even with these v-blocks, the clamp has to be fairly tight, which slightly compresses the pipe, reducing accuracy. Using thicker, stronger plastic would help, as would putting a temporary insert into the pipe.

The edge-finder allows us to match the coordinate system of the mill with the object being milled (in our case, the plastic pipe.) With the mill spindle spinning at about 600 RPM, the edge finder is gradually brought in contact with the object being milled. Within a precision of about .01mm, the tip of the edge finder will jump sideways when it encounters the object. This allows a machinist to zero the digital (or analog) readout, to know exactly where the object is.

The drill bit is a standard type. We are going to be inserting .8mm steel needles, which determines the size of the bit to use. Normally, the drill bit will create a slightly wider hole than itself. This depends on the material, so you will likely have to experiment to determine what size bit to use. Unlike most machining materials, the PVC has a tendency to expand a bit, so requires a slightly larger drill bit than would be expected.

Using a center drill is essential for accurately positioning holes. This is doubly true, due to the small size of the drill bits we are using, and triply true since we will be drilling on a curved surface. A center drill is much more rigid that a standard drill bit, therefore it will flex much less when it contacts the surface. The center drill is used to start the hole, then the standard drill bit is used to finish the hole.

The final part are the welded and drawn, type 304 stainless steel needles. These are similar to the stainless steel needles used for luer lock syringes to dispense adhesive. These needles have approximately a 1.27mm outer diameter, .81 mm inner diameter, 19mm length. We ordered them from Eagle Stainless Tube & Fabrication (although many companies worldwide can do similar work, based on the specifications for standard 18 gauge tubing, .0495/.0505 inch outer diameter, .0315/.0345 inch inner diameter, cut .750 +/- .010 inch length.)

A.1.2 Specific machining procedure

Actually creating the manifolds is a conceptually simple process, although it may take some practice. This section will be explained pictorially, and in the captions of Figures A.2, A.3, A.4, and A.5.

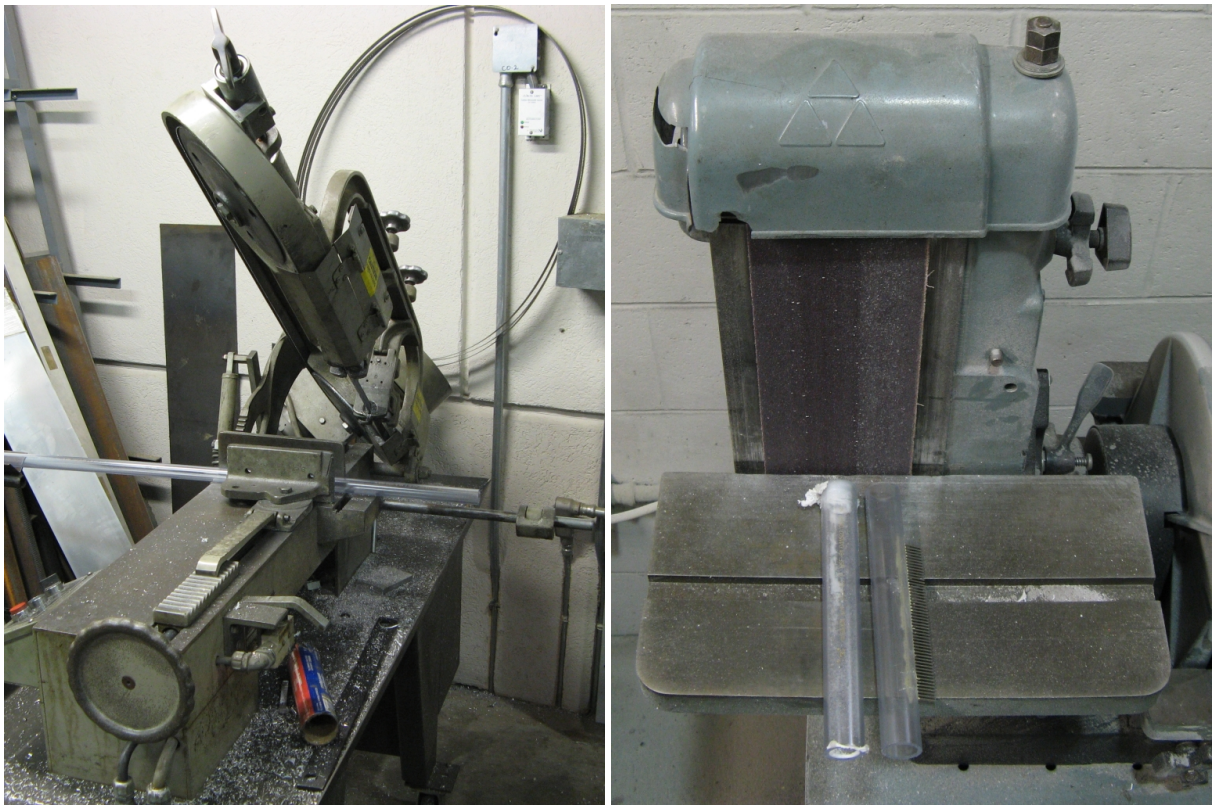
A.2 Adaptive projection testbed hardware

A.2.1 Basic system setup

The illustration and photo in Figure A.6 show the basic hardware setup for the adaptive projection testbed. The large tank of water gives a slight pressure to the manifold, which has an individual tube connection to each valve. The structure is made from the common 80/20 brand extruded aluminum.

A.2.2 Solenoid valves

The adaptive projection testbed uses small solenoid valves (Figure A.7), and low-power switching circuitry (Figure A.8). Each solenoid valve is responsible for a single drop. The solenoid valves have a very light spring to push the valve closed, and the solenoid to open it. There is a small hole in the metal, to allow the pressures of the upper and lower section to equalize. As



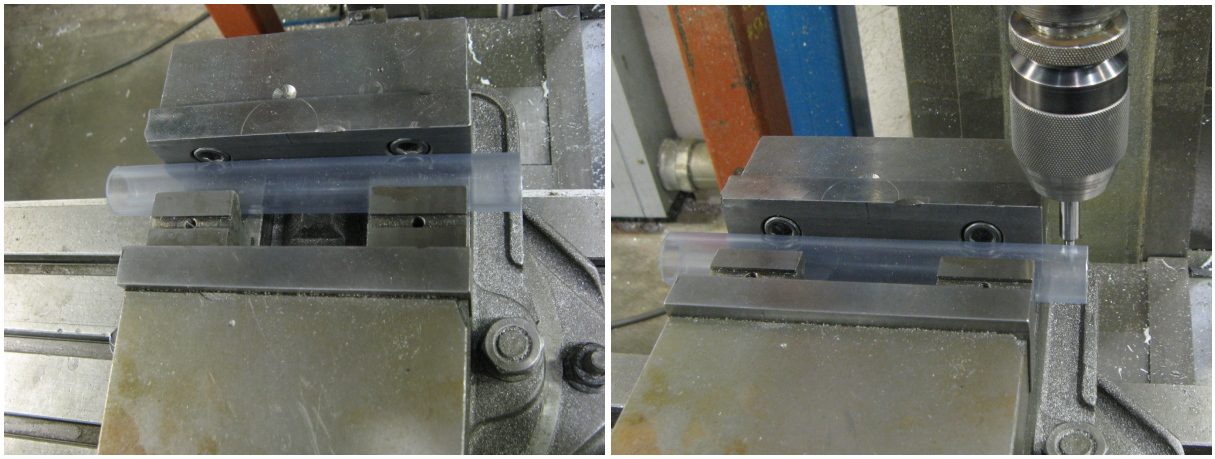
(a) Rough Cut (b) Sand to Length

Figure A.2: The first step is to rough cut the stock with a horizontal band saw (probably the best option if available, although even a hand saw will be fine.) Try to get within 10mm or so. The exact length is not as important as getting all the manifolds to be the same length. Given that, once you rough cut, use a belt sander to get all the tube to be approximately the same length, within a few mm if possible. Using a sander may cause the residue shown on the left pipe in (b), but this can be knocked off by filing very lightly with a metal file.

shown in Figure A.7 (b), the high-pressure water enters through the upper connector, travels to the upper hole in the bronze circular ring. If the valve is open, the water then exits through the hold in the middle of the bronze circular ring, and exits through the bottom connector.

