Image Transformations and Blurring

Justin Domke and Yiannis Aloimonos

***Abstract*—Since cameras blur the incoming light during measurement, different images of the same surface do not contain the same information about that surface. Thus, in general, corresponding points in multiple views of a scene have different image intensities. While multiple view geometry constrains the locations of corresponding points, it does not give relationships between the signals at corresponding locations. This paper offers an elementary treatment of these relationships.**

We first develop the notion of "ideal" and "real" images, corresponding to, respectively, the raw incoming light and the measured signal. This framework separates the filtering and geometric aspects of imaging. We then consider how to synthesize one view of a surface from another; if the transformation between the two views is affine, it emerges that this is possible if and only if the singular values of the affine matrix are positive. Next, we consider how to combine the information in several views of a surface into a single output image. By developing a new tool called "frequency segmentation" we show how this can be done despite not knowing the blurring kernel.

***Index Terms*—Reconstruction, Restoration, Sharpening and deblurring, Smoothing**

I. Introduction

This paper concerns a very basic question: What are the relationships between multiple views of a surface? This question is only partially answered by the geometry of the situation. Consider two points in different images that project to the same 3-D surface point. Even supposing the images are perfectly Lambertian, and disregarding issues like lighting or noise, the image intensities will still in general be different. This is because cameras do not merely *measure* the incoming light. During measurement, the signal is optically filtered or *blurred*. Given the finite resolution of cameras, this is necessary to avoid aliasing effects. However, because this blurring is fixed to the camera's coordinate system, different views of a surface result in different measured signals. This is true even after correcting for the geometrical transformation between the views.

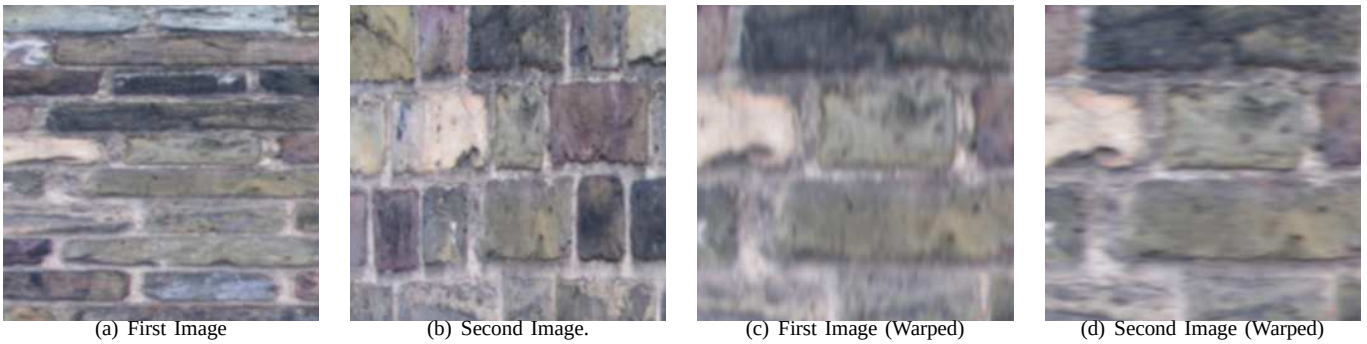
Fig. 1 shows an example of two images of a surface, and the results of warping them to a common coordinate system. To understand this situation, this paper introduces a simple formalism. We contrast between what we call the "ideal image", corresponding to the unblurred incoming light, and the "real image", corresponding to the measured signal. The ideal image cannot be measured, but is useful for analysis because it is immune to blurring. If the ideal image is some function i and the real image is some function j, we write $j = i * e$, where e is the blurring kernel. (Here, i, j, and e are all continuous functions. Assuming that the signal is low-pass filtered appropriately, j can be reconstructed from the discretely sampled pixels.) The advantage of this is that given the transformation between two views, one view's ideal image exactly specifies the other's. Notice that the 3-D structure of the scene does not need to be explicitly considered, only the resulting transformation of surfaces between different views.

Given this setup, it is easy to derive a number of theorems about the relationships between different views. After a few basic results, we will ask the following question: Suppose the real image j_2 for a view is known, as well as the blurring kernel e. Suppose also that the transformation between the views is affine: $\mathbf{x}_1$ in the first view corresponds to $\mathbf{x}_2$ in the second view if and only if $\mathbf{x}_2 = A\mathbf{x}_1$, where A is a 2x2 matrix. (It is convenient to ignore translation, since this does not impact the blurring kernel.) When will it be possible to construct the (real) image for the first view, without inverting the blurring kernel? Our first main result is that this is possible if and only if the singular values of A are greater than one. This is true independently of e, as long as it is a circularly symmetric low-pass filter. This theorem reduces to known results for the special case of Gaussian blurring.

Next, we briefly discuss possible applications of the above theorem to image matching for widely separated views. Suppose that two patches are given, and one wants to check if they might match under some affine motion A. There are three cases of interest. First, if the singular values of A are greater than one, the above theorem applies: One image can be filtered so as above to remove any information that is not present in the other image. After filtering, if the images do match, they should be the same. The second case, where the singular values of A are both *less* than one can be handled in a similar way. However, if one singular value is greater than one, and one is less, the situation is more complex. Nevertheless, we derive filters that will remove the least amount of information, while making it possible to compare the patches.

Finally, as an example of the practical use of this framework, we consider the problem of multiple view image reconstruction. Suppose that we have several views, and the transformations between them. Now, we want to construct an output image containing the best content from all input images. Here, we consider this problem in the context that the blurring kernel e is *unknown* (Though we will make some weak assumptions about its form.). This is motivated by the fact that the blurring kernel in real cameras is difficult to predict and measure, not obeying abstractions like ideal-low-pass or Gaussian [6]. Still, it is possible to reconstruct an output image that is visibly less blurred than any input image. Intuitively, this is done by taking each frequency from the image in which it is least filtered. To do this we develop a tool we call "frequency segmentation". This divides up the space of frequencies into a few regions. In each region, one input image has been least filtered. We first apply this to the case of affine transformations, and then more general transformations through a local linearization strategy.

Figure 1. The motivation for this paper. (a) and (b) show two images of a surface, taken from different views. (c) and (d) show the results of warping each image to a common coordinate system.



(a) First Image (b) Second Image. (c) First Image (Warped) (d) Second Image (Warped)

A. Previous Work

Regarding the application of multiple view image reconstruction, we are aware of only of the paper by Wang et. al [17]. Here, the image blurring is modeled as an ideal low-pass filter. This leads to a linear constraint with the reconstructed image with respect to each observed image. However, their method is limited to the case of a known (ideal low-pass) filter, while the focus in this paper is the case where the filtering is unknown.

The problem of super-resolution [12] has a similar goal to multiple view image reconstruction. Though most work on super-resolution considers single views, some takes multiple views as input [7] [1] [4]. However, super-resolution approaches usually explicitly model the blurring kernel and sampling process. The goal is then essentially to invert this process to obtain a higher resolution image. In contrast, the present paper does not model or invert the blurring process. While these super-resolution approaches would often have difficulty with far-apart views, given close views, the current method would give little improvement. It would make interesting future work to combine the effects we explore here for far-apart views with a strategy for inverting the blurring process. We suspect that such an approach would give better results than either strategy alone.

Other work touches on similar issues. Stone[15] was interested in the problem of shape from texture. He pointed out that in general, a circular filter on the image will project to a non-circular region on the surface in view. As such, he proposed an iterative scheme for computing both the surface shape, and a set of elliptical filters, one for each image point. These filters would each project to an identical area on the object surface. Each iteration consisted of an update to the shape, and then the filters, each using the current best estimates. Later work [16] suggested only updating the size of circularly symmetric filters, arguing that in practice, filters projecting to different shapes on the object surface will yield similar measurements if the areas of projection are the same.

From a more theoretical perspective, Lindeberg [8] considered how to extend linear (symmetric) Gaussian scale-space, arriving at the idea of affine Gaussian scale-space. The principal observation is that affine Gaussian scale-space is *closed* under affine warpings, unlike where the image is filtered only with a symmetric Gaussian. These ideas were later used by Lindeberg and Gårding [8] for 3-D shape estimation from texture. Several authors have used affine Gaussian scale space for different applications. Ravela [13] uses an optimization procedure to select local affine parameters at all points. This is shown to improve performance on recognition problems.

Another area of work that has touched similar issues is affine invariant region detection [11]. Specifically, the Harris-Affine & Hessian Affine detectors [10] search for extrema in the affine Gaussian scale space of an image. These extrema should be the same for images taken under different affine warpings.

II. Setup

Though all proofs are quite simple, for clarity, we postpone them to the appendix whenever they distract from the main ideas of the paper.

A. Preliminaries

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.

Finally, we will make use of the following theorem, which is a special case of the affine Fourier theorem [3][2].

