Establishing Visual Correspondence from Multi-Resolution Graph Cuts for Stereo-Motion

Joshua Worby
Electrical and Computer Engineering
University of Toronto
Toronto, Canada
worbyjos@utoronto.ca

W. James MacLean
Electrical and Computer Engineering
University of Toronto
Toronto, Canada
maclean@eecg.toronto.edu

Abstract

This paper presents the design and implementation of a multi-resolution graph cuts (MRGC) for stereo-motion framework that produces dense disparity maps. Both stereo and motion are estimated simultaneously under the original graph cuts framework [7]. Our framework extends the problem from one to five dimensions, creating a large increase in complexity. Using three different multi-resolution graph cut algorithms, LDNR, EL and SAC, we reduce the number of pixels m and the number of labels n that limit the $\alpha - \beta$ swap algorithm (with complexity $O(mn^2)$) required from the definition of our semi-metric smoothness function. This results in a reduction of computation time and the ability to handle larger images and larger label sets. The choice of the three MRGC algorithms to use in computation determines the appropriate level of accuracy and computation time desired.

1. Introduction

The goal of computer vision is to design an artificial system that models the world through the analysis of information from images. One area of research is to create a model of the world through three-dimensional scene reconstruction. This type of research has applications in areas such as space, medical, entertainment, historical, criminal investigation, and object modeling.

Two areas of focus in computer vision, *stereo-vision* and *motion estimation*, handle scene reconstruction. Stereo algorithms infer depth of perceived scene points from two images taken from different viewpoints. Motion algorithms determine scene structure from two temporally separated images. A key issue in both cases is the problem of *correspondence*: determining matching elements (scene points, features, lines) between images. The position difference between corresponding features is defined as the *disparity*, which has an inverse relationship with depth. Disparity estimates for every single pixel in an image form a dense disparity map. Problems in establishing visual correspondence arise due to lighting and scene assumptions, camera sensor noise, textureless regions, depth discontinuities and occlusions.

1.1. Previous Approaches

Previous approaches involved both local and global methods. Local methods, often called window-based methods or correlation-based methods, find the optimal displacement of a fixed sized region between two consecutive frames. Typically these methods compare intensities within the region according to some likeness measure. Obstacles lie in the choice of window size and the assumption as to which motion model to incorporate over the region. Shiftable windows [3] and windows with adaptive sizes [15] are common techniques for varying the window size. In perhaps the most well known local method, Lucas and Kanade [14] use a local constant motion model for the optical flow and determine a weighted least squares solution.

Global methods provide greater accuracy compared to local methods, but have the disadvantage of creating a higher computation cost. Generally, they formulate a global energy function composed of a data term and a smoothness term.

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d). \quad (1)$$

Measuring the agreement between the disparity function d and the input image pair is encoded in the data term, $E_{data}(d)$. Meanwhile, the smoothness assumptions made by the algorithm are embedded in the smoothness term, $E_{smooth}(d)$, which is made more tractable by restricting the smoothness term to only neighbouring pixels. In regularisation techniques [12], Horn and Schunck are able to overcome the aperture problem but then encounter problems at object boundaries. Brox *et al.*[17] handle object boundaries by employing robust estimators, developed by Black and Anandan [2], on both the data and smoothness terms of their global energy function. This is the current state of the art in

optical flow computation. Other global methods use differing methods to minimize the global energy function. Some methods are simulated annealing [10], highest confidence first [8] and mean-field annealing [9].

Graph cuts [5, 6, 7, 4, 13, 18] is an example of an optimization approach that computes visual correspondence. The method incorporates a global energy function and a maximum flow technique to provide very accurate results. However, like most optimization approaches, it has a high computational cost. The technique depends greatly on the number of pixels in the image, or image size, and the size of the disparity range, which translates directly to the number of labels. Nowadays, image sequences are generally 640 $\times$ 480 pixels in size or greater and typically contain scenes with large object or camera motion, making this method intractable and very slow. Veksler [19] studies the problem of search-space reduction for graph cuts, although she concludes the multiresolution approach is a failure. The work in [19] is limited to stereo only (using the α-expansion algorithm), and reports an average speed up factor of 2.8.