A.2.3 Electronics

The electronics for the valves are fairly simple. As the valves only draw 0.05 amps at 12 volts direct-current, we switch them with a standard Darlington optocoupler and an external flyback diode. The switching signals are sent from an Arduino Mega 1280, which has 54 digital in-



(a) Setup with the v-blocks (b) Edge Finding

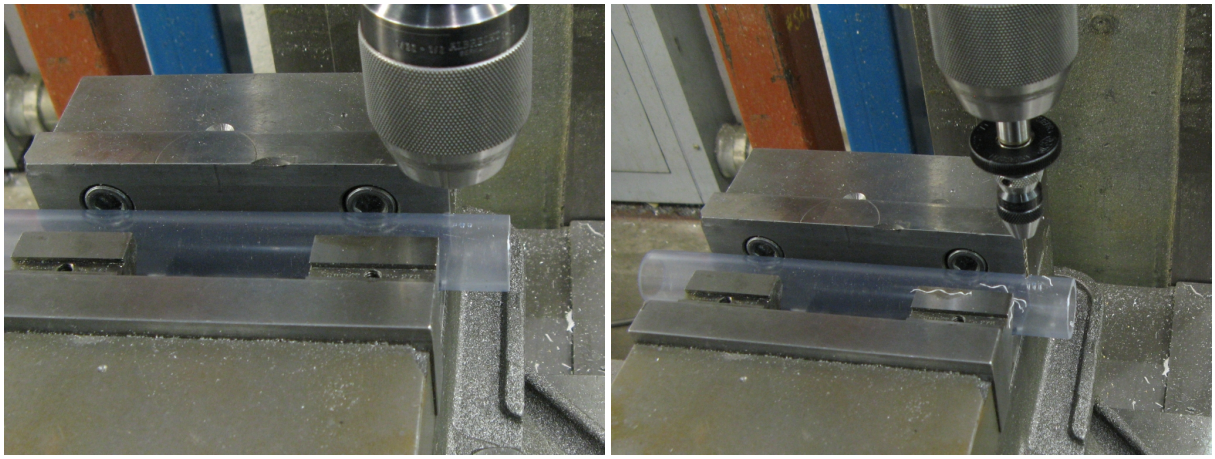
Figure A.3: (a) Secure the pipe with v-blocks on one side. You will likely have to tighten the vise enough that the pipe will distort slightly. (b) To find the center of the short axis of the (slightly) distorted pipe, measure one side, and zero the digital (or analog) readout, without taking the size of the edge finder into account. Then find the edge on the other side of the pipe, and divide by two to find the true center. Find the long axis in the same way (although precision is not as important on the long axis.)

put/output pins.

A.3 Drop display hardware

A.3.1 Solenoid valves

The water drop display uses large solenoid valves (Figure A.9), and high-power switching circuitry (Figure A.10). Due to the large size of the valves relative to the amount of water we need, the valves are never completely opened. Instead, they flutter between fully closed and partially open. This has the advantage of being quiet, as no hard parts of the valves contact each other. In addition, this reduces the amount of turbulence that would be caused by a collision.



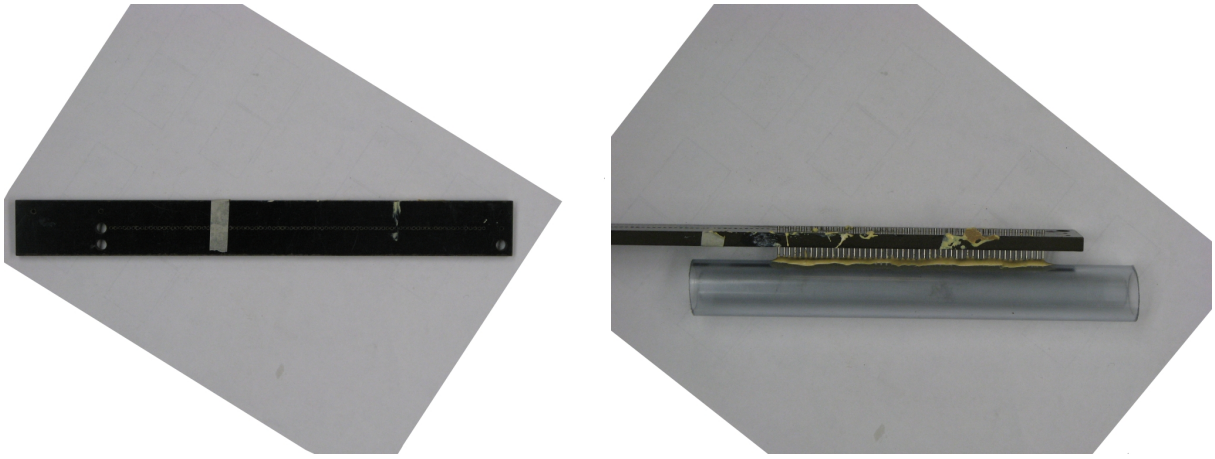
(a) Center drilling (b) Drilling

Figure A.4: Going from the center of the pipe, center drill holes with 2.5mm spacing, then drill with the standard bit. (Note that we are showing holes much closer to the end than you should do, to make it easier to see).

A.3.2 Electronics

The drop display uses large valves that draw approximately 2 amps at 12 volts direct-current. Due to these high-power requirements, we use a MOSEFET switching circuit. The timing signal is processed through two Arduino Duemilanove microcontrollers. The first takes the raw vsync input from the video card, and creates a camera sync signal and a valve-controller sync signal. When the second, valve controller microcontroller receives the sync, it opens each of the valves at the correct time. The timings for each valve are send over an emulated RS-232 connection on the USB connection.

The current valve controller microcontroller uses only a single core, but could benefit from a multi-core implementation. The timing signal should be as accurate as possible, and this would be better assured by using a board like the eight-core Propeller chip. Getting accurate, synchronized, and user-programmable timings would be easier in such a case, especially when parts of the open times for valves occur at the same time.

