Algorithms for Maintaining a High-Resolution Panoramic Display with a Tele-Operated Robotic Camera

Dezhen Song[1], Ni Qin[1], and Ken Goldberg[2]

1: CS Department, Texas A&M University, College Station, TX 77843
2: IEOR and EECS Departments, University of California, Berkeley, CA 94720

***Abstract*— A new class of low-cost teleoperated pan-tilt-zoom robotic video cameras can provide high resolution panoramic displays of remote sites for disaster response, environmental monitoring, and security applications. While the camera is tele-operated, the resulting video is transmitted back and inserted into an evolving panoramic display. Since small errors in camera position can produce large registration errors in the panoramic display, we address the image alignment problem. To quantify alignment error, we introduce a new metric based on motor error and image overlap. We use this metric to develop a fast minimal variance image alignment algorithm. We have implemented the algorithm and describe experiments demonstrating panoramic quality and that optimal alignment can be computed as fast as the camera can be tele-operated.**

***Index Terms*— tele-operation, telerobotics, networked robot, panoramic display, pan-tilt-zoom camera.**

I. Introduction

There are many applications where it is desirable to visually monitor remote environments, for example to observe rescue operations after a natural disaster, to monitor an endangered animal habitat, or to monitor a dangerous zone for security purposes. Recent developments in wireless telecommunications facilitate low-bandwidth connectivity to remote sites and a new class of low-cost teleoperated pan-tilt-zoom robotic video cameras allows fast deployment of systems that can provide high resolution images from a wide field of view in the remote environment.

Driven largely by security applications, several companies have recently introduced low-cost networked tele-operated cameras for remote monitoring. One example is the Panasonic WV-CW864A camera. With 22x zoom motorized optical lens, 360° pan range, and 90° tilt range, this robotic camera can provide resolution up to 500 million pixels per steradian, two orders of magnitude higher than the best available fixed position omnidirectional camera, at a fraction of the cost. Tele-operated cameras provide relatively small "foveal" video sequences that require far less bandwidth than high resolution video of the entire field of view. A major challenge is combining the foveal images together into a coherent panoramic display.

This work was supported in part by the National Science Foundation under IIS-0113147, by Intel Corporation, by Panasonic, and by UC Berkeley's Center for Information Technology Research in the Interest of Society (CITRIS). For more information please contact dzsong@cs.tamu.edu or goldberg@ieor.berkeley.edu.

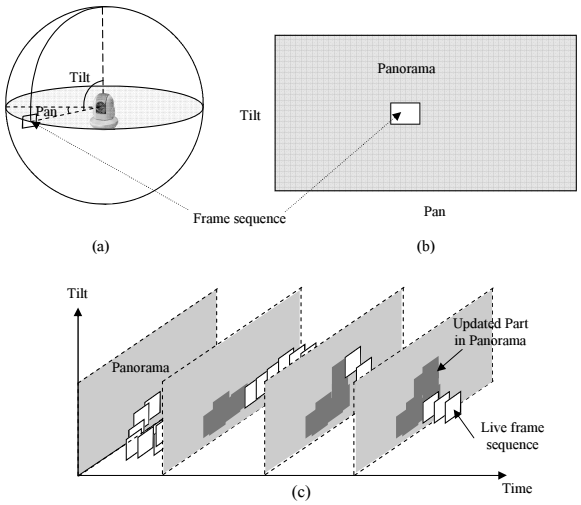

Fig. 1. *A tele-operated robotic camera provides an evolving high-resolution panoramic display of the remote environment. (a) Camera and spherical field of view, (b) Current video image in context of planar panoramic display, (c) Time sequence of video images and evolving panoramic display.*

As illustrated in Figure 1, the camera has a spherical field of view. As the camera is moved by a remote tele-operator, it transmits *frame sequences* over the network back to the tele-operator. (Control of a single camera by multiple tele-operators is addressed in [19], [21]). To provide operator context and archival record, these frame sequences must be inserted into an evolving panoramic display.

Minor errors in camera position can produce large registration errors in the panoramic image. For example, accurate registration of a 640×480 image at zoom = 10x into a panorama requires angular position accuracy within $0.00625°$, 100 times more than the accuracy currently available in commercial robotic cameras.

We assume that motor parameters are approximate and develop an algorithm to optimally insert frame sequences into the evolving panoramic display. The key to our algorithm is a variance based method for identifying a weighted subset of recent overlapping frame sequences. We have implemented the algorithm and report on experiments demonstrating that

image alignment can be computed as fast as the camera can be tele-operated.

II. Related work

1) Multiple-Camera System and Wide Angle System: When low/variable image resolution is acceptable, an evolving panoramic display can be maintained with a single wide-angle camera using a fish eye lens or parabolic mirror [1], [15], [27], [6]. When sufficient bandwidth is available, an evolving high-resolution panorama can be maintained with multiple fixed cameras. Swaminathan and Nayar [22] use four wide angle cameras to monitor a 360° field of view. Similarly, Tan, Hua, and Ahuja [23] combine multiple cameras with a mirror pyramid to create a single-perspective and high resolution panoramic video. Liu, Kimber, and Foote [11] combine four fixed cameras with a robotic camera that can selectively zoom in on details. Our approach could be combined with one or more fixed cameras, but since bandwidth is limited, we focus on using only one robotic camera to monitor the environment.