In this paper, we address the limitations of the graph cuts technique described above, but in the context of the α-β swap algorithm, with potentially much larger label spaces than those found in [19] as we are interested in four-frame stereo-motion. The goal of this work is to provide a framework to compute visual correspondence, or dense disparity maps, with increased speed, reduced computational cost and the ability to handle larger images and disparities, all the while maintaining high accuracy. The key to achieving our goals is the implementation of a multiscale technique for graph cuts that encodes both the combined stereo and motion constraints. The basic assumption is that a multiscale approach allows for a method to quickly initialize the objective function closer to the global minimum than if it were left to its own devices. This speeds up the minimization step and allows for larger disparities and larger images. To improve the accuracy, imposing both stereo and motion constraints should result in greater accuracy than using only stereo or motion constraints separately.

The rest of this paper is organized as follows: Section 2 discusses the problems with the original graph cuts for stereo [7]. Next, we provide a detailed description of the design of an objective function that encodes both stereo and motion constraints and the methodology involved in creating a multiscale method in Section 3. In Section 4, we present three multi-resolution graph cut (MRGC) for stereo-motion algorithms. Finally, we present resulting disparity maps and their analysis in Section 5, with Section 6 providing conclusions and possible directions for future work.

2. Graph Cuts

In this section, we analyze the original graph cuts technique for stereo [5, 6, 7, 4, 13, 18]. Graph cuts takes a graph theory approach to finding a solution to the energy minimization problem. Nodes in the graph represent image pixels while the graph terminals represent every possible label. Edges between nodes are weighted according to our smoothness term $E_{smooth}(d)$, while the data term $E_{data}(d)$ is used to weight the edges between nodes and terminals. Finding a solution is possible by breaking down the graph into many two-terminal graphs, determining which configuration has the lowest energy. Once each pixel is attached to only one terminal, it is assigned that terminal's label or disparity value.

In graph cuts, the global energy function is the sum of our data term plus a weighted smoothness term:

$$\begin{aligned} E(d_p) &= E_{data}(d) + \lambda E_{smooth}(d) \\ &= \sum_{p \in P} D_p(d_p) + \sum_{p,q \in N} V_{p,q}(d_p, d_q) \quad . \end{aligned} \tag{2}$$

where d is a labeling of the image, p and q are neighbouring pixels, N is the neighbourhood of p.

The data term E_{data} evaluates the level of correspondence between values in the input image pair while reducing its sensitivity to image sampling [1, 18]. Since the search is one-dimensional for stereo disparity, we find how well pixel p fits into the real valued range of disparities $(d-\frac{1}{2}, d+\frac{1}{2})$.

$$C_{fwd}(p,d) = \min_{d-\frac{1}{2} \leq x \leq d+\frac{1}{2}} |I_p - I'_{p+x}| \quad .$$

We represent $p + x$ as a pixel which has coordinates of p shifted by disparity x. I' represents intensities in the right image. Fractional values I'_{p+x} are obtained by linear interpolation between discrete pixel values. $C_{rev}(p, d)$ is computed similarly but in the reverse direction. Thus, a single data term is

$$D(p,d) = (\min\{C_{fwd}(p,d), C_{rev}(p,d), const\})^2 \tag{3}$$

where $const$ is a constant value that ensures robustness.

The original graph cuts for stereo algorithm chose a piecewise constant prior as their smoothness term E_{smooth}. Formally, given one-dimensional stereo disparity labels d_1 and d_2, the smoothness term is defined as

$$V_{p,q}(d_1, d_2) = u_{p,q} \cdot (1 - \delta(d_1 - d_2)) \tag{4}$$

where

$$u_{p,q} = U(|I_p - I_q|) = \begin{cases} 2K & \text{if } |I_p - I_q| \leq \tau_I \\ K & \text{if } |I_p - I_q| > \tau_I \end{cases} \tag{5}$$

with K as our penalty constant and τ_I as our intensity difference threshold. τ_I is found experimentally with the optimal setting being 5. This function satisfies the three properties of a metric [7], formally defined as

$$\begin{aligned} V_{p,q}(\alpha,\beta) = 0 &\leftrightarrow \alpha = \beta \\ V_{p,q}(\alpha,\beta) &= V_{p,q}(\alpha,\beta) \geq 0 \\ V_{p,q}(\alpha,\beta) &\leq V_{p,q}(\alpha,\gamma) + V_{p,q}(\gamma,\beta) \end{aligned} \tag{6}$$

With a smoothness term defined as a metric, the use of the much faster α-expansion algorithm for two-terminal graph building $O(mn)$ is allowed (m is the number of pixels and n is the number of disparity values).

