Multiple View Image Reconstruction: A Harmonic Approach

Justin Domke and Yiannis Aloimonos
Center for Automation Research, Department of Computer Science
University of Maryland, College Park, MD, 20742, USA
www.cs.umd.edu/users/domke

Abstract

This paper presents a new constraint connecting the signals in multiple views of a surface. The constraint arises from a harmonic analysis of the geometry of the imaging process and it gives rise to a new technique for multiple view image reconstruction. Given several views of a surface from different positions, fundamentally different information is present in each image, owing to the fact that cameras measure the incoming light only after the application of a low-pass filter. Our analysis shows how the geometry of the imaging is connected to this filtering. This leads to a technique for constructing a single output image containing all the information present in the input images.

1. Introduction

The light field entering a camera is, in general, highly discontinuous. This signal can be said to contain an "infinite" amount of information. All cameras, however, can only measure a finite amount of information. To cope with this, good cameras low-pass filter, or "blur" the incoming light before measuring it. This is sometimes called "anti-aliasing". Note that this filtering is a continuous process that takes place in the optics of the camera. [1]

Now, suppose we have two images of a surface, taken from different positions. If the surface is Lambertian, the discontinuous light fields entering the camera at each position contain the same signals, merely warped depending on the geometry. However, the real, measured signals (after low-pass filtering) are quite different, even accounting for the warping. To warp and then filter is not the same as to filter and then warp.

This effect can be understood in a different way. Take two images of a surface. The incoming light (the "ideal images") contains the same underlying signal. Hence the real, observed images are warped versions of the same (ideal) image, filtered with an identical low-pass filter. We will show in Section 2 that they can be thought of instead as the same (ideal) image, convolved with different, warped versions of the low-pass filter. This insight makes an analysis of these effects much easier.

In Section 3 we will show how this can be understood in the frequency domain. Depending on the effective filtering in each image, different frequencies are attenuated to different degrees. This leads to a simple idea for multiple view image reconstruction- to create an output image, just select each frequency from the input image in which it was least filtered. This leads to several technical problems- what range of frequencies to take from each image, and how to "pull out" a specific range of frequencies from a given image. These problems are addressed in Sections 4 and 5.

Finally, in Section 6 we summarize the algorithm for reconstruction in the case of affine transformations. This is then generalized to arbitrary, continuous transformations in Section 6. Ultimately, we will find that one can reconstruct an output image simply as a sum of the input images (warped to a common frame) convolved with different filters. Fig. 1 summarizes the approach.

1.1. Previous Work

The most closely related previous work is by Wang et al. [6]. There, the authors model the process of creating one image from another through the steps of reconstructing the continuous signal, warping, prefiltering, and resampling. The image filtering process is modeled as an ideal low-pass filter. If the columns of images are stacked together as vectors, this ultimately leads to a numerical linear constraint on the desired image with respect to each observed image. The desired image is then obtained using least squares to find a solution approximately satisfying each of these constraints.

Another related area is super-resolution [5]. In super-resolution, the input images are generally taken from nearby positions, and the goal is to take advantage of aliasing effects to construct a higher-resolution image. Unlike this related work, our method makes no attempt to invert the

[1] If the low-pass filter removes everything above the Nyquist frequency, the continuous signal can be reconstructed from discrete samples. This paper will use continuous mathematics- for standard, discretely sampled images, this can be thought of as referring to the reconstructed signal.

