Spatio-Temporal Frequency Analysis for Removing Rain and Snow from Videos

Peter Barnum Takeo Kanade Srinivasa G Narasimhan
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, USA
{pbarnum,tk,srinivas}@cs.cmu.edu

Abstract

Capturing good videos outdoors can be challenging due to harsh lighting, unpredictable scene changes, and most relevant to this work, dynamic weather. Particulate weather, such as rain and snow, creates complex flickering effects that are irritating to people and confusing to vision algorithms. Although each raindrop or snowflake only affects a small number of pixels, collections of them have predictable global spatio-temporal properties. In this paper, we formulate a model of these global dynamic weather frequencies. To begin, we derive a physical model of raindrops and snowflakes that is used to determine the general shape and brightness of a single streak. This streak model is combined with the statistical properties of rain and snow, to determine how they effect the spatio-temporal frequencies of an image sequence. Once detected, these frequencies can then be suppressed. At a small scale, many things appear the same as rain and snow, but by treating them as global phenomena, we achieve better performance than with just a local analysis. We show the effectiveness of removal on a variety of complex video sequences.

1. Introduction

A movie captured on a day with rain or snow will have images covered with bright streaks from moving raindrops or snowflakes. Not only can they annoy or confuse a human viewer, but they degrade the effectiveness of any computer vision algorithm that depends on small features, such as feature point tracking or object recognition.

Fortunately, the effect of each particle is predictable. Several local removal methods that look at individual pixels or small spatio-temporal blocks have been proposed. But as with the aperture problem of stereo vision, just using local information is problematic. While it is true that rain and snow have a predictable local effect, many other things have a similar appearance, such as panning along a picket fence, viewing a referee's shirt during a football game, or viewing a mailbox during a snowstorm (Figure 1). Even a human viewing a scene with rain would have difficulty pointing out individual streaks, but rather would comment on its general, global properties. In this work, we demonstrate how to use such global properties over space and time to detect and remove rain and snow from videos.

(a) (b) (c)

Figure 1. (a) Part of one image from a video sequence with snow, (b) removing the snow with per-pixel temporal median filtering, and (c) removing by looking at the global frequencies. By examining global patterns, the snow can be removed while leaving the rest of the image unchanged.

1.1. Previous work on local methods

The first methods for removing dynamic weather used a temporal median filter for each pixel [12, 18]. This works because in moderate intensity storms, each pixel is clear more often than corrupted. However, any movement will cause blurring artifacts. Zhang et al [20] extended this idea by correcting for camera motion, although this is only effective when the video frames can be aligned and there are no moving objects in the scene.

Garg and Nayar [8] suggested modifying camera parameters during video acquisition, based on an explicit statistical model. 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, but will also cause unwanted blurring in scenes with moving objects or objects at different depths.

In a different paper, Garg and Nayar [7] suggested that each streak can be individually segmented by finding small blocks of pixels in individual streaks that change over space and time in the same way as rain. Searching for individual streaks can theoretically work for dynamic scenes with a moving camera. In practice, unless the rain is seen against a relatively textureless background, it is hard to segment streaks in this way.

1.2. Using global frequency information

Rain and snow change large regions of the image in a consistent and predictable way. In order to determine the influence of rain and snow on a video, we develop a model in frequency space, based on their physical and statistical characteristics.

The dynamics of falling particles are well understood, and it is simple to determine the general properties of the streak that a given raindrop or snowflake will create. The general shape and appearance of a streak can be approximated with a motion-blurred, circular Gaussian. The statistical characteristics of rain and snow have been studied in the atmospheric sciences, and 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 estimate the appearance of rain and snow in frequency space. This frequency model is fit to the image sequence, from which the rain and snow can be detected and removed.

Sequences with light precipitation or simple motion can be cleared up with many algorithms. To test the robustness of our work, we performed tests on several challenging sequences. Some have several moving objects, others have a cluttered foreground with high frequency textures, and all of them are taken with a moving camera. We present qualitative results of removal accuracy and demonstrate how the removal increases the performance of feature point tracking.

2. The properties of rain and snow streaks

In this section, we derive the distribution of the shapes and sizes of rain and snow streaks in videos. To determine the expected properties of the streaks in an image, we use the physical and statistical properties of raindrops and snowflakes. Later in Section 3.1, we present a frequency space analysis.

2.1. Imaging a single streak

Over a camera's integration time, raindrops and snowflakes move significantly and therefore appear as motion-blurred streaks. For a particle of a given size, speed, and distance from the camera, the length and breadth of the corresponding streak can be computed.

