

Temporally-Consistent Phase Unwrapping for a Stereo-Assisted Structured Light System

Ricardo R. Garcia and Avidesh Zakhor

Department of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, CA USA
{rrgarcia,avz}@eecs.berkeley.edu

Abstract— Phase shifted sinusoidal projection patterns are extensively used in structured light systems due to their low decoding complexity and projection of only three frames per reconstruction. However, they require correct unwrapping of phase images to reconstruct depth accurately. For time varying scenes, it is important for the phase unwrapped results to be temporally coherent. In this paper, we propose a spatio-temporal phase unwrapping algorithm for a structured light system made of a projector and a pair of stereo cameras. Unlike existing frame by frame spatial domain approaches, our proposed algorithm results in a temporally consistent three-dimensional unwrapped phase volume for time varying scenes. We experimentally show the effectiveness of our approach in recovering absolute phase for scenes with multiple disjoint objects, significant motion, and large depth discontinuities.

Keywords- structured light; phase unwrapping; stereo; depth recovery

I. INTRODUCTION

A significant body of work exists on designing effective illumination patterns for structured light systems (SL) that uniquely identify 3D points within the illuminated scene. Examples include temporal and spatial coding of projection patterns or "viewpoint-coded structured light" [1,2,3]. In general, it is desirable to develop illumination patterns which maximize the quality and resolution of each reconstruction while simultaneously minimizing the number of projected frames in the pattern [4].

An important class of projection patterns in SL systems is phase shifted sinusoids. These patterns are robust to depth of field effects and are simple to decode. In the common setup, three sinusoidal images, each phase shifted by $2\pi/3$, are sequentially projected in order to reconstruct the scene [5, 6, 7]. In these systems, it is common to project multiple periods of the sinusoids across the screen. This reduces the number of unique phase values that must be identified, thereby making the decoding process less susceptible to noise. At the same time, the multiple periods create an ambiguity in finding the corresponding point in the projector's image because only the wrapped phase of each pixel can be determined. This ambiguity is removed through the phase unwrapping process [8].

In general, systems requiring phase unwrapping make several assumptions about the content of a scene. First, the

scene is assumed to contain a single contiguous object in space, i.e., all objects in the scene are connected together. Second, phase unwrapping via traditional methods requires most neighboring points to have phase differences that are smaller than $|\pi|$. Otherwise, the true phase differences cannot easily be determined. These assumptions restrict the types of scenes that can successfully be reconstructed with a sinusoidal SL system.

Even though phase unwrapping allows for the recovery of the relative phase between pixels, it does not directly anchor the phase of each pixel to an absolute value. Without absolute phase, there remains a constant unknown offset in the correspondence between camera pixels and projector columns. A variety of techniques have been developed to determine this constant in order to recover absolute depth via triangulation. In [7], Zhang and Yau embed a small cross in the projected pattern to serve as a reference position with a known absolute phase. Once detected, all the remaining points are unwrapped with respect to the marker in order to determine the absolute phase. The embedded marker serves as a valid reference point, but it can be difficult to determine the position of the marker if the projected pattern is out of focus, or if there is significant texture within the scene.

Alternatively, a second camera can be used for stereo matching to remove the relative phase ambiguity [9]. This approach can result in occasional absolute phase errors due to occlusions, calibration errors, "false" matches, and scene motion. In [9], the errors are cleaned up through an energy minimization framework that uses both data and smoothing cost components.

Most existing SL systems for dynamic scenes process the captured data one frame at a time; thus, there is typically no guarantee of temporal coherency in the recovered surfaces potentially resulting in unpleasant flicker artifacts during the rendering process. It is conceivable to both remove flicker and improve phase unwrapping accuracy for scenes with large spatial discontinuities by unwrapping across time as well as space [12].

In this paper, we propose a method to merge stereo phase unwrapping with three-dimensional (3D) phase unwrapping to arrive at temporally consistent absolute phase volumes. By 3D, we refer to (X,Y,T) rather than the commonly used (X,Y,Z) , where T represents time. Our proposed method is a variation of the 3D phase unwrapping algorithm proposed in

[12] whereby each pixel in the wrapped phase image is assigned a quality measure based on its spatio-temporal second difference values. The edges between neighboring spatio-temporal pixels are also assigned quality measures based on the quality measures of the connected pixels. The phase volume is then unwrapped based on rank ordered edge qualities starting from the highest to lowest [12]. Since in our setup we assume a stereo camera pair rather than a single camera, it is possible to develop a framework to integrate stereo phase unwrapping with the basic 3D phase unwrapping method in [12] in order to determine absolute phase. Once unwrapping is completed, the 3D positions of pixels in the phase volume can be determined via triangulation. Besides providing temporal consistency, this approach is also effective in unwrapping scenes with large depth discontinuities or multiple spatially disjoint objects.

The outline of this paper is as follows: Sections 2 and 3 provide an overview of stereo and 3D phase unwrapping algorithms respectively; in Section 4, we describe our approach to merging stereo assisted phase unwrapping with 3D phase unwrapping; Section 5 includes the experimental setup and results; In section 6, we discuss related work; Section 7 offers concluding remarks.

II. OVERVIEW OF STEREO PHASE UNWRAPPING

In stereo phase unwrapping, a second camera is added to the traditional SL system made of a projector and a single camera. To maximize the camera coverage of the illuminated scene, the projector is positioned in between the two cameras, as shown in Figure 1. Before processing captured data, the projector and the pair of cameras must all be calibrated [10].