1-4244-1180-7/07/$25.00 ©2007 IEEE

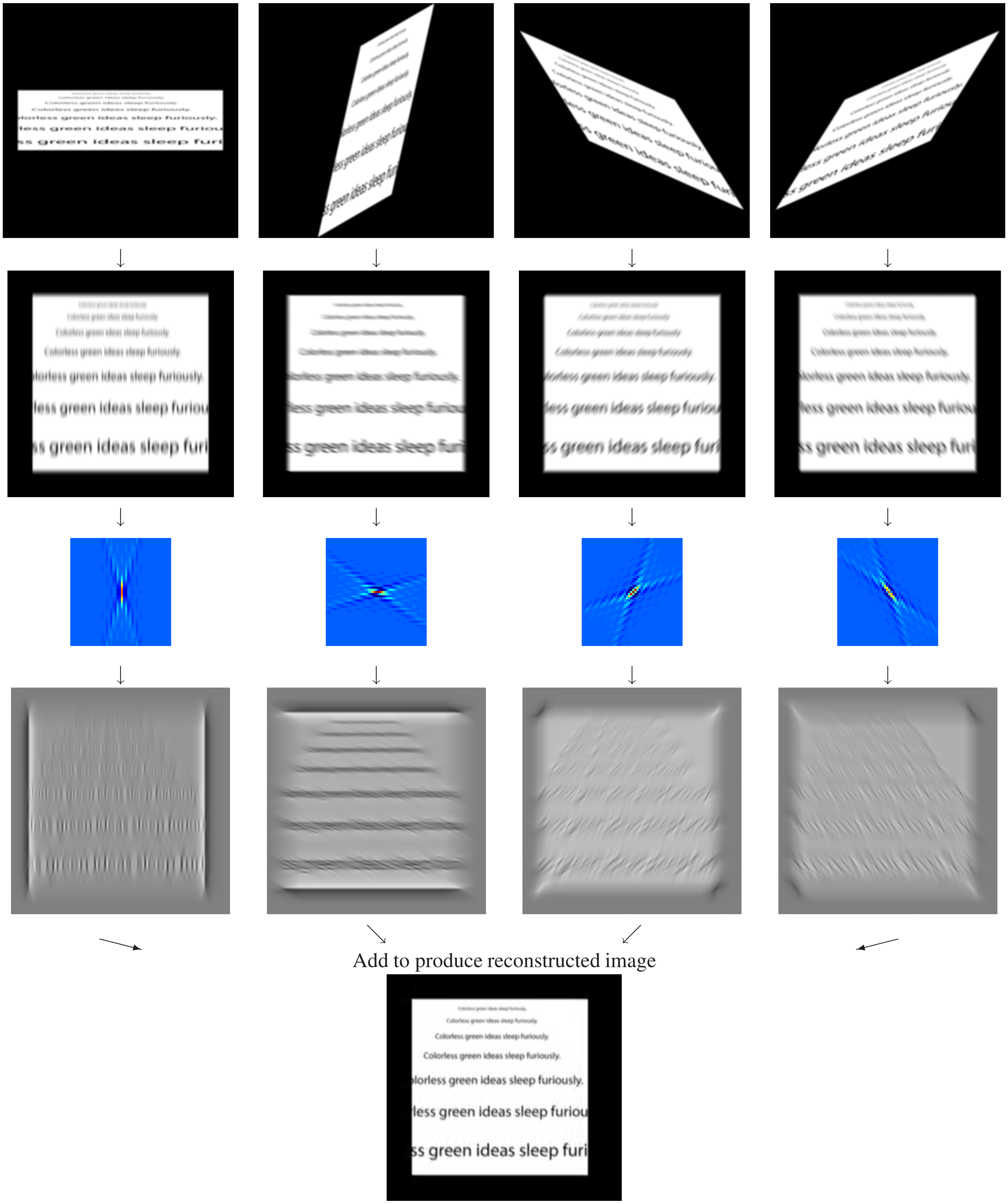


Figure 1. A graphical summary of the method for the affine case. Top row: Input images. Second row: Images warped to a common coordinate system. Third row: Filters computed for each image. Fourth row: Result of convolution of filters with images from the second row. Bottom: Reconstructed image. For non-affine motion, the only change is that the filters become spatially variant.

original blurring kernel. Instead, here we essentially just combine the information in the different input images into one reconstructed image. It turns out that this can still be done, with out modeling the blurring process. Thus, this method would offer little improvement when the input images are nearby. The point-spread function in real cameras is difficult to estimate, and does not appear to exactly obey idealizations like Gaussian or ideal low-pass [4]. If the blurring kernel is modeled, better results (perhaps greatly better) could be obtained by fusing a deconvolution strategy with the constraint we use here.

Other work based on similar principles is in the graphics literature for texture mapping. Greene and Heckbert proposed the elliptical weighted average (EWA) filter [3]. Essentially, a circular filter in the image space is projected to the texture space, resulting in an elliptical filter. Similar principles are used here.

2. Theory

2.1. Notation

Given some function f, and a matrix A, we will use the notation f^A to represent f, warped under the motion A. That is, f^A is the function such that,

$$\forall \mathbf{x}, f^A(\mathbf{x}) = f(A\mathbf{x}). \tag{1}$$

We will use lowercase letters (i, or j) to represent functions in the spatial domain, and uppercase letters (I, or J) for the frequency domain. Boldface letters ($\mathbf{x}$) represent vectors.

2.2. Derivation

Suppose that, if $\mathbf{x}_1$ in the first image corresponds to $\mathbf{x}_2$ in the second image, then $\mathbf{x}_2 = A\mathbf{x}_1$. So if i_1 is a function representing the first "ideal" image, and i_2 is a function representing the second,

$$\mathbf{x}_2 = A\mathbf{x}_1 \implies i_2(\mathbf{x}_2) = i_1(\mathbf{x}_1). \tag{2}$$

Hence, for all $\mathbf{x}$,

$$i_2(\mathbf{x}) = i_1^{A^{-1}}(\mathbf{x}). \tag{3}$$

i_1 and i_2 should be thought of as essentially arbitrary real functions. Now, the fundamental assumption here is that we do not observe these ideal images. One can only measure the ideal image after convolution with some low-pass filter, e. Let $j_1$and j_2 denote the real, measured images.

$$j_1(\mathbf{x}) = [i_1 * e](\mathbf{x}) \tag{4}$$

$$j_2(\mathbf{x}) = [i_2 * e](\mathbf{x}) \tag{5}$$

Now, use Eqn. 3 to change the expression for j_2, to make its relationship to j_1 more explicit.

$$j_2(\mathbf{x}) = [i_1^{A^{-1}} * e](\mathbf{x}) \tag{6}$$

Now, we will use the following Theorem. (see Appendix)

$$[f^A * g](\mathbf{x}) = \frac{1}{|A|}[f * g^{A^{-1}}](A\mathbf{x}) \tag{7}$$

Applying this, we have,

$$j_2(\mathbf{x}) = \frac{1}{|A^{-1}|}[i_1 * e^A](A^{-1}\mathbf{x}). \tag{8}$$

So, finally,

$$\boxed{j_2^A(\mathbf{x}) = |A|[i_1 * e^A](x)} \tag{9}$$

Contrast this with the expression for j_1 in Eqn. 4. It is as if, instead of being filtered with e, it was filtered with $|A|e^A$.

