

## Efficient Feature-based Image Registration by Mapping Sparsified Surfaces

Chun Pang Yung<sup>a,\*</sup>, Gary P. T. Choi<sup>b</sup>, Ke Chen<sup>c</sup>, Lok Ming Lui<sup>a,\*</sup>

<sup>a</sup>*Department of Mathematics, The Chinese University of Hong Kong, Hong Kong*

<sup>b</sup>*John A. Paulson School of Engineering and Applied Sciences, Harvard University, USA*

<sup>c</sup>*Department of Mathematical Sciences, The University of Liverpool, United Kingdom*

---

### Abstract

With the advancement in the digital camera technology, the use of high resolution images and videos has been widespread in the modern society. In particular, image and video frame registration is frequently applied in computer graphics and film production. However, conventional registration approaches usually require long computational time for high resolution images and video frames. This hinders the application of the registration approaches in the modern industries. In this work, we first propose a new image representation method to accelerate the registration process by triangulating the images effectively. For each high resolution image or video frame, we compute an optimal coarse triangulation which captures the important features of the image. Then, we apply a surface registration algorithm to obtain a registration map which is used to compute the registration of the high resolution image. Experimental results suggest that our overall algorithm is efficient and capable to achieve a high compression rate while the accuracy of the registration is well retained when compared with the conventional grid-based approach. Also, the computational time of the registration is significantly reduced using our triangulation-based approach.

*Key words:* Triangulated image, Image registration, Coarse triangulation, Map

---

\*Corresponding author.

*Email addresses:* [lm1ui@math.cuhk.edu.hk](mailto:lm1ui@math.cuhk.edu.hk) (Lok Ming Lui)

interpolation

2000 MSC: 68U10, 68U05

---

## 1 1. Introduction

2 In recent decades, the rapid development of the digital camera hardware has  
3 revolutionized human lives. On one hand, even mid-level mobile devices can easily  
4 produce high resolution images and videos. Besides the physical elements, the  
5 widespread use of the images and videos also reflects the importance of developing  
6 software technology for them. On the other hand, numerous registration techniques  
7 for images and video frames have been developed for a long time. The existing  
8 registration techniques work well on problems with a moderate size. However,  
9 when it comes to the current high quality images and videos, most of the current  
10 registration techniques suffer from extremely long computations. This limitation  
11 in software seriously impedes fully utilizing the state-of-the-art camera hardware.

12 One possible way to accelerate the computation of the registration is to intro-  
13 duce a much coarser grid on the images or video frames. Then, the registration  
14 can be done on the coarse grid instead of the high resolution images or video  
15 frames. Finally, the fine details can be added back to the coarse registration.  
16 It is noteworthy that the quality of the coarse grid strongly affects the quality  
17 of the final registration result. If the coarse grid cannot capture the important  
18 features of the images or video frames, the final registration result is likely to be  
19 unsatisfactory. In particular, for the conventional rectangular coarse grids, since  
20 the partitions are restricted in the vertical and horizontal directions, important  
21 features such as slant edges and irregular shapes cannot be effectively recorded.  
22 By contrast, triangulations allow more freedom in the partition directions as well  
23 as the partition sizes. Therefore, it is more desirable to make use of triangulations  
24 in simplifying the registration problems.

25 In this work, we propose a two-stage algorithm for effective registration of

26 specially large images. In stage 1, a content-aware image representation algorithm  
 27 to *TRiangulate IMages*, abbreviated as *TRIM*, is developed to simplify high  
 28 quality images and video frames. Specifically, for each high quality image or  
 29 video frame, we compute a coarse triangulation representation of it. The aim  
 30 is to create a high quality triangulation on the set of the content-aware sample  
 31 points using the Delaunay triangulation. The computation involves a series of  
 32 steps including subsampling, unsharp masking, segmentation and sparse feature  
 33 extraction for locating sample points on important features. Then in stage 2, using  
 34 coarse triangular representation of the images, the registration is computed by  
 35 a landmark-based quasi-conformal registration algorithm [17] for computing the  
 36 coarse registration. The fine detail of the image or video frame in high resolution  
 37 is computed with the aid of a mapping interpolation. Our proposed framework  
 38 may be either used as a standalone fast registration algorithm or also served as a  
 39 highly efficient and accurate initialization for other registration approaches.