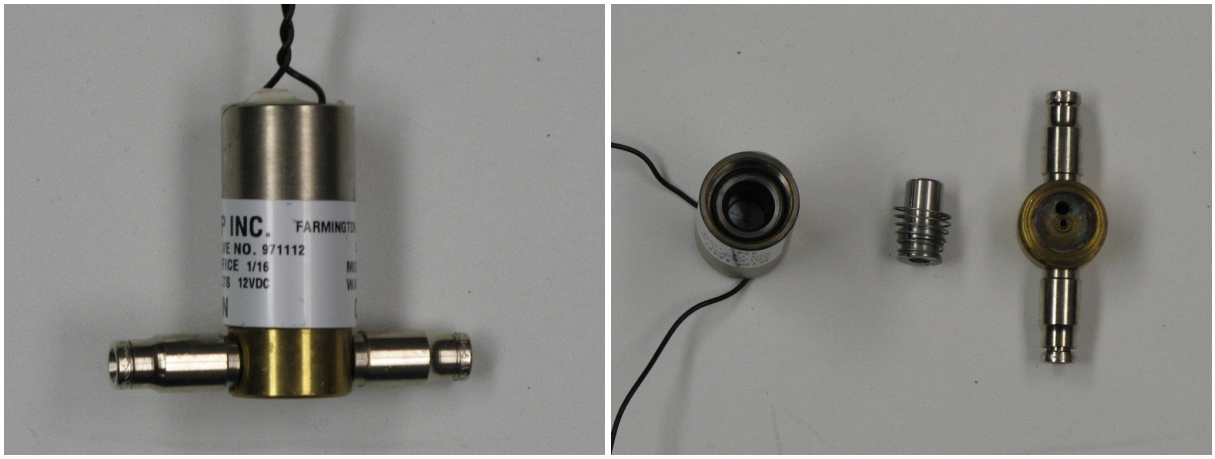


(a) Create a temporary plate to keep the needles vertical (b) Secure with the temporary plate and waterproof glue while the glue dries

Figure A.5: Once the holes are drilled, it is time to place the needles. Using the same technique as the pipe, create a piece of plastic with the same hole size and spacing. This extra piece will be used to keep the needles parallel to each other as the glue dries. Use a glue that is made to dry underwater, as even standard epoxy is liable to degrade due to the constant contact with water.

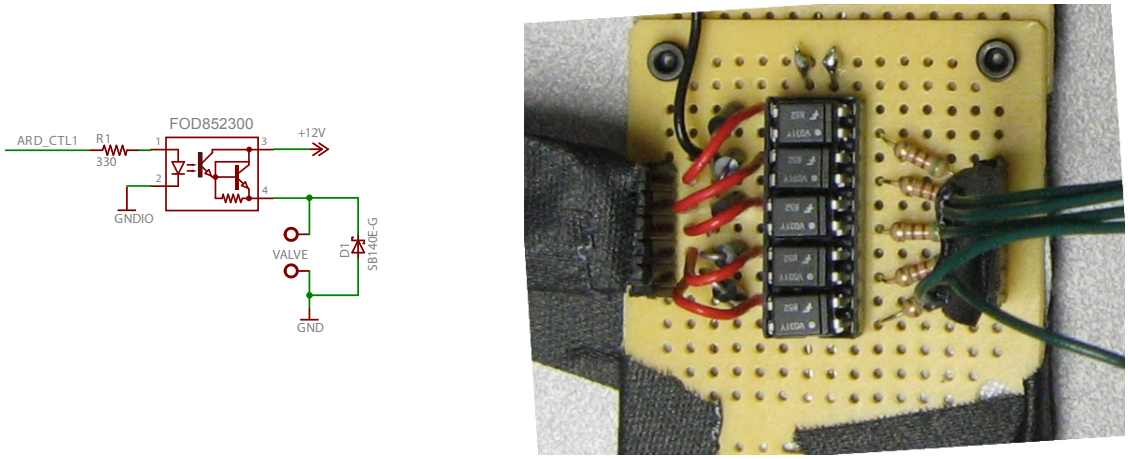
(a) Illustration (b) Photo

Figure A.6: The basic adaptive projection testbed setup.



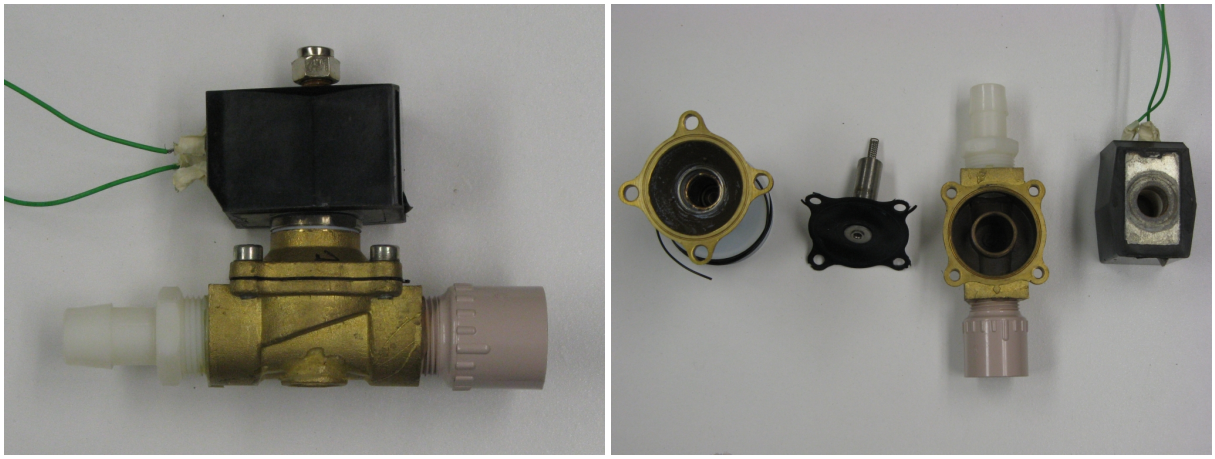
(a) Outside (b) Inside

Figure A.7: The outside and inside of one of the small solenoid valves used to control one drop emitter for the adaptive projection testbed.



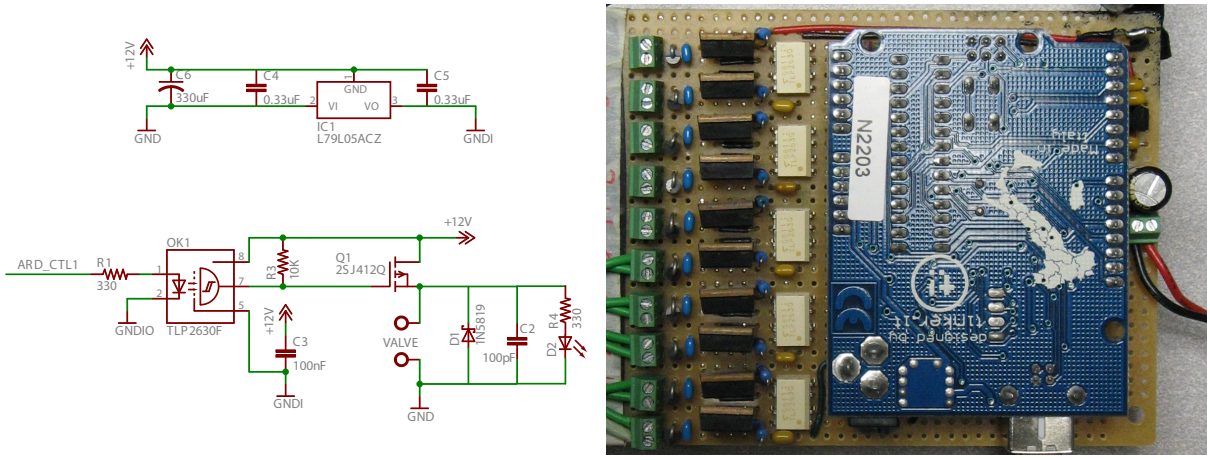
(a) Circuit diagram (b) Picture of circuit

Figure A.8: Details of the adaptive projection testbed electronics. In short, each valve is controlled by a separate Darlington optocoupler, with an external flyback diode.



