

A roadmap to the integration of early visual modules

Abhijit S. Ogale and Yiannis Aloimonos*

Abstract

By examining the problem of image correspondence (binocular stereo and optical flow) and its relationship with other modules such as segmentation, shape and depth estimation, occlusion detection, and local signal processing, we argue that early visual modules are entangled in chicken-and-egg relationships, and unraveling these necessitates a compositional approach. In this paper, we present compositional algorithms which can match images containing slanted surfaces and images having different contrast, while simultaneously solving other problems as part of the same process. Ultimately, our goal is to motivate the application of the compositional approach to unify many other early visual modules. Experimental results have been presented on a large variety of stereo and motion images, including images with contrast mismatch and images containing untextured slanted surfaces.

1. Introduction

Early vision can be thought of as a collection of modules which deal with the estimation of quantities such as binocular disparity, optical flow (image motion), texture, occlusions, depth, shape, and various segmentations. It has long been known that these and other modules are intertwined in a chicken-and-egg fashion. In this paper, we focus on integrating a small subset of such early modules which are related

*A. S. Ogale and Y. Aloimonos are with the Center for Automation Research, University of Maryland, College Park, MD 20742. Email: ogale@cfar.umd.edu, yiannis@cfar.umd.edu

to the problem of image correspondence (i.e., stereo matching and optical flow).

The literature on stereo correspondence is extensive, and recently, Scharstein and Szeliski [1] have provided an exhaustive comparison of many dense stereo correspondence algorithms. Briefly, two categories of algorithms may be identified: (a) window-based approaches [2, 3, 4, 5, 6, 7] that aggregate local information within fixed, adaptive or multiple windows, and (b) global approaches, such as dynamic programming [8], simulated annealing [9, 10], relaxation labeling [11], non-linear diffusion [12], maximum flow [13] and graph cuts [14, 15] that compute the extrema of a global energy function, which includes terms for local data matching, additional smoothness terms, and in some cases, penalties for occlusions. Recently, Egnal and Wildes [16] have also provided comparisons of various methods for finding occlusions.

As is the case with stereo correspondence, there also exists a large body of literature devoted to the understanding of the optical flow problem, including the dense flow estimation problem. Beauchemin et al. [17] and Mitiche et al. [18] provide surveys of the various techniques for optical flow estimation, while Barron et al. [19], and more recently, Galvin et al. [20] and Liu et al. [21], compare various optical flow algorithms. [18] also discusses the problems of finding motion based segmentation and occlusions and surveys related approaches. There also exists some recent work on explicitly finding occlusions and motion discontinuities [22, 23] and studying the spectral properties of occlusions [24].

2. Some problems entangled with correspondence

2.1. Occlusions

Establishing point correspondence between a pair of images involves pairing a point in one image to a unique point in the second image. Due to changes in visibility, there are some points in both images which cannot be mapped to any corresponding point; these points are said to be occluded. *Thus, the problem of establishing image correspondence is intimately connected to the problem of finding the occlusions or points without correspondence.*

2.2. Segmentation

Local evidence, which is obtained by computing some measure of similarity between points in the two images, provides the foundation for computing point correspondence. However, such local evidence on its own is generally noisy and ambiguous and cannot be used directly for making a decision; clearly, the only recourse is to aggregate local information. The central question is how this aggregation (or smoothing) must be performed. In order to answer this, we must first understand that establishing point correspondence is not an end in itself, but a process which leads to descriptions of the scene in terms of smooth surfaces separated by sharp depth discontinuities. In order to obtain such descriptions, we must aggregate local evidence among points which lie on the same surface, but not among points which belong to different surfaces separated by depth discontinuities. *Since we lack knowledge of the depth discontinuities to begin with, finding image correspondence also becomes the problem of finding the depth discontinuities.*

2.3. Shape

Further complicating the task of point correspondence is the fact that real images are made up of pixels which are not ideal points but have a finite size. This fact is often ignored, and several algorithms focus on matching pixels instead of points, while occluded pixels are found by enforcing the constraint that a pixel in the first image cannot correspond to more than one pixel in the other image and vice versa. However, as discussed later in Section 5, when two cameras in a stereo system see a slanted surface, the projection of the surface has a different width in the two images, i.e., the image of the same surface can have a width of 20 pixels in the first image and 30 pixels in the second. It is clear that surface shape directly affects how that surface is sampled (or imaged). *If surface slant at every pixel were known in advance, we would know exactly the number of pixels in the other image which should be matched to a pixel in the first image, and then we would also find the occlusions correctly. Since we do not know the surface slant in advance, it must be estimated along with the image correspondence and occlusions, while concurrently re-interpreting image samples.*

2.4. Texture

It is a well known fact that human observers can easily perceive depth even when the contrast of the image seen by one eye is quite different from the contrast of the image seen by the other eye [25]. Recently, mutual information has been used [26] to perform contrast-invariant matching using graph cuts. Biologically motivated techniques which compute disparity using phase differences [27, 28, 29, 30, 31, 32] also possess this property, but often lack a global mechanism which is essential for finding occlusions and depth discontinuities.

Using phase similarity for contrast-invariant image correspondence requires us to analyse the local image structure in terms of many spatial frequency channels using filters such as Gabor filters. This means that we are essentially using local texture for matching instead of measurements like image brightness. To decide whether two pixels match in a particular frequency channel, their phase difference is usually found for that particular channel, and an overall measure of local matching can be computed using the phase differences from all available channels. We use such an approach but only to provide local evidence which is used by a global framework to make decisions.

As we shall see later in Section 5, slanted surfaces again cause a problem in this case: a texture present on a slanted surface is sampled differently in two images obtained from shifted viewpoints, and therefore interpreting the samples in any region of an image requires knowing the slant in that region. This dependency once again calls for a compositional approach.

3. Compositional Computation of Correspondence, Discontinuities and Occlusions

In this section, we discuss the compositional computation of image correspondence, depth discontinuities and occlusions, in three steps. In the first step, we present a compositional algorithm which relies on a simple binary local matching metric, and illustrates the idea behind concurrent estimation of correspondence, discontinuities and occlusions. In the second and third step, we progressively generalize this algorithm to deal with non-binary local matching metrics while preserving its compositional flavor.

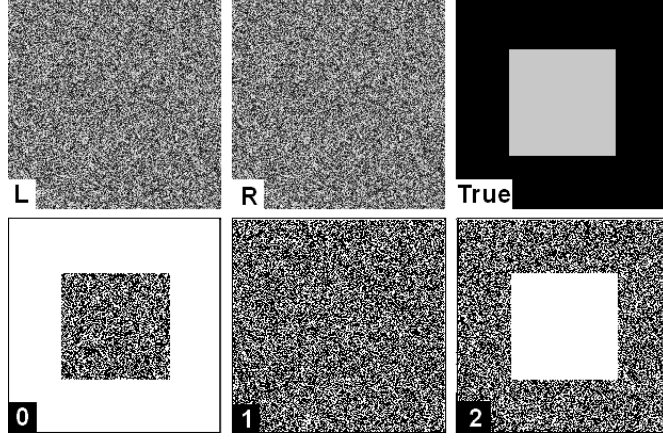


Figure 1. A random dot stereo pair. Top row: Left image, Right image, True disparity (black denotes 0, gray denotes 2). Bottom row: Pixel matches $M(x, y, x + \delta_x, y)$ for horizontal shifts $\delta_x = 0, 1, 2$. White regions denote matching pixels.

3.1. Matching algorithm using binary local evidence

If $I_1(x, y)$ and $I_2(x, y)$ be a given pair of grayscale images, then we can design a simple binary function $M(x, y, x + \delta_x, y + \delta_y)$ described below, which specifies whether a pixel (x, y) in the first image locally matches a pixel $(x + \delta_x, y + \delta_y)$ in the second image.

$$M(x, y, x + \delta_x, y + \delta_y) = |I_1(x, y) - I_2(x + \delta_x, y + \delta_y)| < t \quad (1)$$

Here, t is a thresholding parameter. In Figure 1, the first row shows a random dot stereo pair and the true disparity, while the second row shows $M(x, y, x + \delta_x, y)$ for $\delta_x = 0, 1, 2$ and $\delta_y = 0$. If we observe this function M , we notice that a large connected group of matching pixels (white background) is formed for $\delta_x = 0$, and another large connected group (white central square) is formed for $\delta_x = 2$. We observe that these two groups correctly correspond to two surfaces in the actual scene, and the boundaries of the groups are discontinuities in the true disparity map. This observation leads to the following simple proposition for jointly finding image correspondence and discontinuities if we apply connected components labeling to each binary-valued function $M(x, y, x + \delta_x, y + \delta_y)$.

If $S(x, y, x + \delta_x, y + \delta_y)$ denotes the size of the connected component containing pixel (x, y) obtained

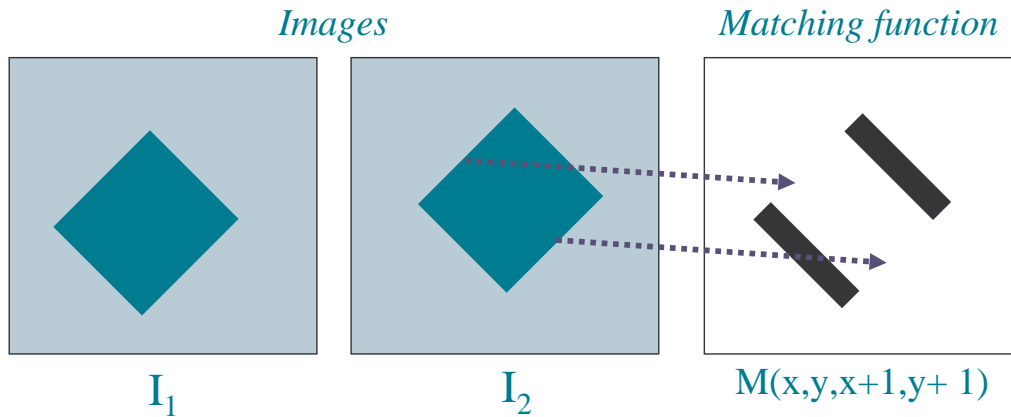


Figure 2. On the left, a pair of images I_1 and I_2 is shown, and on the right, $M(x, y, x + \delta_x, y + \delta_y)$ is shown for $\delta_x = 1, \delta_y = 1$. The dotted arrows show that two depth discontinuities parallel to the direction of (δ_x, δ_y) can be seen in the images, but are not seen in $M(x, y, x + \delta_x, y + \delta_y)$.