For common altitudes and temperatures, a raindrop's speed s can be predicted by a polynomial in its diameter a [6]

$$s(a) = -0.2 + 5.0a - 0.9a^2 + 0.1a^3 \tag{1}$$

Finding the speed of snowflakes is more difficult [14, 1], because of their complex shape. We find that even if the exact characteristics of the snow are unknown, for our removal algorithm, snowflakes can be assumed to fall half as fast as raindrops of similar size.

Now consider a camera imaging this particle. We assume that over the camera's integration time, the camera and the drop are moving with constant velocities, and the drop is at a uniform distance from the camera. Under these conditions, each drop will be imaged as a straight streak with constant breadth. The breadth b and length l of a streak caused by a drop of diameter a, that is z meters away from the camera, is given by

$$b(a, z) = \frac{af}{z} \tag{2}$$

$$l(a, z) = \frac{s(a)ef}{z} + \frac{af}{z} \tag{3}$$

where f is the focal length and e is the exposure time.

2.2. Statistical properties

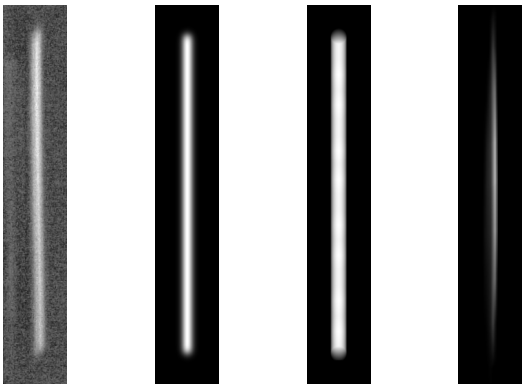
In a single storm, there will be raindrops and snowflakes of various sizes. Size distributions are commonly used for raindrops [15, 19, 4] and snowflakes [11, 16]. Previous works on rain removal [7, 8] have used the Marshall-Palmer [15] distribution.

Unfortunately, as discussed by Desaulniers-Soucy et. al [2, 3], raindrop size distributions are inaccurate. Nevertheless, they give useful general bounds. Drops rarely grow larger than $3mm$, and drops smaller than $.1mm$ cannot be seen individually. Using a uniform distribution between $.1mm$ and $3mm$ allows us to predict the general properties of precipitation. It only requires the estimation of the precipitation rate Λ.

$$N(a) \propto \begin{cases} \Lambda & .1 \leq a \leq 3mm \\ 0 & otherwise \end{cases} \tag{4}$$

3. Finding rain and snow in frequency space

Detecting a single rain or snow streak in an image is difficult. But a large group of streaks makes a predictable difference to an image sequence's spatio-temporal frequencies. Using the properties derived in Section 2, we can calculate how the Fourier transform of the rain and snow will appear.



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

Figure 2. (a) A streak from a real water drop (b) A blurred Gaussian (c) A streak from [9] with environmental lighting (d) A streak from [9] with point lighting. The blurred Gaussian is generally correct, while also being computationally efficient and well-behaved in frequency space.

3.1. A spatio-temporal model for streaks

To compute the frequencies of rain and snow, we need to know not just the length and breadth of a streak, but its complete appearance. Raindrops are mostly spherical. They are transparent and act as lenses, refracting light toward the viewer. As a result, they are generally brighter than other objects [7].

Snowflakes are also bright, for different reasons. As discussed by Koenderink and Richards [13], they appear bright because of the cloud layer's darkening effect, snow's reflectivity, and human's visual system's ability to see contrast.

As a bright sphere falls, it leaves a streak that is brightest in the center and dark toward the edges. The breadth of the streak is the diameter of the sphere projected onto the image, and the length can be calculated using equation 1. We approximate the image of a sphere of unknown color as a Gaussian. As it moves in space, the image it creates is just a linear motion blurred version of the original projection. A visual comparison of this model with others is shown in Figure 2. With environmental lighting from the sky, the appearance variation in streaks due to the drop oscillations discussed in [9] are subtle. Although blurred Gaussians are an inaccurate approximation of a raindrop illuminated with a point light source, most outdoor scenes are not lit with point sources, so the effects of oscillation can be ignored.