On the other hand, for motion estimation disparity values are two-dimensional. A similar data term can be used except that it is now extended to account for the second dimension when reducing the influence of image sampling. The smoothness term, however, requires that we define the function in terms of a semi-metric since it does not satisfy the triangle inequality (third property of Equation 6). Thus, our smoothness term is that of a piecewise smooth prior. Formally, the interaction functional is

$$V_{mot_{p,q}}(d_p, d_q) = \lambda \min(const, A) \tag{7}$$

where

$$A = (d_p^h - d_q^h)^2 + (d_p^v - d_q^v)^2 \quad . \tag{8}$$

We enforce robustness by $const$, λ restricts the influence of the smoothness term and the superscript terms v and h denote the vertical and horizontal components of the motion vector, respectively. As such, we are forced to use the slower α-β swap algorithm for two-terminal graph building $O(mn^2)$. Reducing the number of labels becomes even more vital in effectively decreasing computation time.

3. Multi-Resolution Graph Cuts (MRGC) for Stereo-Motion

Combining both stereo and motion constraints provides tighter constraints on the system than either stereo or motion constraints used separately. In turn, this should have the desired effect of improving accuracy. Assuming the input to the system consists of four rectified images: L_t, R_t, L_{t+1}, R_{t+1}. These are, respectively, the left and right stereo pairs at time t and $t+1$ of a stereo video sequence.

Refer to Figure 1 where we establish *Left*$_t$ as our reference frame. A point in our reference frame p_{L_t} is related to its corresponding points in the other images by the following four relationships

$$\begin{array}{ll} p_{L_t} + d_t = p_{R_t} & \text{for stereo pair at time } t \\ p_{L_t} + \vec{d}_L = p_{L_{t+1}} & \text{for motion left pair} \\ p_{L_{t+1}} + d_{t+1} = p_{R_{t+1}} & \text{for stereo pair at time } t+1 \\ p_{R_t} + \vec{d}_R = p_{R_{t+1}} & \text{for motion right pair} \end{array} \tag{9}$$

where disparity values d_t, d_{t+1}, $\vec{d}_L$ and $\vec{d}_R$ are stereo pairs at time t, $t+1$, left-motion image pairs, and right-motion image pairs, respectively. Thus, we can impose the following circular *combined stereo-motion* constraint.

$$0 = d_t + \vec{d}_R - d_{t+1} - \vec{d}_L \quad . \tag{10}$$

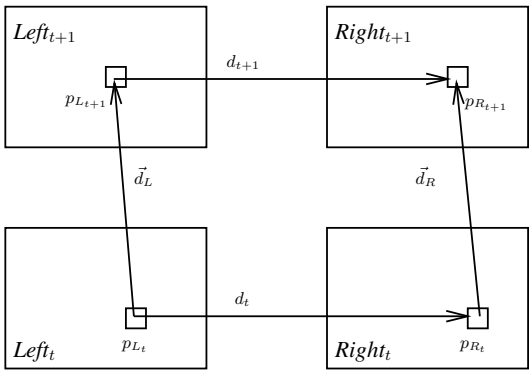

Figure 1. Example of two stereo image pairs ((L_t, R_t) and (L_{t+1}, R_{t+1})) at times t and $t+1$, respectively. A point p_{L_t} in the reference frame *Left*$_t$ is related to its corresponding points in the other images via established disparity values d_t, d_{t+1}, $\vec{d}_L$ and $\vec{d}_R$ for stereo pairs at time t, $t+1$, left-motion image pairs, and right-motion image pairs, respectively.

The downside to this approach is the large increase in the label set. For stereo, labels are only one-dimensional. Motion has two-dimensional labels. However, in our case, the number of labels is now six-dimensional. Given rectified images, it is possible to reduce this down to five dimensions. Once we obtain the vertical direction in one motion pair, we assume that the other motion pair has the same vertical disparity. This would reduce the number of labels to

$$|L| = st_x \times st_x \times (mot_x \times mot_y) \times (mot_x) \quad . \tag{11}$$

where st_x denotes the number of possible labels for stereo computation in the horizontal direction and mot_x and mot_y are the possible number of labels for motion in the horizontal and vertical directions, respectively. The $|\cdot|$ notation signifies the number of elements in the label set rather the absolute value function.