During capture, the scene is illuminated by three consecutive phase shifted sinusoidal images, as in [5]. After each camera captures the scene from its view, a phase wrapped image is computed as:

$$\phi(x, y) = \tan^{-1} \left(\frac{\sqrt{3}(I_1(x, y) - I_3(x, y))}{2I_2(x, y) - I_1(x, y) - I_3(x, y)} \right) \quad (1)$$

where $I_i(x, y)$ represents the intensity of image i at image coordinates (x, y) , and $\phi(x, y)$ represents the phase at (x, y) . If a scene point is visible in both cameras, the phase measurement is independent of the viewing direction, except for saturated image regions corresponding to specular surfaces. Computing phase via (1) requires the projector to accurately project true sinusoidal patterns. Since the projector modifies the projected patterns through its color processing algorithms, we pre-distort the sinusoidal patterns so that the outgoing patterns are truly sinusoidal.

Even though the wrapped phase can be determined at each pixel of the camera, given M periods of sinusoids in the projection pattern, there is an M position ambiguity in each wrapped phase measurement [9]. To determine the absolute phase of a pixel, we search for the single absolute phase offset $2\pi m$ with $m \in \{0, 1, \dots, M-1\}$, which must be added to its wrapped phase to obtain its absolute phase. We refer to m as the offset index. Since the M possible

corresponding positions in the projector are known, the location of the M points can be triangulated, as shown in Fig. 2 [9]. These triangulated points all lie along the ray coming out of the camera, illustrated as camera 1 in Fig. 2.

Fig. 3(a) shows an example of the wrapped phase image of camera 1, from Fig. 2, with a pixel of interest P identified by a red dot. The extrinsic relationship between the stereo cameras in Fig. 2, as well as each camera's intrinsic parameters can be used to project the M 3D positions onto the second camera's image plane, as shown in Fig. 3(b). By comparing the wrapped phase values at the M pixel locations to the first camera's phase value, the absolute phase of P can be estimated. In Fig. 3(b), it is clear that point B is the corresponding match to the pixel P in Fig. 3(a). Points A and C map to blank areas of the scene, and point D is projected outside of the image frame.

Even though this approach works for most pixels, in practice, it does occasionally result in errors. Common causes of incorrect matches are occlusions, calibration errors, motion errors, or noise in the calculated phase. In some cases, the phase of the point in camera 1 is closer to an invalid point than to the correct one. In [9], the stereo unwrapped points are segmented into continuous regions and loopy belief propagation is used to perform an energy minimization on a cost function that attempts to locally smooth segmented regions.

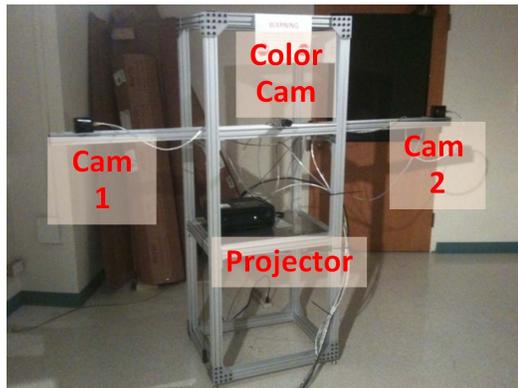


Figure 1. Configuration of stereo SL system.

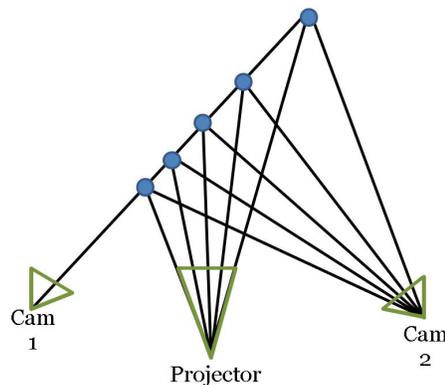


Figure 2. Triangulation of the M possible positions of pixel P in camera 1.

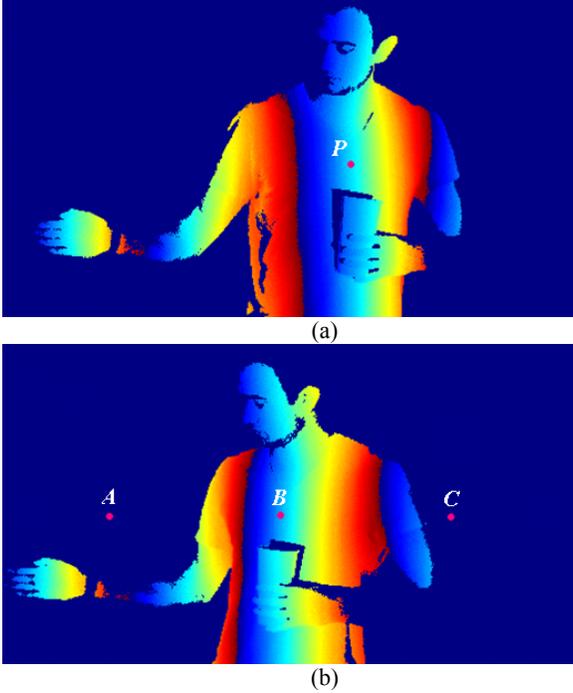


Figure 3. (a) Wrapped phase image of the left camera 1 with the pixel of interest identified by the red dot P ; (b) wrapped phase image for the right camera 2 in system with the M possible points projected onto the image.