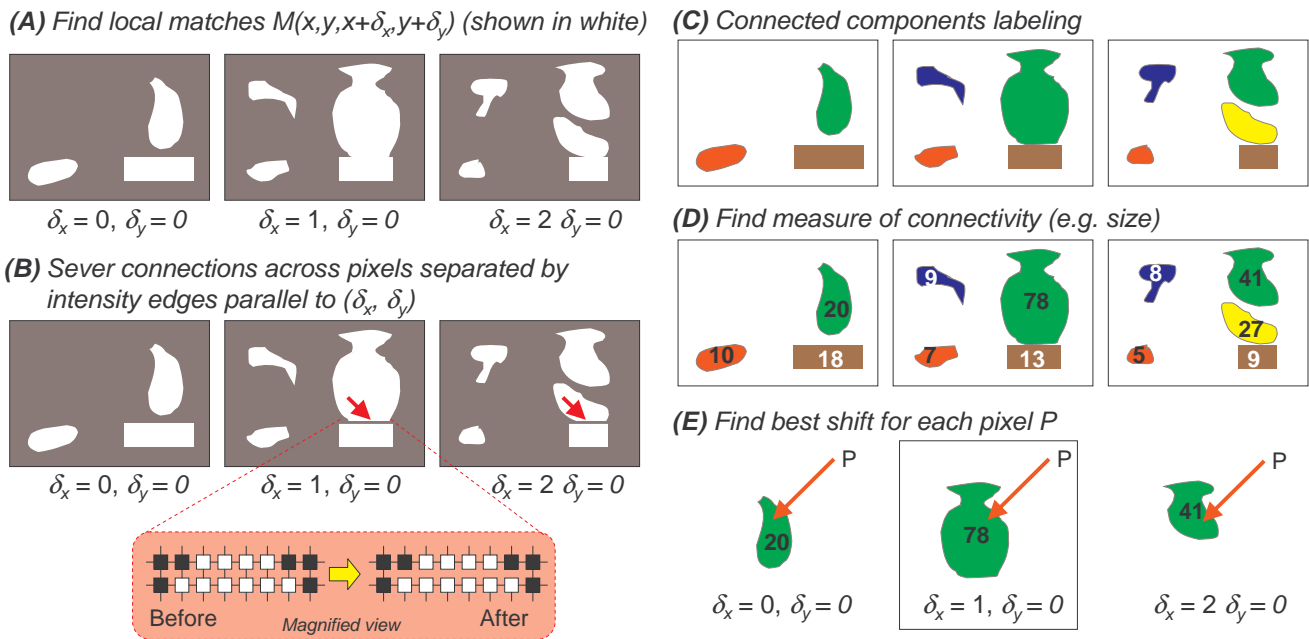


Figure 3. The matching algorithm. (A) Compute local matches M for each shift (B) Sever connections as described in Section 3.1. Notice the removal of vertical connections in the magnified view. (C) Find connected components (D) Find their sizes (E) For each pixel, pick the shift which corresponds to the largest connected component containing that pixel, while respecting the uniqueness constraint.

by labeling $M(x, y, x + \delta_x, y + \delta_y)$, then the correct shift $\vec{\delta} = (\delta_x, \delta_y)$ for this pixel is obtained when $S(x, y, x + \delta_x, y + \delta_y)$ is maximum, i.e.,

$$\vec{\delta}(x, y) = \arg \max_{\vec{\delta}} (S(x, y, x + \delta_x, y + \delta_y)) \quad (2)$$

This assumption holds true when the scene contains surfaces without slant; as we shall see later, it has to be modified for dealing with slanted surfaces. There is also one other problem with this proposition: it relies on discontinuities in the local matching function M to find depth discontinuities. However, there exist some depth discontinuities which are not visible in M : e.g., if two untextured regions are separated by a depth discontinuity which lies along a certain direction $\vec{\eta}$, then for any flow candidate $\vec{\delta}$ parallel to $\vec{\eta}$, the discontinuity will not be visible in $M(x, y, x + \delta_x, y + \delta_y)$. An example of this case is illustrated in Figure 2. Clearly, if we persist in solely using M to find depth discontinuities, the type of discontinuity described above will be missed, the reason being that the *information about such discontinuities is not binocular but resides within an individual image*. To remedy this, whenever we are performing connected components labeling of $M(x, y, x + \delta_x, y + \delta_y)$ for a certain shift $\vec{\delta} = (\delta_x, \delta_y)$, we must first sever connections across any pair of pixels which are separated by an intensity edge exactly parallel to $\vec{\delta}$. Intensity edges can be found by using a standard edge detector, such as the Canny edge detector, on the input images, and edges parallel to each $\vec{\delta}$ may be easily identified.

To find occlusions, the *uniqueness constraint* is used, which enforces a one-to-one correspondence between pixels in the two images. It is enforced within the correspondence search itself as it progresses: whenever we assign a new partner to a given pixel, we make sure that it's previous partner (if it was previously paired) is marked as unpaired. The matching algorithm is presented graphically in Figure 3. The only difference between stereo matching and optical flow computation is that only horizontal shifts δ_x are considered for stereo matching with $\delta_y = 0$. For the case of optical flow, the shifts are two dimensional, and in step (B) of the figure, connections across edges parallel to the flow being considered must be severed. In [33], the same principle of large connected components is used for stereo matching, but the connectivity constraints which we have described above are not imposed.

3.2. Generalization to non-binary local evidence: 1D case

In the previous section, we applied connected components labeling to a binary measure of local matching, which was obtained by thresholding the absolute intensity difference between the input images for various relative positions. There are two problems which arise as a result of using this method: the first problem is the use of a hard threshold to compute the binary measure which makes the result sensitive to threshold selection, and the second problem is the inability to use more general continuous measures of local matching. Therefore, although the method in the previous section transparently demonstrates the ingredients of a compositional strategy, there is a need to generalize it further to address these two problems.

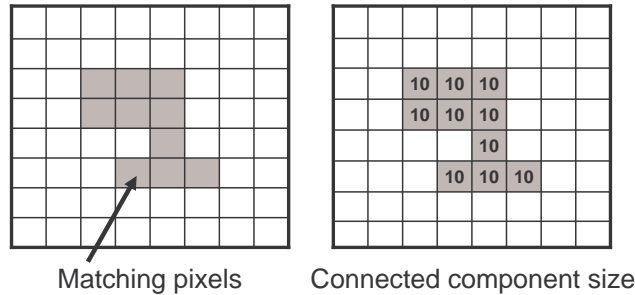


Figure 4. Connected component (left) and its size (right) assigned to each constituent pixel

In order to do this, let us first understand what we really do when we use the connected component size as a decision-making tool for correspondence. In Figure 4, the gray pixels in the left sub-figure denote a connected component of matching pixels, which may be obtained by labeling $M(x, y, x + \delta_x, y + \delta_y)$ for some value of $\vec{\delta}$. Then, the size of the connected component, which is 10 pixels in this case, lends support for assigning this particular flow or disparity $\vec{\delta}$ to each of the pixels within this connected component. To denote this, we have shown the size 10 being assigned as a weight to each pixel in this connected component in the right sub-figure. It represents the following fact:

Each matching pixel receives support from every other matching pixel which is reachable from it. In other words, matching pixels perform two functions: they PROVIDE support to other matching pixels, AND, they TRANSPORT (or CONDUCT) support from one matching pixel to another.

To give a loose physical analogy, we can think of each matching pixel as a provider of heat, as well as a conductor of heat. The quantity of heat it provides as well as its conductivity is directly related to how good the match is. Thus, if we have a matching pixel, it generates heat which is transported to the whole image by neighboring pixels which act as heat conductors. This may be computed explicitly by a computationally expensive diffusion process. However, we present below a less general but efficient method, which has the same computational complexity as the connected components method presented earlier.

To understand this method, let us first consider the simple problem of matching one-dimensional scan-line images $I_1(x)$ and $I_2(x)$, instead of our usual 2D images. At each pixel x and for each flow/disparity d , we wish to compute a goodness measure $G(x, d)$ which mimics the function of the connected components size, but which can utilize non-binary local evidence. We again define $M(x, d)$ as a local measure of how well the pixel x in I_1 matches pixel $x + d$ in I_2 , and can be thought of as the amount of ‘heat’ which each matching pixel *provides*. But since a pixel not only provides but also transports or conducts ‘heat’ provided by other matching pixels, we must specify a conductivity function $C(x, d)$, which specifies how well each matching pixel *conducts*. In practice, both $M(x, d)$ and $C(x, d)$ are closely related, since they both depend on how well the pixel x matches for a given shift d .

In order to create an efficient algorithm for computing the goodness measure $G(x, d)$, we split the computation into two parts. For each pixel, we first compute the contributions coming from its left in $G_{Left}(x, d)$, and then the contributions coming from its right in $G_{Right}(x, d)$. These two contributions are used together to find the complete $G(x, d)$. For the leftmost pixel, G_{Left} is initialized to its value of M , and for the rightmost pixel, G_{Right} is initialized to its value of M .