2) Image Mosaicing Techniques: Generating a single wide-field panoramic image from a set of overlapping images is sometimes referred to as "image mosaicing" [18], [2]. Given a set of overlapping images, the objective is to find the best set of transform parameters for each image. Three approaches have been proposed. The direct method directly matches pixel intensity information using standard least square method or brute force method and requires extensive computation. The second method is frequency domain registration, which uses the fast Fourier transform to maximize alignment in the frequency domain [3], [4], [12], [17]. This method is highly effective when there is substantial overlap between images. The third method is "feature based", using features extracted from the image, such as Harris corner points[7], [25], [29], [31], Moravec's interest operator[8], contour edge[13], convex hull formed from scattered feature points[28], moment invariants[5], and Scale Invariant Feature Transform (SIFT)[14].

3) Constructing a 3D Scene from Video Frames: Constructing a 3D scene from either calibrated or un-calibrated video frames is a very popular problem in both robotics and computer vision [16], [24]. The similarity between this problem and our problem is that both use overlapping frames to establish transformation matrices. The difference is that 3D modeling requires frames captured from different perspectives whereas panorama construction prefers frames from a single perspective. For two given frames, a 3D model can only be constructed for intersection region of the two frames whereas a panorama generated from our problem covers union region of the two frames.

4) Dynamic Panorama: A dynamic panorama refers to a updateable panorama built from a pre-recorded sequence of consecutive video images [9], [26], [30]. Current methods do not take the image registration error into consideration. Therefore, it either has limited number of frames or relies on extensive frame matching computation which can not process live video data. Hence, the dynamic panorama has to be pre-computed off-line before streaming. Our work complements existing work by utilizing camera pan-tilt-zoom values, tracking registration error, and controlling image matching problem size to reduce image registration time and meet the live video requirement.

The idea of dynamic panorama also inspires work on developing panorama video streaming protocol. Kim et al [10] develop a panorama video streaming protocol for a pan-tilt camera system. They capture live video using a fixed lens camera and assume camera pan and tilt readings are accurate enough to register frames. They expand MPEG algorithm by slicing camera horizonal field of view into vertical strips and propose inter-strip and intra-strip compression ideas. Their work do not propose a solution to deal with the problem of image registration error accumulation and can not make good use of camera zooming capability to provide high resolution feedback.

5) Our Previous Related Work and Contribution: In previously reported work, we developed camera control interfaces for multiple simultaneous tele-operators [19], [21]. In [20], we describe a system for remote monitoring of construction sites for dangerous environments such as Iraq. The present paper develops the theory behind a new algorithm that maintains an evolving panorama minimizing image alignment error.

III. Problem Description

A. Inputs and Assumptions

1) Definition of Frame Sequence: When the camera is moving, images are blurred and must be discarded. Once the camera has stopped, we define a *frame sequence* as a sequence of camera frames from some fixed pan-tilt-zoom setting,

$$F = \{C(t_{\text{begin}}, t_{\text{end}}), p, t, z, X, \upsilon\}, \tag{1}$$

where C stands for the frame content data set, t_{begin} and t_{end} are the beginning time and ending time of the frame sequence respectively, (p, t, z) are the approximate pan, tilt, and zoom values obtained from the camera, X is a set of unknown image alignment parameters, and υ is a scalar that indicates how well the frame sequence is aligned with respect to its neighbors as discussed below.

Since the camera does not move for the duration of a frame sequence, we compute the alignment parameters using the first image of each frame sequence and use the same alignment parameters to transform the last image of the sequence to update the panorama. Below, we refer to the "frame" as the first image from a frame sequence.

2) Definition of Panorama: The evolving panorama at time t includes all previous frame sequences,

$$P(t) = \{F | t_{\text{begin}} < t\}$$

inserted in temporal order.

Each panorama has a reference frame. The positional parameters X of other frame sequences are computed with respected to the reference frame. The reference frame is also the first frame of the panorama. Starting with reference frame, the panorama is initialized by commanding the camera to visit a sequence of preset coordinates that cover the field of view as we will show in Section V-A. Actually, the panorama generation and maintenance need the same incremental frame alignment algorithm that will be introduced in Section III-B.

3) Known Camera Intrinsic Parameters: Constructing the panorama requires projection and positional parameters. The projection parameters include image resolution, camera focus length, and CCD sensor size, all of which are known and fixed. We use these to project all images onto a fixed spherical surface. The set of positional parameters X from Equation 1 are unknown and must be computed.

4) Approximate Camera Pan, Tilt, Zoom Position: The teleoperator periodically sends a motion command to the camera, specified as a desired pan, tilt, and zoom (p, t, z) target. After the camera motors servo toward this target, they stop and the camera sends back an estimate of its resulting pan, tilt, and zoom position. As noted above, these estimates are inherently approximate. We use the approximate position for an initial estimate of how many pixels overlap between a pair of frames. Once the alignment parameter X is computed by the algorithm, we use it to refine the number of overlapped pixels.

5) Random Pair-wise Alignment Error: When computing the relative offset between two frames, the matching problem is a nonlinear minimization problem. Introduced by numerical methods for nonlinear optimization like Gaussian-Newton method, Simulated Annealing, or Genetic Algorithms, the error between true optimal and actual solution depends on initial point and truncation error. A good algorithm chooses its initial point randomly, which defines the alignment error to be a random vector. We assume the alignment error random vector has zero mean and variance σ^2, which usually is a function of truncation error and image characteristics and will be discussed in Section IV-A.

6) Errors in Pair-wise Alignment: We assume that the Average Matching Error (AME) A of each pixel (or feature point if using feature-based matching) can be approximated by a quadratic function in the vicinity of its optimal matching location. For the i^{th} pixel in a new frame with its location X_i, this is described by,

$$A(X_i) = a\|X_i - X_i^*\|_2^2 + b, \tag{2}$$

where X_i^* is optimal alignment location, a is a scaling factor, and b is the residual caused by noise. We assume that a and b are the same across all matching pixels.