This simple result has important implications- if we have multiple views of some surface, even after we warp the images into a common coordinate system, different signals are present. However, these signals can be understood simply as the result of the *same* ideal image, convolved with warped versions of the same filter.

This is most intuitive in the frequency domain. Let E denote the Fourier transform of e, J_1that of j_1, and so on. Then, by the convolution theorem,

$$J_1(\mathbf{u}) = [I_1 \cdot E](\mathbf{u}) \tag{10}$$

The analogous equations is also true for J_2, but we will continue our analysis from Eqn. 9. We will use the following theorem, which is essentially a special case of the Affine Fourier Theorem [1] [2].

$$\text{If } \mathcal{F}\{f(\mathbf{x})\} = F(\mathbf{u}) \text{ , then } \mathcal{F}\{f^A(\mathbf{x})\} = \frac{1}{|A|}F^{A^{-T}}(\mathbf{u}) \tag{11}$$

Now apply the theorem to e^A.

$$\mathcal{F}\{e^A(\mathbf{x})\} = \frac{1}{|A|}E^{A^{-T}}(\mathbf{u}) \tag{12}$$

So we can clearly see the contrast between the Fourier transform of the first image, and the Fourier transform of the warped version of the second image

$$\boxed{\mathcal{F}\{j_2^A(\mathbf{x})\} = [I_1 \cdot E^{A^{-T}}](\mathbf{u})} \tag{13}$$

In the frequency domain, it is as if, rather than being filtered with E, it was filtered with $E^{A^{-T}}$.

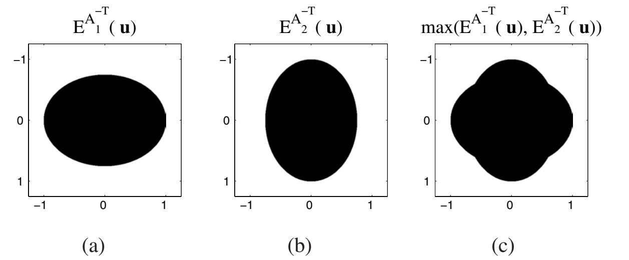


Figure 2. Different filters in the frequency domain.

3. Intuition

Suppose we have many views of a surface, warped under different motions A_i. Suppose also that E is a pure low-pass filter. (This is only for the purpose of explanation. We assume only that E is a symmetric, decreasing function.)

This situation is pictured in Fig. 2 for the two motions,

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & .75 \end{bmatrix}, A_2 = \begin{bmatrix} .75 & 0 \\ 0 & 1 \end{bmatrix}.$$

It is natural to think that if we have these two images, we could reconstruct an image which contains the best frequency information from each. This is shown in part (c).

Now, for images with affine motion, the following simple algorithm will perform multiple view reconstruction. (We emphasize that this is only for the sake of explanation- the results in this paper do not use this algorithm.)

1. Input a set of images, $j_1, j_2, ...\ j_n$, and a corresponding set of motions, $A_1, A_2, ...\ A_n$
2. Warp each image to the central coordinate system, to obtain $j_i^{A_i}(\mathbf{x})$.
3. For each image, j_i compute the Fourier Transform of the warped image, $\mathcal{F}\{j_i^{A_i}\}(\mathbf{u}) = [I \cdot E^{A_i^{-T}}](\mathbf{u})$.
4. Create the reconstructed image in the Fourier domain. For all $\mathbf{u}$, set $K(\mathbf{u}) = \mathcal{F}\{j_l^{A_l}\}(\mathbf{u})$, where $l = \arg\min_i |A_i^{-T}\mathbf{u}|$.
5. Output the image in the spatial domain. $k(\mathbf{x}) = \mathcal{F}^{-1}\{K\}$.

Steps 1-3 simply create the frequency domain representations of the input images in a common coordinate system. To understand step 4, notice that for each frequency, we would like to take it from each image in which it has been filtered least. Now, since we assume that $E(\mathbf{u})$ is a monotonically decreasing function of $|\mathbf{u}|$ only, we would like to select the motion for which $|A_i^{-T}\mathbf{u}|$ is least.

However, this simple algorithm could only work when the global motion is affine. We will present a different algorithm that operates completely in the spatial domain. We will later show that this allows us to do reconstruction for transformations that are only *locally* affine- for example full projective transformations. Nevertheless, the algorithm operates on the same principles as the simple one above. Before we can present the algorithm, we need to develop two tools.