(a) Outside (b) Inside

Figure A.9: The outside and inside of one of the large solenoid valves used to control one linear manifold for the drop display.



(a) Circuit diagram

(b) Picture of circuit

Figure A.10: Details of the water drop display electronics. The circuit is based around MOSFET switching.

Appendix B

Accuracy of the rain/snow model

In this appendix, we derive an expression highlighting the simple structure of Equation 3.9. Rain and snow are fundamentally random, and there is no guarantee if the magnitude of Equation 3.9 will have a simple closed form. However, we can analyze it in the case where each image has the same number of each size of streak, given that each streak is equally likely to appear in any location.

To begin, we show the closed form solution of the Fourier transform of a blurred Gaussian. We show only the case where the streak is completely vertical, and include a normalizing constant for ease of reading. The notation in this section is similar to the rest of the paper, but some symbols are redefined.

B.1 The Fourier transform of a rain streak

To begin, a Gaussian, with width b, blurred over length l is given by:

$$g(x, y; b, l) = \int_{c=0}^{l} exp(-\pi \frac{(x^2 + (y - c)^2)}{b^2}) dc$$

The Fourier transform is given by:

$$G(u,v;b,l) = \mathcal{F}\{\int_{c=0}^{l} exp(-\pi\frac{(x^2+(y-c)^2)}{b^2})dc\}$$

The length integral can be moved outside of the transform:

$$\int_{c=0}^{l} \mathcal{F}\{exp(-\pi\frac{(x^2+(y-c)^2)}{b^2})\}dc$$

Since a shift in space is a multiplication in frequency, the equation becomes:

$$\int_{c=0}^{l} \mathcal{F}\{exp(-\pi\frac{(x^2+y^2)}{b^2})\}exp(2\pi ivc)dc =$$
$$\int_{c=0}^{l} b^2 exp(-\pi b^2(u^2+v^2))exp(2\pi ivc)dc$$

Only the rightmost exponential depends on n, so we can take out the left part and solve the integral:

$$b^2 exp(-\pi b^2(u^2+v^2)) \int_{c=0}^{l} exp(2\pi ivc)dc =$$
$$G(u,v;b,l) = b^2 exp(-\pi b^2(u^2+v^2))i\frac{1-exp(2\pi ivl)}{2\pi v}$$

The magnitude of G is simple itself multiplied by its complex conjugate:

$$
\begin{aligned}
||G(u,v;b,l)|| = \\
(b^2 exp(-\pi b^2(u^2+v^2))i\frac{1-exp(2\pi ivl)}{2\pi v})(-b^2 exp(-\pi b^2(u^2+v^2))i\frac{1-exp(-2\pi ivl)}{2\pi v}) = \\
\frac{b^4 exp(-2\pi b^2(u^2+v^2))}{4\pi^2 v^2}(1-exp(2\pi ivl))(1-exp(-2\pi ivl)) = \\
\frac{b^4 exp(-2\pi b^2(u^2+v^2))}{4\pi^2 v^2}(2-2cos(2\pi vl)) = \\
\frac{b^4 exp(-2\pi b^2(u^2+v^2))}{4\pi^2 v^2}(4sin^2(\pi vl)) = \\
||G(u,v;b,l)|| = \frac{b^4 sin^2(\pi vl) exp(-2\pi b^2(u^2+v^2))}{\pi^2 v^2}
\end{aligned}
\tag{B.1}
$$

B.2 The Fourier transform of multiple identical streaks

The next step is to determine the magnitude of multiple streaks. To begin, we assume that all streaks are identical, but in different locations, $\boldsymbol{\mu} = [\mu_x, \mu_y]$. The image of all streak is then:

$$\sum_n g(x,y;b,l,\boldsymbol{\mu_n})$$

The Fourier transform is:

$$
\begin{aligned}
\mathcal{F}\{\sum_n g(x,y;b,l,\boldsymbol{\mu_n})\} = \\
\sum_n \mathcal{F}\{g(x,y;b,l,\boldsymbol{\mu_n})\} = \\
\sum_n G(u,v;b,l) exp(2\pi i(u\mu_{xn} + v\mu_{yn})) = \\
G(u,v;b,l)\sum_n exp(2\pi i(u\mu_{xn} + v\mu_{yn}))
\end{aligned}
\tag{B.2}
$$

The magnitude is given by:

$$(G(u,v;b,l)\sum_n exp(2\pi i(u\mu_{xn}+v\mu_{yn})))(G^*(u,v;b,l)\sum_n exp(-2\pi i(u\mu_{xn}+v\mu_{yn}))) = \quad \text{(B.3)}$$
$$G(u,v;b,l)G^*(u,v;b,l)(\sum_n exp(2\pi i(u\mu_{xn}+v\mu_{yn})))(\sum_n exp(-2\pi i(u\mu_{xn}+v\mu_{yn})))$$

The two sums will multiply into pairs of exponentials, which can be converted into cosines:

$$G(u,v;b,l)G^*(u,v;b,l)(N+\sum_{a=1}^{N-1}\sum_{b=a+1}^{N} cos(2\pi(u(\mu_{xa}-\mu_{xb})+v(\mu_{ya}-\mu_{yb})))) \quad \text{(B.4)}$$

The first part with G multiplied by its conjugate is the magnitude from Equation B.1:

$$\frac{b^4 sin^2(\pi vl)exp(-2\pi b^2(u^2+v^2))}{\pi^2 v^2}(N+\sum_{a=1}^{N-1}\sum_{b=a+1}^{N} cos(2\pi(u(\mu_{xa}-\mu_{xb})+v(\mu_{ya}-\mu_{yb})))) \quad \text{(B.5)}$$

The resulting equation is the magnitude of the blurred Gaussian, multiplied by the addition of N and a sum of cosines term.

B.3 The Fourier transform of multiple streaks of different sizes

In the general case of streaks of many different sizes, the resulting expression does not reduce as cleanly as in Equation B.5. But it does in the special case where at a given location, there are the same number of each size. Starting from Equation B.2, but with multiple bs and ls:

$$\sum_n(\sum_l\sum_b G(u,v;b,l))exp(2\pi i(u\mu_{xn}+v\mu_{yn}))$$

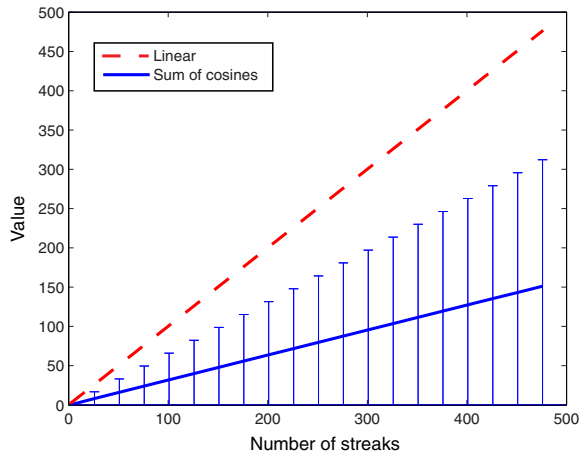


Figure B.1: To determine the value of the sum of cosines in Equation B.5 compared to the number of streaks N, we ran simulations with different numbers of streaks. The dotted line is the value of the linear component. The solid line is the mean across trials of the median of the absolute value of the sum of cosines.