Theorem 1 (Translation-Free Affine Fourier Theorem):

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

where $\mathcal{F}(\cdot)$ denotes the Fourier transform.

Proof: (Postponed) ■

B. Basic Results

Assume that we have two images of the same surface, taken under an affine motion. Without loss of generality, assume that the transformation sends the origin to the origin. Then if i_1 and i_2 are the ideal images, there exists some A such that

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

or equivalently,

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

Here, $\mathbf{x}_1$and $\mathbf{x}_2$ are a two dimensional vectors, and A is a two by two matrix.

Eq. 3 makes several simplifying assumptions. First, it does not model issues such as sensor noise. Second, if the images are taken of a static scene from different camera positions, this equation assumes Lambertian reflectance. Nevertheless, Eq. 3 is sufficient to study the changes to the signals introduced by camera blurring.

Let j_1 and j_2 be the real, observed images. If e is the low-pass filter applied by the optics of the camera, the real image is, by definition, the result of convolving the ideal image with e.

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

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

The results in this paper are independent of the particular form of e. However, we make two assumptions.

1) e is circularly symmetric. Formally, $e(\mathbf{x})$ is a function of $|\mathbf{x}|$ only. Notice that if $E(\mathbf{u})$ is the Fourier Transform of $e(\mathbf{x})$, this also implies that E is circularly symmetric.
2) e is a monotonically decreasing low-pass filter. Formally, if $|\mathbf{u}_2| \geq |\mathbf{u}_1|$, then $E(\mathbf{u}_2) \leq E(\mathbf{u}_1)$. (It is easy to show that E will be real, since $e(\mathbf{x}) = e(-\mathbf{x})$.)

While these assumptions are close enough to give useful results, we note that real cameras do not have perfectly circularly symmetric blurring, nor do they exactly obey the assumption of translation invariance.

Notice that unlike the ideal images, the real images need not match at corresponding points.

$$\mathbf{x}_2 = A\mathbf{x}_1 \not\Rightarrow j_2(\mathbf{x}_2) = j_1(\mathbf{x}_1) \tag{6}$$

The key result driving this paper is the following. Contrast this with Eqn. 3 for the ideal images.

Theorem 2 (Affine Blurring Theorem):

$$\boxed{j_2^A(\mathbf{x}) = |A|[i_1 * e^A](\mathbf{x})} \tag{7}$$

Proof: (Postponed) ■

When warped into the coordinate system of the first image, the second image is equivalent to the first ideal image, convolved with a warped version of the low-pass filter. $|A|$ acts as a normalization factor so that e^A integrates to the same value as e.

Another useful result is given by taking the Fourier Transform of both sides of Eqn. 7.

Theorem 3 (Fourier Affine Blurring Theorem):

$$\boxed{\mathcal{F}\{j_2^A(\mathbf{x})\} = [I_1 \cdot E^{A^{-T}}](\mathbf{u})} \tag{8}$$

Proof: (Postponed) ■

C. Affine Gaussian scale-space

As an example, suppose the blurring kernel is a Gaussian. Eqn. 7 can be used to get the relationship between j_1 and j_2^A more explicitly. Let $\mathcal{N}_\Sigma$ denote an origin-centered Gaussian with covariance matrix Σ. Then, $e = \mathcal{N}_{\sigma I}$.

$$j_1(\mathbf{x}) = [i_1 * \mathcal{N}_{\sigma I}](\mathbf{x}) \tag{9}$$

The following lemma specifies how a Gaussian behaves under affine warping. (This result follows easily by substituting $A\mathbf{x}$ in place of $\mathbf{x}$ in the definition of a Gaussian.)

Lemma 1:

$$\mathcal{N}_\Sigma^A(\mathbf{x}) = \frac{1}{|A|}\mathcal{N}_{A^{-1}\Sigma A^{-T}}(\mathbf{x}) \tag{10}$$

From this, the following relationship follows immediately.

$$j_2^A(\mathbf{x}) = [i_1 * \mathcal{N}_{\sigma A^{-1}A^{-T}}](\mathbf{x}) \tag{11}$$

Notice that the constant of $|A|$ was absorbed into the Gaussian. Essentially this same relationship is known in the literature on affine Gaussian scale-space. [9]

D. Parameterization of A

Though A has four parameters, it turns out that there are only three parameters of interest to us here. Consider the following decomposition of A [5].

$$A = R(\theta)R(-\phi)\Lambda R(\phi) \tag{12}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{13}$$

This decomposition is visualized in Fig. 2. It will be convenient later to make reference to the parameters θ, ϕ, λ_1, and λ_2 without explicitly specifying that they correspond to some matrix A.

The parameter θ has no role in the blurring behavior of the images. Intuitively, θ is just a rotation, which has no effect on the filtering since the low-pass filter has no preferred direction. This is formalized in the following theorem.

Theorem 4:

$$j_2^A \text{ is independent of } \theta.$$

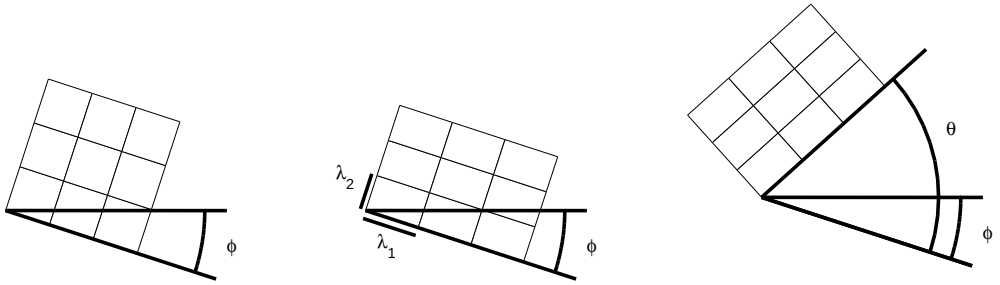
Proof: Notice that the expression for j_2 in Eqn. 7 depends on A only through $|A|$, and e^A. By our assumption that e is circularly symmetric, $e^A(\mathbf{x}) = e(A\mathbf{x}) = e(|A\mathbf{x}|) = e(|\Lambda R(\phi)\mathbf{x}|)$ does not depend on θ. It is also easy to see that $|A|$ is independent of θ. ■

III. Two-View Accessibility

Suppose we are given only A and j_2. A question arises: is it possible to filter j_2^A so as to synthesize j_1? As we will show below, this is only sometimes possible.

In this paper, we want to avoid the problem of deconvolution, or deblurring. In theory, if the low-pass filter E were nonzero for all frequencies, it would be possible to increase the magnitude of all frequencies to completely cancel out the effects of the filtering. With the ideal image in hand, any other

Figure 2. Visualization of the four parameters of A. We can see that the effect of $R(-\phi)\Lambda R(\phi)$ is to stretch by factors of λ_1, and λ_2 along axes an angle of ϕ away from the starting axes.



(a) Initial Points. (b) After $R(-\phi)\Lambda R(\phi)$. (c) After rotation by θ.

view could be synthesized. However, in practice, this can be done only for a limited range of frequencies, because for high frequencies the noise in the observed image will usually be much larger than the signal remaining after filtering [14]. If deconvolution is practical, we can imagine that it has already been applied, and the problem restated with an appropriately changed blurring kernel and input images. Hence, we will not attempt to invert the blurring kernel.

Suppose that we apply some filter d to one of the images. If D is the Fourier Transform of d, we require that D does not increase the magnitude of any frequency. Formally, for all $\mathbf{u}$, we must have $D(\mathbf{u}) \leq 1$. (Again, D will be real. See the discussion in the following theorem.)

We will say that a view "is accessible" from another view taken under motion A if there exists a valid filter d such that $e = |A|e^A * d$ (Recall Eqn. 7). Notice that if this is the case, then

$$\begin{aligned} [j_2^A * d](\mathbf{x}) &= |A|[(i_1 * e^A) * d](\mathbf{x}) && (14)\\ &= |A|[i_1 * (e^A * d)](\mathbf{x}) && (15)\\ &= [i_1 * e](\mathbf{x}) && (16)\\ &= j_1(\mathbf{x}). && (17)\end{aligned}$$

The following theorem formalizes exactly when one view is accessible from another. It is worth sketching the proof in some detail.

Theorem 5 (Two-View Accessibility Theorem): j_1 is accessible from j_2 if and only if $\lambda_1 \geq 1$ and $\lambda_2 \geq 1$.