For horizontal and vertical frequencies u, v, a blurred Gaussian G with breadth b and length l is given by

$$G(u,v;b,l) = \int_0^l exp(-b^2(u^2+v^2))exp(2\pi iyv)dy$$
$$= \begin{cases} i\frac{exp(-b^2(u^2+v^2))(1-exp(2\pi ilv))}{2\pi v} & v \neq 0 \\ exp(-b^2u^2)l & v = 0 \end{cases} \quad (5)$$

In the notation, a semicolon is simply to differentiate between the parameters due to location versus the others. For example, $(u, v; b, l)$ means at location (u, v), with parameters (b, l). Since streaks are often not completely vertical, the coordinate space can be rotated, giving the more general form $G(u, v; b, l, \theta)$.

All of the components can now be combined to make a model of how rain and snow appear in frequency space, for spatial frequencies u and v, temporal frequency w, and a set of the different streak orientations $\hat{\theta}$. We do not need to consider raindrops or snowflakes that are extremely far from the camera and thus individually invisible. To speed up sampling, we specify a reasonable range of distances, $z_{min} = .5m$ to $z_{max} = 10m$.

The streak model can then be written as

$$R(u,v,w;\alpha,\hat{\theta}) = \alpha\ R(u,v,w;\hat{\theta}) \quad (6)$$

$$R(u,v,w;\hat{\theta}) = \sum_{\theta\in\hat{\theta}} \int_{.1}^{3} \int_{z_{min}}^{z_{max}} G(u,v;b(a,z),l(a,z),\theta)dz\ da \quad (7)$$

where the scaling factor α is given by the rainfall rate Λ and a normalization component.

$$\alpha = \frac{\Lambda}{\int_u \int_v \int_w R(u,v,w;\hat{\theta})dw\ dv\ du} \quad (8)$$

The amount of rain or snow is different between frames, but the exact change is more difficult to estimate than the expected amount in any one frame. Therefore, we model the change between frames by setting R to be constant in temporal frequency w.

3.2. Estimating model parameters

The model in Equation 6 depends on two physical quantities: the precipitation rate α and the streak orientation θ. In this section, we describe techniques to estimate these quantities from an input video.

A movie m, acquired in rain, can be decomposed into two components: a clean image c and a rain/snow component r.

$$m(x,y,t) = c(x,y,t) + r(x,y,t) \quad (9)$$

We only have m from which to estimate α and θ, but only a single precipitation rate and streak orientation needs to estimated per frame.

The streak orientation can be computed, if there is a short segment of the video where most of the change is due to rain or snow (i.e. c is mostly constant with respect to time). A rough estimate $\tilde{R}$ of the frequencies can be obtained by computing the standard deviation over time for each spatial

frequency, for T frames.

$$\tilde{R}(u,v) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(||M(u,v,t)|| - ||\overline{M}(u,v)||)^2} \quad (10)$$

where $M(u,v,t)$ is the 2D Fourier transform of $m(x,y,t)$ and $\overline{M}$ is the mean across time.

The correct θ is found by minimizing the following equation, for all $w \neq 0$.

$$\underset{\theta}{\operatorname{argmin}} \int_{u,v,w} (||R(u,v,w;\theta)|| - \tilde{R}(u,v))^2 du\ dv\ dw \quad (11)$$

An exhaustive search in this space can be performed. Rain generally falls in almost the same directions, so only one θ is needed. But since snow has a less consistent pattern, a set of orientations $\hat{\theta}$ centered around θ is used (a set of θs $\pm.1$ radians about the mean is effective for most videos).

A rough estimate of the precipitation rate is found by taking a ratio of the median of all frequencies, again where $w \neq 0$.

$$\alpha \approx \frac{median(|M(u,v,w)|)}{median(|R(u,v,w;\hat{\theta})|)} \quad (12)$$

Because rain and snow cover such a broad part of the frequency space, the median of all frequencies is a robust estimate of α.

4. Removing rain and snow in image space

We are now ready to find rain and snow in image space. The removal step has similarities to notch filtering, as used in many image processing applications [10]. In notch filtering, frequencies corresponding to repetitive structures such as lines or grids are suppressed or eliminated.

Intuitively, we want to suppress unwanted frequencies yet keep those that are due to objects in the scene. If the model predicts that a given frequency should be much lower magnitude, then it is probably cluttered with non-rain. The frequencies that are definitely rain should be removed first, as estimated by the ratio of the predicted value to the true value. This can be done for individual frames, but since rain is dynamic, the best performance comes from performing a 3D Fourier transform on blocks of consecutive images.

Similar to [20], basic image alignment is performed as a preliminary step. Since we are interested in sequences with moving objects and a lot of rain and snow, it is difficult to align exactly without human intervention. However, we can automatically find strong corners, then eliminate poor matches with RANSAC [5] on the fundamental matrix. Even with heavy noise, pure translational motion can corrected, even when a complete alignment is difficult.