Thus, the steps for computing $G(x, d)$ are as follows:

1. From the left to the right, we can compute $G_{Left}(x, d)$ as follows:

$$G_{Left}(x, d) = G_{Left}(x - 1, d)C(x, d) + M(x, d) \tag{3}$$

Absolute intensity differences for some shift d

2	3	1	21	4	1	3	3
---	---	---	----	---	---	---	---

Diffusion

Connected components

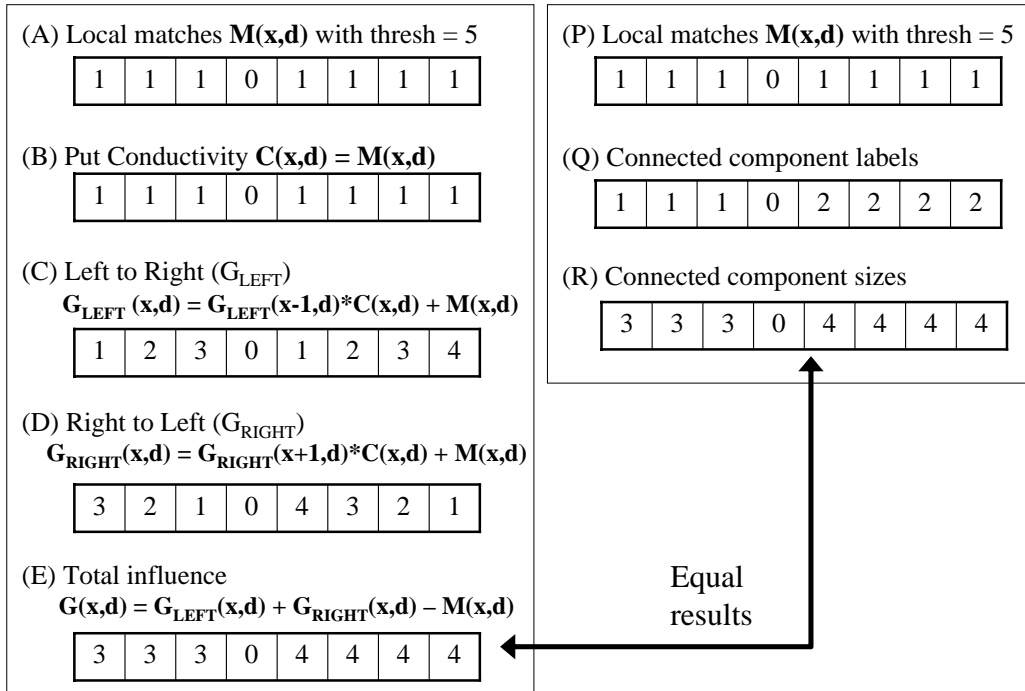


Figure 5. Connected components as a special case of the *provider-conductor* process. On the top, we show absolute intensity differences for some relative shift d . On the left (A to E), we show how a measure of the influence of matching pixels on each other is computed using conduction. Note that for simplicity, we have set $C(x, d) = M(x, d)$. On the right, we see how the same result is obtained using connected component sizes.

2. Similarly, from the right to the left,

$$G_{Right}(x, d) = G_{Right}(x + 1, d)C(x, d) + M(x, d) \quad (4)$$

3. Hence, the total influence from the left and the right is

$$G(x, d) = G_{Left}(x, d) + G_{Right}(x, d) - M(x, d) \quad (5)$$

In the first two steps, we can see that the reason for the efficiency of the algorithm is that each matching pixel performs its two functions of being a *conductor* and a *provider* simultaneously. Note that in (5), we subtract $M(x, d)$ since it was counted twice, once in G_{Left} and once in G_{Right} .

Figure 5 shows an example computation of $G(x, d)$ for given values of $M(x, d)$ and $C(x, d)$; by deliberately choosing the input $M(x, d)$ to be binary, the figure also shows how this *provider-conductor* process gives the same result as the previous connected components process if M is binary. But we can see immediately that unlike the connected components, the *provider-conductor* process is more general since it can utilize non-binary continuous-valued M functions as well. The final matching algorithm for the 1D case is shown below. It takes as input the two scanline images $I_1(x)$ and $I_2(x)$, and set of possible shifts S , and outputs the result $\delta(x)$. The uniqueness constraint is also enforced to find the half-occlusions in a single-pass through all the shifts.

1. for each shift $d \in S$, do

(a) Find $M(x, d)$ and $C(x, d)$ using the input images.

(b) Compute $G(x, d)$ using (3), (4), (5)

2. For each pixel x , find $\delta(x) = \underset{d}{argmax} [G(x, d)]$ subject to the uniqueness constraint (to find occlusions).

3.3. Generalization to non-binary local evidence: 2D case

To generalize the above approach to 2D images without losing its computational edge, we have experimented with a few variants of the theme which will be discussed below, but the gains in the quality of results have been modest at the cost of increased running time; therefore, we shall present here only the simplest 2D variant.

At each pixel (x, y) , when we consider a disparity/flow $\vec{\delta} = (\delta_x, \delta_y)$, we must first use the above 1D algorithm in a line of pixels parallel to $\vec{\delta}$ to compute $G_{\parallel}(x, y, \vec{\delta})$, and then in a line of pixels perpendicular to $\vec{\delta}$ to compute $G_{\perp}(x, y, \vec{\delta})$. Instead of adding the two contributions together to get $G(x, y, \vec{\delta})$, we multiple them together to avoid the effects of unequal image height and width which complicate direct addition. (In practice, we add their logarithms together.) Thus,

$$G(x, y, \vec{\delta}) = G_{\parallel}(x, y, \vec{\delta}) * G_{\perp}(x, y, \vec{\delta})$$

The computation of G_{\parallel} is performed for a line of pixels parallel to $\vec{\delta}$ exactly as in the 1D case. In the direction perpendicular to $\vec{\delta}$, things are slightly different. Recall from Section 3.1 that depth discontinuities which are parallel to $\vec{\delta}$ are not visible in the local matching function $M(x, y, x+\delta_x, y+\delta_y)$, and therefore we had explicitly severed connections between pixels which were separated by intensity edges parallel to $\vec{\delta}$ before forming connected components. Likewise, to compute a goodness function $G_{\perp}(x, y, \vec{\delta})$ for a line of pixels in a direction perpendicular to $\vec{\delta}$, we must *weaken* connections across neighboring pixels which are separated by intensity edges parallel to $\vec{\delta}$.

To achieve this, we first define $E_{\vec{p}, \vec{q}}(\vec{\delta})$, which denotes the strength of the link joining two neighboring pixels \vec{p} and \vec{q} for a given value of $\vec{\delta}$, where \vec{p} and \vec{q} lie on a line perpendicular to $\vec{\delta}$. We want $E_{\vec{p}, \vec{q}}(\vec{\delta})$ to be weak if the intensity gradient at location $(\vec{p} + \vec{q})/2$ is perpendicular to $\vec{\delta}$, which means that the edge direction is parallel to $\vec{\delta}$. We shall discuss the computation of $E_{\vec{p}, \vec{q}}(\vec{\delta})$ below, but for the moment, assume that we have such a function. Since this function $E_{\vec{p}, \vec{q}}(\vec{\delta})$ is like a conductivity for the edges, we can use our previous 1D algorithm again for this case by substituting *ghost pixels* in place of the edges. This process, which is illustrated in Figure 6, shows how we perform the construction for a line of pixels

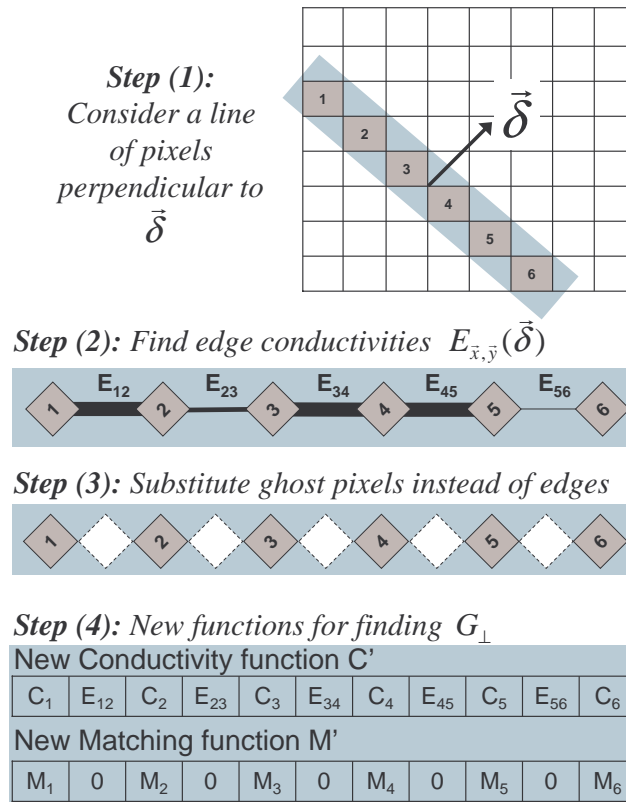


Figure 6. Creating ghost pixels for computing G_{\perp} : (1) Select a line of pixels perpendicular to $\vec{\delta}$. (2) Assign edge weights to edges connecting nearest pixels. Weights are weaker if image edges parallel to $\vec{\delta}$ pass between two pixels. (3) Substitute ghost pixels in place of edges in between regular pixels. (4) Using the regular pixel local matching M and conductivity C , and edge weights E , create M' and C' . At ghost pixels, M' is 0 and $C' = E$, while at normal pixels, these values are the same as their usual M and C values. The new M' and C' values can now be used in the 1D algorithm to find G_{\perp} .

perpendicular to $\vec{\delta}$. In the first step, we pick our line of pixels, while in the second step, we have arranged these pixels in a row, with the edges connecting neighboring pixels. The thickness of the edges has been used to denote the strength of $E_{\vec{p},\vec{q}}$. Given local match values M and conductivity C for each pixel as in the 1D case, we now have these extra edge weights $E_{\vec{p},\vec{q}}$ which control the transport or conduction along edges. In the third and fourth steps, we show that if we create ghost pixels which alternate between real pixels, then we can create new match and conductivity values M' and C' which can be used as an input for the 1D algorithm. For regular pixels, M' and C' are the same as their regular M and C values, and at the ghost pixels, $M' = 0$ (since an edge only *transports* and does not *provide*) while C' is set to the value of $E_{\vec{p},\vec{q}}$ for that edge. The result of the 1D algorithm using M' and C' as input, yields $G_{\perp}(x, y, \vec{\delta})$ as the output. Obviously, the values of G at the ghost pixels are discarded in the final output, before storing it in $G_{\perp}(x, y, \vec{\delta})$.