Proof: We will develop several conditions, each of which is equivalent to the assertion "j_1 is accessible from j_2". By definition, this means there is a valid d such that

$$\forall \mathbf{x}, [|A|e^A * d](\mathbf{x}) = e(\mathbf{x}). \quad (18)$$

Apply the translation-free affine Fourier theorem to e^A.

$$\mathcal{F}\{e^A(\mathbf{x})\} = |A^{-1}|E^{A^{-T}}(\mathbf{u}) \quad (19)$$

Now, if we also take the Fourier transform of d and e, the condition is, by the convolution theorem,

$$\forall \mathbf{u}, E(A^{-T}\mathbf{u}) \cdot D(\mathbf{u}) = E(\mathbf{u}). \quad (20)$$

Since E is real, clearly D must also be. Since we need that $D(\mathbf{u}) \leq 1$ we require that for all $\mathbf{u}$, $E(A^{-T}\mathbf{u}) \geq E(\mathbf{u})$. This will be the case when

$$\forall \mathbf{u}, |A^{-T}\mathbf{u}| \leq |\mathbf{u}|. \quad (21)$$

It is not hard to show that this is equivalent to λ_1 and λ_2 being at least one. ■

When j_1 is in fact accessible from j_2, constructing the filter is trivial. By Eqn. 20, simply set $D(\mathbf{u}) = E(\mathbf{u})/E(A^{-T}\mathbf{u})$ for all $\mathbf{u}$, and obtain d by the inverse Fourier transform. The exact form of d will of course depend on the form of e.

For example, suppose e is a Gaussian filter. Let $\mathcal{N}_\Sigma$ denote a Gaussian function with covariance matrix Σ. We need that $\mathcal{N}_{\sigma I} = \mathcal{N}_{\sigma A^{-1}A^{-T}} * d$. (Recall Eqn. 11.) Since $\mathcal{N}_{\Sigma_1} * \mathcal{N}_{\Sigma_2} = \mathcal{N}_{\Sigma_1+\Sigma_2}$, we can see that $d = \mathcal{N}_{\sigma(I-A^{-1}A^{-T})}$. For this to be a valid Gaussian, the covariance matrix $I - A^{-1}A^{-T}$ must be positive definite. This is true if and only if the singular values of A are at least one, confirming the above theorem.

For another example, suppose e is an ideal low pass filter. That is, suppose $E(\mathbf{u}) = 1$ when $|\mathbf{u}| < r$ for some threshold r, and zero otherwise. In this case, we can use $d = e$. The effect is to filter out those frequencies in j_2^A that are not in j_1. However, if the singular values of A are not both at least one, there will be some frequencies in j_1 that are not in j_2^A, and again j_1 would be inaccessible.

A. Accessibility and Matching

The accessibility theorem has some possible applications to image matching that we briefly discuss here. This section can be skipped without loss of continuity.

Suppose images j_1 and j_2, are hypothesized to vary by some motion A. How can this be checked? Intuitively, one could do filtering as in the previous section to create a "possible copy" of j_1 from j_2^A, under the assumption that the image regions do correspond. This can then be compared to j_1 to determine if the regions match. However, this may or may not be possible, depending on A. There are three situations.

1) $\lambda_1 \geq 1$, $\lambda_2 \geq 1$. Here filtering can be done exactly as described above.
2) $\lambda_1 \leq 1$, $\lambda_2 \leq 1$. Now, j_1 is not accessible from j_2. However, by symmetry, one can instead try to synthesize

j_2 from j_1, under a hypothesized motion of A^{-1}. (The singular values of A^{-1} will both be at least one.)

3) Otherwise. In this situation, neither image is accessible from the other. Intuitively, this means that both images contain some frequencies that are reduced (or eliminated) in the other image.

The first two situations can be dealt with by the methods above, but the third needs more discussion. The problem is how to filter the images to remove any extra information in one image that is not present in the other. At the same time, to match successfully, it is better to remove as little information as possible. The goal is to create filters d_1 and d_2 such that

$$\forall \mathbf{x}, [j_1 * d_1](\mathbf{x}) = [j_2^A * d_2](\mathbf{x}). \tag{22}$$

Consider this in the frequency domain. For the left hand side, we have

$$\mathcal{F}\{j_1 * d_1\} = J_1(\mathbf{u}) \cdot D_1(\mathbf{u}) = I_1(\mathbf{u}) \cdot E(\mathbf{u}) \cdot D_1(\mathbf{u}). \tag{23}$$

For the right hand side, recall the Fourier Transform of j_2^A from Eqn. 8.

$$\mathcal{F}\{j_2^A * d_2\} = I_1(\mathbf{u}) \cdot E^{A^{-T}}(\mathbf{u}) \cdot D_2(\mathbf{u}). \tag{24}$$

So filters need to be constructed such that

$$\forall \mathbf{u}, E(\mathbf{u}) \cdot D_1(\mathbf{u}) = E^{A^{-T}}(\mathbf{u}) \cdot D_2(\mathbf{u}). \tag{25}$$

The optimal solution is to set the filters so as to minimize the amount of information removed.

$$D_1(\mathbf{u}) = \begin{cases} \frac{E^{A^{-T}}(\mathbf{u})}{E(\mathbf{u})} & \text{if } E(\mathbf{u}) \geq E^{A^{-T}}(\mathbf{u}) \\ 1 & \text{if } E(\mathbf{u}) \leq E^{A^{-T}}(\mathbf{u}) \end{cases} \tag{26}$$

$$D_2(\mathbf{u}) = \begin{cases} 1 & \text{if } E(\mathbf{u}) \geq E^{A^{-T}}(\mathbf{u}) \\ \frac{E(\mathbf{u})}{E^{A^{-T}}(\mathbf{u})} & \text{if } E(\mathbf{u}) \leq E^{A^{-T}}(\mathbf{u}) \end{cases} \tag{27}$$

The problem remains how to divide up the space of $\mathbf{u}$ into those where E or $E^{A^{-T}}$ is greater. This problem will be addressed in a more general context in Section IV.

B. Example

This example uses synthetically generated images consisting of white noise, because the broad frequency content makes filtering easy to see. In this case, the low pass filter is a symmetric Gaussian. Hence, $e = \mathcal{N}_{\sigma I}$.

In Fig. 3, the first image is observed with a motion consisting of a large vertical stretch, and a mild horizontal stretch. Since here, both λ_1 and λ_2 are greater than one, the first image is in fact accessible from the second.

IV. Frequency Segmentation

Suppose we have a set of images warped to a common coordinate system, $\{j_i^{A_i}(\mathbf{x})\}$, as well as the set of motions that produced those images, $\{A_i\}$. The question is: for a given frequency $\mathbf{u}$, in what image is it least filtered? This section develops a method to segment frequency space into labeled regions. In each region, the label gives the index of the image with the least filtering for that region's frequencies.

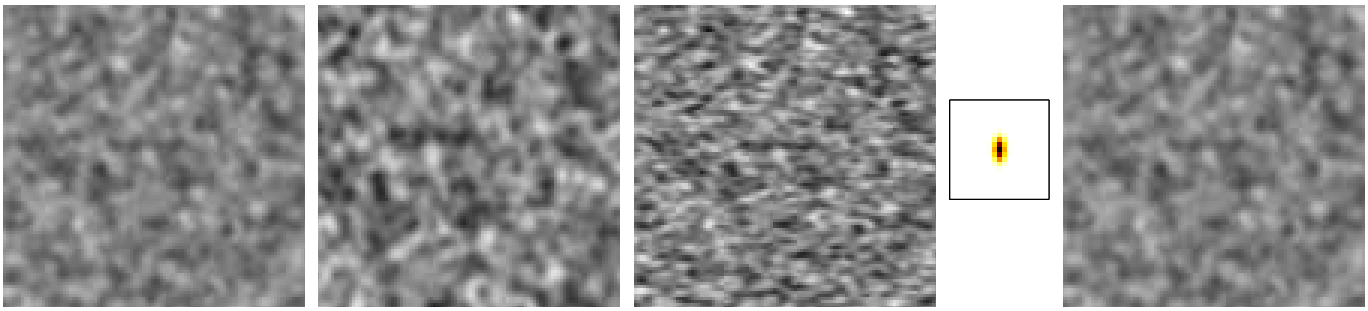
One can write the Fourier Transform of each of the images as the ideal image with a warped version of the low-pass filter. (Recall Eqn. 8.)

$$\mathcal{F}\{j_i^{A_i}(\mathbf{x})\} = [I \cdot E^{A_i^{-T}}](\mathbf{u}) \tag{28}$$

The problem is to determine, for each $\mathbf{u}$, for which A_i is $E(A_i^{-T}\mathbf{u})$ maximum, or equivalently, for which A_i is $|A_i^{-T}\mathbf{u}|$ minimum. Notice that this will depend only on the "angle" of $\mathbf{u}$, and not on its magnitude; if $|A_i^{-T}\mathbf{u}|$ is minimum for some $\mathbf{u}$, it will also be minimum for $a\mathbf{u}$, for any a. So, for each angle θ, we would like to choose the motion such that $|A_i^{-T}[\cos\theta, \sin\theta]^T|$ is minimum. One 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 4(a)). To "carve up" the space of frequencies, requires two steps.