B. Incremental Frame Alignment Problem

The incremental Frame Alignment problem is: *given a set of n existing frame sequences, find X for a newly arrived frame sequence.*

We solve it in two steps. The first step is to identify a subset of past frame sequences and decompose the alignment problem into multiple pair-wise alignment problems and give each an appropriate weight. In the second step, the pair-wise alignment problems are solved by applying standard image mosaicing methods from Section II-.2. We use the direct matching method throughout the rest of the paper.

We focus on step one: identify a subset of past frames sequences that provide an optimal tradeoff between quality of the panorama and computation time.

IV. Algorithms

We've assumed that error of X is a random vector with zero mean. Therefore, the magnitude of error variance determines the quality of alignment. To analyze the error variance, we first propose a quality metric to measure how sensitive an image alignment method is to errors. We study how error variance gets accumulated and propagated in the alignment process using a simple 1D example. Based on the analysis, we propose a minimum variance approach to select an optimal set of existing frames to register a newly arrived frame. We begin with definition of the quality metric.

A. Quality Metric for Image Alignment

We propose the following quality metric υ to quantify alignment error. The scalar υ measures average pixel-wise alignment variance and will be defined for each frame sequence. Since image alignment is not perfect due to round off

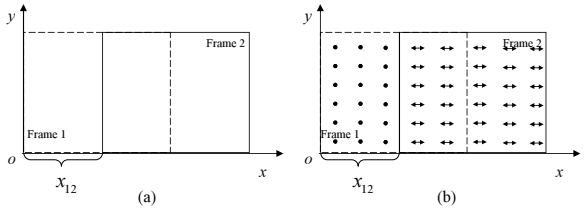

Fig. 2. *An illustration of Metric υ using a panorama composed by two equally sized frames with equal number of pixels. Frame 1 is the reference image in the alignment.*

errors and image characteristics, the displacement between the actual coordinate X_i of the i^{th} pixel and its ideal coordinate X_i^* is a random vector $D_i = X_i - X_i^*$. Let n_p be the number of pixels in panorama P. For P, metric υ is,

$$\upsilon(P) = \frac{1}{n_p}\sum_{i=1}^{n_p} Var(D_i) \tag{3}$$

Metric υ is defined for a frame sequence as the average alignment variance of all pixels in its first frame.

Figure 2 illustrates how to compute υ using a panorama with two equally sized frames. The displacement between the two frames is caused by camera pan motion so that the only alignment parameter is the horizontal displacement, x_{12}, between the two frames. Frame 1 enters the system first, then Frame 2 is captured. Frame 2 will be put on the top of frame 1. Define x_{12}^* as the optimal displacement. Random displacement error is $d_{12} = x_{12} - x_{12}^*$. Since frame 1 is the reference frame, all its pixels have zero variance. Alignment variance of each pixel in frame 2 is σ^2. Figure 2(b) uses arrows to indicate variance amplitude. Let m, $m \leq n_p$, be number of pixels in each frame and m_{12}, $0 < m_{12} \leq m$, be number of overlapping pixels. Metric υ of the panorama can be computed as

$$\upsilon = \frac{1}{n_p}((m - m_{12}) \times 0 + m\sigma^2) = \frac{m}{n_p}\sigma^2, \tag{4}$$

where frame 1 contributes $m - m_{12}$ pixels to the panorama and frame 2 contributes m pixels to the panorama.

B. Analyzing Alignment Errors

In this section we use statistical metric υ to compare the quality of image alignment methods. We begin with the simplest pair-wise alignment operation.

1) Error Variance in Pair-wise Alignment: Define O as the set of the overlapped pixels. According to the assumption in Section III-A.6, the Total Matching Error (TME) T over O becomes,

$$T = \sum_{i \in O} (a\|X_i - X_i^*\|_2^2 + b) \quad (5)$$

$$= |O|a\|X_i - X_i^*\|_2^2 + |O|b. \quad (6)$$

The image alignment is an optimization problem,

$$\arg \min_{\{X_i, i \in O\}} T,$$

subject to image integrity constraint, which actually reduces the unknown set $\{X_i, i \in O\}$ to the single vector X defined in Equation 1. We must find X such that

$$T(X) \leq |O|b + \epsilon,$$

where ϵ is the truncation error from the minimization problem. Inserting it into Equation 5, all possible solutions must be inside the ball,

$$\|X - X^*\|_2 \leq \sqrt{\frac{\epsilon}{|O|a}}, \quad (7)$$

where X^* is the optimal solution. Recall that AME is an approximation of real matching function in the vicinity of the optimal. AME is unknown during the problem solving process. Therefore, we can not directly use X^* deducted from AME as the solution. Any point in the ball with radius $r = \sqrt{\frac{\epsilon}{|O|a}}$ is a possible solution. To solve the matching problem is just to sample a point from the ball with a unknown location. Any point in the ball is likely to be a solution if the matching algorithm chooses its initial point randomly. The dimensionality of the ball depends on the dimensionality of X.