40 The rest of this paper is organized as follows. In Section 2, we review the  
 41 literature on image and triangular mesh registration. Our proposed method is  
 42 explained in details in Section 3. In Section 4, we demonstrate the effectiveness of  
 43 our approach with numerous real images. The paper is concluded in Section 5.

## 44 **2. Previous works**

45 In this section, we describe the previous works closely related to our work.

46 Image registration have been widely studied by different research groups.  
 47 Surveys on the existing image registration approaches can be found in [39, 6, 16,  
 48 38]. In particular, one common approach for guaranteeing the accuracy of the  
 49 registration is to make use of landmark constraints. Bookstein [1, 2, 3] proposed  
 50 the unidirectional landmark thin-plate spline (UL-TPS) image registration. In [13],  
 51 Johnson and Christensen presented a landmark-based consistent thin-plate spline  
 52 (CL-TPS) image registration algorithm. In [14], Joshi et al. proposed the Large

53 Deformation Diffeomorphic Metric Mapping (LDDMM) for registering images  
 54 with a large deformation. In [10, 11], Glaunès et al. computed large deformation  
 55 diffeomorphisms of images with prescribed displacements of landmarks.

56 A few works on image triangulations have been reported. In [8], Gee et al.  
 57 introduced a probabilistic approach to the brain image matching problem and  
 58 described the finite element implementation. In [15], Kaufmann et al. introduced a  
 59 framework for image warping using the finite element method. The triangulations  
 60 are created using the Delaunay triangulation method [31] on a point set distributed  
 61 according to variance in saliency. In [18, 19], Lehner et al. proposed a data-  
 62 dependent triangulation scheme for image and video compression. Recently, Yun  
 63 [35] designed a triangulation image generator called DMesh based on the Delaunay  
 64 triangulation method [31].

65 In our work, we handle image registration problems with the aid of triangula-  
 66 tions. Numerous algorithms have been proposed for the registration of triangular  
 67 meshes. In particular, the landmark-driven approaches use prescribed landmark  
 68 constraints to ensure the accuracy of mesh registration. In [34, 22, 33], Wang et  
 69 al. proposed a combined energy for computing a landmark constrained optimized  
 70 conformal mapping of triangular meshes. In [23], Lui et al. used vector fields to  
 71 represent surface maps and computed landmark-based close-to-conformal map-  
 72 pings. Shi et al. [32] proposed a hyperbolic harmonic registration algorithm with  
 73 curvature-based landmark matching on triangular meshes of brains. In recent years,  
 74 quasi-conformal mappings have been widely used for feature-endowed registration  
 75 [36, 37, 24, 26]. Choi et al. [5] proposed the FLASH algorithm for landmark  
 76 aligned harmonic mappings by improving the algorithm in [34, 22] with the aid  
 77 of quasi-conformal theories. In [17], Lam and Lui reported the quasi-conformal  
 78 landmark registration (QCLR) algorithm for triangular meshes.

79 **Contributions.** Our proposed approach for fast registration of high resolution  
 80 images or video frames is advantageous in the following aspects:



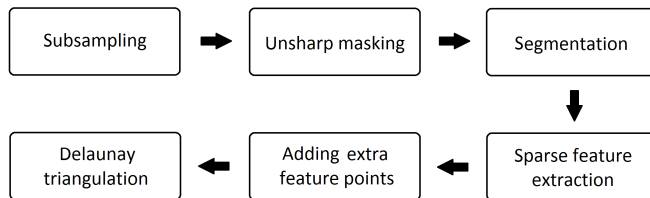


Figure 1: The pipeline of our proposed *TRIM* algorithm for accelerating image registration via coarse triangulation.