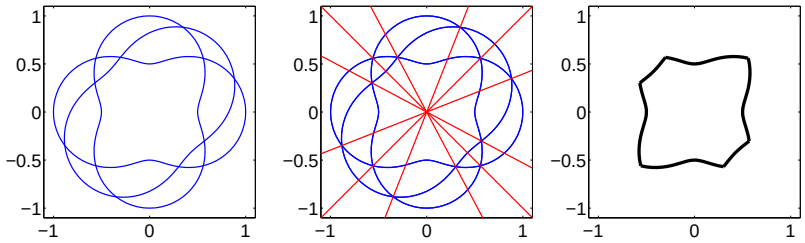
1) For each pair of motions A_i and A_j, find points for which the curves meet. That is, find $\mathbf{u}$ such that
$$|A_i^{-T}\mathbf{u}| = |A_j^{-T}\mathbf{u}|. \tag{29}$$
This is equivalent to finding $\mathbf{u}$ such that
$$\mathbf{u}^T(A_i^{-1}A_i^{-T} - A_jA_j^{-T})\mathbf{u} = 0. \tag{30}$$
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 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 remains to find the motion that has the smallest value in each region. The simplest procedure would simply be to test all motions, and explicitly find
$$\arg\min_i |A_i^{-T} \begin{bmatrix} \cos(.5(\theta_j + \theta_{j+1})) & \sin(.5(\theta_j + \theta_{j+1})) \end{bmatrix}^T|. \tag{31}$$
This works, but has a worse-case time complexity of $O(n^3)$, where n is the number of input images. However, an optimization can reduce this to $O(n^2 \log n)$, the complexity of the sorting step: 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 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. 31. This means that we only need to test at most two motions in each region.

Figure 3. A demonstration of 2-view accessibility, with an image of white noise. (a) j_1. (b) j_2. (c) j_2^A. Notice that there is visual content clearly visible in j_2^A, but not in j_1. (d) the filter d. (e) the result of convolving j_2^A with d. Notice that this is almost identical to j_1. For this experiment, $A = \begin{bmatrix} 1.1 & 0 \\ 0 & 2.2 \end{bmatrix}$.



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

Figure 4. An intuitive description of Frequency Segmentation. See text for a full description.



(a) (b) (c)

The entire frequency segmentation process is illustrated in Fig. 4. Part (a) shows the magnitude $|A_i^{-T}[\cos\theta, \sin\theta]^T|$ plotted for each angle θ. In part (b), the angles are found where the motions "intersect". In part (c), the shape is shown, where for each angle, the magnitude is taken from the best motion.

V. Reconstruction with an Unknown Kernel

At this point, one might proceed to extend the results of Section III-A to the case of many views. This is certainly possible. One could define alternate conditions to those in the two-view accessibility theorem– rather than insisting that each frequency in the desired view is less filtered in the other, the condition would be that for each frequency in the desired view there is at least one image in which it is less filtered. Given this, one could use frequency segmentation to build appropriate filters and synthesize new views. However, this will only be possible if the blurring kernel e is known. Given the difficulty of measuring the blurring kernel in real cameras, in this paper we instead focus on what to do if the blurring kernel is unknown. It might initially seem that the blurring kernel must be known in order to do multiple view reconstruction. However, notice that the frequency segmentation method does not make use of the particular form of e– only the assumption that it is circularly symmetric and monotonically decreasing.

Still, it is impossible to synthesize a view without knowing the blurring kernel that view is the result of. Hence, we cannot require this as output. Instead, we simply combine the frequencies in the input images, *at their observed levels*, into one output image. This output will be visibly "less blurred" than any input images, but does not correspond to what would be measured from any real view. To do this, it is not necessary to know the blurring kernel. After frequency segmentation is done, one only needs to design a filter that will take a certain range of frequencies from each image. This problem is addressed in the Section VI.

It is also important to consider what will happen if our assumptions on e violated. Suppose e is not exactly circularly symmetric, or suppose that it is spatially variant. (In real cameras these are both likely to be true to some degree.) As long as these conditions are not dramatically violated, the frequency segmentation boundaries will still be found nearby the optimal ones, and hence the reconstruction will still be similar to the optimal one.

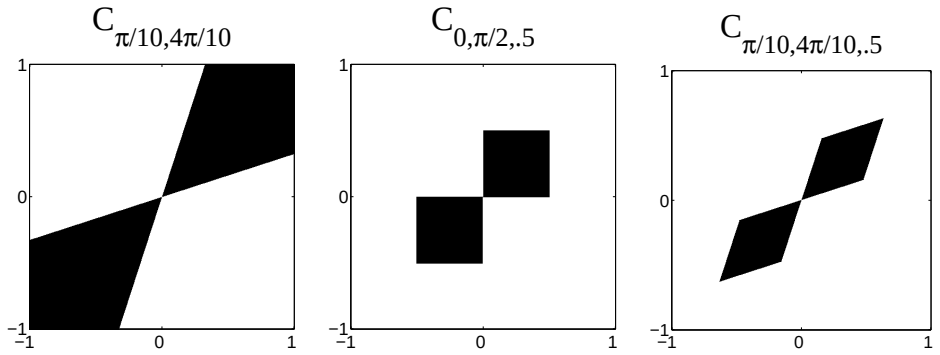
VI. The Frequency Slice Filter

The overall goal here is to create a filter which 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. 5(a)),

$$C_{\theta_1,\theta_2}(\mathbf{u}) = \begin{cases} 1 & \theta_1 \leq \arg \mathbf{u} \leq \theta_2 \\ 0 & \text{else} \end{cases}. \tag{32}$$

First, consider the case where $\theta_1 = 0$, and $\theta_2 = \pi/2$. 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

Figure 5. The frequency slice filter in the Fourier domain. The filter in (a) is ideal, but cannot be used in the spatial domain since its inverse Fourier transform does not converge. (b) The filter with angles $\theta_1 = 0$, and $\theta_2 = \pi/2$, and threshold $r = .5$. (c) The filter with angles $\theta_1 = \pi/10$, and $\theta_2 = \pi/2$, and threshold $r = .5$.



(a) (b) (c)

what the filter does to very high frequencies. So, instead, define (Fig. 5(b))

$$C_{0,\pi/2,r}(\mathbf{u}) = \begin{cases} 1 & 0 \leq \arg \mathbf{u} \leq \pi/2, \text{ and } |\mathbf{u}| \leq r \\ 0 & \text{else} \end{cases}. \quad (33)$$

Before extending this to the case of other θ_1, θ_2, consider the inverse Fourier transform.

$$\begin{aligned} c_{0,\pi/2,r}(\mathbf{x}) =& \mathcal{F}^{-1}\{C_{0,\pi/2,r}(\mathbf{u})\} && (34)\\ =& \int_{\mathbf{u}} C_{0,\pi/2,r}(\mathbf{u}) \exp(2\pi i \mathbf{u}^T \mathbf{x}) d\mathbf{u} && (35)\\ =& \int_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{r}} [\exp(2\pi i \mathbf{u}^T \mathbf{x}) + \exp(-2\pi i \mathbf{u}^T \mathbf{x})] d\mathbf{u} && (36)\\ =& \int_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{r}} 2\cos(2\pi \mathbf{u}^T \mathbf{x}) d\mathbf{u} && (37)\\ =& \frac{1}{2\pi^2 xy}[\cos(2\pi r x) + \cos(2\pi r y) && (38)\\ & - \cos(2\pi r(x+y)) - 1] \end{aligned}$$

Examples of c in the spatial domain are shown in Fig. 6. Notice specifically that this function decays by the inverse of xy, hence it has limited spatial extent. This will be important in extending the approach to non-affine transformations. To define the filter for arbitrary angles, define the matrix V.

$$V = \begin{bmatrix} \cos\theta_1 & \cos\theta_2 \\ \sin\theta_1 & \sin\theta_2 \end{bmatrix} \quad (39)$$

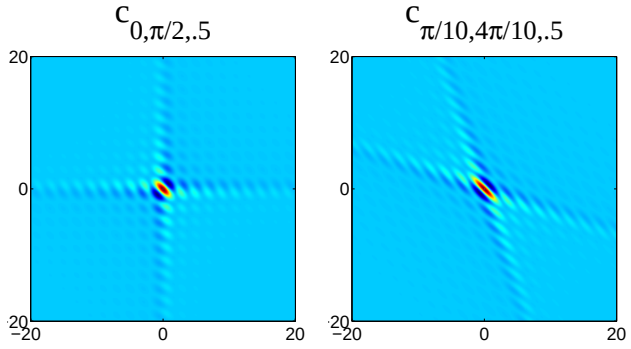
Now, we can define the frequency slice filter for arbitrary angles.

$$c_{\theta_1,\theta_2,r}(\mathbf{x}) = |V| c^{V^T}_{0,\pi/2,r}(\mathbf{x}) \quad (40)$$

To see this, apply the translation-free affine Fourier theorem to the right hand side.

$$\begin{aligned} \mathcal{F}\{|V| c^{V^T}_{0,\pi/2,r}(\mathbf{x})\} &= |V| \cdot |V^{-T}| C^{(V^T)^{-T}}_{0,\pi/2,r}(\mathbf{u}) && (41)\\ &= C^{V^{-1}}_{0,\pi/2,r}(\mathbf{u}) && (42) \end{aligned}$$

Figure 6. The frequency slice filter in the spatial domain. (a) The filter with angles $\theta_1 = 0$, and $\theta_2 = \pi/2$, and threshold $r = .5$. (c) The filter with angles $\theta_1 = \pi/10$, and $\theta_2 = \pi/2$, and threshold $r = .5$.