Since it does not depend on n, the double sum of blurred Gaussians can be pulled out in the same way, yielding:

$$(\sum_{l}\sum_{b} G(u,v;b,l)) \sum_{n} exp(2\pi i(u\mu_{xn} + v\mu_{yn})) \tag{B.6}$$

And the remaining steps are the same as Equations B.3 to B.5.

B.4 Approximating the Fourier transform of multiple streaks

The value of the sum of cosines in Equation B.5 varies depending on the frequency (u, v) and the distribution of streak locations μ. It has a minimum of zero and a very large maximum at $(u, v) = (0, 0)$, but streaks at different locations tend to cancel each other out, so it is generally low. We ran simulations to find the kind of values that the sum of cosines tends to have.

In our experiments, we computed the value by sampling different numbers of streaks with uniformly distribution locations. Figure B.1 shows the value for the median of all frequencies, for

one to five hundred streaks, with one thousand trials each. For each number of streaks N from one to five hundred, we ran one thousand trials of randomly sampled streak locations. The dotted line is the value of N from the first term in the addition in Equation B.5. The solid line is for the sum of cosines term. For all frequencies u and v, we compute the absolute value of the sum of cosines. For each trial, we then compute the median of the value for all frequencies except $(u, v) = (0, 0)$. This median is computed for each of the one thousand trials, and the solid line is the mean across trials of the medians. The error bars represent the mean across trials of the standard deviation for all frequencies. The results show that although the value for some frequencies is large, the median value is low enough that $N + \sum_{a=1}^{N-1} \sum_{b=a+1}^{N} cos(2\pi(u(\mu_{xa} - \mu_{xb}) + v(\mu_{ya} - \mu_{yb}))$ can be approximated as being linear in N, which validates our use of Equation 3.10.

Bibliography

Davide Agnelli, Dario Buzzini, and Tal Drori. Fashion victims. In *Designing interactive systems: processes, practices, methods, and techniques*, 2004.

Bala Ambravaneswaran, Edward D. Wilkes, and Osman A. Basarana. Drop formation from a capillary tube: Comparison of one-dimensional and two-dimensional analyses and occurrence of satellite drops. *Physics of Fluids*, 14(8):2606–2621, 2002.

Toshimitsu Araki, Fumihiko Kawamata, Masayuki Ogino, Hitoshi Miyagawa, Toshio Kamata, Mitsuo Watanabe, and Kunihiro Miyashita. US Patent 5,067,653: Screen forming apparatus and method. 1991.

August H. Auer Jr. Distribution of graupel and hail with size. *Monthly Weather Review*, 100(5): 325–328, 1972.

August H. Auer Jr. and Donald L. Veal. The dimension of ice crystals in natural clouds. *Journal of the Atmospheric Sciences*, 27(6):919–926, 1970.

Ian Austen. Illuminating road hazards that lurk beyond lights. In *The New York Times*, October 31, 2005.

Donald G. Bailey. Raster based region growing. In *New Zealand Image Processing Workshop*, 1991.

Peter C. Barnum, Mei Chen, Hiroshi Ishikawa, Gadi Wollstein, and Joel Schuman. Local quality assessment for optical coherence tomography. In *ISBI*, 2008a.

Peter C. Barnum, Mei Chen, Hiroshi Ishikawa, Gadi Wollstein, and Joel Schuman. Pathology insensitive quality assessment for optical coherence tomography. In *ARVO*, 2008b.

Peter C. Barnum, Srinivasa G. Narasimhan, and Takeo Kanade. A projector-camera system for creating a display with water drops. In *PROCAMS*, 2009a.

Peter C. Barnum, Yaser Sheikh, Ankur Datta, and Takeo Kanade. Dynamic seethroughs: Synthesizing hidden views of moving objects. In *ISMAR*, 2009b.

Peter C. Barnum, Srinivasa G. Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2-3):256–274, 2010a.

Peter C. Barnum, Srinivasa G. Narasimhan, and Takeo Kanade. A multi-layered display with water drops. In *SIGGRAPH*, 2010b.

Kenneth V. Beard. Terminal velocity and shape of cloud and precipitation drops aloft. *Journal of the Atmospheric Sciences*, 33(5):851–864, 1976.

Massimo Bertozzi and Alberto Broggi. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7(1):62–81, 1998.

A. C. Best. Empirical formulae for the terminal velocity of water drops falling through the atmosphere. *Quarterly Journal of the Royal Meteorological Society*, 76(329):302311, 1950.

Black Rock City, llc. Burning man.

Duncan C. Blanchard. Raindrop size-distribution in Hawaiian rains. *Journal of Atmospheric Sciences*, 10(6):457–473, 1953.

Hannes P. Böhm. A general equation for the terminal fall speed of solid hydrometeors. *Journal of the Atmospheric Sciences*, 46:2419–27, 1989.

Jèrèmie Bossu, Nicolas Hautière, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use ofahistogram of orientation of streaks. *International Journal of Computer*

Vision, 2011. ISSN 0920-5691.

Jean-Yves Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. Intel Corporation, Microprocessor Research Labs, 2000.

Nathan Brewer and Nianjun Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Structural, Syntactic, and Statistical Pattern Recognition*, 2008.

James Bruce, Tucker Balch, and Manuela Veloso. Fast and inexpensive color image segmentation for interactive robots. In *IROS*, 2000.

Wang Changbo, Zhangye Wang, Xin Zhang, Lei Huang, Zhiliang Yang, and Qunsheng Peng. Real-time modeling and rendering of raining scenes. *The Visual Computer*, 24(7-9):605–616, 2008.

G.F. Christopher and S.L. Anna. Microfluidic methods for generating continuous droplet streams. *Journal of Physics D Applied Physics*, 40:R319–R336, 2007.

Fabio Cozman and Eric Krotkov. Depth from scattering. In *International Conference on Computer Vision*, 1997.

Andrew Dahley, Craig Wisneski, and Hiroshi Ishii. Water lamp and pinwheels: Ambient projection of digital information into architectural space. In *CHI*, 1998.

Fernando de la Torre and Michael L. Black. Robust principal component analysis for computer vision. In *International Conference on Computer Vision*, 2001.

Roger Lannes de Montebello. US Patent 3,428,393: Optical dissector.

Nicolas Desaulniers-Soucy. *Empirical test of the multifractal continuum limit in rain*. PhD thesis, Mcgill, 1999.

Nicolas Desaulniers-Soucy, Shaun Lovejoy, and Daniel Schertzer. The HYDROP experiment: an empirical method for the determination of the continuum limit in rain. *Atmospheric Research*, 59-60:163–197, 2001.

Ernst D. Dickmanns. The development of machine vision for road vehicles in the last decade. In *IEEE Intelligent Vehicle Symposium*, 2002.

Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *SIGGRAPH*, 2003.

Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 1, 1999.

Shinichiro Eitoku, Kotaro Hashimoto, and Tomohiro Tanikawa. Controllable water particle display. In *SIGCHI international conference on Advances in computer entertainment technology*, 2006a.

Shinichiro Eitoku, Tomohiro Tanikawa, and Yasuhiro Suzuki. Display composed of water drops for filling space with materialized virtual three-dimensional objects. In *Virtual Reality*, 2006b.