For the simple 1D case in Figure 2, the ball degrades to a line segment. If we assume the solution is uniformly distributed, then its variance is

$$\sigma^2 = \frac{(2r)^2}{12} = \frac{r^2}{3} = \frac{\epsilon}{3|O|a}. \quad (8)$$

Inserting Equation 8 into Equation 4 and defining $\alpha = m_{12}/m$, we obtain the Metric υ for pair-wise image alignment:

$$\upsilon = \frac{\epsilon}{3n_p a \alpha}. \quad (9)$$

For the general $d-$dimension case $X = \{x_1, x_2, ..., x_d\}$, we have variances of the marginal distributions along each dimension, $\{\sigma_{x_1}^2, \sigma_{x_2}^2, ..., \sigma_{x_d}^2\}$. We define

$$\sigma^2 = \max\{\sigma_{x_1}^2, \sigma_{x_2}^2, ..., \sigma_{x_d}^2\}.$$

Interestingly, though the distribution of the solution point in the ball is unknown, the $d-$dimension case has a similar format with the 1-dimensional case in Equation 8 with a different constant factor k_d, as summarized as the following theorem.

Theorem 1: Using AME approximation of image matching function in the vicinity of the optimal solution, the variance of alignment displacement error is

$$\sigma^2 = \frac{r^2}{k_d} = \frac{\epsilon}{k_d |O| a}, \quad (10)$$

where $k_d \geq 1$ and d is the problem dimensionality. The exact value of k_d depends on d and the joint probability distribution function of the solution distribution over the ball defined by Equation 7.

Proof: Define the joint probability density function as $f(x_1, x_2, ..., x_d)$, we have

$$\underbrace{\int_{-r}^{r} ... \int_{-r}^{r}}_{d} f(x_1, x_2, ..., x_d) dx_1 dx_2 ... dx_d = 1. \quad (11)$$

Without loss of generality, we assume $\sigma_{x_1}^2 = \sigma^2$. We compute $\sigma_{x_1}^2$ in the rest of the proof. Because x_1 has zero mean, we know

$$\sigma_{x_1}^2 = E(x_1^2) - E^2(x_1) = E(x_1^2).$$

We define,

$$f_1(x_1) = \underbrace{\int_{-r}^{r} ... \int_{-r}^{r}}_{d-1} f(x_1, x_2, ..., x_d) dx_2 ... dx_d, \quad (12)$$

and

$$F_1(y) = \int_{-r}^{y} f_1(x_1) dx_1, \quad (13)$$

as the marginal probability density function and the cumulative probability function for x_1 respectively. Now we are ready to compute σ^2,

$$\begin{aligned}
\sigma^2 &= \int_{-r}^{r} x_1^2 f_1(x_1) dx_1 \\
&= \int_{-r}^{r} x_1^2 dF_1(x_1) \\
&= x_1^2 F_1(x_1)|_{-r}^{r} - \int_{-r}^{r} 2x_1 F_1(x_1) dx_1 \\
&= r^2 - \int_{-r}^{r} 2x_1 F_1(x_1) dx_1 \\
&= r^2 - \int_{-r}^{0} 2x_1 F_1(x_1) dx_1 - \int_{0}^{r} 2x_1 F_1(x_1) dx_1 \\
&= r^2 + \int_{-r}^{0} (-2x_1) F_1(x_1) dx_1 - \int_{0}^{r} 2x_1 F_1(x_1) dx_1
\end{aligned}$$

Applying the Second Mean Value Theorem for Integrals, we know $\exists \xi \in [-r, 0], \exists \zeta \in [0, r]$ such that,

$$\int_{-r}^{0} (-2x_1) F_1(x_1) dx_1 = F_1(\xi) \int_{-r}^{0} (-2x_1) dx_1 = F_1(\xi) r^2,$$

and

$$\int_{0}^{r} (2x_1) F_1(x_1) dx_1 = F_1(\zeta) \int_{0}^{r} (2x_1) dx_1 = F_1(\zeta) r^2.$$

Therefore,

$$\sigma^2 = (1 + F_1(\xi) - F_1(\zeta))r^2,$$

and

$$k_d = 1/(1 + F_1(\xi) - F_1(\zeta))$$

is the constant. ■

As summarized in Theorem 1 the quality of the solution is determined by how many pixels are involved in the matching, $|O|$, and the image characteristics a.

2) Insertion Without Updating Panoramic Display: A naive approach is to insert new frames using one panoramic image that is never updated. We can use use metric υ to analyze the resulting performance.

Consider inserting a new frame 3 with the same size into the panorama in Figure 2. Define m_{23}, $0 \leq m_{23} \leq m$, as number of overlapping pixels between frame 2 and frame 3. To simplify the notation, we also define $\beta = \frac{m_{23}}{m}$. Hence $m_{23} = \beta m$ as illustrated in Figure 3.

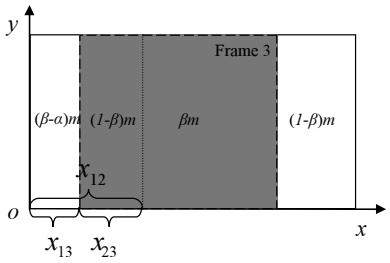

Fig. 3. *Insertion of a new frame into the panorama generated by frame 1 and frame 2 in Figure 2.*

Define x_{13} as the offset of frame 3 and x^*_{13} as the corresponding optimal offset. Recall that x_{12} is the offset of frame 2. Because frame 2 carries displacement error $d_{12} = x_{12} - x^*_{12}$, the TME in Equation 5 becomes,

$$\begin{aligned} T &= (1-\beta)m\big(a(x_{13} - x^*_{13})^2 + b\big) \\ &+ \beta m\big(a(x_{13} - x^*_{13} + d_{12})^2 + b\big). \end{aligned}$$