4. The Frequency Slice Filter

Here, the goal is to create a filter that will pass a certain range of frequencies. More specifically, given $\theta_1$and θ_2, we would like a filter c such that, in the frequency domain (Fig. 3(a)), (where $\angle\mathbf{u}$ denotes the "angle" of $\mathbf{u}$: if $\mathbf{u} = r[\cos\theta, \sin\theta]^T$, then $\angle\mathbf{u} = \theta$)

$$C_{\theta_1,\theta_2}(\mathbf{u}) = \begin{cases} 1 & \theta_1 \leq \angle\mathbf{u} \leq \theta_2 \\ 0 & \text{else} \end{cases} \tag{14}$$

First, suppose we would like to create a filter that passes exactly those frequencies in the first and third quadrants. (That is, $\theta_1 = 0$, $\theta_2 = \pi/2$.) Naively, we would just plug these values into the above equation. However, if we do this, the inverse Fourier Transform will not converge. A convenient fact is useful here- the images have already been low-pass filtered. Hence, it does not matter what the filter does to very high frequencies. So, instead, we will define the following filter, cutting off higher frequencies. (Fig. 3(b))

$$C_{0,\pi/2,r}(\mathbf{u}) = \begin{cases} 1 & 0 \leq \angle\mathbf{u} \leq \pi/2, \text{ and } |u| \leq r \\ 0 & \text{else} \end{cases} \tag{15}$$

The filters corresponding to different r will always be different. Nevertheless, once r is sufficiently high (above the highest frequency present in an image), the *result* of applying the different filters to an image will be the same.

Before extending this to the case of other θ_1, θ_2, we will find the inverse Fourier transform.

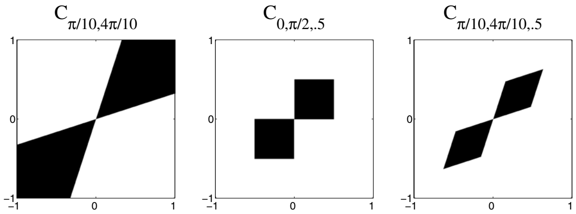
$$c_{0,\pi/2,r}(\mathbf{x}) = \mathcal{F}^{-1}\{C_{0,\pi/2,r}(\mathbf{u})\} \tag{16}$$

$$= \int_{\mathbf{u}} C_{0,\pi/2,r}(\mathbf{u}) \exp(2\pi i \mathbf{u}^T\mathbf{x}) d\mathbf{u} \tag{17}$$

$$= \int_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{r}} [\exp(2\pi i \mathbf{u}^T\mathbf{x}) + \exp(-2\pi i \mathbf{u}^T\mathbf{x})] d\mathbf{u} \tag{18}$$

$$= \int_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{r}} 2\cos(2\pi\mathbf{u}^T\mathbf{x}) d\mathbf{u} \tag{19}$$

$$\boxed{\begin{array}{l} c_{0,\pi/2,r}(\mathbf{x}) = \\ \frac{1}{2\pi^2 xy}[\cos(2\pi r x) + \cos(2\pi r y) - \cos(2\pi r(x+y)) - 1] \end{array}} \tag{20}$$



(a) (b) (c)

Figure 3. The frequency slice filter in the frequency domain.

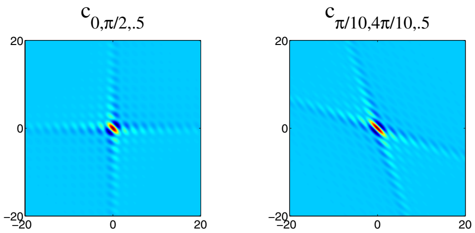


Figure 4. The frequency slice filter in the spatial domain.

Examples are shown in Fig. 4. Notice that this function has linear decay in the spatial domain. This is a potential problem, since the filter will need to be large to capture the entire profile. In practice, however, we have found that this was not a problem. We used matrices of size 51 by 51 to represent the filters, and found that using larger sizes resulted in little change to the output. This will be important later, because in projective reconstruction, the transformation must be close to affine in a window the size of this filter. Regardless, it would certainly be better to use a filter with faster decay. This is left as future work.

Now, to define the filter for arbitrary angles, we will again use the special case of the Affine Fourier Theorem, as in Eqn. 11. Given θ_1, and θ_2, define the following matrix:

$$V = \begin{bmatrix} \cos\theta_1 & \cos\theta_2 \\ \sin\theta_1 & \sin\theta_2 \end{bmatrix} \tag{21}$$

Now, we can define the frequency slice filter for arbitrary angles.