(a) (b)

Notice that V^{-1} 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. 5 (c) shows an example in the frequency domain with $\theta_1 = \pi/10$, $\theta_2 = 4\pi/10$.

Different filters could also be used to perform reconstruction. It is only necessary that the filters behave as indicator functions for low frequencies. For example, instead of "cutting" the frequencies off at the threshold of r as done here, the falloff could be smooth. However, we have found that the above filter is sufficient for our experiments. For our experiments, r was set to .3. Increasing it further does not visibly change the results.

VII. Affine Reconstruction

The algorithm for affine reconstruction is given as Alg. 1. The intuition behind the method is quite simple. The images are warped to a common coordinate system, and then convolved with a filter calculated for each image. This filter depends on the results of frequency segmentation. Finally, the results of the convolutions are added together to produce the output image.

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 IV to segment 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 $k = \sum_i [j_i^{A_i} * c_{\theta_{i1},\theta_{i2},r}]$.

A. Example

Fig. 7 shows an example reconstruction, contrasted with a simple averaging scheme. For comparison, the algorithm is run with various subsets of the images as input. The relatively modest improvements of the reconstruction over the input are typical of what will be encountered in practice. However, to give a sense of what is theoretically possible, Fig. 8 shows an experiment with five images taken of a painting with extreme foreshortening. In this idealized case, the improvement of the reconstruction is quite dramatic.

VIII. 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 by locally approximating the transformations as affine. We will give experimental results 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

$$\text{if } \mathbf{x}_2 = \mathbf{t}(\mathbf{x}_1) \text{ , then } i_2(\mathbf{x}_2) = i_1(\mathbf{x}_1). \tag{43}$$

It follows that

$$\forall \mathbf{x},\ i_2(\mathbf{x}) = i_1(\mathbf{t}^{-1}(\mathbf{x})). \tag{44}$$

Now, write j_2 in the usual way.

$$\begin{aligned} j_2(\mathbf{x}) &= [i_2 * e](\mathbf{x}) && (45)\\ &= \int_{\mathbf{x}'} i_1(\mathbf{t}^{-1}(\mathbf{x}-\mathbf{x}'))e(\mathbf{x}')d\mathbf{x}' && (46)\end{aligned}$$

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

$$\mathbf{t}^{-1}(\mathbf{x}-\mathbf{x}') \approx \mathbf{t}^{-1}(\mathbf{x}) - J_{\mathbf{t}}^{-1}(\mathbf{x})\mathbf{x}' \tag{47}$$

Where $J_{\mathbf{t}}(\mathbf{x})$ denotes the Jacobian of $\mathbf{t}$, evaluated at the point $\mathbf{x}$. Substitute this in the above expression for j_2.

$$j_2(\mathbf{x}) = \int_{\mathbf{x}'} i_1(\mathbf{t}^{-1}(\mathbf{x}) - J_{\mathbf{t}}^{-1}(\mathbf{x})\mathbf{x}')e(\mathbf{x}')d\mathbf{x}' \tag{48}$$

Now, change variables. Set $\mathbf{y} = J_{\mathbf{t}}^{-1}(\mathbf{x})\mathbf{x}'$.

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 $\mathbf{x}$, $J_{\mathbf{t}_i}(\mathbf{x})$.
   b) Use the method described in Section IV to segment frequency space. Obtain a set of pairs of angles, along with the best motion in that region, $< \theta_{i1}, \theta_{i2}, J_{\mathbf{t}_i}(\mathbf{x}) >$.
   c) Set $k(\mathbf{x}) = \sum_i [j_i^{J_{\mathbf{t}_i}(\mathbf{x})} * c_{\theta_{i1},\theta_{i2},r}](\mathbf{x})$
4) Output k.

$$\begin{aligned} j_2(\mathbf{x}) &= \int_{\mathbf{y}} i_1(\mathbf{t}^{-1}(\mathbf{x}) - \mathbf{y})e(J_{\mathbf{t}}(\mathbf{x})\mathbf{x}')|J_{\mathbf{t}}(\mathbf{x})|d\mathbf{y} && (49)\\ &= |J_{\mathbf{t}}(\mathbf{x})|[i_1 * e^{J_{\mathbf{t}}(\mathbf{x})}](\mathbf{t}^{-1}(\mathbf{x})) && (50)\end{aligned}$$

So finally, we have a simple local approximation.

$$j_2(\mathbf{t}(\mathbf{x})) = |J_{\mathbf{t}}(\mathbf{x})|[i_1 * e^{J_{\mathbf{t}}(\mathbf{x})}](x) \tag{51}$$

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.

A. Projective Reconstruction

It is most common to write a projective transformation in homogeneous coordinates, in which case it is just a linear transformation. Here, however, we must use non-homogeneous coordinates. Let $\mathbf{x}_1 = [x_1, y_1]^T$ and $\mathbf{x}_2 = [x_2, y_2]^T$ be corresponding points in two images. Then, for some parameters $h_1, h_2, ..., h_9$,

$$x_2 = \frac{h_1 x_1 + h_2 y_1 + h_3}{h_7 x_1 + h_8 y_1 + h_9}, \tag{52}$$

$$y_2 = \frac{h_4 x_1 + h_5 y_1 + h_6}{h_7 x_1 + h_8 y_1 + h_9}. \tag{53}$$

The Jacobian is simply a matrix containing four partial derivatives.

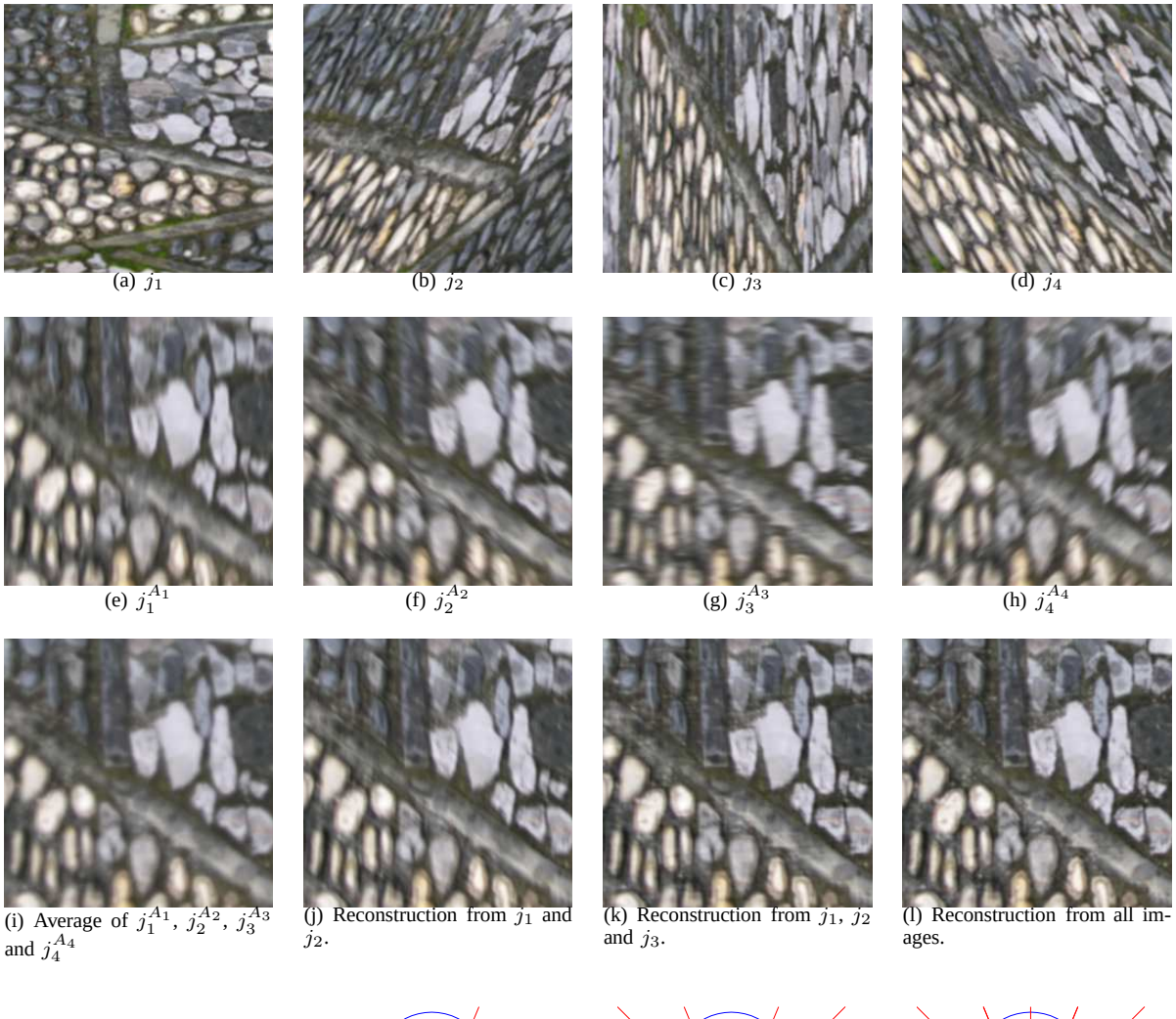
$$J_{\mathbf{t}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial x_2}{\partial x_1} & \frac{\partial x_2}{\partial y_1} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial y_1} \end{bmatrix} \tag{54}$$