This equation can be simplified as,

$$\begin{aligned} T &= ma(x_{13} - x^*_{13} + \beta d_{12})^2 \\ &+ m\big(ad^2_{12}(\beta - \beta^2) + b\big). \end{aligned} \quad (14)$$

It is not surprising that its residual $m\big(ad^2_{12}(\beta - \beta^2) + b\big)$ gets bigger because of the displacement error in frame 2. Using the result from Equation 7, the radius of the ball that covers possible solution is $\sqrt{\frac{\epsilon}{ma}}$. The variance of the solution for a given d_{12} is,

$$Var(x_{13}|d_{12}) = \frac{\epsilon}{3ma}.$$

Equation 14 also tells us the expected solution for a given d_{12} is,

$$E(x_{13}|d_{12}) = x^*_{13} - \beta d_{12}.$$

From knowledge of conditional variance, we know that

$$Var(x_{13}) = E(Var(x_{13}|d_{12})) + Var(E(x_{13}|d_{12})).$$

Therefore, we can get the variance of displacement for each pixel in frame 3,

$$Var(x_{13}) = \frac{\epsilon}{3ma}(1 + \frac{\beta^2}{\alpha}). \quad (15)$$

Now, we can compute metric υ for this case. Figure 3 also tells us that frame 1 contributes $(1-\alpha)m - (1-\beta)m = (\beta - \alpha)m$ pixels to the panorama, frame 2 contributes $(1-\beta)m$ to the panorama, and frame 3 contributes m pixels to the panorama. Plug them in to Equation 3,

$$\begin{aligned} \upsilon &= \frac{1}{n_p}\big(m\frac{\epsilon}{3ma}(1 + \frac{\beta^2}{\alpha}) + (1-\beta)m\frac{\epsilon}{3\alpha ma}\big) \\ &= \frac{\epsilon}{3n_p a}(1 + \frac{\beta^2}{\alpha} + \frac{1-\beta}{\alpha}). \end{aligned} \quad (16)$$

Comparing to υ from Equation 9, the result in Equation 16 may grow; the panoramic display deteriorates over time due to deterioration of the matching function, which decreases the subsequent alignment accuracy. This can also be seen in the increase of the residual in Equation 14, which indicates a decrease in the signal/noise ratio. Since the panorama is not updated, the deteriorating trend continues as new frames are inserted. To address this, we must update the panorama as frames are inserted. However, as shown in next section, this may suffer from error propagation if it is not designed properly.

3) Insertion With Updating Panoramic Display: Instead of aligning frame 3 with respect to a fixed panorama, we can align it with respect to the existing frames including either frame 1 or frame 2 or both. The choice depends on a tradeoff between reducing

- variance, and
- computation time.

We use the example in Figure 3 to illustrate different outcomes for different approaches. As shown in the figure, there are three unknown variables: x_{12}, x_{13}, and x_{23}. The last variable x_{23} is defined as the offset between frame 2 and frame 3. We know that $x_{13} + x_{23} = x_{12}$ under ideal settings. Due to this relationship, we only need two out of three variables. Since x_{12} is known when the third frame enters the system, we first match frame 2 with frame 3.

Since there are βm pixels overlapped between the two images, the TME function T is,

$$T = \beta ma\|x_{23} - x^*_{23}\|^2_2 + \beta mb.$$

The corresponding variance is

$$Var(x_{23}) = \frac{\epsilon}{3\beta ma}.$$

However, we need to know $Var(x_{13})$, because frame 1 is the reference coordinate. We know that x_{12} and x_{23} are independent random variables. Therefore,

$$Var(x_{13}) = Var(x_{12}) + Var(x_{23}) = \frac{\epsilon}{3ma}(\frac{1}{\alpha} + \frac{1}{\beta}). \quad (17)$$

The variance from x_{12} propagates to x_{13} and can grow with each new insertion unless we choose the right images to align with as follows.

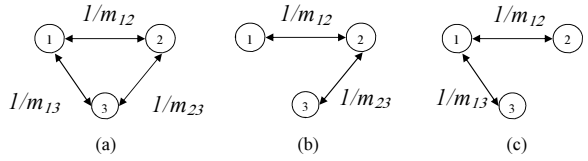

Fig. 4. *Graphical representation of alternate methods. Each node represents a camera frame. Each edge represents an overlap between two frames. With edge length proportion to the inverse of the number of overlapping pixels, selective pair-wise matching finds the shortest path from node 3 to node 1 (the reference node).*

C. Image Alignment Methods

1) Selective Pair-wise Matching (SPM): An alternative is to align frame 3 with frame 1. Define m_{13}, $0 \leq m_{13} \leq m$, as number of pixels between frame 1 and frame 3. To simplify the notation, we define $\gamma = m_{13}/m$. Following a similar derivation, we obtain

$$Var(x_{13}) = \frac{\epsilon}{3ma\gamma}. \tag{18}$$

Although Equation 18 does not contain variance from frame 2, $Var(x_{13})$ is not necessarily smaller than that of Equation 17. If we limit ourselves to pair-wise matching, the choice of matching depends on which pair yields smaller variance,

$$\begin{aligned} Var(x_{13}) &= \frac{\epsilon}{3ma} \min\{\frac{1}{\gamma}, \frac{1}{\alpha} + \frac{1}{\beta}\} \\ &= \frac{\epsilon}{3a} \min\{\frac{1}{m_{13}}, \frac{1}{m_{12}} + \frac{1}{m_{23}}\}. \end{aligned}$$

Figure 4 uses a graph to illustrate the selective pair-wise matching process. With each node represents a frame and each edge represents the overlapping relationship between frames, the choice of the least variance matching is to find the shortest path from the new node to the reference node.