We formulate our global energy function in the same form as Equation 2. Since we need two stereo estimations and two motion estimations in the combined stereo-motion constraint, we combine the data terms for graph cuts for stereo and for motion. The stereo terms are $C_{st}(p_{L_t}, f_p^{d_t})$ and $C_{st}(p_{L_{t+1}}, f_p^{d_{t+1}})$, and the motion terms are $C_{mot}(p_{L_t}, f_p^{\vec{d}_L})$ and $C_{mot}(p_{R_t}, f_p^{\vec{d}_R})$, Our resulting data term for pixel p given label f_p is formulated in a similar fashion to a L_2 norm.

$$\begin{aligned} D_p(f_p) &= D_p(f_p^{d_t}, f_p^{d_{t+1}}, f_p^{\vec{d}_L}, f_p^{\vec{d}_R}) \\ &= \min(\tau_{D_{cutoff}}, A) \end{aligned} \tag{12}$$

edge	weight	for
t_p^{α}	$D_p(\alpha) + \sum_{\substack{q \in N_p \\ q \notin S}} V_{p,q}(\alpha, f_q)$	$p \in S$
t_p^{β}	$D_p(\beta) + \sum_{\substack{q \in N_p \\ q \notin S}} V_{p,q}(\beta, f_q)$	$p \in S$
$e_{p,q}$	$V_{p,q}(\alpha, \beta)$	$\{p,q\} \in N$ $p,q \in S$

Table 1. Edge weight assignments for the α-β swap algorithm. (Reproduced from [18].)

where

$$A = (C_{st}(p_{L_t}, f_p^{d_t})^2 + C_{mot}(p_{R_t}, f_p^{\vec{d}_R})^2 + C_{st}(p_{L_{t+1}}, f_p^{d_{t+1}})^2 + C_{mot}(p_{L_t}, f_p^{\vec{d}_L})^2)^{\frac{1}{2}} \quad (13)$$

and $\tau_{D_{cutoff}}$ is a constant used to make the data term robust to outliers.

We also use a set of *super-labels* to index into this large six-dimensional set of labels.

$$f = \{f^{d_t}, f^{d_{t+1}}, \vec{f}^{d_L}, \vec{f}^{d_R}\} \quad (14)$$

In a similar fashion, we formulate our smoothness term to arrive at

$$V_{p,q}(f_p, f_q) = \lambda \min(\tau_{S_{cutoff}}, B) \quad (15)$$

where

$$B = (V_{st}(f_p^{d_t}, f_q^{d_t})^2 + V_{mot}(f_p^{\vec{d}_R}, f_q^{\vec{d}_R})^2 + V_{mot}(f_p^{\vec{d}_L}, f_q^{\vec{d}_L})^2 + V_{st}(f_p^{d_{t+1}}, f_q^{d_{t+1}})^2)^{\frac{1}{2}} \quad . \quad (16)$$

The constant $\tau_{S_{cutoff}}$ performs the same function as in the data term; providing robustness to outliers, while the λ term weights the influence of the smoothness term. The stereo interaction functions $V_{st}(f_p^{d_t}, f_q^{d_t})$ and $V_{st}(f_p^{d_{t+1}}, f_q^{d_{t+1}})$ are described in Equations 4 and 5 without the presence of their respective λ terms. The motion interaction functions are taken from Equation 7.

The second problem with this approach is that we are forced to use the slower α-β swap algorithm $O(mn^2)$ for graph building with our smoothness function not satisfying the triangle inequality. Edges in the graph are weighted according to Table 1. Then the solution to each graph is found using an effecient maximum flow algorithm [4].

To account for the large increase in the label set, we use a multi-resolution approach to reduce computation time. This allows a reduction of the number of labels and the number of pixels within a known sampling factor. The second benefit arises from level initialisation. Finding the solution at one level of the pyramid initialises the next level of the pyramid closer to its ideal solution. This benefit repeats itself at every level of the pyramid until we reach the finest level of the pyramid. This process is best illustrated in Figure 2.

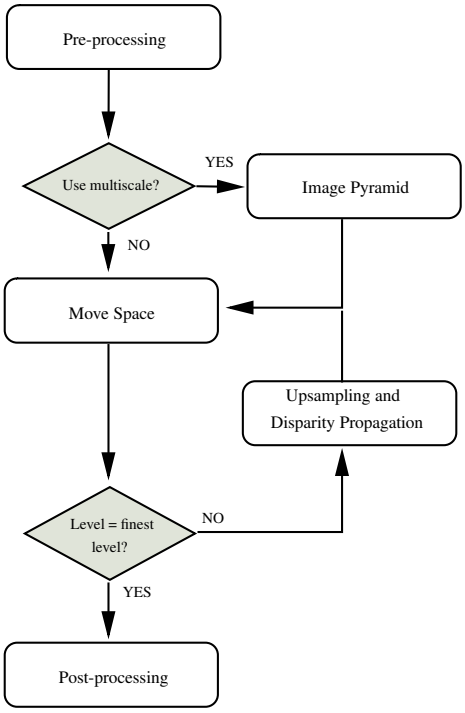

Figure 2. Flow diagram of multi-resolution graph cuts using stereo-motion constraints system.