III. OVERVIEW OF THREE-DIMENSIONAL PHASE UNWRAPPING

Multidimensional phase unwrapping has been an area of active research for many years, with a substantial body of work in two dimensions [11, 12]. Applications for these methods include synthetic aperture radar, medical imaging, and SL systems [11]. In recent years, many of the two dimensional techniques have been extended to three and higher dimensions [12].

The basic idea behind most multi-dimensional phase unwrapping algorithms is to integrate along paths of neighboring phase values in order to determine the relative phase between all pixels. This is a trivial problem when the true phase difference between neighboring values is less than $|\pi|$ and noise does not push the resulting phase values outside this range. Since, in most applications, this assumption is invalid, phase unwrapping algorithms are designed to minimize the impact of these incorrect regions.

Most 3D phase unwrapping algorithms can be classified into one of three groups: 1) a global-error minimization, 2) residue detection, and 3) quality guided methods [11]. Of these basic approaches, quality guided methods have proven to be both computationally efficient and robust [12]. The basic idea behind them is to quantify the likelihood that two neighboring pixels can be correctly unwrapped with respect to each other. An edge is defined to exist between all neighboring pixels and a measure of quality is assigned to each edge [12]. If two neighboring pixels have similar phase values, the quality of that edge is considered to be high. In contrast, two pixels with greater phase difference would have

a lower quality edge. Although, basic gradient calculations between pixels have been used to generate quality maps [11], other methods, such as a second difference quality map, have been found to be more robust [12].

The second difference value for each pixel (i, j, k) can be computed as:

$$SD_{ijk} = \sqrt{H_{ijk}^2 + V_{ijk}^2 + N_{ijk}^2 + \sum_{n=1}^{10} D_{n,ijk}^2} \quad (2)$$

$$H_{ijk}^2 = \gamma[\phi_{i-1,j,k} - \phi_{i,j,k}] - \gamma[\phi_{i,j,k} - \phi_{i+1,j,k}] \quad (3)$$

$$V_{ijk}^2 = \gamma[\phi_{i,j-1,k} - \phi_{i,j,k}] - \gamma[\phi_{i,j,k} - \phi_{i,j+1,k}] \quad (4)$$

$$N_{ijk}^2 = \gamma[\phi_{i,j,k-1} - \phi_{i,j,k}] - \gamma[\phi_{i,j,k} - \phi_{i,j,k+1}] \quad (5)$$

D

where H , V , and N represent the horizontal, vertical, and normal, i.e., temporal, second differences via discrete Laplacians respectively. In addition, $\phi_{i,j,k}$ represents the wrapped phase at (i, j, k) and γ is a function that wraps its parameter to a value within $[-\pi, \pi)$. The ten diagonal second difference components $D_{n,ijk}^2$, $n \in \{1, \dots, 10\}$ in (2) are computed through all of the diagonal paths in the $3 \times 3 \times 3$ volume centered on the pixel of interest. The squared second derivative can identify pixels where a discontinuity occurs. If there are no discontinuities, the phase increase should be approximately linear, resulting in a squared second difference value near zero; on the other hand, if the squared second difference value is large, there is a strong likelihood of a discontinuity. Using this quality map, it is possible to identify pixels with spatial or temporal discontinuities.

The quality of a pixel is inversely related to the second difference value. Once a quality value is assigned to each pixel in the 3D phase volume, the quality of each edge is simply defined as the sum of the quality of each pixel connected to the edge [12]. The quality of edges determines the order in which the edges in the volume are unwrapped by sorting them from highest to lowest.

When an edge is evaluated, the pixels connected to the edge are unwrapped with respect to one another, forming links in a chain. Each chain has a pixel that is designated the lead pixel. When the first link of a chain is created, the lead pixel is defined as the pixel with the highest quality. As each new pixel is added to a chain, the new pixel is unwrapped with respect to the connected pixel along the evaluated edge. Each newly added pixel is then given a phase value relative to the lead pixel in the chain. When two chains are merged, each pixel in the smaller chain is assigned a phase relative to the lead pixel in the larger chain [12]. Chains grow in connected 3D volumes and their actual shape is unimportant as long as the phase of each pixel in the chain is defined with respect to the lead pixel.

While this 3D algorithm is capable of accurately unwrapping the phases of all points within a volume, it does not necessarily result in absolute phase. Since each absolute phase is assigned to a column of pixels within the sinusoidal

pattern, correct triangulation is only possible when the absolute phase is known; thus without a method to recover the absolute phase, the 3D unwrapping algorithm is useless in the SL system.

Additionally, the algorithm in [12] is only capable of unwrapping scenes with one contiguous, connected region. While this works in MRI, the application domain this algorithm was originally designed for, it is not applicable to depth recovery for scenes with multiple disjoint objects as it could assign incorrect relative phases to each object. This is because in [12], all edges between pixels with a wrapped phase must be unwrapped with respect to one another regardless of the quality of the edges. For example, consider a scene with a ball traveling in front of a flat board. As the ball moves, the pixels that fall along the edge of the ball and the background receive low quality measures due to the motion and depth discontinuity. Therefore the algorithm [12] is likely to assign erroneous phases to those pixels. However, within the ball, the local depth is not changing significantly; we would expect these pixels to have high quality edges. Unwrapping the background and ball independently prevents the two regions from being incorrectly unwrapped. As long as the absolute phase of each region can be determined, e.g., through stereo phase unwrapping, there is no need to unwrap across low quality edges.