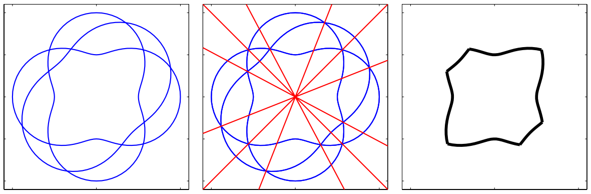
$$\boxed{c_{\theta_1,\theta_2,r}(\mathbf{x}) = |V| c_{0,\pi/2,r}^{V^T}(\mathbf{x})} \tag{22}$$

To see this, apply Eqn. 11 to the right hand side

$$\mathcal{F}\{|V| c_{0,\pi/2,r}^{V^T}(\mathbf{x})\} = |V| \cdot |V^{-T}| C_{0,\pi/2,r}^{(V^T)^{-T}}(\mathbf{u}) \tag{23}$$

$$\mathcal{F}\{|V| c_{0,\pi/2,r}^{V^T}(\mathbf{x})\} = C_{0,\pi/2,r}^{V^{-1}}(\mathbf{u}) \tag{24}$$

To understand the presence of V^{-1}, notice that it will send $[\cos\theta_1, \sin\theta_1]^T$ to $[1,0]^T$, and $[\cos\theta_2, \sin\theta_2]^T$ to $[0,1]^T$. Hence only those frequencies in the correct range of angles will be passed. Fig. 3 (c) shows an example with $\theta_1 = \pi/10, \theta_2 = 4\pi/10$.



(a) (b) (c)

Figure 5. Frequency Segmentation.

5. Partitioning Frequency Space

Given the frequency slice filters, we are nearly prepared to do multiple view reconstruction for affine transformations. The main remaining problem is the following: Suppose we are given a set of images warped to a common frame, $\{j_i^{A_i}(\mathbf{x})\}$, as well as the set of motions producing those images, $\{A_i\}$. For a given frequency $\mathbf{u}$ having angle θ, in which image is that frequency least filtered? (Notice that this will depend only on the angle of $\mathbf{u}$, and not on its magnitude.)

Recall from above that

$$\mathcal{F}\{j_i^{A_i}(\mathbf{x})\} = [I \cdot E^{A_i^{-T}}](\mathbf{u}).$$

So, for each angle θ, we would like to choose a motion A_i, such that $E(A_i^{-T}[\cos\theta, \sin\theta]^T)$ is maximum. This will be the case when $|A_i^{-T}[\cos\theta, \sin\theta]^T|$ is minimum. We can picture the situation by drawing a curve, where for each angle θ, we use the length $|A_i^{-T}[\cos\theta, \sin\theta]^T|$. (Fig 5 (a).) To "carve up" the space of frequencies we need two steps.

1. For each pair of motions A_i, and A_j, find points for which the curves meet. That is, find $\mathbf{u}$ such that

$$|A_i^{-T}\mathbf{u}| = |A_j^{-T}\mathbf{u}|.$$

$$\mathbf{u}^T(A_i^{-1}A_i^{-T} - A_j^{-1}A_j^{-T})\mathbf{u} = 0$$

   Notice that this does not depend on the magnitude of $\mathbf{u}$. If we assume that either the first or second component of $\mathbf{u}$ is nonzero, this can be solved by setting $\mathbf{u} = [u_x, 1]^T$, or $\mathbf{u} = [1, u_y]^T$, and solving a quadratic equation. Complex values as a solution indicate that the two curves do not intersect.

2. Find the angles of all the points $\mathbf{u}$ found in the previous step and sort them. Now, form pairs from all adjacent angles. This results in a sequence of pairs of angles, $<\theta_1, \theta_2>, <\theta_2, \theta_3> \ldots <\theta_{m-1}, \theta_m>$. It now remains to find the motion that has the smallest value

in each region. The simplest procedure is simply to test all motions and find,

$$\arg\min_i |A_i^{-T} \begin{bmatrix} \cos(.5(\theta_j + \theta_{j+1}) \\ \sin(.5(\theta_j + \theta_{j+1})) \end{bmatrix}|. \tag{25}$$

This works, but has a worst-case time complexity of $O(n^3)$, where n is the number of input images. However, an optimization can reduce this to $O(n^2 \log n)$: for each angle θ_j keep track of the motions whose intersection produced that angle. Then, if some motion A_i is minimum in the region $< \theta_{j-1}, \theta_j >$, A_i will also be the minimum in the region $< \theta_j, \theta_{j+1} >$, unless θ_j was produced by the intersection of A_i with some other motion A_k, and A_k is "smaller" than A_i in the sense of Eqn. 25. This means that we only need to test at most two motions in each region.

The frequency segmentation process is illustrated in Fig. 5. (a) shows the magnitude of $|A_i^{-T}[\cos\theta, \sin\theta]^T|$, for each motion A_i, and each angle θ. In (b), the angles are found where the motions "intersect". In (c), the shape is shown, where for each angle, the magnitude is taken from the best motion.

6. Affine Reconstruction

Algorithm 1 summarizes the method for reconstruction of images warped under affine transformations. The method is fairly simple- first the frequency space is partitioned, then the final image is reconstructed as the sum of the input images convolved with different filters. Notice that this algorithm takes place entirely in the spatial domain.

In our implementation, the convolution in step 4 takes place discretely, rather than as an integration. For the results shown in this paper, we use $r = .3$. This is high enough to avoid filtering out frequencies that are present in the input images, but low enough to allow for the filters to be reasonably approximated with discrete samples.