These are easily evaluated by differentiating Eqs. 52 and 53.

$$\frac{\partial x_2}{\partial x_1} = \frac{y_1(h_1 h_8 - h_7 h_2) + h_1 h_9 - h_7 h_3}{(h_7 x_1 + h_8 y_1 + h_9)^2} \tag{55}$$

$$\frac{\partial x_2}{\partial y_1} = \frac{x_1(h_2 h_7 - h_8 h_1) + h_2 h_9 - h_8 h_3}{(h_7 x_1 + h_8 y_1 + h_9)^2} \tag{56}$$

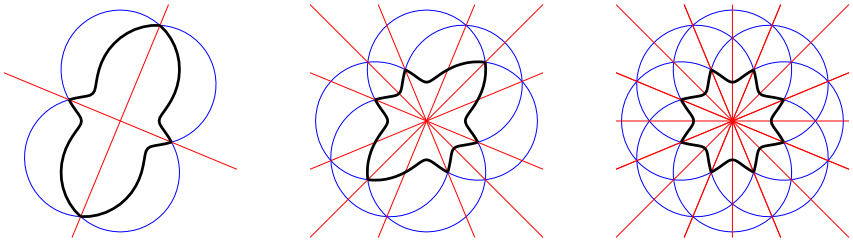
Figure 7. Affine Reconstruction (Synthetically warped images)



(a) j_1 (b) j_2 (c) j_3 (d) j_4

(e) $j_1^{A_1}$ (f) $j_2^{A_2}$ (g) $j_3^{A_3}$ (h) $j_4^{A_4}$

(i) Average of $j_1^{A_1}$, $j_2^{A_2}$, $j_3^{A_3}$ and $j_4^{A_4}$ (j) Reconstruction from j_1 and j_2. (k) Reconstruction from j_1, j_2 and j_3. (l) Reconstruction from all images.



(m) Frequency segmentation for j_1 and j_2. (n) Frequency segmentation for j_1, j_2, and j_3. (o) Frequency segmentation for all images.

$$\frac{\partial y_2}{\partial x_1} = \frac{y_1(h_4h_8 - h_7h_5) + h_4h_9 - h_7h_6}{(h_7x_1 + h_8y_1 + h_9)^2} \tag{57}$$

$$\frac{\partial y_2}{\partial y_1} = \frac{x_1(h_5h_7 - h_8h_4) + h_5h_9 - h_8h_6}{(h_7x_1 + h_8y_1 + h_9)^2} \tag{58}$$

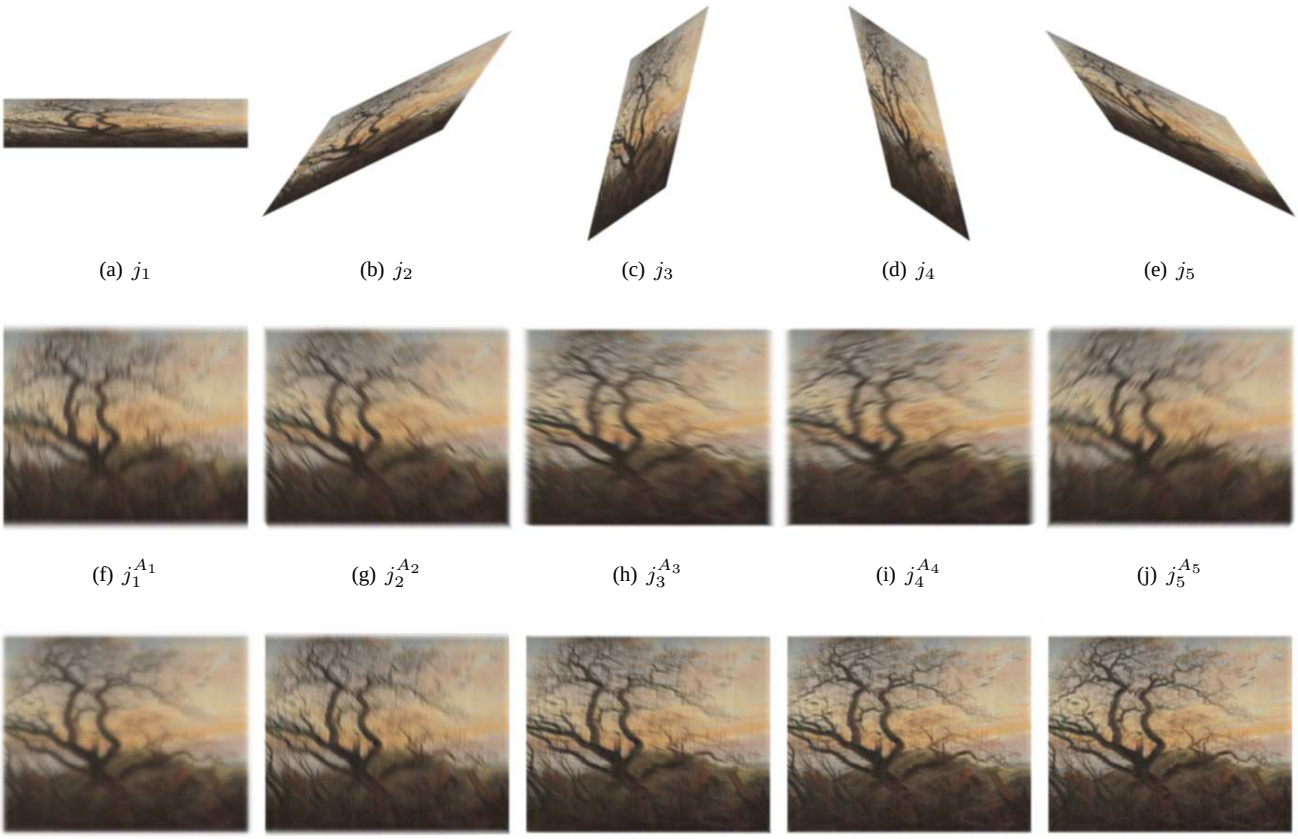
Though all experiments here will be for projective transformations, a much larger class of transformations can be locally approximated as affine. To do reconstruction for any such transformation is it only necessary to calculate the partial derivatives as above.

B. Making Reconstruction Faster

For affine reconstruction, the algorithm is extremely fast– the computation is dominated by the convolutions in step 4. In the algorithm for general reconstruction, however, the filters are spatially varying, and need to be recalculated at each pixel.

A simple trick can speed this up. Instead of calculating the filters for each pixel, calculate them for some small patch of pixels. If the affine approximation is slowly changing, this will result in no visible change to the output, while hugely reducing the overhead for recomputing filters. However, this strategy could introduce artifacts on some images, at the boundary where the filters change. To reduce this, we allow

Figure 8. Affine Reconstruction with extreme foreshortening. (Synthetically warped images)



(a) j_1 (b) j_2 (c) j_3 (d) j_4 (e) j_5

(f) $j_1^{A_1}$ (g) $j_2^{A_2}$ (h) $j_3^{A_3}$ (i) $j_4^{A_4}$ (j) $j_5^{A_5}$

(k) Average of $j_1^{A_1}$ ... $j_4^{A_5}$ (l) Reconstruction from j_1 and j_2. (m) Reconstruction from j_1, j_2 and j_3. (n) Reconstruction from j_1 ...j_4. (o) Reconstruction from all images.

the patches to overlap by one pixel, and use a small amount of blending. For pixels in overlapping regions, the output is set to the average of the results of the different filters. In our experiments, we used patches of size 20 by 20.