The shortcomings of the phase unwrapping algorithm in [12] are a direct consequence of the fact that it has been developed for applications in which only a single observation exists for each phase value. In an application such as MRI, since each point is only measured or observed once there is no choice but to unwrap all pixels with respect to one another. However, in a SL system with a second camera to disambiguate the absolute phase of each pixel, unwrapping can be restricted to high-quality edges, i.e., those connecting pixels with similar phase values. In the following section, we exploit this to develop a method for combining stereo phase unwrapping with 3D phase unwrapping to overcome some of the shortcomings of [12].

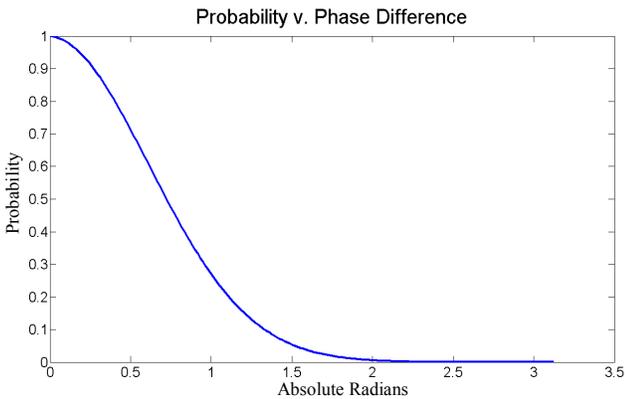


Figure 4. Converting absolute phase difference to probabilities.

IV. MERGING STEREO TECHNIQUES AND THREE-DIMENSIONAL METHODS

In this section, we develop a probabilistic framework for stereo phase unwrapping in order to augment the 3D phase unwrapping algorithm of [12] to determine absolute phase.

A. Determining a Pixel's Absolute Phase Probabilities

As mentioned earlier, the likelihood of a pixel having a specific absolute phase offset is determined by triangulating the M possible corresponding points between camera 1 and the projector, as shown in Fig. 2, projecting these M points onto the second camera's image, as shown in Fig. 3(b), and comparing the phases of the M projected points to the phase of the original pixel P in camera 1. With pixel's taking on wrapped phase values from $-\pi$ to π , if the original pixel P and one of the corresponding M projected pixels have a phase difference of 0, they are likely to be a correct match; likewise, if the pair differ by $|\pi|$ then they are unlikely to be a correct match¹. One way to compute the likelihood for each of the M possible absolute phases offsets is to divide the phase difference of the corresponding pixels by π . However, this likelihood measure does not sufficiently penalize large phase differences. Rather, we propose using a Gaussian function to map the likelihood falloff, as shown in Fig. 4.

In recovering the absolute phase of each pixel, not all of the M corresponding points necessarily project onto valid regions in the second image as shown earlier in Fig. 3. Some could fall on portions of the image where there are no wrapped phase values, as indicated by dots A and C in Fig. 3(b), and some might fall outside the viewable region of the camera, as indicated by dot D in Fig. 3(b). We assign σ_B and σ_E as the probabilities of 3D points projected outside the boundary and to an empty region of the image, respectively. In general, $\sigma_B < \sigma_E$ since an empty position could be due to occlusions, whereas an out of bounds pixel should not exist within the reconstruction area. We note that some pixels near the edges of camera 1's image may not be visible from the second view. For these cases, we assign a probability between σ_B and σ_E . Once the probabilities for all of the possible absolute phase offsets are determined, they are normalized to generate a probability distribution for the absolute phase offset of pixel P in camera 1.

B. Using Pixel Probabilities to Build Chains

In [12], each time a pixel is connected to an existing chain, or when two chains are combined, the newly added pixels are unwrapped relative to their linked neighbors. The relative phase between all pixels in a chain is used to determine the phase difference between the lead pixel and each pixel in the chain. Since we only connect "high quality" pixels with reliable edges, all the pixels in a chain require the same phase offset to convert their relative

¹ When calculating the difference between the phase values, the wrapping function γ , from equations (3)-(5), is used to ensure the difference is in the range $[-\pi, \pi)$.

unwrapped phase to the absolute phase². To determine the absolute phase of the chain, we search for the single absolute phase offset $2\pi m$, where $m \in \{0, 1, \dots, M-1\}$, which must be added to the chain's lead pixel's wrapped phase to obtain its absolute phase. We refer to m as the offset index for that chain. If the absolute phase of the lead pixel and the relative phase difference between the lead pixel and each pixel in the chain are known, then the absolute phase for each pixel in the chain can be determined.

The key to ensuring the pixels in the chain have the correct absolute phase is to use the phase probability distributions of each pixel in the chain to determine the absolute phase of the lead pixel. As explained shortly, the basic idea is to use the phase probability distribution of each newly added pixel in the chain to update the probability distribution of the offset index for the lead pixel of the chain. Once the chain is completed, the lead pixel offset with the highest likelihood is chosen to be the phase offset for the entire chain.