Algorithm 1 Affine Reconstruction

```
1. Input a set of images, $j_1, j_2, ... j_n$, and a corresponding set of motions, $A_1, A_2, ... A_n$
2. Warp each image to the central coordinate system, to obtain $j_i^{A_i}(\mathbf{x})$.
3. Use the method described in Section 5 to partition the frequency space. Obtain a set of pairs of angles, along with the best motion in that region, $< \theta_{i1}, \theta_{i2}, A_i >$.
4. Output $k = \sum_i [j_i^{A_i} * c_{\theta_{i1},\theta_{i2},r}]$
```

7. General Reconstruction

The theory developed thus far has all been for the case of affine motion. We can observe, however, that it is essentially a local process- the filters have a small area of support. It turns out that we can extend the method to essentially arbitrary differentiable transformations. This is because any differentiable transformation can be locally approximated as affine. We will give examples here for projective transformations, but it is simplest to first show how to approximate a general transformation. Suppose that some function $\mathbf{t}(\mathbf{x})$ gives the transformation, so

$$\text{if } \mathbf{x}_2 = \mathbf{t}(\mathbf{x}_1) \text{ , then } i_2(\mathbf{x}_2) = i_1(\mathbf{x}_1). \tag{26}$$

It follows that

$$\forall \mathbf{x},\ i_2(\mathbf{x}) = i_1(\mathbf{t}^{-1}(\mathbf{x})). \tag{27}$$

Now, write j_2 in the usual way.

$$j_2(\mathbf{x}) = [i_2 * e](\mathbf{x}) \tag{28}$$

$$j_2(\mathbf{x}) = \int_{\mathbf{x}'} i_1(\mathbf{t}^{-1}(\mathbf{x} - \mathbf{x}'))e(\mathbf{x}')d\mathbf{x}' \tag{29}$$

Notice here that $e(\mathbf{x}')$ will be zero unless $\mathbf{x}'$ is small. So, we will use a local approximation for the transformation.

$$\mathbf{t}(\mathbf{x} - \mathbf{x}') \approx \mathbf{t}(\mathbf{x}) - J_{\mathbf{x}}\mathbf{x}' \tag{30}$$

Where $J_{\mathbf{x}}$ denotes the Jacobian of $\mathbf{t}$, evaluated at the point $\mathbf{x}$. Now, take the inverse.

$$\mathbf{t}^{-1}(\mathbf{x} - \mathbf{x}') \approx \mathbf{t}^{-1}(\mathbf{x}) - J_{\mathbf{x}}^{-1}\mathbf{x}' \tag{31}$$

Substitute this in the above expression for j_2.

$$j_2(\mathbf{x}) = \int_{\mathbf{x}'} i_1(\mathbf{t}^{-1}(\mathbf{x}) - J_{\mathbf{x}}^{-1}\mathbf{x}')e(\mathbf{x}')d\mathbf{x}' \tag{32}$$

Now, change variables. Set $\mathbf{y} = J_{\mathbf{x}}^{-1}\mathbf{x}'$.

$$j_2(\mathbf{x}) = \int_{\mathbf{y}} i_1(\mathbf{t}^{-1}(\mathbf{x}) - \mathbf{y})e(J_{\mathbf{x}}\mathbf{x}')|J_{\mathbf{x}}|d\mathbf{y} \tag{33}$$

$$j_2(\mathbf{x}) = |J_{\mathbf{x}}|[i_1 * e^{J_{\mathbf{x}}}](\mathbf{t}^{-1}(\mathbf{x})) \tag{34}$$

So finally, we have a simple local approximation.

$$j_2(\mathbf{t}(\mathbf{x})) = |J_{\mathbf{x}}|[i_1 * e^{J_{\mathbf{x}}}](x) \tag{35}$$

The method for general reconstruction is given as Algorithm 2. Conceptually, the only difference with affine reconstruction is that the final image k is the sum of *spatially varying* filters convolved with the input images.

Algorithm 2 General Reconstruction

```
1. Input a set of images, $j_1, j_2, ... j_n$, and a corresponding
   set of transformations, $\mathbf{t}_1, \mathbf{t}_2, ... \mathbf{t}_n$
2. Warp each image to the central coordinate system, to
   obtain $j_i^{\mathbf{t}_i}(\mathbf{x})$.
3. For each point $\mathbf{x}$,
    (a) For each transformation $\mathbf{t}_i$, compute the Jacobian
        at $\mathbf{x}$, $J_{\mathbf{x},i}$.
    (b) Use the method described in Section 5 to parti-
        tion the frequency space. Obtain a set of pairs of
        angles, along with the best motion in that region,
        $< \theta_{i1}, \theta_{i2}, J_{\mathbf{x},i} >$.
    (c) Set $k(\mathbf{x}) = \sum_i [j_i^{J_{\mathbf{x},i}} * c_{\theta_{i1},\theta_{i2},r}](\mathbf{x})$
4. Output $k$.
```