- 81 (1). The triangulation algorithm is fully automatic. The important features of the  
 82 input image are well recorded in the resulting coarse triangulation.
- 83 (2). The algorithm is fast and robust. The coarse triangulation of a typical high  
 84 resolution image can be computed within seconds.
- 85 (3). The registration algorithm for the triangulated surfaces by a Beltrami frame-  
 86 work incorporates both the edge and landmark constraints to deliver a better  
 87 quality map as fine details are restored. By contrast, for regular grid-based  
 88 approaches, the same landmark correspondences can only be achieved on the  
 89 high resolution image representation.
- 90 (4). Using our approach, the problem scale of the image and video frame registra-  
 91 tion is significantly reduced. Our method can alternatively serve as a fast and  
 92 accurate initialization for the state-of-the-art image registration algorithms.

### 93 3. Proposed method

94 In this section, we describe our proposed approach for efficient image registra-  
 95 tion in details.

#### 96 3.1. Stage 1 – Construction of coarse triangulation on images

97 Given two high resolution images  $I_1$  and  $I_2$ , our goal is to compute a fast and  
 98 accurate mapping  $f : I_1 \rightarrow I_2$ . Note that directly working on the high resolution

99 images can be inefficient. To accelerate the computation, the first step is to  
 100 construct a coarse triangular representation of the image  $I_1$ . In the following, we  
 101 propose an efficient image triangulation scheme called *TRIM*. The pipeline of our  
 102 proposed framework is described in Figure 1.

103 Our triangulation scheme is content-aware. Specifically, special objects and  
 104 edges in the images are effectively captured by a segmentation step, and a suitable  
 105 coarse triangulation is constructed with the preservation of these features. Our  
 106 proposed *TRIM* method consists of 6 steps in total.

### 107 3.1.1. Subsampling the input image without affecting the triangulation quality

108 Denote the input image by  $I$ . To save the computational time for triangulating  
 109 the input image  $I$ , one simple remedy is to reduce the problem size by performing  
 110 certain subsampling on  $I$ . For ordinary images, subsampling unavoidably creates  
 111 adverse effects on the image quality. Nevertheless, it does not affect the quality of  
 112 the coarse triangulation we aim to construct on images.

113 In our triangulation scheme, we construct triangulations based on the straight  
 114 edges and special features on the images. Note that straight edges are preserved  
 115 in all subsamplings of the images because of the linearity. **More specifically,**  
 116 **if we do subsampling on a straight line, the subsampled points remain to be**  
 117 **collinear.** Hence, our edge-based triangulation is not affected by the mentioned  
 118 adverse effects. In other words, we can subsample high resolution images to a  
 119 suitable size for enhancing the efficiency of the remaining steps for the construction  
 120 of the triangulations. We denote the subsampled image by  $\tilde{I}$ . **In practice, for**  
 121 **images larger than  $1000 \times 1000$ , we subsample the image so that it is smaller than**  
 122  **$1000 \times 1000$ .**

### 123 3.1.2. Performing unsharp masking on the subsampled image

124 After obtaining the subsampled image  $\tilde{I}$ , we perform an unsharp masking  
 125 on  $\tilde{I}$  in order to preserve the edge information in the final triangulation. More

126 specifically, we first transform the data format of the subsampled image  $\tilde{I}$  to  
 127 the CIELAB standard. Then, we apply the unsharp masking method in [27] on  
 128 the intensity channel of the CIELAB representation of  $\tilde{I}$ . The unsharp masking  
 129 procedure is briefly described as follows.

130 By an abuse of notation, we denote  $\tilde{I}(x, y)$  and  $\bar{I}(x, y)$  the intensities of the  
 131 input subsampled image  $\tilde{I}$  and the output image  $\bar{I}$  respectively, and  $G_\sigma(x, y)$  the  
 132 Gaussian mean of the intensity of the pixel  $(x, y)$  with standard derivation  $\sigma$ .  
 133 Specifically,  $G_\sigma(x, y)$  is given by

$$G_\sigma(x, y) \triangleq \frac{1}{\sigma\sqrt{2\pi}} \int_{(u,v) \in \Omega} e^{-\frac{(u-x)^2+(v-y)^2}{2\sigma^2}}. \quad (1)$$