Similar to the unwrapping method in [12], we start by assigning quality values to all of the pixels and edges. The quality value of an edge is the sum of the quality of the two pixels connected to the edge. Once the quality of all edges has been determined, they are ranked in quality from highest to lowest.

To start the unwrapping process, the most reliable edge is identified and the pixels connected to the edge are unwrapped with respect to each other to form a two pixel chain. We set the pixel with higher quality as the lead pixel of the chain. We need to combine the probability distributions of the absolute phase offset for these two pixels in order to determine the probability distribution for the absolute phase offset of the leading pixel, or equivalently whole chain. One way to do this is to add the log probabilities of the two pixels. However, this can only be done after accounting for the phase difference between the two connected pixels.

To illustrate this, consider two possible chains, each with two elements. The first chain, shown in Fig. 5(a), has a lead pixel with a wrapped phase of $\pi - 2\delta$ and a second pixel with a phase of $\pi - \delta$, where $0 < \delta \ll \pi$. Local unwrapping would imply that the two points are from the same sinusoidal period. This means that the \log^3 probability distributions could be directly added together to refine the offset index state of the lead pixel:

$$\log P(c_i = m) \propto \log P(p_j = m) + \log P(p_k = m) \quad (6)$$

where c_i is the offset index state of the lead pixel of the i^{th} chain given the influence from the connected pixels, p_j is the offset index state of the lead pixel and p_k is the offset index

state of the second pixel, with offset index $m \in \{0, 1, \dots, M-1\}$.

Now consider the second case, shown in Fig. 5(b), where the lead pixel has a value of $\pi - \delta$ and the second pixel has a value of $-\pi + \delta$, where $0 < \delta \ll \pi$. For this case, the second pixel is unwrapped into a period that is outside the period corresponding to the lead pixel. Specifically, the second pixel has an absolute phase offset that is 2π greater than the lead pixel, or equivalently an offset index that is one greater than the lead pixel. Therefore when we add the log probabilities to determine the absolute phase distribution for the whole chain, we must shift the individual pixel probability distributions, as shown in Fig. 5(b):

$$\log P(c_i = m) \propto \log P(p_j = m) + \log P(p_k = m + 1) \quad (7)$$

The same procedure is used when adding a single pixel to an already formed chain. As each new pixel is added to the chain, the absolute phase offset distribution of the new pixel is first properly shifted to reflect the relative phase difference between the lead pixel and the new pixel; then the log probabilities for the new pixel are added to a running sum for the chain. In essence, this represents the contribution of phase probability distribution of the new pixel to the overall phase offset of the chain it joins.

In summing log probabilities as pixels are added, we keep a count of the number of pixel probabilities added to each possible offset index. Due to the shifting, each possible offset index might have a different number of log probabilities summed together. For example, in Fig. 5(b), the lead, left, pixel in the chain has two log probabilities summed together for $k_1 \in [0 \dots M-2]$, shown in blue, but only one for $k_1 = M-1$, shown in pink. We know the second pixel in the chain cannot have an offset index of 0, shown in green, since this would imply the lead pixel to have an offset index of -1. This is not possible since $m \in \{0, 1, \dots, M-1\}$. To determine the final absolute phase offset for the lead pixel in the chain, the average log probability for each absolute phase offset is computed and the absolute phase offset with the maximum average log probability is selected. This absolute phase offset is then used to determine the absolute phase for all the pixels in the chain.

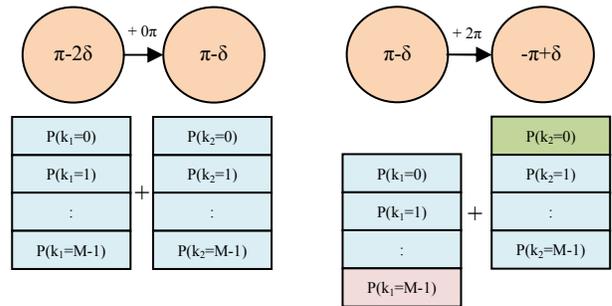


Figure 5. The addition of log probabilities for two pixels (a) in the same period and (b) from different periods.

² In essence, we are assuming all the pixels in a chain are part of a contiguous object in a scene without depth discontinuities.

³ Due to numerical stability issues, we opt to add the log of probabilities rather than multiply them.

For the case where two chains are connected together, the smaller chain is unwrapped relative to the larger chain. That is, each pixel in the smaller chain is unwrapped relative to the lead pixel in the larger chain. In addition, the set of summed log probabilities for the smaller chain is shifted to reflect the relative phase offset between the lead pixels of the two chains. The shifted log probabilities are then added to the summed log probabilities of the larger chain. This is similar to the probability distribution shifting and summing explained in the two pixel chain examples in equations (6) and (7) except that it is being carried out with respect to the lead pixels in the two chains rather than individual pixels.

C. Removing Edges

The advantage of using stereo phase unwrapping to anchor each chain is that it is no longer necessary to unwrap across all edges as it is in [12]. This allows us to determine the absolute phase of spatially disjoint objects in the scene; in addition it prevents us from unwrapping across unreliable low quality edges that may have true phase differences greater than $|\pi|$. Local unwrapping fails when the true phase differences between neighboring pixels are greater than $|\pi|$.

By limiting unwrapping to edges with quality greater than a minimum threshold, we reduce the probability of unwrapping across spatial or temporal discontinuities. By adding the time dimension to unwrapping, small regions are unwrapped more accurately since more observations are merged over time.