Shinichiro Eitoku, Tomohiro Tanikawa, Yasuhiro Suzuki, Koichi Hirota, Toshio Iwai, and Michitaka Hirose. Display using water drops for displaying vr objects. *Trans. on the Virtual Reality Society of Japan*, 12(3):425–435, 2007.

Shinichiro Eitoku, Kunihiro Nishimura, Tomohiro Tanikawa, and Michitaka Hirose. Study on design of controllable particle display using water drops suitable for a light environment. In *Proc. of ACM VRST*, 2009.

Shinichiro Eitoku, Kunihiro Nishimura, Tomohiro Tanikawa, and Michitaka Hirose. Method to increase refresh rate of display using water drops featured three-dimensional and color representation. *Trans. on the Virtual Reality Society of Japan*, 15(3):439–448, 2010.

Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision*, 2000.

Graham Feingold and Zev Levin. The lognormal fit to raindrop spectra from frontal convective clouds in Israel. *Journal of Climate and Applied Meteorology*, 25:1346–63, 1986.

Zhong-Xin Feng, Min Tang, Jin-Xiang Dong, and Shang-Ching Chou. Real-time rain simulation. In *Computer Supported Cooperative Work in Design II*, 2006.

Martin Fishler and Robert Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 1981.

Robert P. Fleishauer, Vincent E. Larson, and Thomas H. Vonder Haar. Observed microphysical structure of midlevel, mixed-phase clouds. *Journal of the Atmospheric Sciences*, 59(11): 17791804, 2002.

G. B. Foote and P. S. duToit. Terminal velocity of raindrops aloft. *Journal of Applied Meteorology*, 8(2):249–53, 1969.

Arnold Frohn and Norbert Roth. *Dynamics of Droplets*. Springer, 2000.

J.G. Fryer and D.C. Brown. Lens distortion for close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 52:51–58, 1986.

Kshitiz Garg and Shree K. Nayar. Detection and removal of rain from videos. In *Computer Vision and Pattern Recognition*, 2004.

Kshitiz Garg and Shree K. Nayar. When does a camera see rain? In *International Conference on Computer Vision*, 2005.

Kshitiz Garg and Shree K. Nayar. Photorealistic rendering of rain streaks. In *SIGGRAPH*, 2006.

Dariu M. Gavrila. Sensor-based pedestrian protection. *IEEE Intelligent Systems*, 16(6):77–81, 2001.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing 2nd Edition*. Prentice Hall, 2002.

Eugenio Gorgucci, Luca Baldini, and V. Chandrasekar. What is the shape of a raindrop? an answer from radar measurements. *Journal of the Atmospheric Sciences*, 63(11):30333044,

2006.

Theodore O. Grosch. Radar sensors for automotive collision warning and avoidance. In *Proceedings of SPIE*, 1995.

Stefan Grundhoff. Technik: Hightech im 7er bmw. In *Alle Autos In*, 2008.

K.L.S. Gunn and J.S. Marshall. The distribution with size of aggregate snowflakes. *Journal of Meteorology*, 15:452–61, 1958.

Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa Narasimhan. Flexible voxels for motion-aware videography. In *European Conference on Computer Vision*, 2010.

Harold Hall. *Milling: A Complete Course (Workshop Practice)*. Trans-Atlantic Publications, Inc., 2004.

Takenori Hara, Hideo Saito, and Takeo Kanade. Removal of glare caused by water droplets. In *Conference for Visual Media Production*, 2009.

James A Harvey. *Machine Shop Trade Secrets: A Guide to Manufacturing Machine Shop Practices*. Industrial Press, Inc., 2005.

Hiroyuki Hase, Kazunaga Miyake, and Masaaki Yoneda. Real-time snowfall noise elimination. In *International Conf. on Image Processing*, 1999.

Akira Hasegawa. Digital kakejiku.

Kaiming He, Jian Sun, and Xiaou Tang. Single image haze removal using dark channel prior. In *Computer Vision and Pattern Recognition*, 2009.

David J. Heeger. Optical flow from spatiotemporal filters. In *International Conference on Computer Vision*, 1987.

Jeremy M. Heiner, Scott E. Hudson, and Kenichiro Tanaka. The information percolator: Ambient information display in a decorative object. In *Symposium on User Interface Software and Technology*, 1999.

Shiho Hirayama and Yasuaki Kakehi. Shaboned display. In *SIGGRAPH Emerging Technologies*, 2010.

Peter V. Hobbs and Arthur L. Rangno. Microstructures of low and middle-level clouds over the Beaufort Sea. *Quarterly Journal of the Royal Meteorological Society*, 124(550): Microstructures of low and middle–level clouds over the Beaufort Sea, 1998.

Honda. Intelligent night vision system. In *Honda Press Release*, September, 2004.

JG Hosking. The arrival rate of raindrops at the ground. *Journal of Applied Meteorology*, 26(4): 433–442, 1987.

Hiroshi Ishii, Craig Wisneski, Scott Brave, Andrew Dahley, Matt Gorbet, Brygg Ullmer, and Paul Yarin. ambientroom: Integrating ambient media with architectural space. In *CHI*, 1998.

A. R. Jameson and A. B. Kostinski. What is a raindrop size distribution? *Bulletin of the American Meteorological Society*, 82(6):1169–1177, 2001.

A. R. Jameson and A. B. Kostinski. When is rain steady? *Journal of Applied Meteorology*, 41: 83–90, 2002.

Willie D. Jones. Safer driving in the dead of night. In *IEEE Spectrum*, March 2006.

Akira Kataoka and Yasuhiro Kasahara. US Patent 5,270,752 : Method and apparatus for a fog screen and image-forming method using the same. 1993.

Qifa Ke and Takeo Kanade. A robust subspace approach to layer extraction. In *IEEE Workshop on Motion and Video Computing*, 2002.

Hidei Kimura, Taro Uchiyama, and Hiroyuki Yoshikawa. Laser produced 3D display in the air. In *SIGGRAPH*, 2006.

A. B. Kostinski and A. R. Jameson. On the spatial distribution of cloud particles. *Journal of the Atmospheric Sciences*, 57(7):901–915, 2000.

Rodney J. Kubesh and Kenneth V. Beard. Laboratory measurements of spontaneous oscillations

of moderate-size raindrops. *Journal of the Atmospheric Sciences*, 50:1089–1098, 1993.

Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *SIGGRAPH*, 2003.

Michael S. Langer and Richard Mann. Optical snow. *International Journal of Computer Vision*, 55(1):55–71, 2003.

Michael S. Langer and Linqiao Zhang. Rendering falling snow using and inverse fourier transform. In *ACM SIGGRAPH Technical Sketches Program*, 2003.

Michael S. Langer, Linqiao Zhang, A.W. Klein, A. Bhatia, J. Pereira, and D. Rekhi. A spectral-particle hybrid method for rendering falling snow. In *Eurographics Symposium on Rendering*, 2004.

M. P. Langleben. The terminal velocity of snowflakes. *Quarterly Journal of the Royal Meteorological Society*, 80(344):174181, 1954.

Yves Lauzière, Denis Gingras, and Frank P. Ferrie. A model-based road sign identification system. In *Computer Vision and Pattern Recognition*, 2001.

J. Lavergnat and P. Golé. A stochastic raindrop time distribution model. *Journal of Applied Meteorology*, 37:805–818, 1998.