The removal is a two-step process. First, each image is considered independently. To denote the probability of rain at a given (x,y,t) or (u,v,w), we use an upper-case P for probability in Fourier space and a lower case one in image space. Each equation is for the range of frames from times $t-k$ to $t+k$. $p(rain|k)$ is used as a short form.

$$p(rain|k) \equiv p(rain|x,y,t,\{m(x,y,\tau) : t-k \leq \tau \leq t+k\}) \quad (13)$$

$$P(rain|k) \equiv P(rain|u,v,w,\{m(x,y,\tau) : t-k \leq \tau \leq t+k\}) \quad (14)$$

For a single image, the probability that a given pixel is rain is

$$p(rain|k=0) = \mathcal{F}^{-1}\{\frac{|R(u,v;\alpha,\hat{\theta})|}{|M(u,v)|} exp(i\phi\{M(u,v)\})\} \quad (15)$$

The right hand side needs be normalized from zero to one.

Second, we examine several consecutive single frame estimates together. From the single frame removal, we have a probability for each pixel of each frame. This is the same as having a video with pixels of values from zero to one. The same probability calculation is now run using temporal information. In this case, a 3D Fourier transform is performed on blocks of consecutive frames, to get a refined estimate.

$$p(rain|k=1) = \mathcal{F}^{-1}\{\frac{|R(u,v,w;\alpha,\hat{\theta})|}{|P(rain|k=0)|} exp(i\phi\{P(rain|k=0)\})\} \quad (16)$$

Note that $P(rain|k=0)$ is the 3D transform of several per-frame estimates. And as before, the right hand side is normalized.

If the pixel values in the original image are scaled from zero to one, $p(rain|k=1)$ is the amount to subtract from each pixel, and the clear video estimate $c(x,y,t)$ is

$$c(x,y,t) = m(x,y,t) - p(rain|k=1) \quad (17)$$

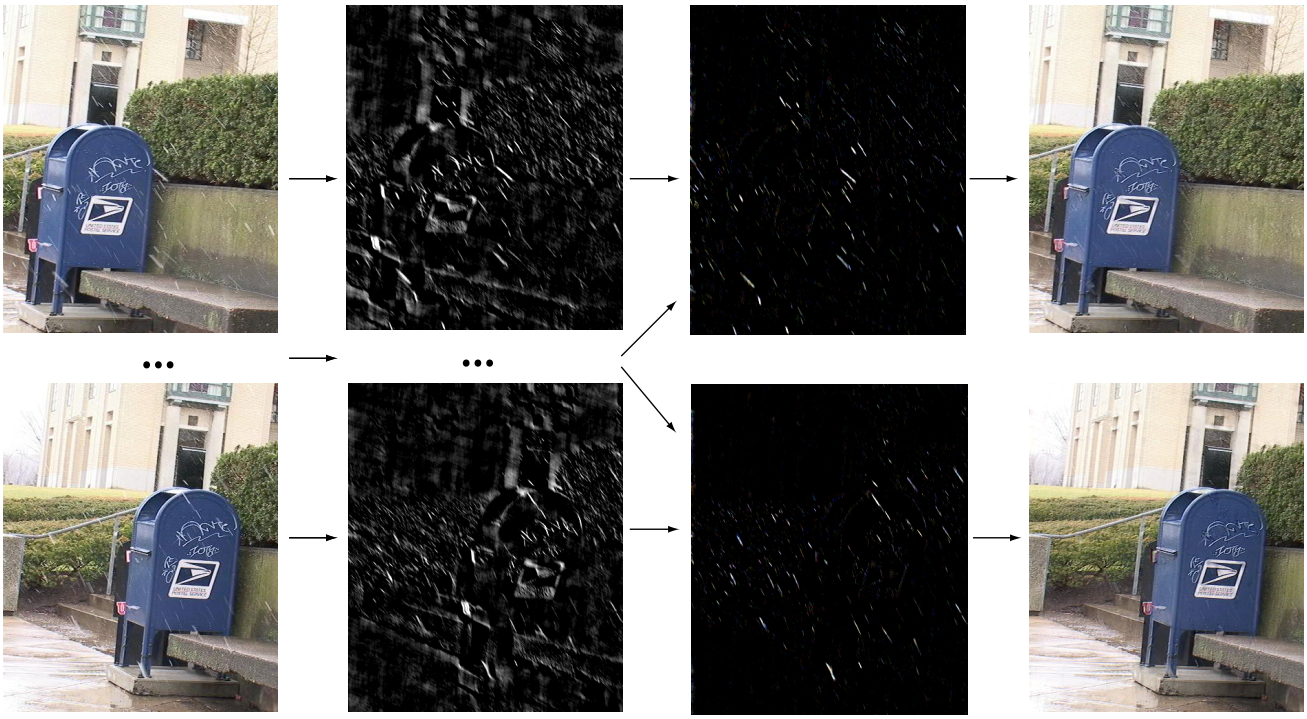
Alternately, p can be used as a mixing weight for any simple estimate of the clear video $\tilde{c}$, such as a temporal mean or median.