2) Minimum Variance Matching (MVM): In Figure 3, another possible method is to simultaneously align the third frame with both frame 1 and frame 2. This is different from the result in Equation 15, because more pixels are involved in the matching process. In Equation 15, part of frame 1 has been covered by frame 2 in the fixed panorama and hence can not participate the alignment process. Equation 10 shows that variance declines as more pixels are involved in the matching. However, it also could increase the chance of error propagation and increase the variance. The minimum variance matching approach is to find the best set of matching images so that the variance of matching is the smallest.

Let us consider a general case. Assume that the j^{th} frame enters the system, it intersects with a set of existing frames M_j. For the l^{th} frame in M_j, we also know that the number of pixels in frame j intersecting with frame l is m_{jl}. Define X_j and X_l as the vectors that describe the location of image j and image l with respect to the reference image respectively.

Define X_{jl} and X^*_{jl} as the relative offset and the optimal relative offset between frame j and frame l. Then the TME formulation of the matching between frame j and all images in set M_j is,

$$T = \sum_{l \in M_j} \big(am_{jl} \|X_{jl} - X^*_{jl}\|_2^2 + bm_{jl}\big).$$

Since we are looking for the absolution location $X_j = X_l + X_{jl}$, we change the equation above to,

$$T = \sum_{l \in M_j} \big(am_{jl} \|X_j - X_l - X^*_{jl}\|_2^2 + bm_{jl}\big).$$

Apply the same approach we did for Equation 14, we get

$$E(X_j | \{X_l, l \in M_j\}) = \frac{\sum_{l \in M_j} \big(m_{jl}(X_l + X^*_{jl})\big)}{\sum_{l \in M_j} m_{jl}} \tag{19}$$

and

$$Var(X_j | \{X_l, l \in M_j\}) = \frac{\epsilon}{k_d a \sum_{l \in M_j} m_{jl}}.$$

Therefore,

$$\begin{aligned} Var(X_j) &= Var(E(X_j | \{X_l, l \in M_j\})) \\ &+ E(Var(X_j | \{X_l, l \in M_j\})) \\ &= \frac{\sum_{l \in M_j} m_{jl}^2 Var(X_l)}{(\sum_{l \in M_j} m_{jl})^2} \\ &+ \frac{\epsilon}{k_d a \sum_{l \in M_j} m_{jl}}. \end{aligned}$$

From Theorem 1, we know that $Var(X_l) = \frac{\epsilon}{k_d a} w_l$, where w_l has been computed when the l^{th} image entered the system. Inserting them into $Var(X_j)$, we get

$$Var(X_j) = \frac{\epsilon}{k_d a} \big(\frac{1}{\sum_{l \in M_j} m_{jl}} + \frac{\sum_{l \in M_j} m_{jl}^2 w_l}{(\sum_{l \in M_j} m_{jl})^2}\big). \tag{20}$$

Matching over all overlapping frames may not provide us with the smallest variance. What we want is an optimal set of overlapping frames. If the l^{th} image is not used in the matching, we can simply set $m_{jl} = 0$ in Equation 20 to get the new variance. This defines a minimization problem. Define $I_l, l \in M_j$ as the image choice variable, we get the following optimization problem,

$$\min F(\{I_l, l \in M_j\}) = \frac{1}{\sum_{l \in M_j} I_l} + \frac{\sum_{l \in M_j} I_l^2 w_l}{(\sum_{l \in M_j} I_l)^2} \tag{21}$$

subject to

$$\sum_{l \in M_j} I_l \leq \bar{m}_j, \tag{22}$$

$$I_l = \{0, m_{jl}\}, \forall l \in M_j \tag{23}$$

where $\bar{m}_j$ is the maximum limit for number of pixels involved in the matching problem. The constraint in Equation 22 controls the size of the subsequent matching problem to limit computation time. We solve this optimization problem to derive the optimal set of matching images.

3) Minimum Variance Matching Algorithm (MVMA): The optimal solution of Equation 21 yields the minimum variance. However, this is a nonlinear combinatorial problem, which could be very computationally expensive. Though the number of overlapping images $k = |M_j|$ is usually a small number, solving it exhaustively requires time exponential in k.

Looking closer, we observe that when the constraint in Equation 22 is binding,

$$\sum_{l \in M_j} I_l = \bar{m}_j,$$

the objective function in Equation 21 becomes

$$F(\{I_l, l \in M_j\}) = \frac{1}{\bar{m}_j} + \frac{\sum_{l \in M_j} I_l^2 w_l}{(\bar{m}_j)^2}.$$

Then the minimization problem is simplified as,

$$F' = \min_{\{I_l, l \in M_j\}} \sum_{l \in M_j} I_l^2 w_l \tag{24}$$

subject to the constraint in Equation 23. The l^{th} candidate matching image takes m_{jl}-pixel space in total $\bar{m}_j$ pixels and contributes $m_{jl}^2 w_l$ to variance if it is selected. The variance per pixel is $m_{jl}^2 w_l / m_{jl} = m_{jl} w_l$. Define candidate solution set as $\hat{M}_j \subseteq M_j$, sum of pixels in $\hat{M}_j$ as $s_1 = \sum_{l \in \hat{M}_j} m_{jl}$, and partial variance sum as $s_2 = \sum_{l \in \hat{M}_j} I_l^2 w_l$. We propose an approach that is based on the order of the variance density and solves the problem for the case that the constraint in Equation 22 is binding. This algorithm takes the images that contribute less variance first and gradually expands the set until it reaches the constraint.