Now we give an example of how to compute the function $E_{\vec{p},\vec{q}}(\vec{\delta})$. If $\vec{g}_1(\vec{r}) \equiv (g_1(\vec{r}), \phi_1(\vec{r}))$ and $\vec{g}_2(\vec{r}) \equiv (g_2(\vec{r}), \phi_2(\vec{r}))$ denote the intensity gradient (magnitude, direction) computed from images I_1 and I_2 respectively at location \vec{r} , and if $\vec{\delta}$ has direction ϕ , then one possible choice of $E_{\vec{p},\vec{q}}(\vec{\delta})$ is shown below:

$$F_i(\vec{r}, \vec{\delta}) = \cos^2(\phi - \phi_i(\vec{r}))(1 - e^{-\lambda g_i(\vec{r})}) + e^{-\lambda g_i(\vec{r})} \quad ; \quad \text{where } i = 1, 2$$

$$E_{\vec{p},\vec{q}}(\vec{\delta}) = \min \left\{ F_1(\vec{r}, \vec{\delta}), F_2(\vec{r} + \vec{\delta}, \vec{\delta}) \right\} \quad ; \quad \vec{r} = (\vec{p} + \vec{q})/2$$

More sharply tuned functions can be obtained by substituting $(1 - \sin^{2n}(\phi - \phi_i(\vec{r})))$ for $n > 1$, in place of $\cos^2(\phi - \phi_i(\vec{r}))$ in the above equation. Note that λ is a parameter which controls the trustworthiness of the gradient angle according to the gradient magnitude.

4. Designing a contrast-invariant Local Matching Measure

In the previous section, we have presented compositional algorithms for correspondence which can utilize continuous-valued measures of local matching within a global framework, in order to compute correspondence, discontinuities and occlusions. In this section, we develop a local measure of matching which is insensitive to changes in contrast between the two images, which can be provided to the

above algorithms as an input. To create such a measure, phase difference information obtained from Gabor filters at various scales, spatial frequencies and orientations is used, which is reminiscent of early processes in biological vision. Unlike most phase-based algorithms (see Section 2.4), we use phase differences only as a local measure of matching, while we have a separate global framework to make decisions about the correspondence and the occlusions. One advantage of this approach is that since we are not using the phase to explicitly compute correspondences but only as a local measure, we do not need filters with large spatial extent in order to deal with large disparities, hence depth discontinuities can be detected with greater accuracy.

Let us understand the process by considering the problem in one dimension since the extension to two dimensions is straightforward. Given two 1D images $I_a(x)$ and $I_b(x)$, our task is to assign a shift d to each pixel, where $d \in \{d_1, d_2, \dots, d_n\}$, with the range of possible shifts specified as an input. We begin by applying complex valued Gabor filters of the form $g_{x_0, \omega}(x)$ to both the images, where the filter is centered at x_0 in space, and at ω in the frequency domain. For example, the filter centered at $x_0 = 0$ having a frequency ω is given by:

$$g_{0, \omega}(x) = e^{-x^2/2\sigma^2} e^{i\omega x} \quad (6)$$

The real and imaginary parts of this complex-valued filter form a quadrature pair. We select σ to ensure a constant one octave bandwidth in the frequency domain. The space frequency domain can be sampled by a complete set of functions obtained by translation and scaling of this basic filter. In the two dimensional case (which we use in our implementation), rotation is also present, since the filters are oriented. The output of the filter with frequency ω (written as a convolution) on the two images I_a and I_b is denoted by

$$\begin{aligned} A_\omega(x) &= g_{0, \omega}(-x) \otimes I_a(x) \\ B_\omega(x) &= g_{0, \omega}(-x) \otimes I_b(x) \end{aligned} \quad (7)$$

Now assume that we are dealing with some shift d . If the phase difference computed between the two images using the responses of a particular filter with frequency ω at position x and disparity d is denoted by $\Delta\phi_{\omega,d}(x)$, then

$$e^{i\Delta\phi_{\omega,d}(x)} = \frac{A_{\omega}(x)B_{\omega}^*(x+d)}{|A_{\omega}(x)B_{\omega}^*(x+d)|} \quad (8)$$

Notice that by using $B_{\omega}^*(x+d)$ in the above equation, we are explicitly shifting the image I_b by an amount d , and hence if the two images locally match for this shift d in this frequency channel, we would expect the phase difference to be close to zero. The deviation of the phase difference from zero can therefore be implicitly used as a measure of local matching. This is unlike most of the previous approaches which utilize phase, since they use the phase differences either explicitly or through phase correlation to directly find the unknown disparity.

Fleet [30] discusses how phase correlation can be thought of as a voting scheme, such that when we take the inverse Fourier transform, each channel casts a vote in a sinusoidal manner in the spatial domain. The inverse Fourier transform using the outputs of filters centered at a spatial position x_0 is given by (ignoring the spatial extents of the filters for the moment):

$$F_{x_0,d}(x) = \int e^{i\Delta\phi_{\omega,d}(x_0)} e^{i\omega(x-x_0)} d\omega \quad (9)$$

Ideally, the real parts of all the sinusoids would sum up to create a single peak at a certain position, and the imaginary parts would all cancel out. To find the degree of local matching for a pixel x_0 in image I_a for the disparity d , we want to measure how close this peak lies to the center position x_0 of the applied filter. To achieve this, we can simply use the real part of the function $F_{x_0,d}(x_0)$ computed at $x = x_0$ as a measure of closeness of the peak to x_0 .

Thus, if we are applying N filters to the images, and the phase difference at location x from channel ω for disparity d is denoted by $\Delta\phi_{\omega,d}(x)$ derived in (8), then we can define a function $H(x, d)$ which sums the real parts of the inverse Fourier transform in the discrete case:

$$H(x, d) = \frac{1}{N} \sum_{N \text{ channels}} \cos(\Delta\phi_{\omega,d}(x)) \quad (10)$$

The factor $1/N$ is used to ensure that $H(x, d)$ becomes an average over cosines of all the phases, and therefore lies in the range $[-1, 1]$. Since phase relationships can become unreliable if the power in the selected frequency channels is close to zero, we define another function $J(x, d)$ as follows:

$$J(x, d) = W(x, d) \cdot H(x, d) + (1 - W(x, d)) \cdot (1) \quad (11)$$

$$\text{where } W(x, d) = \exp(-\alpha P(x, d)) \quad (12)$$

$$\text{and } P(x, d) = \sum_{\omega} |A_{\omega}(x)B_{\omega}^*(x + d)| \quad (13)$$

Here, $P(x, d)$ denotes the sum of the magnitudes (power) of all the filter response products, and $W(x, d)$ is a weight which exponentially decays to zero as $P(x, d)$ increases. Hence, as $P(x, d)$ tends to zero, the weighting function reduces the importance of the phase difference function $H(x, d)$ and forces $J(x, d)$ closer to the value 1, which indicates good matching. Thus, if the phase responses are unreliable due to the nonexistence of local variations in the intensity, we take the default position that there exists a local match. Note that $J(x, d)$ also lies in the range $[-1, 1]$. Since we would like to construct a non-negative measure $M(x, d)$ of local matching, we define

$$M(x, d) = \frac{J(x, d) + 1}{2} \quad (14)$$

where $M(x, d)$ lies in the range $[0, 1]$ and measures how well a pixel x in image I_a matches pixel $x + d$ in image I_b . In practice, we perform the Gabor filtering using 2D oriented filters using an efficient implementation by Nestares et al [34] at four different scales and four different orientations. To achieve

contrast invariant matching, we use the method defined in Sections 3.2 and 3.3, along with the values of $M(x, d)$ from eqn. (14) and with $C(x, d) = M(x, d)$.

5. Shape

In [35, 36], we have extensively discussed the effects of shape on the correspondence problem, and shown that shape estimation must be an integral part of a compositional solution to the correspondence problem. In this section, we shall very briefly review the basic ideas from these papers, and present some additional implications of shape which are relevant to the methods presented in this paper.

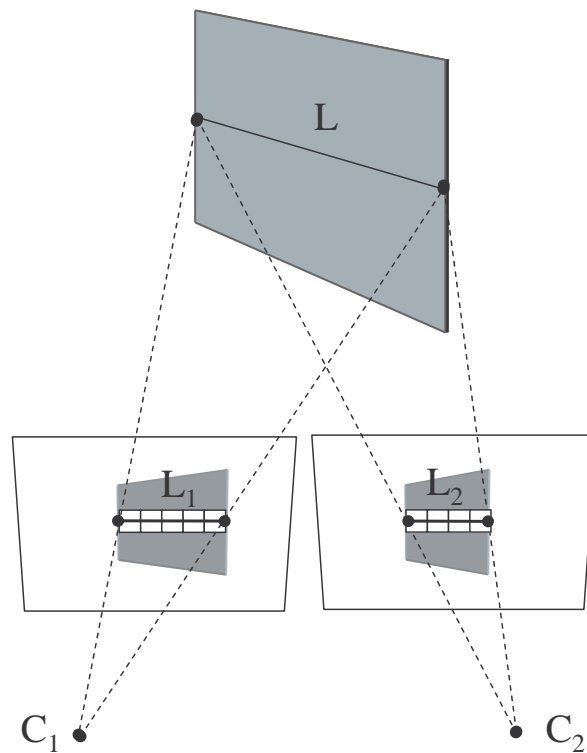


Figure 7. Images of a horizontally slanted surface seen by two horizontally separated cameras. A line segment L on the surface projects to a line segment L_1 of length 5 pixels in *Camera 1* and to a line segment L_2 in *Camera 2* of length 4 pixels.