4. Three MRGC for Stereo-Motion algorithms

As is common in most stereo algorithms, there is a trade-off between computation time and accuracy. Finding the right balance between the two depends on the problem at hand. We have formulated three different algorithms [20] under the multi-resolution graph cuts (MRGC)for stereo-motion framework with a different balance in mind. All three algorithms have similar goals:

1. **Reduce the number of pixels**
2. **Reduce the number of labels**
3. **Improve computation time**

The general form of the multi-resolution approach is described in Listing 2, called the *Level Seeding MRGC (LS)* algorithm. $Dmap_i$ represents the disparity map obtained for level i with $Dmap_{final}$ referring to the final solution obtained at the finest level. The energy minimisation algorithm (EMA in step 3c) is the step that varies among all the algorithms described in this section.

4.1. Level Disparity Neighbourhood Restricted (LDNR)

To greatly reduce the number of labels, we introduce the *Label Disparity Neighbourhood Restricted MRGC (LDNR)*

1.	Create Gaussian image pyramid
2.	Determine disparity range for pyramid levels
3.	FOR i = numLevels-1 to 0
	IF i = (numLevels-1)
3a.	$Dmap_i$ = Normal graph cuts for stereo-motion
	ELSE
3b.	Upsample disparity map
3c.	$Dmap_i$ = EMA($Dmap_{i+1}$)
4.	$Dmap_{final}$ = $Dmap_0$

Table 2. General outline of all multiscale methods. $Dmap_i$ represents the disparity map obtained for level i, while EMA refers to the energy minimization algorithm used in computation.

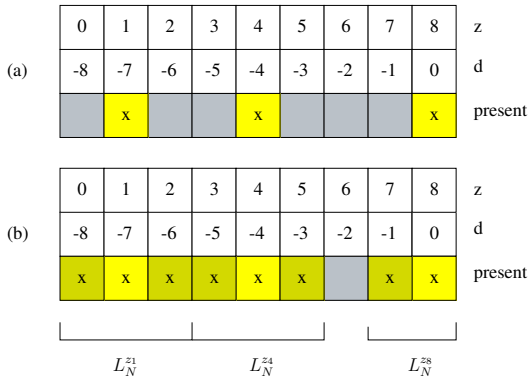

Figure 3. The (b) label disparity neighbourhood of (a) the labels present (in yellow).

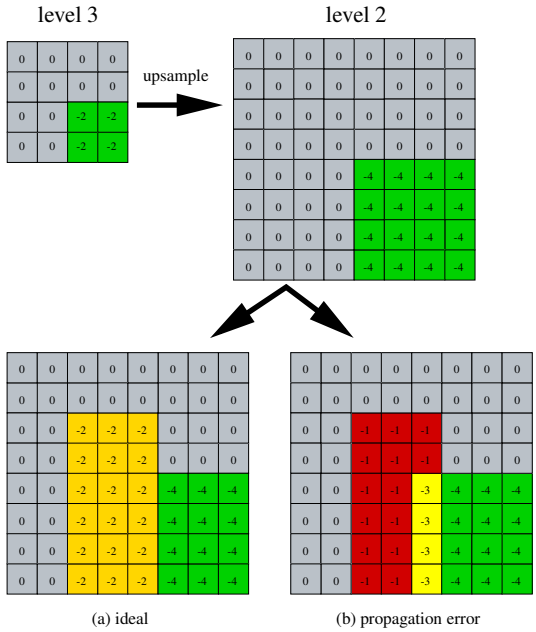

Figure 4. After upsampling a disparity map from level 3 to level 2, we find the presence of label error propagation between levels of the pyramid. The ideal solution for level 2 should be (a). However, a disparity value of -2 is not in the label disparity neighbourhood of disparity values of 0 or 4. Therefore, using LDNR, the disparity map achieved at level 2 would be as in (b) where the disparity values of -1 (red) and -3 (yellow) are errors that propagate to subsequent levels.

algorithm. This algorithm uses the notion of a *label disparity neighbourhood* to restrict the size of the label sets. For each disparity label L present in the upsampled map, we define a neighbourhood in the range $\pm\delta$ around L (see Figure 3, where $\delta = 1$). Label swaps are restricted to being between a label and the other labels in its neighbourhood. The neighbourhoods are determined once at the start, so the set of possible disparity values is fixed through all iterations. We justify this restriction by assuming that disparity estimates are already close to their ideal value, and as such, have a very short disparity distance to travel. Now, rather than swapping all possible label combinations, we swap between a reduced set.