134 We perform an unsharp masking on the image using the following formula

$$\bar{I}(x, y) = \tilde{I}(x, y) - \lambda \begin{cases} G_\sigma * \tilde{I}(x, y) & \text{if } V_s(x, y) > \theta, \\ 0 & \text{if } V_s(x, y) < \theta, \end{cases} \quad (2)$$

135 where

$$V_s(x, y) \triangleq \sqrt{\frac{1}{Area(M_s)} \int_{(u,v) \in M_s} (\tilde{I}(u, v) - \tilde{I}_{mean}(x, y))^2} \quad (3)$$

136 and

$$\tilde{I}_{mean}(x, y) = \frac{1}{Area(M_s)} \int_{(u,v) \in M_s} \tilde{I}(u, v). \quad (4)$$

137 Here, **\* is the convolution operator and**  $M_s$  is the disk with center  $(x, y)$  and  
 138 radius  $s$ . The effect of the unsharp masking is demonstrated in Figure 2. With  
 139 the aid of this step, we can highlight the edge information in the resulting image  
 140  $\bar{I}$  for the construction of the triangulation in the later steps. For simplicity we  
 141 set  $s = \sigma$ . In our experiment, we choose  $\lambda = 0.5$ ,  $\sigma = 2$ ,  $s = 2$ , and  $\theta = 0.5$ . **An**  
 142 **analysis on the choice of the parameters is provided in Section 4.**

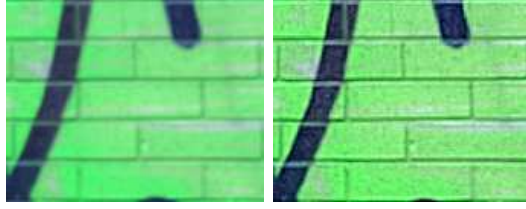


Figure 2: An illustration of unsharp masking. Left: the input image. Right: the resulting image. The unsharp masking procedure helps preserving the edge information of the input image to ensure that the vertices in unclear edges can also be extracted.

### 143 3.1.3. Segmenting the image

144 After obtaining the image  $\bar{I}$  upon unsharp masking, we perform a segmentation  
 145 in this step in order to optimally locate the mesh vertices for computing the coarse  
 146 triangulation. Mathematically, our segmentation problem is described as follows.

147 Suppose the image  $\bar{I}$  has  $L$  intensity levels in each RGB channel. Denote  $i$   
 148 as a specific intensity level (*i.e.*  $0 \leq i \leq L - 1$ ). Let  $C$  be a color channel of the  
 149 image (*i.e.*  $C \in \{R, G, B\}$ ), and let  $h_i^C$  denote the image histogram for channel  $C$ ,  
 150 in other words, the number of pixels which correspond to its  $i$ -th intensity level.

151 Define  $p_i^C \triangleq \frac{h_i^C}{N}$ , where  $N$  represents the total number of pixels in the image  $\bar{I}$ .  
 152 Then we have

$$\sum_{\substack{i=0, \\ C \in \{R, G, B\}}}^L p_i^C = 1 \quad \text{and} \quad \mu_T^C = \sum_{\substack{i=0, \\ C \in \{R, G, B\}}}^L i p_i^C. \quad (5)$$

153 Suppose that we want to compress the color space of the image  $\bar{I}$  to  $l$  intensity  
 154 levels. Equivalently,  $\bar{I}$  is to be segmented into  $l$  classes  $D_1^C, \dots, D_l^C$  by the ordered  
 155 threshold levels  $x_j^C, j = 1, \dots, l - 1$ . We define the best segmentation criterion to  
 156 be maximizing the inter-class intensity-mean variance. More explicitly, we define  
 157 the cost

$$\sigma^C \triangleq \sum_{\substack{j=1, \\ C \in \{R, G, B\}}}^l w_j^C (\mu_j^C - \mu_T^C)^2, \quad (6)$$

158 where the probability  $w_j^C$  of occurrence of a pixel being in the class  $D_j^C$  and the