Figure 7 shows how a slanted surface, which has depth variation in the direction of the separation of the two camera positions, appears stretched in one image as compared to the other. In this figure,

the cameras are horizontally separated, and the surface in view is slanted about a vertical axis (hence depth varies horizontally on the surface). The projected images in the two cameras clearly have different width. This observation brings out several important drawbacks of our previous approach (and of many other existing approaches) plus the required modifications:

- *Unequal correspondence*: The algorithms we have presented so far treated pixels as points, and aimed to correspond a pixel in one image to at most one pixel in the other. However, in this figure, four pixels in the second image are in correspondence with five pixels in the first image. Thus, it is clear that in the presence of slant, a pixel in one image can correspond to more than one pixel in the other image. As we demonstrate in [35], untextured slanted surfaces further compound this problem. Therefore, our *algorithms must be suitably modified to allow unequal numbers of pixels to correspond*.
- *Occlusions*: Previously, we had used the uniqueness constraint to enforce a one-to-one matching between pixels in the two images, which resulted in unpaired pixels being labeled as occluded. If we now allow different numbers of pixels from the two images to correspond to each other to correctly handle slant, our basis for finding occlusions falls apart. Detecting occlusions now necessitates the *formulation of a different uniqueness constraint* which does not apply to pixels, but to line segments.
- *Sampling*: Methods such as [37] are widely used by the stereo vision community as local measures of matching which are insensitive to image sampling, since they lead to dramatic improvements in the results. These methods correct for errors due to fractional pixel misalignment of the sampling grid in the two images. However, the sampling grids of the two images are still assumed to sample a scene surface with identical local density, i.e., the same number of equally spaced pixels. But as we have seen in Figure 7, this assumption is not valid for slanted surfaces, since they are sampled with an unequal number of pixels by the two cameras. This means that before using the samples to make local comparisons with misalignment corrections (as in [37]), we must use the slant to resample the images, so that local neighborhoods being compared are represented by the same

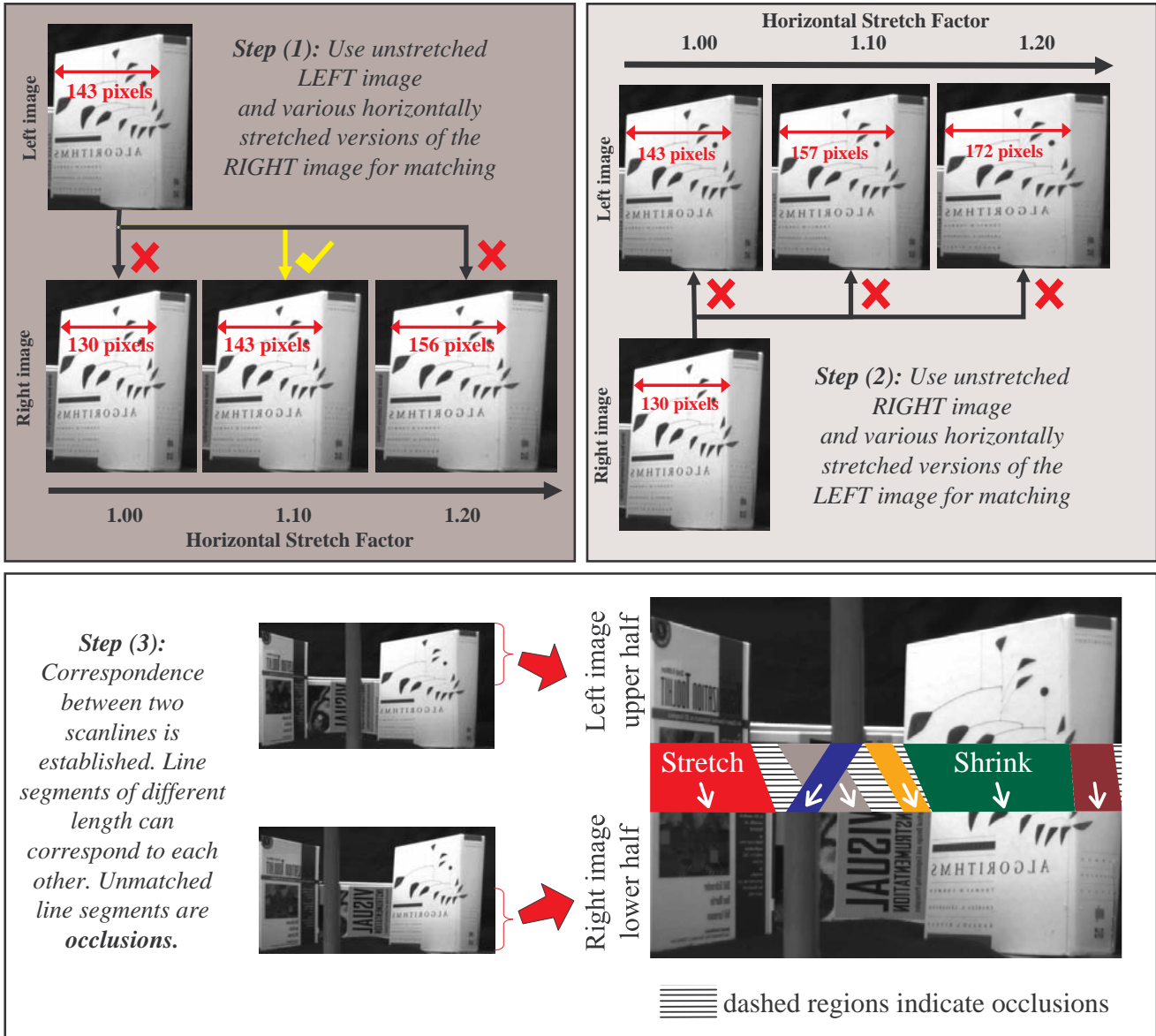


Figure 8. In the stereo case, exploring the space of horizontal slants involves three steps. *Step 1:* Run the algorithm in Section 3 with the original left image and with various horizontally stretched versions of the right image. *Step 2:* Then run the algorithm with the original right image and different stretched versions of the left image. In the above example, we can see that the width of the left and right images of a slanted book is same only in one of the trials (see the left pane above) where the right image is stretched by a factor 1.10 horizontally. *Step 3:* Correspondence is established between line segments, not pixels, while occlusions are unmatched line segments. For complete details, refer to [35].

number of samples. To do this, we need to know the local slant before we do correspondence; since we do not have this knowledge, *the only way is to try all possible slants*. Thus, *slant estimation must be performed concurrently with correspondence since it influences the interpretation and comparison of the local evidence itself*.

To address all these problems, the only thing we have to change in our algorithms from Section 3 is to search over the space of possible slants as well. For each possible slant, we stretch (resample) the images and apply the same matching algorithm as before from Section 3 to concurrently find the best slant and shift. Even the step of local matching is performed after the stretching step.

Figure 8 illustrates the essence of this algorithm by using two stereo images containing some slanted books. Horizontal slant is estimated in the correspondence process concurrently. For further details regarding this algorithm and the effects of slant in directions other than the horizontal, the interested reader is referred to [35]. It is important to mention here that one useful piece of information which we have not utilized in the current algorithm is the occurrence of orientation disparity in the presence of slant; we plan to include this in future work.

6. Experiments

In this section, we present experimental results obtained for several stereo and motion pairs and some quantitative evaluations. In Figure 9, we see the results obtained by running the algorithm discussed in Section 3.3 on three stereo pairs, and in Figure 10, we see the results of finding optical flow on four motion sequences. Occlusions can also be seen in these figures. Intensity differences were used as local evidence and computed using the method in [37]. Quantitative comparisons were performed on stereo pairs using the standard testbed provided at www.middlebury.edu/stereo by Scharstein and Szeliski [1]; the error percentages and ranks for the stereo pairs in this test suite are shown in Table 1.

Figure 11 shows the results on five stereo pairs with different left and right image contrasts. The local matching metric defined in Section 4 was used, and we performed Gabor filtering in two dimensions using an efficient implementation by Nestares et al [34], which uses an odd and even pair of filters with