C. Experiments

Figure 9 shows a reconstruction from three synthetically generated images of a radius pattern. We compare our results to a simple averaging process.

Figures 10 and 11 show the reconstruction process for images taken of two different textures. After reconstruction, for presentation, we manually segmented out the background. To better understand how using more images leads to better reconstruction, we also include the results of reconstruction using only the first two of the three images.

IX. Conclusions

This paper introduced the formalism of the "ideal image", consisting of the unblurred incoming light, and the "real image" consisting of the blurred, measured image. Because this framework separates the filtering and geometrical aspects, it makes it easy to derive several results. The notion of the ideal image might be useful for deriving other results.

More formally, a relation between the outputs of filters applied at corresponding patches in different views was developed. This result was used to formulate and prove the Accessibility Theorem, which states the conditions under which, given two images of the same scene, one is accessible by the other (i.e., one image can be obtained by appropriately filtering the other).We then discussed the consequences of this result to the understanding of the space of all images of a scene. As an application of this framework, we showed that it is possible to perform multiple view image reconstruction, even with an unknown blurring kernel, through the new tool of frequency segmentation.

X. Acknowledgements

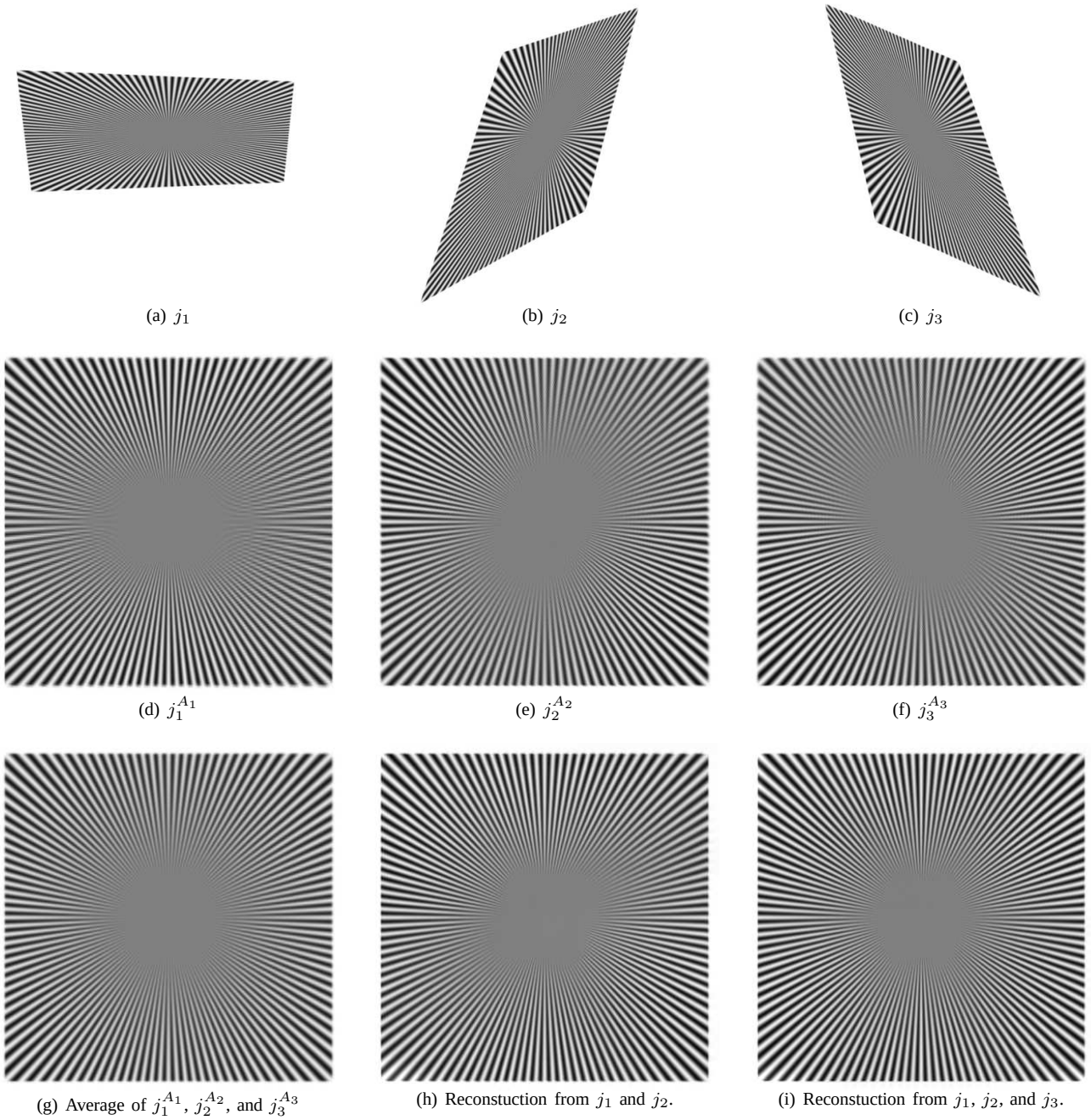
The support of (a) NSF under a collaborative project with Stanford University (EMT Bioinspired computing) and (b) the European Union's 7th Framework program on Cognitive Systems under the project POETICON, is gratefully acknowledged.

References

[1] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, pages 573–582, 1996.

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

Figure 9. Projective Reconstruction (Synthetic images) This figure is best viewed in the printed version.



(a) j_1 (b) j_2 (c) j_3

(d) $j_1^{A_1}$ (e) $j_2^{A_2}$ (f) $j_3^{A_3}$

(g) Average of $j_1^{A_1}$, $j_2^{A_2}$, and $j_3^{A_3}$ (h) Reconstuction from j_1 and j_2. (i) Reconstuction from j_1, j_2, and j_3.

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

[4] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame super-resolution. *IEEE Trans. Image Processing*, 13(10):1327–1344, Oct. 2004.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[6] 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.

[7] H. Ji and C. Fermuller. Wavelet-based super-resolution reconstruction: Theory and algorithm. In *ECCV*, 2006.

[8] T. Lindeberg. On the axiomatic foundations of linear scale-space: Combining semi-group structure with causality vs. scale invariance. Technical report, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, 1994.

[9] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure. *Image Vision Comput.*, 15(6):415–434, 1997.

[10] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

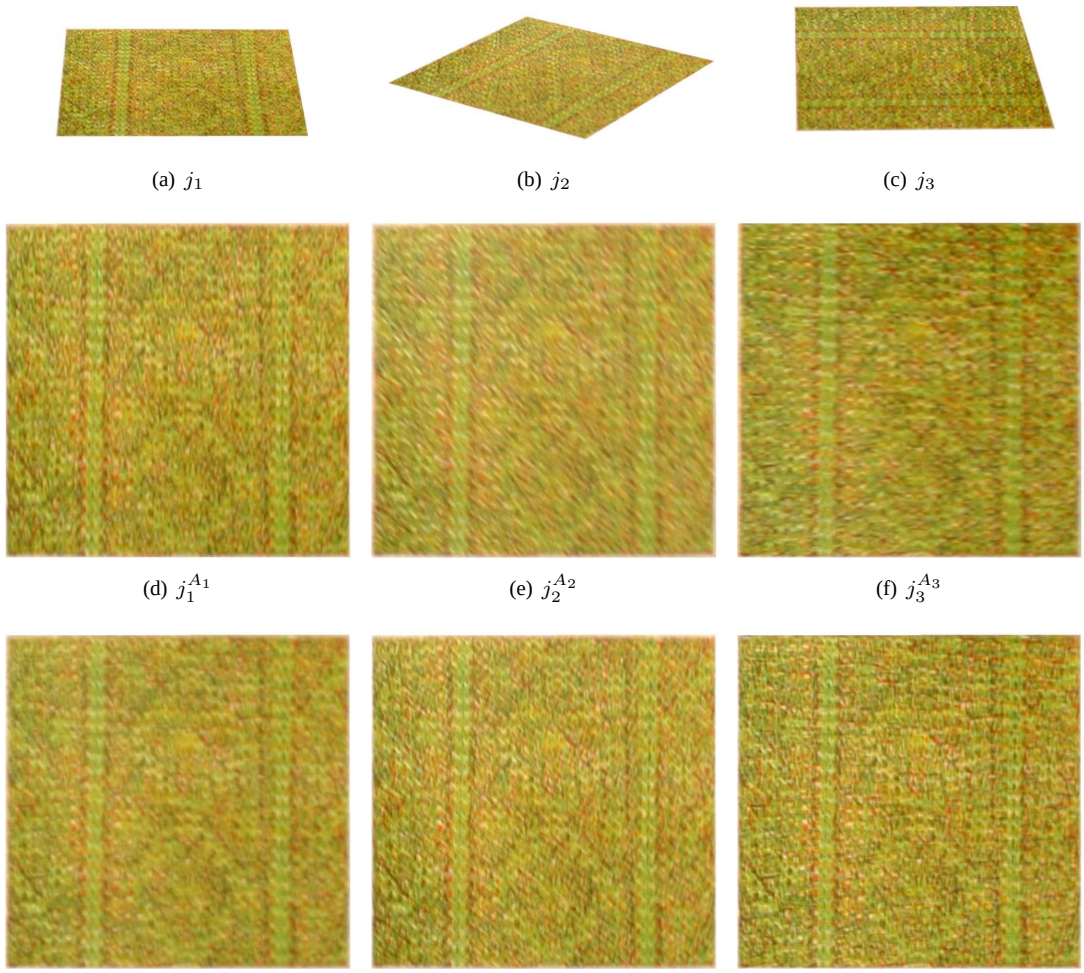
[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

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

[13] S. Ravela. Shaping receptive fields for affine invariance. In *CVPR*, volume 02, pages 725–730, 2004.