In our implementation, we choose a minimum quality value to be met in order for the edge to be considered valid. Even though this results in several smaller chains at the end of the unwrapping process, the points within each chain are more likely to be correctly unwrapped. In general, incorrect unwrapping only occurs along the edges between discontinuous regions.

V. EXPERIMENTAL SETUP AND RESULTS

To test the presented algorithm, we have developed an SL system, similar to the one in [9], but without the color camera for texture capture. It consists of two PointGrey Dragonfly Express cameras with a resolution of 640×480 and 1394b ports. We use an Optoma TX780 projector, with a resolution of 1024×768 operating at 60 Hz. Similar to many other SL systems utilizing a DLP projector, the color wheel has been removed to increase the rate at which grayscale patterns are projected to 180 Hz [6].

To illustrate the effectiveness of our proposed algorithm, we have captured several data sets. Once the correspondences have been found between the camera and projector, the depth of each pixel is determined via triangulation. We present results on the accuracy of the estimated absolute phase of the scene, which when accurate, results in accurately reconstructed point clouds.

We show our algorithm's ability to handle scenes with disjoint objects. Fig. 6(a) shows the unwrapped phase of a scene with disjoint objects using our proposed algorithm. Fig. 6(b) illustrates how the algorithm in [12] fails to correctly unwrap the disconnected regions inside the ellipse.

In the phase image, a cone shaped object is lifted off of a box. The cone and box objects are positioned at approximately the same depth, so they should have nearly the same unwrapped phase values as shown in Fig. 6(a). Since there are no connections between the two regions, there is no way for the algorithm in [12] to determine the phase difference between the isolated segments. Our method handles this by using the added information from the stereo camera as long as the points are visible in both cameras.

While existing SL approaches using stereo cameras enforce only spatial smoothness during the phase unwrapping process [9], our approach enforces both spatial and temporal smoothing. To characterize the advantage of temporal smoothing, we have compared our algorithm with the stereo phase unwrapping method presented in [9]. Since our method only determines the unwrapped phase of one camera at a time, to do a fair comparison we do not implement the left-to-right and right-to-left consistency check described in [9]. This type of consistency check could easily be applied to our method to provide the same benefits as the approach in [9]. In the loopy belief propagation implementation, we process 10 iterations before generating the final labeling for all segments. Fig. 7(a) shows two consecutively unwrapped phase images generated using the algorithm in [9]. In Fig. 8(a) the same images are processed using our proposed algorithm. The successive images in Fig. 8(a) resulting from our approach are more temporally consistent than those in Fig. 7(a). The unwrapped phase difference between the successive images in Fig. 7(a) and 8(a) are shown in Fig. 7(b) and 8(b), respectively. As seen, the areas with phase difference in Fig. 7(b) are significantly smaller and less noticeable than in 8(b). Specifically, in Fig. 7(b) 15.3% of pixels are not consistent over time as compared to 3.4%⁴ in Fig. 8(b). Visual inspection of the two phase video sequences also confirms the temporal coherency of our approach as compared to spatial only smoothing which exhibits a significant amount of flicker. Video results can be found at http://www-video.eecs.berkeley.edu/research/4D_SL/

In [12], the edges are sorted and unwrapped according to the quality of the connection between pixels. Since the lower quality pixels are unwrapped at the end, any potential errors affect only the few pixels that follow in unwrapping order. By combining the absolute phase positioning from the stereo cameras, it is not necessary for every pixel to be unwrapped relative to all of its neighboring pixels. Therefore, our proposed algorithm eliminates edges between pixels with low quality edges. Fig. 9 shows the same scene unwrapped with our proposed algorithm and that of [12]. In Fig. 9(b), the black circled regions contain points that are unwrapped incorrectly. The phase in these regions changes rapidly both spatially and temporally due to the discontinuity between the two planes and the high motion of the plane. As seen, the low quality edges do not allow for accurate unwrapping.

⁴ Some of the 3.4% corresponds to actual motion in the scene.

VI. RELATED WORK

Our proposed setup is quite similar to the stereo assisted structured light system in [9]. In [9], the stereo matching information is used to select an initial offset for each pixel. It is acknowledged that the stereo information for a single pixel does not always indicate the correct absolute phase for a pixel. However, hard decisions are made with these stereo measurements and the resulting errors are later corrected by segmenting the initial unwrapped phase image and by applying an energy minimization on the mislabeled unwrapped image. In contrast, rather than selecting an offset for each pixel, our method selects an offset for a group of pixels that are first locally unwrapped according to an edge quality measure. By restricting local unwrapping to edges with high measures of reliability, the correct relative phase between all pixels in a chain can be determined with high confidence. Once the relative phase of all pixels within the chain is known, only a single offset needs to be chosen in order to find the absolute phase for all pixels within the chain. The final probability distribution for the chain is generated by combining the “noisy” probability distributions of each pixel. In this detection/estimation problem, many probability distributions are combined to generate a distribution that has a lower likelihood of error.

In [9], the segmentation of the estimated absolute phase is done by combining pixels that have a phase difference of less than $|\pi|$. Phase unwrapping based on local gradients have been shown to be less accurate than methods using other edge quality measures such as discrete Laplacians [12]. In addition, once segmentation is completed, the local smoothing component of the energy cost function in [9] only operates along the edges of segments. If incorrect labeling occurs within the interior of a segment, the local smoothing cost component cannot detect the error. In contrast, all edges between pixels in our method are examined, and they are only connected if the edge is reliable.