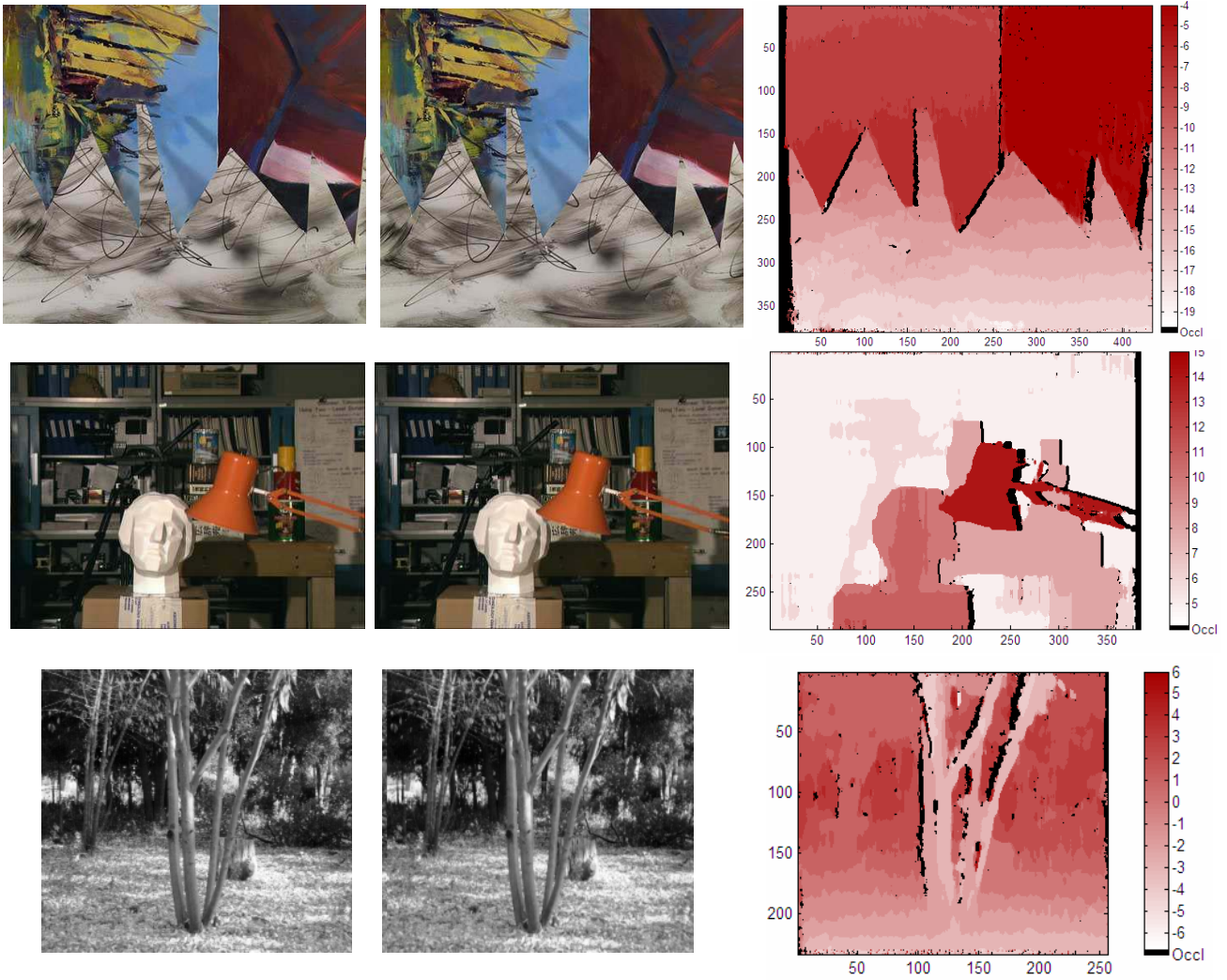


Figure 9. Results on three stereo pairs: *Sawtooth*, *Tsukuba*, and *SRI Trees*. Black regions are the occlusions.

Table 1. Results of the Middlebury stereo evaluation (error percentages and ranks).

Tsukuba			Sawtooth		
all	untex	disc.	all	untex	disc.
1.77	0.95	9.48	0.61	0.17	5.05
(6)	(5)	(7)	(4)	(11)	(6)
Venus			Map		
all	untex	disc.	all	disc.	
3.00	5.22	7.63	0.21	3.01	
(20)	(20)	(8)	(2)	(4)	

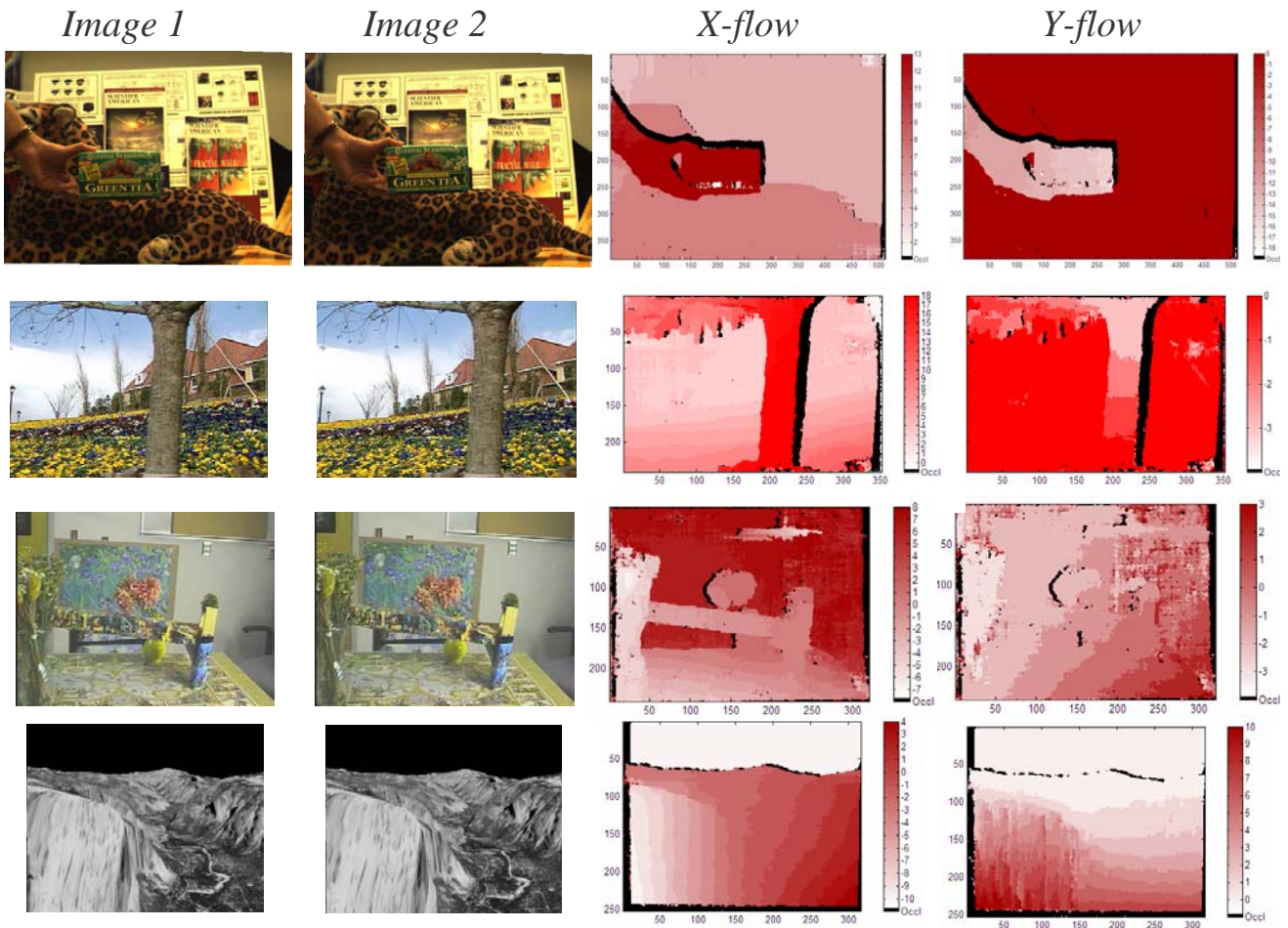


Figure 10. Results on four motion sequences: *leopard*, *flower-garden*, *table-vase*, and *yosemite*. Two frames are shown along with color-coded images of the x and y components of the optical flow. Black regions are the occlusions.

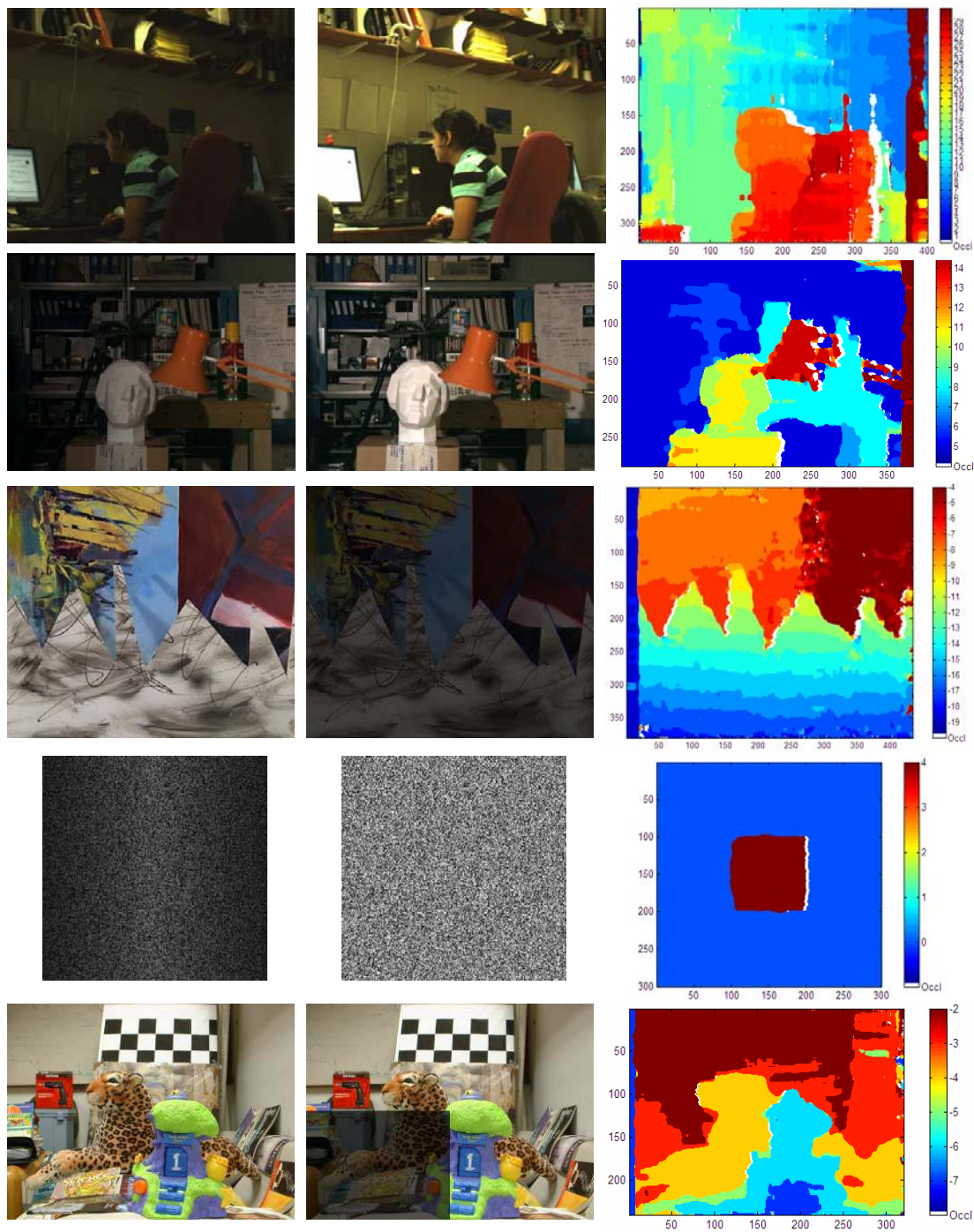


Figure 11. Row 1: True contrast variation: stereo images taken from cameras with different apertures and exposures. The disparity map is shown on the right. Row 2: *Tsukuba* stereo pair with a quadratic contrast variation across the left image. Row 3: *Sawtooth* stereo pair. Row 4: *Random dot* pair with a Gaussian contrast variation across the left image. Row 5: *Leopard* stereo pair with different contrast in a single square patch in the right image. In all cases, the occlusions are shown in white color.