4.2. Expanding Label Dispariy Neighbourhood at Every Iteration (EL)

The LDNR algorithm has one downside to it: its overall accuracy suffers. It introduces an error propagation problem, best illustrated in Figure 4. Once we find the solution to level 3 of the pyramid, we upsample it to get the disparity map at level 2, where one pixel is upsampled to four. It is in the uncertainty of these extra three pixels and restricting swaps to among label disparity neighbourhoods, that the error propagation problem is introduced. Ideally, the solution would be (a), but what really results for that level is (b), where the pixels with a disparity of -1 are incorrect. This error propagates further down the pyramid until the finest level of the pyramid is reached, where the accuracy will greatly suffer.

To remedy this problem, we propose the EL algorithm which allows the label set to grow at every iteration, while it remained constant in the LDNR algorithm. Label swaps are still done within label disparity neighbourhoods, but as new labels are added during swaps, neighbourhoods are created for them also, and swaps allow using values in these new neighbourhoods. Over time the disparity values slowly

propagate towards their ideal values. This method slowly increases the number of labels, but in practice, it rarely expands to all labels.

4.3. Swap All Combinations LDNR-MRGC (SAC)

The third algorithm takes a different approach to solving the error propagation problem. Called *Swap All Combinations LDNR-MRGC* (SAC), it removes the restriction that swaps must occur within label disparity neighbourhoods. The algorithm initially determines the set of labels present, $L_{present}$, in the upsampled disparity map, without growing throughout the algorithm. Then we determine the label disparity neighbourhoods for each of the labels in $L_{present}$, adding these labels to a new label set L_{todo}. Then the algorithm computes α-β swaps for all of the combinations of labels in L_{todo}. This allows for larger disparity distances travelled, improving the accuracy. However, there is a large increase in label combinations, causing a large increase in computation time. This allows pixels at discontinuities to take values of neighbouring pixels in a fashion similar to the filtering operation described in [19].

5. Computational Examples

In order to demonstrate the feasibility of the approach we evaluate the performance of the *Multi-Resolution Graph Cuts for Stereo-Motion* system in comparison to the original *graph cuts for stereo* technique of [5, 7, 4, 13, 18]. Without the presence of any known ground truth data for stereo image sequences with motion, we are forced to test all algorithms against ground truth data obtained from the Middlebury Stereo webpage [16]. All image sequences are rectified stereo image sequences that contain static scenes and robust objects.

The system implementation was done in C++ on a Pentium IV 3GHz machine with 4GB of memory, running the Linux operating system. We test our algorithms using the Tsukuba sequence, which has a translating camera from right to left. Our stereo-motion framework is tested using three consecutive images of the sequence. For example, the first stereo pair of images L_t and R_t correspond to images 3 and 4. The second stereo pair of images at time $t+1$, L_{t+1} and R_{t+1}, correspond to images 4 and 5.

Both the MRGC stereo algorithms and the MRGC motion algorithms generate Gaussian pyramids for two input images. Figure 5 shows a sample of these Gaussian pyramids generated by the *Image Pyramid* stage. In the pyramid, the original input images are shown at level 0. For stereo, $image\ 1$ and $image\ 2$ correspond to images *Left*$_t$ and *Right*$_t$, respectively. When computing motion, $image\ 1$ and $image\ 2$ correspond to the im_t and im_{t+1} image pairs, respectively.

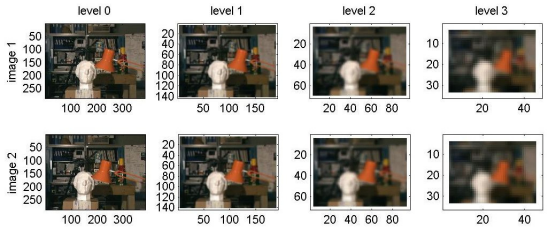

Figure 5. Image pyramids for an image pair. From left to right, the images proceed up the pyramid, passing from the finest to the coarsest level of the pyramid with each image axis indicating the image dimensions in pixels.