Hue P. Le. Progress and trends in ink-jet printing technology. *Journal of Imaging Science Technology*, 42:49–62, 1998.

Johnny C. Lee, Paul H. Dietz, Dan Maynes-Aminzade, Ramesh Raskar, and Scott E. Hudson. Automatic projector calibration with embedded light sensors. In *ACM symposium on User interface software and technology*, 2004.

Hao Li, H.-M Rein, T. Suttorp, and J. Bock. Fully integrated SiGe VCOs with powerful output buffer for 77-GHz automotive radar systems and applications around 100 GHz. *IEEE Journal of Solid-State Circuits*, 39(10):1650 – 1658, 2004.

Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150, 2001.

Peng Liu, Jing Xu, Jiafeng Liu, and Xianglong Tang. Pixel based temporal analysis using chromatic property for removing rain from videos. In *Computer and Information Science*, 2009.

Martin Löffler-Mang and Jürg Joss. An optical disdrometer for measuring size and velocity of hydrometeors. *Journal of Atmospheric and Oceanic Technology*, 17(2):130–139, 2000.

S. Lovejoy, M. Lilley, N. Desaulniers-Soucy, and D. Schertzer. Large particle number limit in rain. *Physical Review*, 68(2):–, 2003.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 20:91–110, 2004.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, 1981.

C. Magono and T. Nakamura. Aerodynamic studies of falling snowflakes. *Journal of the Meteorological Society of Japan*, 43:139–47, 1965.

Frank M. Marlow. *Machine Shop Essentials: Questions and Answers*. Metal Arts Press, 2nd edition, 2008.

J.S. Marshall and W.M. Palmer. The distribution of raindrops with size. *Journal of Meteorology*, 5:165–6, 1948.

B. J. Mason and J. B. Andrews. Drop-size distributions from various types of rain. *Quarterly Journal of the Royal Meteorological Society*, 86:346353, 1960.

Christoph Mertz, Sue McNeil, and Charles Thorpe. Side collision warning systems for transit buses. In *IEEE Intelligent Vehicles Symposium*, 2000.

C. Metz, J. Grubert, J. Heyen, A.F. Jacob, S. Janot, E. Lissel, G. Oberschmidt, and L.C. Stange. Fully integrated automotive radar sensor with versatile resolution. *IEEE Transactions on Mi-*

crowave Theory and Techniques, 49(12):2560 – 2566, 2001.

Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, 2001.

Pranav Mistry and Pattie Maes. Sixthsense a wearable gestural interface. In *Proceedings of SIGGRAPH Asia 2009, Sketch*, 2009a.

Pranav Mistry and Pattie Maes. Sixthsense - a wearable gestural interface. In *Proceedings of SIGGRAPH Asia 2009, Emerging Technologies*, 2009b.

David L. Mitchell. Use of mass- and area-dimensional power laws for determining precipitation particle terminal velocities. *Journal of the Atmospheric Sciences*, 53(12):1710–1723, 1996.

Andrew Vande Moere. Beyond the tyranny of the pixel: Exploring the physicality of information visualization. In *Information Visualisation*, 2008.

John L. Muerle and Donald C. Allen. Experimental evaluation of techniques for automatic segmentation of objects in a complex scene. In *Pictorial Pattern Recognition*, 1968.

Bruce R. Munson. *Fundamentals of Fluid Mechanics*. John Wiley & Sons, 6th edition, 2009.

Masahiro Nakamura, Go Inaba, Jun Tamaoki, Kazuhito Shiratori, and Junichi Hoshino. Bubble cosmos. In *SIGGRAPH Emerging Technologies*, 2006.

NAO Design. infernoptix - Digital Pyrotechnic Matrix. www.infernoptix.com.

Srinivasa G. Narasimhan and Shree K. Nayar. Chromatic framework for vision in bad weather. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

Srinivasa G. Narasimhan and Shree K. Nayar. Removing weather effects from monochrome images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

Srinivasa G. Narasimhan and Shree K. Nayar. Vision and the atmosphere. *International Journal of Computer Vision*, 48(3):233–254, 2002a.

Srinivasa G. Narasimhan and Shree K. Nayar. Vision and the atmosphere. *International Journal*

of Computer Vision, 48(3):233–254, 2002b.

Srinivasa G. Narasimhan and Shree K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):713 – 724, 2003.

Shree K. Nayar and Srinivasa G. Narasimhan. Vision in bad weather. In *International Conference on Computer Vision*, 1999.

Shree K. Nayar, Vlad Branzoi, and Terry E. Boult. Programmable imaging using a digital micromirror array. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

Erik Oberg. *Machinery's Handbook*. Industrial Press, 28th edition, 2008.

Takeshi Ohtake. Preliminary observations on size distribution of snowflakes and raindrops at just above and below the melting layer. In *International Conference on Cloud Physics*, 1965.

C. Onof, R. E. Chandler, A. Kakou, P. Northrop, H. S. Wheater, and V. Isham. Rainfall modelling using Poisson-cluster processes: a review of developments. *Stochastic Environmental Research and Risk Assessment*, 14:384–411, 2000.

Oerlikon Optics. Oerlikon optics: For the first time, clear nighttime vision on the road. In *Oerlikon Press Releases*, May 23, 2007.

Chandrashekhar N. Padole and Vinay G. Vaidya. Image restoration using Wigner distribution for night vision system. In *Signal Processing*, 2008.

Karri Palovuori and Ismo Rakkolainen. US Patent 6,819,487: Method and apparatus for forming a projection screen or a projection volume.

E. Parker and P.R. Wallis. Three-dimensional cathode-ray tube displays. *Journal of the Institution of Electrical Engineers*, 95:371–389, 1948.

Stephen H. Pevnick. US Patent 4,294,406: Program controllable free falling water drop fountain. 1981.

H. R Pruppacher and R. L. Pitter. A semi-empirical determination of the shape of cloud and rain drops. *Journal of Atmospheric Sciences*, 28(1):86–94, 1971.

Hans R. Pruppacher and James D. Klett. *Microphysics of Clouds and Precipitation*. Kluwer Academic Publishers, second revised and enlarged edition, 1997.

Anna Puig-Centelles, Oscar Ripolles, and Miguel Chover. Creation and control of rain in virtual environments. *The Visual Computer*, 25(11):1037–1052, 2009.

Ismo Rakkolainen and Karri Palovuori. WAVE a walk-thru virtual environment. In *CD-Proc. Immersive Projection Technology Symposium*, 2002.

Ismo Rakkolainen and Karri Palovuori. Interactive digital fogscreen. In *Third Nordic conference on Human-computer interaction*, 2004.

Ismo Rakkolainen and Karri Palovuori. Laser scanning for the interactive walk-through fogscreen. In *ACM symposium on Virtual reality software and technology*, 2005.

Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, and Henry Fuchs. Multi-projector displays using camera-based registration. In *Proceedings of the conference on Visualization*, 1999.

R. H. Rasshofer and K. Gresser. Automotive radar and lidar systems for next generation driver assistance functions. *Advances in Radio Science*, 3:205209, 2005.

J. Rayner. Aquascript - information waterfall. 2007.

William T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, 1983.

D.C. Rogers. An observational study of aggregation. In *Proceedings of the International Conference on Cloud Physics*, 1974.

Justin Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, pages 14–20, 2008.

Martin Roser and Andreas Geiger. Video-based raindrop detection for improved image registration. In *IEEE 12th International Conference on Computer Vision Workshops*, 2009.