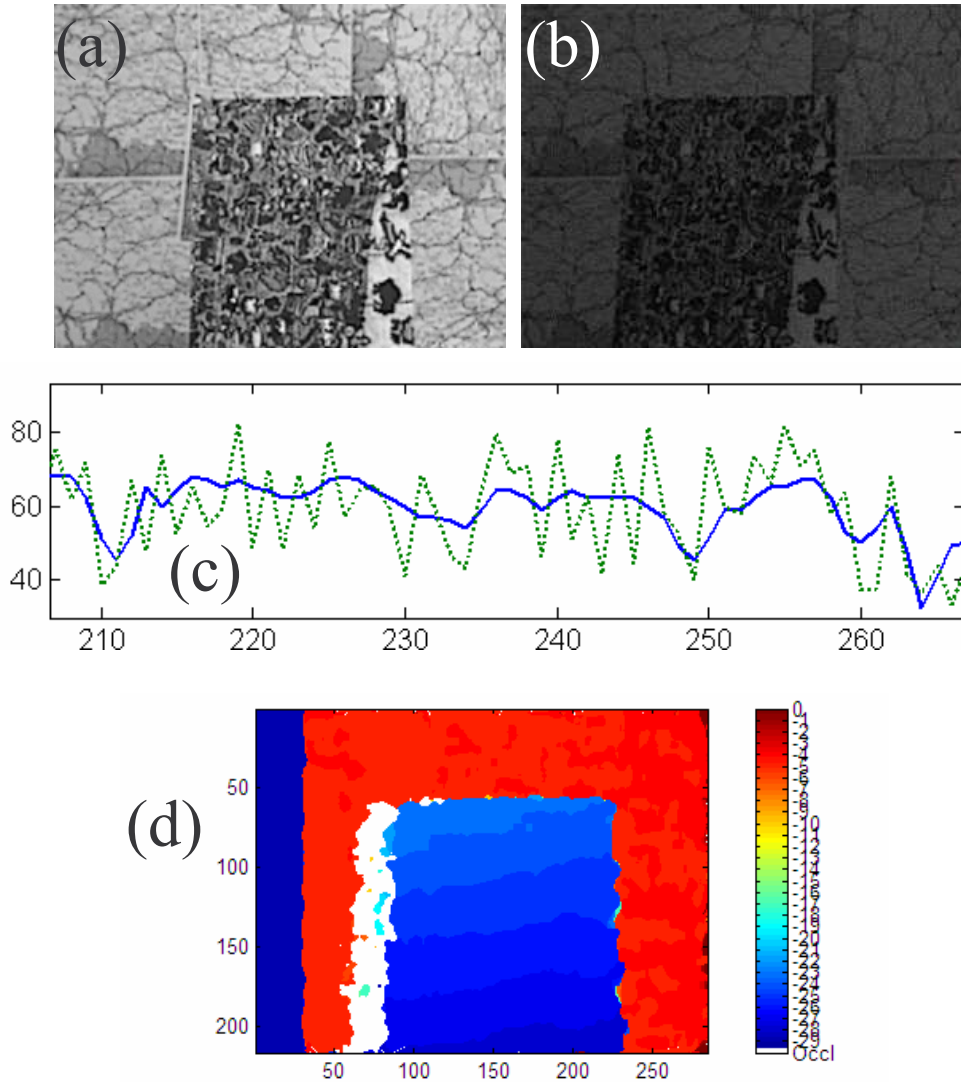


Figure 12. (a) Left image from the *map* sequence. (b) Right image with lower contrast and the addition of noise in the high frequency channel. The noise causes upto 25% variation in the intensity. (c) shows a portion of a scanline in the right image, where the solid line shows the intensity values before addition of the noise, and the dotted line shows the values after addition of the noise. (d) shows the results. Occlusions are shown in white.

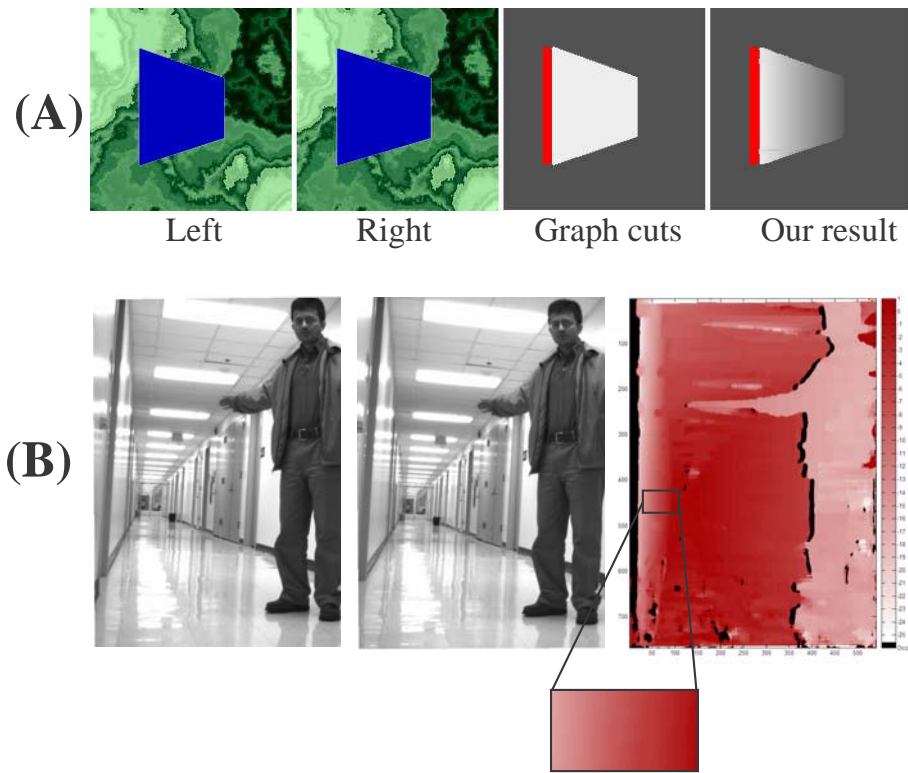


Figure 13. (A) Results for an untextured slanted surface: a synthetic stereo pair is displayed along with results using graph cuts [15] which yields the same disparity value throughout the slanted surface, and our result which finds the slant. Occlusions are colored red to match the output of the graph cuts software. (B) Results for a scene with untextured slanted surfaces and plenty of specularities. Occlusions are colored black. A small portion of the result has been magnified to show the computed disparity variation on the left wall.

a constant one octave bandwidth, at four scales (wavelengths are multiplied by two from one scale to the next) and four orientations (separated by 45°). The first pair in this figure was obtained from a real stereo system, in which the contrast mismatch is due to the different aperture and exposure settings of the two cameras. Each subsequent row shows stereo pairs with a different type of contrast mismatch, such as a spatially uniform mismatch, a smooth spatially varying mismatch, or a contrast mismatch which occurs only in one patch in the image. In Figure 12, we have added 25% noise in the high frequency region of the right image in addition to a change in contrast. The results show that the algorithm is relatively stable even if significant amounts of noise are added to one of the frequency channels. Existing quantitative comparisons of correspondence algorithms and the associated datasets do not deal with the issue of contrast changes, hence such comparisons will have to form a part of future work.

Finally, Figure 13 shows the results obtained on slanted surfaces. The top portion of the figure shows our results on a synthetic scene, and also the output of a graph cuts algorithm [15] for comparison. It can be seen that our approach recovers the slant, whereas the graph cuts computes a fronto-parallel surface. The bottom portion of this figure shows results on a difficult corridor scene with untextured slanted walls and several specularities.