[14] J. Starck, E. Pantin, and F. Murtagh. Deconvolution in astronomy: A review.

Figure 10. Projective Reconstruction (Real images)



(a) j_1 (b) j_2 (c) j_3

(d) $j_1^{A_1}$ (e) $j_2^{A_2}$ (f) $j_3^{A_3}$

(g) Average of $j_1^{A_1}$, $j_2^{A_2}$, and $j_3^{A_3}$ (h) Reconstuction from j_1 and j_2. (i) Reconstuction from j_1, j_2, and j_3.

[15] J. V. Stone. Shape from texture: Textural invariance and the problem of scale in perspective images of textures surface. In *BMVC*, pages 181–187, 1990.

[16] J. V. Stone and S. D. Isard. Adaptive scale filtering: A general method for obtaining shape from texture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(7):713–718, 1995.

[17] 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.

APPENDIX

Theorem 6 (Translation-Free Affine Fourier Theorem):

If $\mathcal{F}\{f(\mathbf{x})\} = F(\mathbf{u})$ then $\mathcal{F}\{f^A(\mathbf{x})\} = |A^{-1}|F^{A^{-T}}(\mathbf{u})$.

Proof:

$$\mathcal{F}\{f^A(\mathbf{x})\}(\mathbf{u}) = \int f(A\mathbf{x})e^{-2\pi\mathbf{u}^T\mathbf{x}}d\mathbf{x} \tag{59}$$

Now, define $\mathbf{x}' = A\mathbf{x}$, and change variables.

$$\mathcal{F}\{f^A(\mathbf{x})\}(\mathbf{u}) = |A^{-1}| \int f(\mathbf{x}')e^{-2\pi(A^{-T}\mathbf{u})^T\mathbf{x}'}d\mathbf{x}' \tag{60}$$

This is just the Fourier transform of f, evaluated at $A^{-T}\mathbf{u}$.

$$\mathcal{F}\{f^A(\mathbf{x})\}(\mathbf{u}) = |A^{-1}|F(A^{-T}\mathbf{u}) \tag{61}$$

■

Lemma 2 (Affine Convolution Lemma):

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

Proof:

By definition,

$$[f^A * g](\mathbf{x}) = \int f(A\mathbf{x} - A\mathbf{x}')g(\mathbf{x}')d\mathbf{x}'. \tag{62}$$

Now, define $\mathbf{y} = A\mathbf{x}'$. Then, we can re-write the integral.

$$[f^A * g](\mathbf{x}) = |A^{-1}| \int f(A\mathbf{x} - \mathbf{y})g(A^{-1}\mathbf{y})d\mathbf{y} \tag{63}$$

$$= |A^{-1}|[f * g^{A^{-1}}](A\mathbf{x}) \tag{64}$$

■

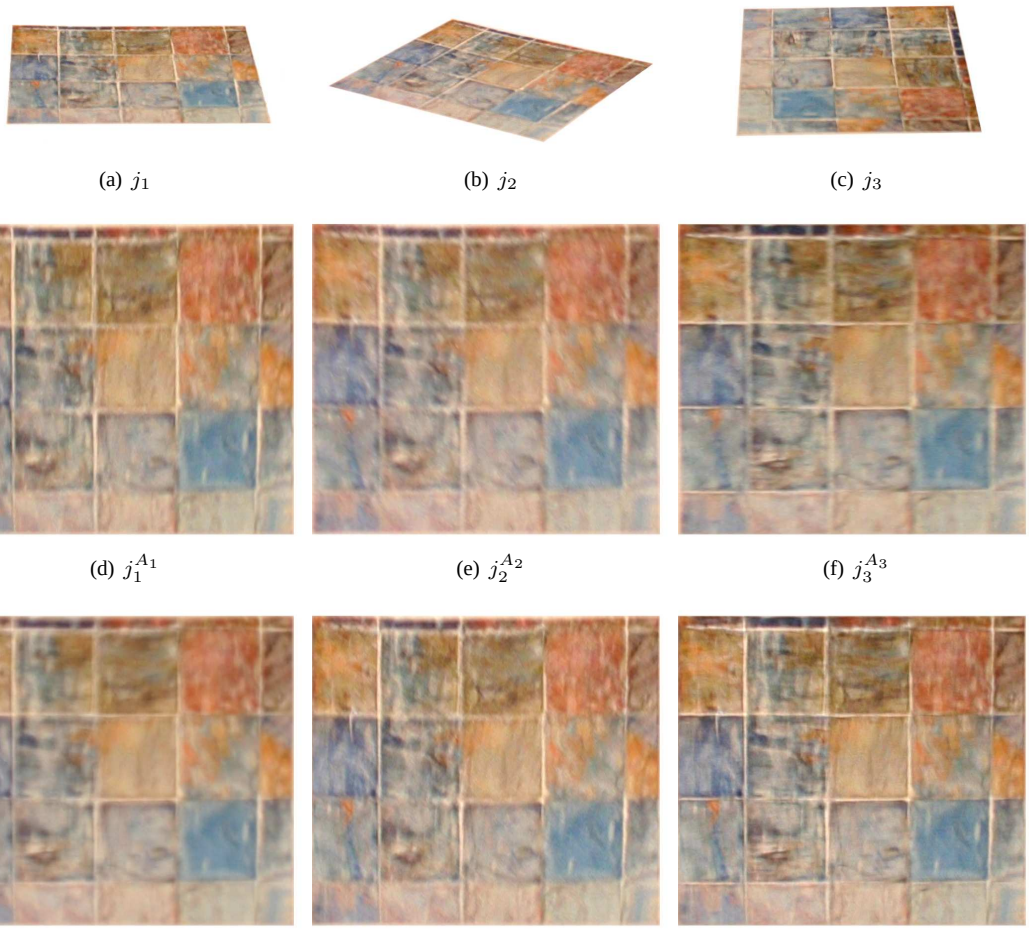
Theorem 7 (Affine Blurring Theorem):

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

Proof:

By definition, $i_2(\mathbf{x}) = i_1^{A^{-1}}(\mathbf{x})$. Hence,

Figure 11. Projective Reconstruction (Real images.)



(a) j_1 (b) j_2 (c) j_3

(d) $j_1^{A_1}$ (e) $j_2^{A_2}$ (f) $j_3^{A_3}$

(g) Average of $j_1^{A_1}$, $j_2^{A_2}$, and $j_3^{A_3}$ (h) Reconstuction from j_1 and j_2. (i) Reconstuction from j_1, j_2, and j_3.

$$j_2^A(\mathbf{x}) = [i_2 * e](A\mathbf{x}) = [i_1^{A^{-1}} * e](A\mathbf{x}). \tag{65}$$

Now, apply the affine convolution lemma to the right hand side.

$$[i_1^{A^{-1}} * e](A\mathbf{x}) = |A|[i_1 * e^A](A^{-1}A\mathbf{x}) \tag{66}$$

■

Theorem 8 (Fourier Affine Blurring Theorem):
$$\mathcal{F}\{j_2^A(\mathbf{x})\} = [I_1 \cdot E^{A^{-T}}](\mathbf{u})$$

Proof:
Start with the result of the affine blurring theorem.

$$j_2^A(\mathbf{x}) = |A|[i_1 * e^A](\mathbf{x}) \tag{67}$$

Now, apply the affine Fourier theorem to e^A.

$$\mathcal{F}\{e^A(\mathbf{x})\} = |A^{-1}|E^{A^{-T}}(\mathbf{u}) \tag{68}$$

The result follows from applying the convolution theorem to both sides of the affine blurring theorem. ■