159 intensity-mean  $\mu_j^C$  of the class  $D_j^C$  are respectively given by

$$w_j^C = \begin{cases} \sum_{i=0, C \in \{R,G,B\}}^{t_j^C} p_i^C & \text{if } j = 1, \\ \sum_{i=t_{j-1}^C+1, C \in \{R,G,B\}}^{t_j^C} p_i^C & \text{if } 1 < j < l, \\ \sum_{i=t_j^C+1, C \in \{R,G,B\}}^{L-1} p_i^C & \text{if } j = l, \end{cases} \quad \text{and} \quad \mu_j^C = \begin{cases} \sum_{i=0, C \in \{R,G,B\}}^{t_j^C} \frac{ip_i^C}{w_j^C} & \text{if } j = 1, \\ \sum_{i=t_{j-1}^C+1, C \in \{R,G,B\}}^{t_j^C} \frac{ip_i^C}{w_j^C} & \text{if } 1 < j < l, \\ \sum_{i=t_j^C+1, C \in \{R,G,B\}}^{L-1} \frac{ip_i^C}{w_j^C} & \text{if } j = l. \end{cases} \quad (7)$$

160 Hence, we maximize three objective functions of each RGB channel

$$\arg \max_{1 < x_1^C < \dots < x_{l-1}^C < L} \sigma^C(\{x_j^C\}_{j=1}^{l-1}), \quad (8)$$

161 where  $C \in \{R, G, B\}$ . Our goal is to find a set of  $\mathbf{x} = \{x_j^C\}_{j=1}^{l-1}$  such that above  
 162 function is maximized for each RGB channel.

163 To solve the aforementioned segmentation optimization problem, we apply the  
 164 Particle Swarm Optimization (PSO) segmentation algorithm [9] on the image  $\bar{I}$ .  
 165 The PSO method is used in this segmentation optimization problem for reducing  
 166 the chance of trapping in local optimums.

167 An illustration of the segmentation step is provided in Figure 3. After perform-  
 168 ing the segmentation, we extract the boundaries of the segments. Then, we can  
 169 obtain a number of large patches of area in each of which the intensity information  
 170 is almost the same. They provide us with a reasonable edge base for constructing  
 171 a coarse triangulation in later steps.

#### 172 3.1.4. Sparse feature extraction on the segment boundaries

173 After computing the segment boundaries  $\mathcal{B}$  on the image  $\bar{I}$ , we aim to extract  
 174 sparse feature points on  $\mathcal{B}$  in this step. For the final triangulation, it is desirable  
 175 that the edges of the triangles are as close as possible to the segment boundaries  $\mathcal{B}$ ,



Figure 3: An illustration of the segmentation step for compressing the color space to achieve a sparse intensity representation. Left: the original image. Right: the segmentation result.

176 so as to preserve the geometric features of the original image  $I$ . Also, to improve  
 177 the efficiency for the computations on the triangulation, the triangulation should  
 178 be much coarser than the original image. To achieve the mentioned goals, we  
 179 consider extracting sparse features on the segment boundaries  $\mathcal{B}$  and use them as  
 180 the vertices of the ultimate triangulated mesh.

181 Consider a rectangular grid table  $G$  on the image  $\bar{I}$ . Apparently, the grid table  
 182  $G$  intersects the segment boundaries  $\mathcal{B}$  at a number of points. Denote  $\mathcal{P}$  as our  
 183 desired set of sparse features. Conceptually,  $\mathcal{P}$  is made up of the set of points at  
 184 which  $\mathcal{B}$  intersect the grid  $G$ , with certain exceptions.

185 In order to further reduce the number of feature points for a coarse triangulation,  
 186 we propose a merging procedure for close points. Specifically, let  $g_{i,j}$  be the vertex  
 187 of the grid  $G$  at the  $i$ -th row and the  $j$ -th column. We denote  $\mathcal{P}_{i,j}^1$  and  $\mathcal{P}_{i,j}^2$   
 188 respectively as the set of points at which  $\mathcal{B}$  intersect the line segment  $\overline{g_{i,j}g_{i,j+1}}$   
 189 and the line segment  $\overline{g_{i,j}g_{i+1,j}}$ . See Figure 4 for an illustration of the parameters.