$$c(x,y,t) = (1-p(rain))m(x,y,t) + p(rain)\tilde{c}(x,y,t) \quad (18)$$

With either method, the process is iterated until the video is clean. Figure 3 illustrates the two-step estimation.

5. Results and applications

We have selected several rain and snow filled sequences, each with a combination of natural and artificial objects, and taken with a moving camera. Not only are they difficult for image-based algorithms, but due to the wide variety of



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

Figure 3. An example of snow removal. The original image (a) and after per-frame removal (b). The snow is found, but there are misclassifications. With temporal information, an improved estimate is obtained (c). The result after 3 removal iterations is shown in (d).

low and high frequency textures, the frequency domain is cluttered as well.

The removal process will often work for snow as well. During very windy days or if the snowflakes are large, the streaks will be oriented in many directions and cannot be segmented by our algorithm. However, if the snow moves in a fairly consistent direction, then the same technique can be used, by modifying the model to predict that snowflakes move half the speed of similar sized raindrops.

A few frames and difference images are shown in Figures 5-8. (Note that the images have been brightened, so that details can be seen when this paper is printed.)

We use two methods to evaluate the removal quantitatively. The first is to compare the overall amount of rain removed versus background incorrectly removed. The second is to evaluate the accuracy of feature point tracking.

Removal accuracy: Since rain has similar spatio-temporal frequencies as certain other objects, we cannot remove rain without partially disrupting the rest of the image. However, we can iterate for different amounts of time, to achieve different levels of rain-removal to background-corruption. We use rain rendered by the photorealistic process of Garg and Nayar [9] as ground truth. We use two sequences, one of 230 frames from a 503x376 sequence with a moving car and a panning camera, and one of 120 frames from a 504x400 sequence with a camera zooming in on a window. In each video, the shutter speed is assumed to be one sixtieth of a second.

The total amount of rain or non-rain is defined as the sum of pixel intensity across all images and color channels. To measure removal accuracy, we compute the difference D between the true rain component r and the estimate obtained from our removal algorithm $m-c$. The error is given as the percentage of rain that is not removed H and the non-rain that is incorrectly removed E.

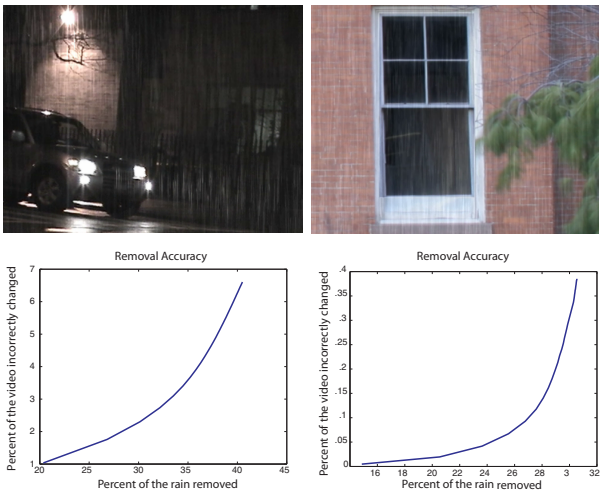
$$D(x,y,t) = r(x,y,t) - (m(x,y,t) - c(x,y,t)) \quad (19)$$

$$H = 100 \sum_{x,y,t} \frac{\{D(x,y,t) : D(x,y,t) > 0\}}{r(x,y,t)} \quad (20)$$

$$E = 100 \sum_{x,y,t} \frac{\{D(x,y,t) : D(x,y,t) < 0\}}{c(x,y,t)} \quad (21)$$

Figure 4 shows an example frame for each sequence, and a curve comparing the percentage of the rain removed H to the percentage of the background corrupted E. Although only some of the rain is removed, there is very little accidental removal.