MVM Algorithm

$\hat{M}_j = \emptyset$, $s_1 = 0$, $s_2 = 0$	$O(1)$
Compute $m_{jl} w_l$, $l \in M_j$,	$O(k)$
Sort $\{m_{jl} w_l, l \in M_j\}$ in ascending order,	$O(k \log k)$
For each l in the ascending sequence of $m_{jl} w_l$,	$O(k)$
if $s_1 + m_{jl} \leq \bar{m}_j$,	
$s_1 = s_1 + m_{jl}$, $s_2 = s_2 + m_{jl}^2 w_l$, $\hat{M}_j = \hat{M}_j \cup \{l\}$	
else	
Break for loop	
end if	
End for	
$F(\hat{M}_j) = \frac{1}{s_1} + \frac{s_2}{s_1^2}$	$O(1)$
Output $\hat{M}_j$ and $F(\hat{M}_j)$	$O(1)$

The algorithm above does not directly offer a solution when $\sum_{l \in M_j} m_{jl} < \bar{m}_j$. This is not a problem, because we can treat $\bar{m}_j$ as a variable to perform a search over it. Recall the F' defined in Equation 24, this new optimization problem is,

$$\min_{\bar{m}_j} \frac{1}{\bar{m}_j} + \frac{F'}{\bar{m}_j^2}, \tag{25}$$

which can be solved straightforwardly by keeping tracking of F value in the for loop of the MVM algorithm. Instead of using the final $F(\hat{M}_j)$, we output the smallest F and its corresponding set of frames. With this modification, we have

Theorem 2: The MVM algorithm finds the optimal set of overlapping frames in $O(k \log k)$ time for a image with k overlapping frames.

D. Pair-wise Matching

As stated in Section III-B, with an optimal set of existing frames, the resulting pair-wise alignment sub problems can be solved using any image mosaicing methods in Section II-.2. Equation 19 also tells us that the optimal alignment parameter, X, is a weighted average of the pair-wise matching results using the number of overlapping pixels as the weight.

V. Experiments and Results

We have installed a Canon VCC3 Pan-Tilt-Zoom camera at the UC Berkeley campus. The camera has a pan range of $180°$ and a tilt range of $55°$. It features an 1/4-inch CCD sensor with a maximum resolution of 768×576. Its horizontal field of view ranges from $4°$ to $46°$. Our processor is a 2.53Ghz Intel Pentium 4 PC with 1GB RAM and an 80GB hard drive. We have conducted two phases of tests including construction phase and update phase.

A. Construction Phase

In construction phase, we construct a panorama by directing the camera to visit a set of predefined coordinates, each of which defines a composing frame of the panorama. We have taken 21 320×240-pixel frames. During the construction process, we combine our MVM Algorithm with Breadth First Search (BFS) to generate a panorama. The BFS starts with camera home position frame, which also our reference frame. It is node 0 in Figure 5. The BFS incrementally covers all 21 points represented by the 21 nodes in the graph illustrated in Figure 5. The pair-wise matching algorithm is a feature-based algorithm. The overall computation time to generate such a panorama is 9.7 seconds, which is even less than the camera travel time. The VCC3 camera can only travel with a maximum speed of $70°$ per second. To cover all 21 points, it takes about 30 seconds because of frequent stops. Since our algorithm generates the panorama incrementally, it can compute the panorama as the camera travels around. It outputs the full panorama 331 milliseconds after the camera completes its travel.

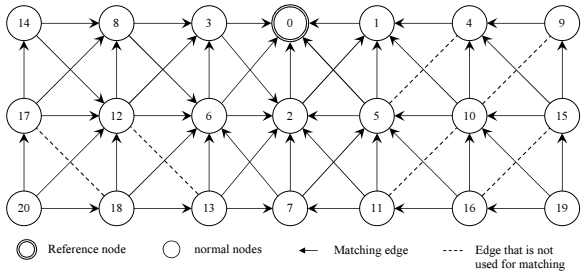

Fig. 5. *Resulting matching sequence from MVM-BFS using the 21 frames. Each node represents a frame and node numbers are corresponding to BFS frame capturing order. The distribution of matching edges is determined by image alignment mechanisms. The alignment edges are directional: node a → node b means frame a is captured later and uses the existing frame b for alignment.*

B. Update Phase

We next test how long it takes to update an existing panoramic display. Based on results of 1000 test runs, the algorithm required an average of 331 milliseconds to update the panorama. The parameter $\bar{m}_j$ in Equation 22 determines the trade-off between panorama quality and computation time. In our settings, $\bar{m}_j = 90000$ offers the best trade-off. The update operation is activated when the camera leaves for a new pan-tilt-zoom setting. Since camera travel and stabilization time usually requires more than 331 milliseconds, image

alignment can be computed as fast as the camera can be tele-operated.

VI. Conclusions and Future Work

We present algorithms for maintaining a high resolution panoramic display for disaster response, environmental monitoring, and security applications using a tele-operated robotic camera. Since the robotic camera can cover a large region of interest by adjusting its pan-tilt-zoom parameters, it is difficult to keep track of where and when the camera has visited. We construct a updated spherical panoramic display from the time stamped frame sequences. Whenever the camera changes its pan-tilt-zoom settings, we update the panorama by inserting a new frame sequence.

We propose a variance-based quality metric to analyze how errors get accumulated and use it to show that arbitrarily selecting a set of existing frames to register new frames can cause registration errors to grow out of control. We then propose a minimum variance alignment algorithm. Our algorithm can register a new frame in $O(k \log k)$ time for a panorama with k overlapping frames.