7. Conclusion

We have presented compositional algorithms which concurrently solve related chicken-and-egg problems such as image correspondence, depth segmentation, slant estimation and occlusion detection. We have further extended this framework to obtain contrast-invariant correspondence by performing local matching using phase information from a bank of Gabor filters. Since we use phase differences for local matching only and not for explicitly computing the correspondence, we do not need filters of large spatial extent in order to compute large shifts, which prevents degradation of boundaries. The algorithm is able to handle significant changes in contrast between the two images even if the changes vary spatially over the image, and performs well in the presence of noise. These advantages were demonstrated by performing experiments under various input conditions. Overall, it has been the aim of this paper to

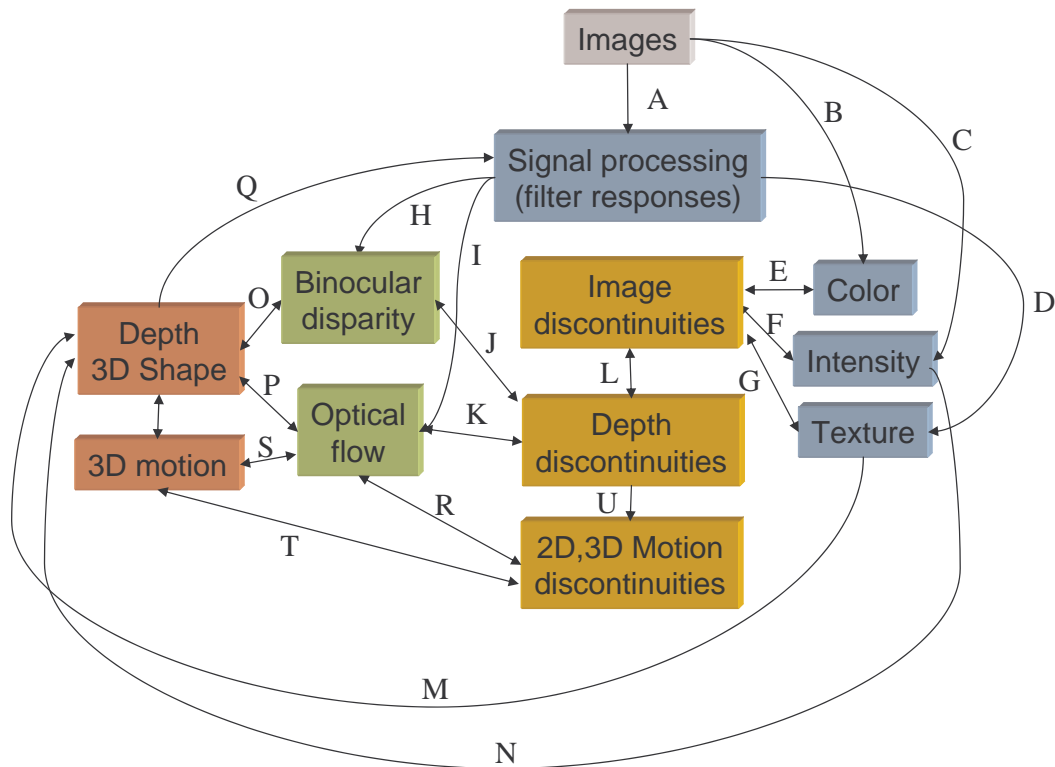


Figure 14. A Compositional Viewpoint. Explanation of each edge in the graph is as follows: (A,B,C) Image measurements such as filter responses, color or brightness (D) Filter responses collectively describe textures (E,F,G) Color, Intensity and Texture are used to identify discontinuities in a single image, which in turn influence perception of color or intensity. (H,I) Local evidence is used for binocular and motion correspondence (J,K) Binocular disparity and optical flow signal depth discontinuities (L) Image discontinuities are also needed when depth discontinuities are parallel to local flow/disparity. (M,N) Shape from texture and shading. (O,P) Disparity and flow yield structure (Q) Shape feeds back to influence local signal processing (R,S,T) Optical flow, 3D motion estimation and motion discontinuities influence each other. (U) Depth discontinuities may affect motion discontinuities.

motivate the integration of early visual modules with the hope of improving our understanding of vision by recognizing and harnessing the interdependencies among problems. Keeping this in mind, let us conclude by presenting a scenario of possible interactions among various visual processes.

Figure 14 describes the flow of information among various early modules related to the correspondence problem. At the beginning, various measurements such as color, intensity, and filter responses are gathered from the input images. The filter responses can be used as inputs to a texture analysis module, as well as local evidence for binocular stereo and motion correspondence modules. Stereo disparity and optical flow both affect and are affected by the estimation of depth discontinuities, shape and depth; therefore, stereo and optical flow will indirectly interact with each other. Shape from X (texture, shading) also influences the estimation of 3D shape. The estimation of depth discontinuities is also affected in certain directions by image discontinuities such as intensity, color or texture discontinuities, as we have seen in Section 3.1. Image discontinuities in turn may affect the perception of intensity, color and texture as well, which is indicated by certain visual illusions. Besides depth discontinuities, optical flow is also related to 3D motion discontinuities (i.e., boundaries of independently moving objects) and 3D motion estimation. (In [38], we also show how occlusions in the optical flow can independently be used to compute a depth ordering.) Ultimately, computed shape feeds back to affect the interpretation of local measurements and samples, and we return back to where we started.

The above scenario can be expanded to contain many more modules, interactions and dependencies than those we have just described, which only furthers the need for compositional solutions and takes us closer to our goal of an integrated approach [39] to early vision.

8. Acknowledgements

The support of the National Science Foundation is gratefully acknowledged.

References

- [1] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7 – 42, April 2002.
- [2] D. Geiger, B. Ladendorf, and A. Yuille, “Occlusions and binocular stereo,” *European Conference on Computer Vision*, pp. 425–433, 1992.
- [3] M. Okutomi and T. Kanade, “A multiple baseline stereo,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 353–363, April 1993.
- [4] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: theory and experiment,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.
- [5] A. Fusiello, V. Roberto, and E. Trucco, “Efficient stereo with multiple windowing,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 858–863, June 1997.
- [6] A. F. Bobick and S. S. Intille., “Large occlusion stereo,” *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, Sept 1999.
- [7] H. Tao, H. Sawhney, and R. Kumar, “A global matching framework for stereo computation,” *International Conference on Computer Vision*, vol. 1, pp. 532–539, July 2001.
- [8] Y. Ohta and T. Kanade, “Stereo by intra- and inter-scanline search using dynamic programming,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 139–154, March 1985.
- [9] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, Nov 1984.

- [10] S. T. Barnard, “Stochastic stereo matching over scale,” *International Journal of Computer Vision*, vol. 3, no. 1, pp. 17–32, 1989.
- [11] R. Szeliski, “Bayesian modeling of uncertainty in low-level vision,” *International Journal of Computer Vision*, vol. 5, no. 3, pp. 271–302, Dec 1990.
- [12] D. Scharstein and R. Szeliski, “Stereo matching with nonlinear diffusion,” *International Journal of Computer Vision*, vol. 28, no. 2, pp. 155–174, 1998.
- [13] S. Roy and I. Cox, “A maximum-flow formulation of the n-camera stereo correspondence problem,” *International Conference on Computer Vision*, pp. 492–499, 1998.
- [14] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, Nov 2001.
- [15] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph cuts,” *International Conference on Computer Vision*, pp. 508–515, July 2001.
- [16] G. Egnal and R. Wildes, “Detecting binocular half-occlusions: empirical comparisons of five approaches,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1127–1133, Aug 2002.
- [17] S. S. Beauchemin and J. L. Barron, “The computation of optical flow,” *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–467, 1995.
- [18] A. Mitiche and P. Bouthemy, “Computation and analysis of image motion: a synopsis of current problems and methods,” *International Journal of Computer Vision*, vol. 19, no. 1, pp. 29–55, July 1996.
- [19] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, “Performance of optical flow techniques,” *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 92, pp. 236–242, 1994.

- [20] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, “Recovering motion fields: An evaluation of eight optical flow algorithms,” *Proceedings of the British Machine Vision Conference*, Sept 1998.
- [21] H. Liu, T. Hong, M. Herman, T. Camus, and R. Chellappa, “Accuracy vs efficiency trade-offs in optical flow algorithms,” *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 271–286, 1998.
- [22] M. Black and D. Fleet, “Probabilistic detection and tracking of motion discontinuities,” *International Journal of Computer Vision*, vol. 38, no. 3, pp. 231–245, 2000.
- [23] L. Alvarez, R. Deriche, T. Papadapoulo, and J. Sanchez, “Symmetrical dense optical flow estimation with occlusion detection,” *European Conference on Computer Vision*, p. I: 721 ff., 2002.
- [24] S. Beauchemin and J. Barron, “On the fourier properties of discontinuous visual motion,” *Journal of Mathematical Imaging and Vision*, vol. 13, no. 3, pp. 155–172, 2000.
- [25] B. Julesz, *Foundations of Cyclopean Perception*. University of Chicago Press, Chicago., 1971.
- [26] J. Kim, V. Kolmogorov, and R. Zabih, “Visual correspondence using energy minimization and mutual information,” *International Conference on Computer Vision*, vol. 2, pp. 1033–1040, 2003.
- [27] T. Sanger, “Stereo disparity computation using gabor filters.” *Biological Cybernetics*, vol. 59, pp. 405–418, 1988.
- [28] M. Jenkin and A. Jepson, *Computational processes in Human Vision*. (ed.) Z. Pylyshn, Ablex Press, NJ, 1988, ch. The measurement of binocular disparity.
- [29] D. Fleet, A. Jepson, and M. Jenkin, “Phase-based disparity measurement.” *CVGIP: Image Understanding*, vol. 53, pp. 198–210, 1991.
- [30] D. Fleet, “Disparity from local weighted phase-correlation,” *IEEE International Conference on SMC*, pp. 48–56, October 1994.

- [31] J. Weng, "Image matching using windowed fourier phase," *International Journal of Computer Vision*, vol. 11, pp. 211–236, 1994.
- [32] N. Qian, "Computing stereo disparity and motion with known binocular cell properties," *Neural Computation*, vol. 6, pp. 390–404, 1994.
- [33] Y. Boykov, O. Veksler, and R. Zabih, "A variable window approach to early vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1283–1294, Dec 1998.
- [34] O. Nestares, R. Navarro, J. Portilla, and A. Taberero, "Efficient spatial-domain implementation of a multiscale image representation based on gabor functions," *J. Electronic Imaging*, vol. 7, pp. 166–173, 1998.
- [35] A. Ogale and Y. Aloimonos, "Shape and the stereo correspondence problem," *International Journal of Computer Vision*, vol. 65, no. 1, Oct 2005.
- [36] A. Ogale and Y. Aloimonos, "Stereo correspondence with slanted surfaces: critical implications of horizontal slant," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 568–573, June 2004.
- [37] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, 1998.
- [38] A. Ogale, C. Fermuller, and Y. Aloimonos, "Motion segmentation using occlusions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 988–992, 2005.
- [39] Y. Aloimonos and D. Shulman, *Integration of Visual Modules: An Extension of the Marr Paradigm*. Academic Press, 1989.