Improvements in feature point tracking: Rain and snow can degrade any vision algorithm that depends on small features or high frequencies, such as feature point tracking, unstructured noise removal, and edge detection.



(a) Driving car (b) Zooming in on a window

Figure 4. Quantitative results on two sequences of rendered rain from [9]. An example frame and the removal accuracy of our method is shown for each of the two videos. In each of the two cases, it is possible to remove much of the rain with only a slight change to the image.

To demonstrate the importance of removing rain and snow, we compare the results of feature point tracking between several weather-corrupted sequences and their corresponding de-weathered versions. We use the same evaluation method as [17], which is to track points while each sequence is played forward then backwards. Because each 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 each sequence, the number of points correctly tracked within five pixels is mentioned in the captions of Figures 5-8.

In general, the strongest feature points can be tracked even in the presence of heavy rain. It may not be necessary to always remove rain in scenes with many strong corners, such as buildings in dense urban areas. But for many cases, these results have implications for not just point tracking, but for any algorithm that uses small details.

6. Summary

We have demonstrated a method for globally detecting and removing rain and snow, by using a physical and statistical model to suppress certain spatio-temporal frequencies. On several challenging sequences, we show that rain and snow can be reduced by looking at their global properties. 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.

7. Acknowledgements

This work is supported in part by the National Science Foundation under Grant No. EEEC-0540865, NSF CAREER Award No. IIS-0643628, NSF Award No. CCF-0541307, ONR Award No. N00014-05-1-0188, and Denso Corporation. The authors thank Kshitiz Garg and Shree Nayar for the rendered videos of rain.

References

[1] H. P. Böhm. A general equation for the terminal fall speed of solid hydrometeors. *J. of the Atmospheric Sciences*, 46:2419–27, 1989.

[2] N. Desaulniers-Soucy. *Empirical test of the multifractal continuum limit in rain*. PhD thesis, Mcgill, 1999.

[3] N. Desaulniers-Soucy, S. Lovejoy, and D. Schertzer. The hydrop experiment: an empirical method for the determination of the continuum limit in rain. *Atmospheric Research*, 59-60:163–197, 2001.

[4] G. Feingold and Z. Levin. The lognormal fit to raindrop spectra from frontal convective clouds in Israel. *J. of Climate and Applied Meteorology*, 25:1346–63, 1986.

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

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

[7] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *CVPR*, 2004.

[8] K. Garg and S. K. Nayar. When does a camera see rain? In *ICCV*, 2005.

[9] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *SIGGRAPH*, 2006.

[10] R. C. Gonzalez and R. E. Woods. *Digital Image Processing 2nd Edition*. Prentice Hall, 2002.

[11] K. Gunn and J. Marshall. The distribution with size of aggregate snowflakes. *J. of Meteorology*, 15:452–61, 1958.

[12] H. Hase, K. Miyake, and M. Yoneda. Real-time snowfall noise elimination. In *ICIP*, 1999.

[13] J. J. Koenderink and W. A. Richards. Why is snow so bright? *J. of the Optical Society of America*, 9(5):643–8, 1992.

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

[15] J. Marshall and W. Palmer. The distribution of raindrops with size. *J. of Meteorology*, 5:165–6, 1948.

[16] T. Ohtake. Preliminary observations on size distribution of snowflakes and raindrops at just above and below the melting layer. In *Int'l. Conference on Cloud Physics*, 1965.

[17] P. Sand and S. Teller. Particle video: long-range motion estimation using point trajectories. In *CVPR*, 2006.

[18] S. Starik and M. Werman. Simulation of rain in videos. In *Int'l. Workshop on Texture Analysis and Synthesis*, 2003.

[19] C. W. Ulbrich. Natural variations in the analytical form of the raindrop size distribution. *J. of Applied Meteorology*, 22(10):1764–75, 1983.

[20] X. Zhang, H. Li, Y. Qi, W. Kheng, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *ICME*, 2006.

Figure 5. 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. In the original sequence, 81 feature points are tracked correctly, compared with 102 in the removed version.

Figure 6. 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 their legs are misclassified. Due to the pedestrians moving across the lower part of the video, only points in the upper half can be tracked correctly, with 54 correct in the original and 77 in the removed version.

Figure 7. 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. Because the streaks are so large and bright, even some strong corners are mistracked, allowing for 69 points to be tracked in the removed version, compared to 58 in the original.

Figure 8. 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. But even with the errors, point tracking is improved, from 144 to 168.