7.1. Projective Transformations

We will give specific results for the case of projective transformations. The only issue is how to compute the local Jacobian of the transformation. In non-projective coordinates, let $\mathbf{x}_1 = [x_1, y_1]^T$, $\mathbf{x}_2 = [x_2, y_2]^T$. Then we can write the transformation as

$$x_2 = (h_1 x_1 + h_2 y_1 + h_3)/(h_7 x_1 + h_8 y_1 + h_9), \tag{36}$$

$$y_2 = (h_4 x_1 + h_5 y_1 + h_6)/(h_7 x_1 + h_8 y_1 + h_9). \tag{37}$$

The Jacobian is simply a matrix containing four partial derivatives. These are easily evaluated. (See additional material.)

$$J_{\mathbf{x}_1} = \begin{bmatrix} \frac{\partial x_2}{\partial x_1} & \frac{\partial x_2}{\partial y_1} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial y_1} \end{bmatrix}$$

8. Discussion

Figure 6 shows an example reconstruction for affine motion. In this experiment, images are of the same surface under four different transformations. Notice that when warped to a common coordinate system, each image is visibly blurred. Close observation reveals that this blurring is different in each of the warped images. Still, the reconstructed image is quite sharp. The meaning of this is perhaps surprising- all the high-frequency content in the reconstructed image is present in the input, just distributed among the different images. Figure 7 shows a similar example for reconstruction under projective motion. In this case all four input images are shown. (Please see the extra material for more experiments.)

This paper has presented an analysis of the filtering done by the imaging process in multiple views taken of a surface. We have seen that if we warp images to a common frame, they can be understood as the same ideal image, convolved with distorted versions of the same low-pass filter. Here this has led to an image reconstruction technique, but the same ideas could be useful in different areas. One important application is the problem of image matching. To correspond surface regions, it is necessary to understand the relationship between the underlying signals. This analysis could be useful in creating invariant region descriptors for matching when there is large motion between the images.

References

[1] R. Bracewell, K.-Y. Chang, A. Jha, and Y.-H. Wang. Affine theorem for two dimensional fourier transform. *Electronics Letters*, 29(3):304, 1993.

[2] R. N. Bracewell. *Two-dimensional imaging*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[3] N. Greene and P. S. Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Comput. Graph. Appl.*, 6(6):21–27, 1986.

[4] D. Hong and V. Kenneth. Effects of point-spread function on calibration and radiometric accuracy of ccd camera. *Applied optics*, 43(3):665–670, 2004.

[5] S. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003.

[6] L. Wang, S. B. Kang, R. Szeliski, and H.-Y. Shum. Optimal texture map reconstruction from multiple views. In *CVPR (1)*, pages 347–354, 2001.

A. Appendix

Theorem

$$[f^A * g](\mathbf{x}) = \frac{1}{|A|}[f * g^{A^{-1}}](A\mathbf{x}) \tag{38}$$

Proof

$$[f^A * g](\mathbf{x}) = \int_{\mathbf{x}'} f(A\mathbf{x} - A\mathbf{x}')g(\mathbf{x}')d\mathbf{x}' \tag{39}$$

We would like to change variables. Define $\mathbf{y} = A\mathbf{x}'$. Then, we can re-write the integral.

$$[f^A * g](\mathbf{x}) = \frac{1}{|A|}\int_{\mathbf{y}} f(A\mathbf{x} - \mathbf{y})g(A^{-1}\mathbf{y})d\mathbf{y} \tag{40}$$