5.1. Number of Labels

In the graph cuts framework, factors influencing computation time are the number of pixels m and the number of labels n. This is evident in the time complexity equations $O(mn)$ for the α-expansion algorithm and $O(mn^2)$ for the α-β swap algorithms.

Figure 6 shows the results of varying the number of labels and their effect on processing time. In the top graph, we compare the original stereo graph cuts algorithm and the MRGC stereo algorithm. The bottom graph compares the standard motion graph cut algorithm and the MRGC motion version. As the number of labels increases, the computation time increases. During computation, we only compute unique label combinations, which explains the near linear curve of α-expansion $O(mn)$ for stereo. For motion, it appears to follow $O(mn^2)$ for the α-β swap algorithm. However, no regression analysis has been done to confirm these interpretations. While it may seem odd to explore label sets larger than those known to be present, it should be noted that we will not always know the correct disparity range in advance. Also, another important result is that the MRGC algorithm for motion starts to out-perform the standard motion graph cuts algorithm in terms of computation time once the label set sizes start to become large.

Figure 7 demonstrates this characteristic for all algorithms implemented. We can see that we exhibit near linear computation time for stereo graph cuts algorithms, while we do better than quadratic computation time for the motion and combined stereo-motion algorithms.

5.2. Accuracy

For accuracy measurements, disparity values are only considered in regions of non-occlusion. From top to bottom, left to right, Figure 8 shows the disparity maps for the MRGC motion algorithm, LDRN, EL and SAC stereo-motion algorithms (where we only show the first motion pair of images in the four image set). Similarly, Table 3

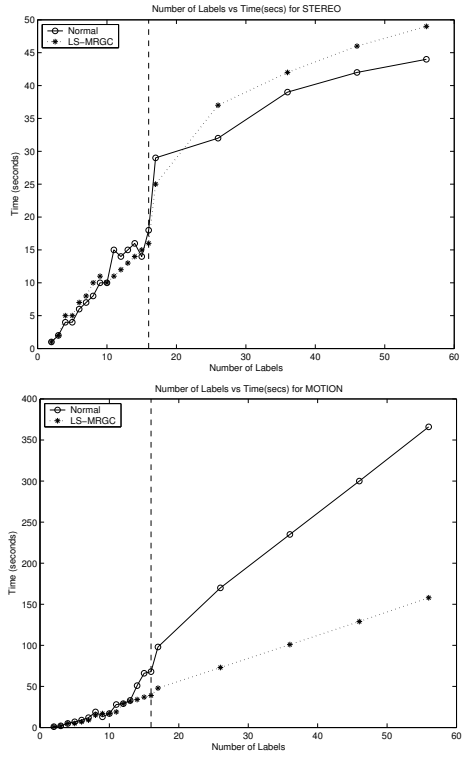

Figure 6. Number of labels versus time to convergence for the original stereo graph cuts algorithm and the LS stereo graph cuts algorithm (top) and the motion graph cuts algorithm (bottom). The dotted vertical line indicates the range of disparities known to be actually present in the data.

shows the accuracy and time results we obtained for each algorithm. In the disparity maps, we can clearly see that LDNR displays the error propagation problem, while EL and SAC recover from it. However, both these algorithms increase in computation because they compute more label combinations. One thing to notice when looking at the motion results is that the data supports the notion that computation time will decrease using a multi-resolution approach.

6. Conclusions and Future Work

We have developed a a multi-resolution graph cuts for stereo-motion system to address the visual correspondence problem for stereo vision and motion estimation that is able to handle larger images and larger label sets, as well as, reduce the computation time.

The stereo-motion framework increased the time complexity of the problem by increasing the number of dimensions from one to five (one-dimension for each of the two

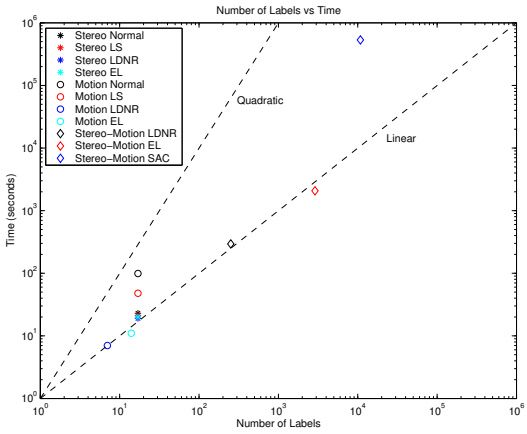

Figure 7. The relationship between the number of labels used in an algorithm and the time to convergence for all graph cut algorithms (stereo, motion and combined stereo-motion).

stereo image pairs and two-dimensions for the motion image pairs). This required the use of the much slower α-β swap algorithm for graph building. However, in this framework we are still able to establish the trade-off between an increase in comptuation time that results in an increase in accuracy. This is best exemplified in our three MRGC algorithms: LDNR, EL and SAC; where LDNR is the fastest but least accurate algorithm and SAC is on the opposite end with the slowest but most accurate algorithm.