190 There are 3 possible cases for  $\mathcal{P}_{i,j}^k$ , where  $k = 1, 2$ :

- 191 (i) If  $|\mathcal{P}_{i,j}^k| = 0$ , then there is no intersection point between the line segment and  
 192  $\mathcal{B}$  and hence we can neglect it.
- 193 (ii) If  $|\mathcal{P}_{i,j}^k| = 1$ , then there is exactly one intersection point  $p_{i,j}^k$  between the line  
 194 segment and  $\mathcal{B}$ . We include this intersection point  $p_{i,j}^k$  in our desired set of  
 195 sparse features  $\mathcal{P}$ .

196 (iii) If  $|\mathcal{P}_{i,j}^k| > 1$ , then there are multiple intersection points between the line  
 197 segment and  $\mathcal{B}$ . Since these multiple intersection points lie on the same line  
 198 segment, it implies that they are sufficiently close to each other. In other  
 199 words, the information they contain about the segment boundaries  $\mathcal{B}$  is highly  
 200 similar and redundant. Therefore, we consider merging these multiple points  
 201 as one point.

202 More explicitly, for the third case, we compute the centre  $m_{i,j}^k$  of the points in  $\mathcal{P}_{i,j}^k$   
 203 by

$$m_{i,j}^k = \text{mean}_{\{p|p \in \mathcal{P}_{i,j}^k\}} p. \quad (9)$$

204 The merged point  $m_{i,j}^k$  is then considered as a desired feature point. In summary,  
 205 our desired set of sparse features is given by

$$\mathcal{P} = \bigcup_i \bigcup_j \{p_{i,j}^1, p_{i,j}^2, m_{i,j}^1, m_{i,j}^2\}. \quad (10)$$

206 An illustration of the sparse feature extraction scheme is given in Figure 4.  
 207 However, one important problem in this scheme is to determine a suitable size  
 208 of the grid  $G$  so that the sparse feature points are optimally computed. Note  
 209 that to preserve the regularity of the extracted sparse features, it is desirable  
 210 that the elements of the grid  $G$  are close to perfect squares. Also, to capture the  
 211 important features as complete as possible, the elements of  $G$  should be small  
 212 enough. Mathematically, the problem can be formulated as follows.

213 Denote  $w$  as the width of the image  $\bar{I}$ ,  $h$  as the height of the image  $\bar{I}$ ,  $w'$  as  
 214 the number of columns in  $G$ ,  $h'$  as the number of rows in  $G$ ,  $l_w$  as the horizontal  
 215 length of every element of  $G$ , and  $l_h$  as the vertical length of every element of  $G$ .  
 216 See Figure 4 for a geometric illustration of  $l_w$  and  $l_h$ . We further denote  $p$  as the  
 217 percentage of grid edges in  $G$  which intersect the segment boundaries  $\mathcal{B}$ , and  $n$  as  
 218 the desired number of the sparse feature points. Given the two inputs  $p$  and  $n$ , to

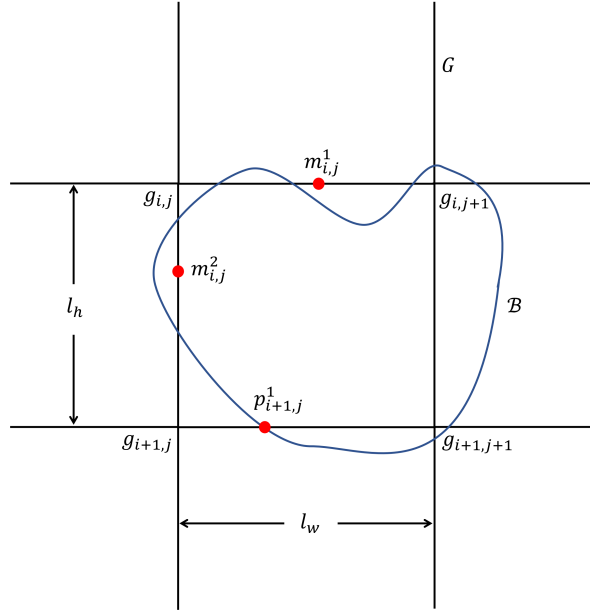


Figure 4: An illustration of our sparse feature extraction scheme. The chosen sparse feature points are represented by the red dots. If the segment boundary does not intersect an edge, no point is selected. If the segment boundary intersects an edge at exactly one point, the point is selected as a feature point. If the segment boundary intersects an edge at multiple points, the centre of the points is selected as a feature point.