The most significant difference between our method and existing phase unwrapping methods for SL, including [9], is the enforcement of temporal consistency in the unwrapped phase maps. In doing so, we take advantage of the inherent temporal correlation between successive captured phase images. If the capture rate of a system captures is high enough to ensure that the phase at a single point does not change by more than $|\pi|$ in successive frames⁵, then the same local smoothing assumptions applied spatially during phase unwrapping can ideally also be applied in the temporal domain as well. In our case, consistent labeling over time is ensured by building larger chains that span multiple frames. The offset for each of these larger chains is determined more accurately as long as only high quality edges are used during local unwrapping.

Finally, our method improves the results presented in [12] for application to a stereo structured light system. The primary motivation of the phase unwrapping approach in [12] is magnetic resonance imaging applications. By using

⁵ For scenes with high speed motion, phase recovery at each pixel is erroneous if not compensated [9].

the stereo observations from two cameras, our algorithm can find absolute rather relative phase values.

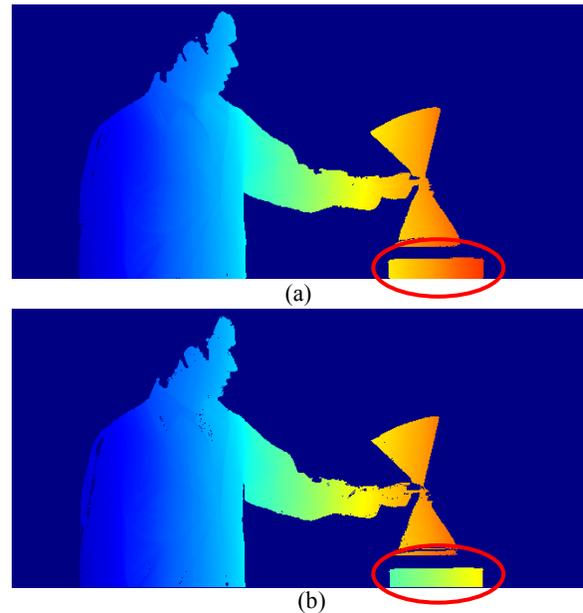


Figure 6. (a) Correctly unwrapped scene using our proposed algorithm; (b) incorrect unwrapping for the algorithm in [12].

VII. CONCLUSIONS

We have proposed a scheme which combines stereo phase unwrapping and 3D temporal phase unwrapping. By using stereo, our algorithm is able to recover the absolute phase for disjoint objects in a scene. In addition, by incorporating time in the unwrapping process, we obtain temporally coherent phase images and depth maps, and avoid unwrapping across spatial discontinuities. Finally, our edge removal process reduces the potential errors from unwrapping across edges with large discontinuities.

Future work includes unwrapping the images from camera 1 and camera 2 simultaneously. In the results we have presented so far, camera 2 aids in unwrapping of the phase image of camera 1. With our current method, we would need to switch the role of the cameras to unwrap the phase image of camera 2. We envision modifying the 3D phase unwrapping method to unwrap both images in a single pass. Additionally, a streaming implementation of our algorithm could unwrap across a small temporal window, rather than the entire video segment.

REFERENCES

- [1] O. Hall-Holt, S. Rusinkiewicz, "Stripe boundary codes for real-time structured-light range scanning of moving objects," (ICCV'01) - Volume 2, 2001.
- [2] H. Kawasaki, R. Furukawa, R. Sagawa, Y. Yasushi, "Dynamic scene shape reconstruction using a single structured light pattern," Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on , vol., no., pp.1-8, 23-28 June 2008.
- [3] M. Young, E. Beeson, J. Davis, S. Rusinkiewicz, R. Ramamoorthi, "Viewpoint-Coded Structured Light," Computer Vision and Pattern

Recognition, 2007. CVPR '07. IEEE Conference on , vol., no., pp.1-8, 17-22 June 2007.

- [4] J. Salvi, J. Pages, J. Batlle, "Pattern codification strategies in structured light systems", Pattern Recognition, Volume 37, Issue 4, Agent Based Computer Vision, April 2004, Pages 827-849.
- [5] S. Zhang, P. Huang, "High-resolution, real-time 3-D shape acquisition," IEEE Computer Vision and Pattern Recognition Workshop, 03(3), 28-35, 2004.
- [6] P. Huang, S. Zhang, "Fast three-step phase-shifting algorithm," Appl. Opt., 45(21), 5086-5091, 2006.
- [7] S. Zhang, S. Yau, "Absolute phase assisted three-dimensional data registration for a dual-camera structured light system", Appl. Opt., 47(17), 3134-3142, 2008.
- [8] S. Zhang, X. Li, S.T. Yau, "Multilevel quality-guided phase unwrapping algorithm for real-time three-dimensional shape reconstruction," Appl. Opt. 46, 50-57 (2007).
- [9] T. Weise, B. Leibe, L. Van Gool, "Fast 3D Scanning with Automatic Motion Compensation," Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on , vol., no., pp.1-8, 17-22 June 2007.
- [10] J.Y. Bouguet. *Camera Calibration Toolbox for Matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc/, 2010.
- [11] D.C. Ghiglia, M.D. Pritt, Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software (Wiley, 1998).
- [12] H. Abdul-Rahman, M.A. Gdeisat, D.R. Burton, M.J. Lalor, F. Lilley, C.J. Moore, "Fast and robust three-dimensional best path phase unwrapping algorithm," Appl. Opt. 46, 6623-6635 (2007).