The system developed has many areas to investigate further to decrease computation time and increase accuracy. The following are issues for future work:

- Finding the fundamental matrix between motion image pairs, according to [11], so that the search space for correspondence becomes one-dimensional, reducing the number of dimensions from five to four.
- Formulate a smoothness term in the form of a metric, allowing for the use of the much faster α-expansion algorithm.
- Determine an improved convergence criteria for the energy minimization techniqe since the system energy has nearly stabilized after 4-5 iterations.
- Compute sub-pixel accuracy disparity measurements by breaking down our label sets even further.

References

[1] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on*

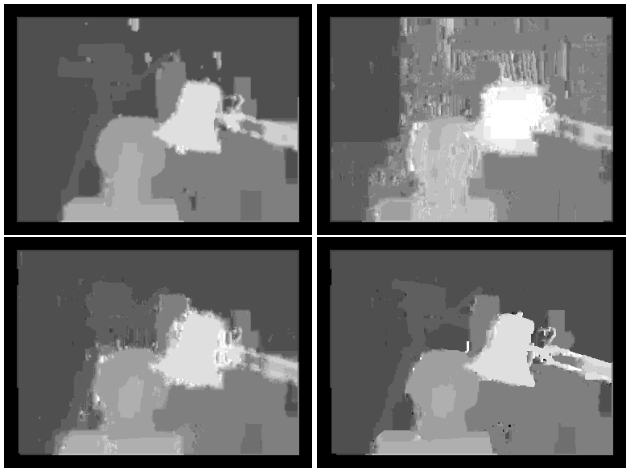

Figure 8. The disparity maps for the left motion pair of images: MRGC (top-left), LDNR (top-right), EL (bottom-left), SAC(bottom-right).

Type	Alg.	#Labels	Time(s)	Acc%
ST	Normal	17	23	97.406
ST	MRGC	17	23	97.399
ST	LDNR	17	23	97.402
ST	EL	17	23	97.399
MOT	Normal	17	99	93.671
MOT	MRGC	17	48	90.168
MOT	LDNR	7	7	84.577
MOT	EL	14	11	97.329
SM	Normal	83521	A	—
SM	LS	83521	B	—
SM	LDNR	251	294	80.71
SM	EL	2890	2073	87.18
SM	SAC	10798	533593	90.96

Table 3. The relationship between the number of labels and the time to convergence for all graph cuts algorithms. A is a very large time (estimated as approximately 6 months) and B is also large and is less than A.

Pattern Analysis and Machine Intelligence, 20(4):401–406, 1998.

[2] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *European Conference on Computer Vision*, pages 231–236, 1993.

[3] A. Bobick and S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[5] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with effecient approximations. In *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, page 648, 1998.

[6] Y. Boykov, O. Veksler, and R. Zabih. A new algorithm for energy minimization with discontinuities. *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, pages 205–220, July 1999.

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[8] P. Chou and C. Brown. The theory and practice of bayesian image labeling. In *International Journal of Computer Vision*, volume 4, pages 185–210, 1990.

[9] D. Geiger and F. Girosi. Mean field theory for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):401–412, 1991.

[10] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2nd edition, 2003.

[12] B. Horn and B. Schunck. Determining optic flow. *Artificial Intelligence*, 17(1-3):185–203, August 1981.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, January 2004.

[14] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 674–679, 1981.

[15] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162, 1992.

[16] D. Scharstein and R. Szeliski. Middlebury college - stereo vision page research. http://cat.middlebury.edu/stereo/.

[17] N. P. T. Brox, A. Bruhn and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, volume 3024, pages 25–36, 2004.

[18] O. Veksler. *Effecient Graph-Based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, August 1999.

[19] O. Veksler. Reducing search space for stereo correspondence with graph cuts. In *British Machine Vision Conference*, volume 2, pages 709–718, September 2006.

[20] J. Worby. Multi-resolution graph cuts for stereo-motion estimation. Master's thesis, University of Toronto, 2007.