$$[f^A * g](\mathbf{x}) = \frac{1}{|A|}[f * g^{A^{-1}}](A\mathbf{x}) \tag{41}$$

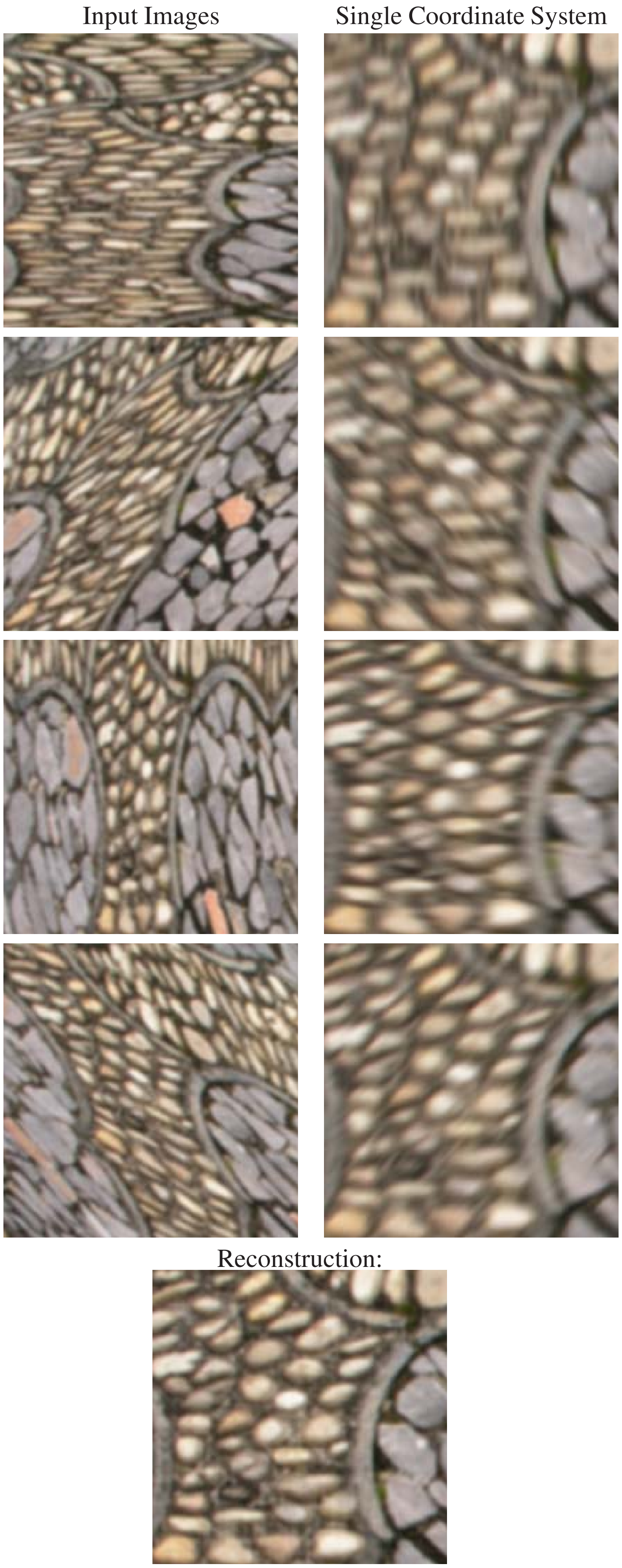


Figure 6. An example affine reconstruction. Left column: Input images. Right column: Input images warped to a common coordinate system. Bottom: Reconstructed output image.

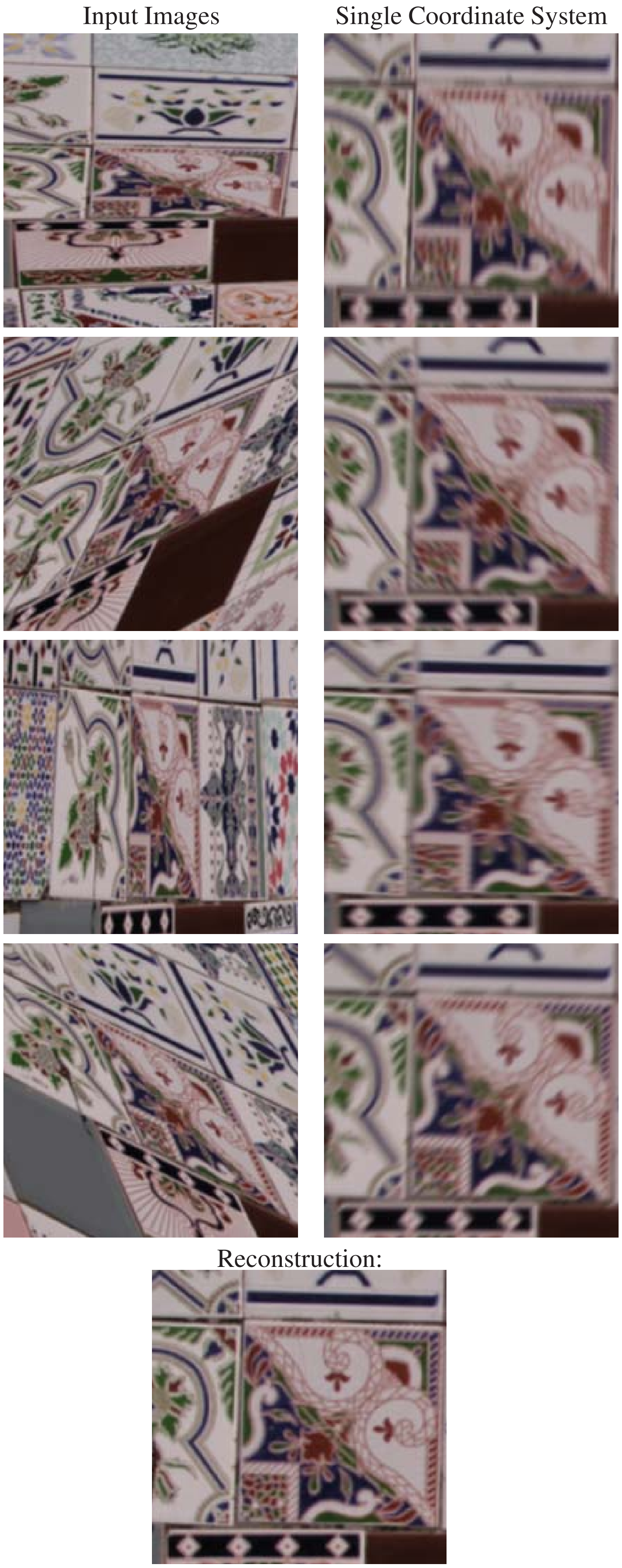


Figure 7. An example projective reconstruction. Left column: Input images. Right column: Input images warped to a common coordinate system. Bottom: Reconstructed output image.