Peter Sand and Seth Teller. Particle video: long-range motion estimation using point trajectories. In *Computer Vision and Pattern Recognition*, 2006.

Conrad Sanderson and Kuldip K. Paliwal. Polynomial features for robust face authentication. In *IEEE International Conference on Image Processing (ICIP)*, 2002.

Yoav Y. Schechner, Srinivasa G. Narasimhan, and Shree K. Nayar. Instant dehazing of images using polarization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

Arno Schödl, Richard Szeliski, David Salesin, and Irfan Essa. Video textures. In *SIGGRAPH*, 2000.

M. Schönhuber, H.E. Urban, J.P.V. Poiares Baptista, W.L. Randeu, and W. Riedler. Measurements of precipitation characteristics by a new distrometer. In *Conf. on Atmospheric Physics and Dynamics in the Analysis and Prognosis of Precipitation Fields*, 1994.

M. Schönhuber, G. Lammer, and W. L. Randeu. One decade of imaging precipitation measurement by 2D-video-distrometer. *Advances in Geosciences*, 10:85–90, 2007.

E.D.R. Shearman, E.G. Hoare, and A. Hutton. Trials of automotive radar and lidar performance in road spray. *IEE Colloquium on Automotive Radar and Navigation Techniques (Ref. No. 1998/230)*, pages 10/1 – 10/7, 1998.

Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, 1994.

James A. Smith and Richard D. De Veaux. A stochasic model relating rainfall intensity to raindrop processes. *Water Resources Research*, 30(3):651–664, 1994.

A. F. Spilhaus. Raindrop size, shape and falling speed. *Journal of Atmospheric Sciences*, 5(3): 108–110, 1948.

Sonia Starik and Michael Werman. Simulation of rain in videos. In *International Workshop on Texture Analysis and Synthesis*, 2003.

Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, 1998.

Rahul Sukthankar, Robert G. Stockton, and Matthew D. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *International Conference on Computer Vision*, 2001.

Risa Suzuki, Taro Suzuki, Seiichi Ariga, Makoto Iida, and Chuichi Arakawa. "ephemeral melody": music played with wind and bubbles. In *SIGGRAPH Posters*, 2008.

Martin Szummer and Rosalind W. Picard. Temporal texture modeling. In *IEEE Intl. Conf. Image Processing*, volume 3, pages 823–826, September 1996.

Dharmpal Takhar, Jason Laska, Michael Wakin, Marco Duarte, Dror Baron, Shriram Sarvotham, Kevin Kelly, and Richard Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Proceedings of Computational Imaging IV at SPIE Electronic Imaging*, 2006.

Sarah Tariq. Rain. Technical report, Nvidia, 2007.

Natalya Tatarchuk and John Isidoro. Artist-directable real-time rain rendering in city environments. In *Eurographics Workshop on Natural Phenomena*, 2006.

Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, , Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, , Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, , Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661692, 2006.

M. Thurai, V. N. Bringi, and W. A. Petersen. Rain microstructure retrievals using 2-D video disdrometer and C-band polarimetric radar. *Advances in geosciences*, 20:13–18, 2009.

Ali Tokay and Kenneth Beard. A field study of raindrop oscillations. part I: Observation of size spectra and evaluation of oscillation causes. *Journal of Applied Meteorology*, 35:16711687, 1996.

Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, 1991.

Phillip H. S. Torr, Richard Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. In *International Conference on Computer Vision*, 1999.

Abhishek Kumar Tripathi and Sudipta Mukhopadhyay. A probabilistic approach for detection and removal of rain from videos. In *IETE Journal of Research*, 2011.

Joel Tropp, Michael Wakin, Marco Duarte, Dror Baron, and Richard Baraniuk. Random filters for compressive sampling and reconstruction. In *Proceedings of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2006.

R Uijenhoet, JNM Stricker, P Torfs, and JD Creutin. Towards a stochastic model of rainfall for radar hydrology: testing the poisson homogenity hypothesis. *Physica and Chemistry of the Earth (B)*, 24(6):747–755, 1999.

Remko Uijlenhoet. *Parameterization of rainfall microstructure for radar meteorology and hydrology*. PhD thesis, Wageningen University, 1999.

Carlton W. Ulbrich. Natural variations in the analytical form of the raindrop size distribution. *Journal of Applied Meteorology*, 22(10):1764–75, 1983.

Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt

McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William Red Whittaker, Ziv Wolkowicki, , Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, , Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425466, 2008.

Francis Vale. Cadillac night vision: Greed & speed are good. In *vxm*, 2001.

HC Van de Hulst. *Light Scattering by Small Particles*. John Wiley and Sons, NY, 1957.

Ting-i Wang, G. Lerfald, R. S. Lawrence, and S. F. Clifford. Measurement of rain parameters by optical scintillation. *Applied Optics*, 16(8):2236–2241, 1977.

Ting-i Wang, R. S. Lawrence, and M. K. Tsay. Optical rain gauge using a divergent beam. *Applied Optics*, 19(21):3617–3621, 1980.

Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, 2000.

J. Wenger. Automotive radar - status and perspectives. In *IEEE Compound Semiconductor Integrated Circuit Symposium*, 2005.

J. Wenger. Short range radar - being on the market. In *European Radar Conference*, 2007.

J. Wenger and S. Hahn. Long range and ultra-wideband short range automotive radar. In *IEEE International Conference on Ultra-Wideband*, 2007.

Glenn R. Widmann, Michele K. Daniels, Lisa Hamilton, Lawrence Humm, Bryan Riley, Jan K. Schiffmann, David E. Schnelker, and William H. Wishon. Comparison of lidar-based and radar-based adaptive cruise control systems. Technical report, Delphi Automotive Systems, 2000.

Paul T. Willis and John Hallett. Microphysical measurements from an aircraft ascending with a

growing isolated maritime cumulus tower. *Journal of Atmospheric Sciences*, 48(2):283–299, 1991.

Craig Wisneski, Hiroshi Ishii, Andrew Dahley, Matt Gorbet, Scott Brave, Brygg Ullmer, and Paul Yarin. Ambient displays: Turning architectural space into an interface between people and digital information. In *CoBuild*, 1998.

World Meteorological Organization. Climatological information. As of May 25, 2011.

John L. Wyatt Jr., Craig Keast, Mark Seidel, David Standley, Berthold Horn, Tom Knight, Charles Sodini, Hae-Seung Lee, and Tomaso Poggio. Analog VLSI systems for image acquisition and fast early vision processing. *International Journal of Computer Vision*, 8(3): 217–230, 1992.

Atsushi Yamashita, Yuu Tanaka, and Toru Kaneko. Removal of adherent waterdrops from images acquired with stereo camera. In *International Conference on Intelligent Robots and Systems*, 2005.

Atsushi Yamashita, Isao Fukuchi, Toru Kaneko, and Kenjiro T. Miura. Removal of adherent noises from image sequences by spatio-temporal image processing. In *IEEE International Conference on Robotics and Automation*, 2008.

MK Yau and RR Rogers. *A Short Course in Cloud Physics*. Butterworth-Heinemann, third edition, 1998.

Lihi Zelnik-Manor, Moshe Machline, and Michal Irani. Multi-body factorization with uncertainty: Revisiting motion consistency. *International Journal of Computer Vision*, 68(1):27–41, 2006.

Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *ICME*, 2006.