In the future, we will develop new data structures for image alignment and storage. We know that after a new frame is inserted into the system, it may provide a better alignment choice for existing frames. Adjustment of existing frames to improve the quality of the panorama is an interesting problem. The new data structure and its corresponding algorithms can also help us to efficiently move old frames to hard disk storage. We are also developing methods that allow queries into the time history of panoramas.

Acknowledgments

Thanks Carlo Séquin for bringing evolving panorama problem into attention. Thanks are given to Q. Hu, B. Lin, X. Ling, and V. Jan for implementing part of the project. We thank H. Lee, A. Dahl, J. Schiff, I. Chen, K. Paulsen, J. Young, M. Gosalia, T. Shlain, G. Gershoni, J. Lecavalier for their contributions in Demonstrate system development. Our thanks to J. Luntz, P. Wright, R. Bajcsy, D. Plautz, C. Cox, D. Kimber, Q. Liu, J. Foote, L. Wilcox, Y. Rui, K. "Gopal" Gopalakrishnan, R. Alterovitz, and I. Y. Song for insightful discussions and feedback.

References

[1] S. Baker and S. K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175 – 196, November 1999.

[2] R. Benosman and S. B. Kang. *Panoramic Vision*. Springer, New York, 2001.

[3] R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 1965.

[4] E. Castro and C. Morandi. Registration of translated and rotated images using finite fourier transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):700–703, Sept. 1987.

[5] X. Dai and S. Khorram. A featured-based image registration algorithm using improved chain-code representation combined with invariant moments. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5):2351–2363, 1999.

[6] J. Foote and D. Kimber. Enhancing distance learning with panoramic video. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, 2001.

[7] C. J. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. 4th Alvey Vision Conference, Manchester*, pages 147–151, 1988.

[8] H. Hu, L. Yu, P. W. Tsui, and Q. Zhou. Internet-based robotic systems for teleoperation. *Assemby Automation*, 21(2):143–151, May 2001.

[9] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Mosaic representations of video sequences and their applications. *Signal Processing: Image Communication*, 8(4):327–351, May 1996.

[10] B. Y. Kim, K. H. Jang, and S. K. Jung. Adaptive strip compression for panorama video streaming. In *Computer Graphics International (CGI'04), Crete, Greece*, June 2004.

[11] D. Kimber, Q. Liu, J. Foote, and L. Wilcox. Capturing and presenting shared multi-resolution video. In *SPIE ITCOM 2002. Proceeding of SPIE, Boston*, volume 4862, pages 261–271, Jul. 2002.

[12] C. Kuglin and D. Hines. The phase correlation image alignment method. In *IEEE International Conference on Cybernet Society, New York*, 1975.

[13] H. Li, B.S. Manjunath, and S.K. Mitra. A contour-based approach to multisensor image registration. *IEEE Trans on Image Processing*, 4(3):320–334, March 1995.

[14] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[15] S. K. Nayar. Catadioptric omnidirectional camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, June 1997.

[16] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Metric 3D surface reconstruction from uncalibrated image sequences. In *Proc. SMILE Workshop (post-ECCV'98)*, pages 138–153. Springer-Verlag, June 1998.

[17] B. S. Reddy and B. N. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. In *IEEE Transactions on Image Processing*, volume 5, pages 1266–1271, August 1996.

[18] Y. Y. Schechner and S. K. Nayar. Generalized mosaicing. In *Proceedings of the 8th IEEE International Conference on Computer Vision, Vancouver, British Columbia, Canada*, volume 1, pages 17–24, July 2001.

[19] D. Song and K. Goldberg. Sharecam part I: Interface, system architecture, and implementation of a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference on Intelligent Robots (IROS)*, Nov. 2003.

[20] D. Song, Q. Hu, N. Qin, and K. Goldberg. Automating high resolution panoramic inspection and documentation of construction using a robotic camera. In *(Submitted to) IEEE Conference on Automation Science and Engineering, 2005*, Aug. 2005.

[21] D. Song, A. Pashkevich, and K. Goldberg. Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference on Intelligent Robots (IROS)*, Nov. 2003.

[22] R. Swaminathan and S. K. Nayar. Nonmetric calibration of wide-angle lenses and polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1172–1178, October 2000.

[23] K.-H. Tan, H. Hua, and Ahuja N. Multiview panoramic cameras using mirror pyramids. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):1941– 946, July 2004.

[24] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137 54, 1992/11/. Copyright 2005, IEE.

[25] P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 278–294. Springer-Verlag, 2000.

[26] E. Trucco, A. Doull, F. Odone, A. Fusiello, and D. M. Lane. Dynamic video mosaics and augmented reality for subsea inspection and monitoring. In *Oceanology International, United Kingdom*, March 2000.

[27] Y. Xiong and K. Turkowski. Creating image-based VR using a self-calibrating fisheye lens. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 237–243, June 1997.

[28] Z. Yang and F. S. Cohen. Image registration and object recognition using affine invariants and convex hulls. *IEEE Transactions on Image Processing*, 8(7):934–846, 1999.

[29] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

[30] Z. Zhu, G. Xu, E. M. Riseman, and A. R. Hanson. Fast generation of dynamic and multi-resolution 360-degree panorama from video sequences. In *IEEE International Conference on Multimedia Computing and Systems, Florence, Italy*, volume 1, pages 9400–9406, June 1999.

[31] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *IEEE International Conference on Computer Vision and Pattern Recognition, Puerto Rico*, June 1997.