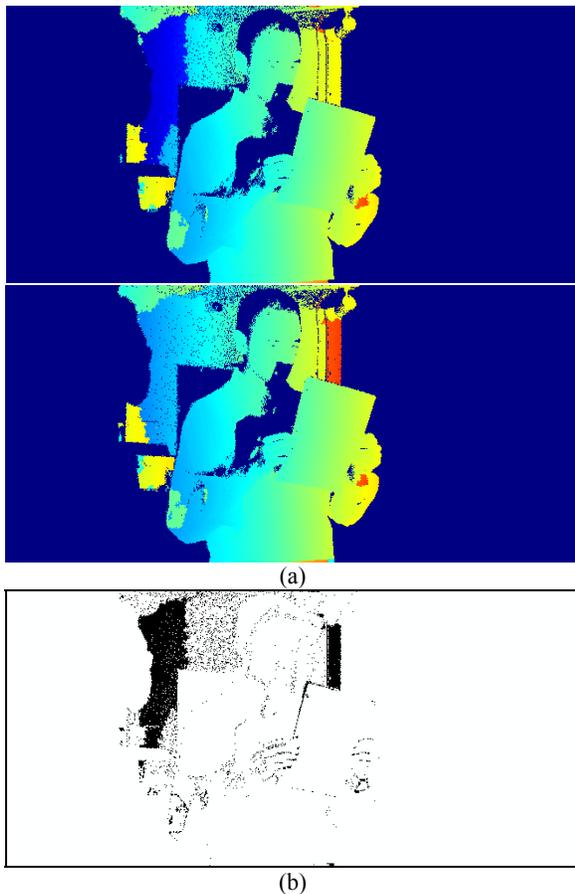


Figure 7. (a) Two consecutively unwrapped phase images using the method in [9]; (b) regions of phase difference in consecutive images.

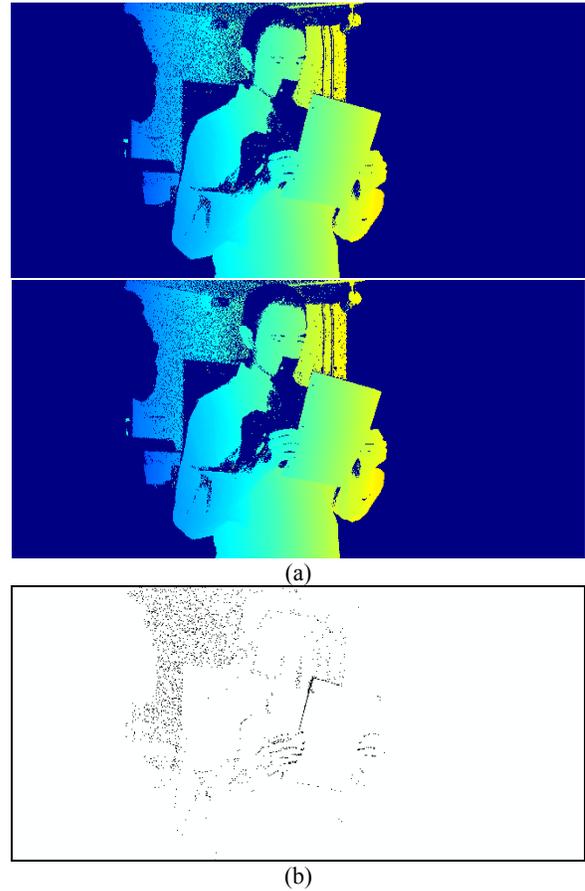


Figure 8. (a) Two consecutively unwrapped phase images with our proposed algorithm; (b) regions of phase difference in consecutive images.

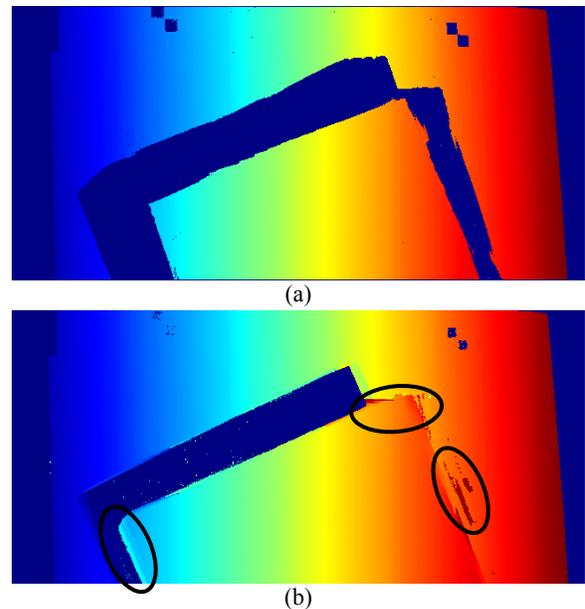


Figure 9. Front board is moving fast in front of back flat surface; unwrapping by (a) our proposed algorithm; (b) the algorithm in [12].