219 find a suitable grid size of  $G$ , we aim to minimize the cost function

$$c(l_w, l_h) = |l_w - l_h|^2 \quad (11)$$

220 subject to

$$(i) \quad h = h' l_h, \quad (12)$$

$$(ii) \quad w = w' l_w, \quad (13)$$

$$(iii) \quad p(w' + h' + 2w'h') = n. \quad (14)$$

221 Here, the first and the second constraint respectively correspond to the hori-  
 222 zontal and vertical dimensions of the grid  $G$ , and the third constraint corresponds



223 to the total number of intersection points. To justify Equation (14), note that

$$\begin{aligned}
 & \text{Total \# of line segments} \\
 &= \text{Total \# of horizontal line segments} + \text{Total \# of vertical line segments} \quad (15) \\
 &= h'(w' + 1) + w'(h' + 1) \\
 &= w' + h' + 2w'h'.
 \end{aligned}$$

224 Note that this minimization problem is nonlinear. To simplify the computation,  
 225 we assume that  $w', h'$  are very large, that is, the grid  $G$  is sufficiently dense. Then,  
 226 from Equation (14), we have

$$\frac{p}{n} = \frac{1}{w' + h' + 2w'h'} \approx \frac{1}{2w'h'} = \frac{1}{2\left(\frac{w}{l_w}\right)\left(\frac{h}{l_h}\right)} = \frac{l_w l_h}{2wh}. \quad (16)$$

227 By further assuming that the grid  $G$  is sufficiently close to a square grid, we have  
 228  $l_w \approx l_h$ . Then, it follows that

$$\frac{p}{n} \approx \frac{l_w^2}{2wh}, \quad l_w \approx \sqrt{\frac{2pwh}{n}}. \quad (17)$$

229 Similarly,

$$l_h \approx \sqrt{\frac{2pwh}{n}}. \quad (18)$$

230 To satisfy the integral constraints for  $w'$  and  $h'$ , we make use of the above  
 231 approximations and set

$$h' = h'_0 := \left\lfloor \frac{h}{\sqrt{\frac{2pwh}{n}}} \right\rfloor = \left\lfloor \sqrt{\frac{nh}{2pw}} \right\rfloor. \quad (19)$$

232 Similarly, we set

$$w' = w'_0 := \left\lfloor \frac{w}{\sqrt{\frac{2pwh}{n}}} \right\rfloor = \left\lfloor \sqrt{\frac{nw}{2ph}} \right\rfloor. \quad (20)$$

233 Finally, we take

$$l_h = \frac{h}{h'_0} \quad \text{and} \quad l_w = \frac{w}{w'_0}. \quad (21)$$

234 To summarize, with the abovementioned strategy for the feature point extrac-  
 235 tion, we obtain a set of sparse feature points which approximates the segment  
 236 boundaries  $\mathcal{B}$ . Specifically, given the inputs  $p$  and  $n$ , the rectangular grid  $G$  we  
 237 introduce leads to approximately  $n$  regularly-extracted sparse feature points. An  
 238 illustration of the sparse feature extraction scheme is shown in Figure 5 (left). In  
 239 our experiments,  $p$  is set to be 0.2, and  $n$  is set to be 10% of the number of pixels  
 240 in the segmentation result. A denser triangulated representation can be achieved  
 241 by increasing the value of  $p$ .

### 242 3.1.5. Adding landmark points to the vertex set of the desired coarse triangulation

243 This step is only required when our *TRIM* algorithm is used for landmark-  
 244 constrained registration. For accurate landmark-constrained registration, it is  
 245 desirable to include the landmark points in the vertex set of the coarse represen-  
 246 tations of the input image  $I$ . One of the most important features of our coarse  
 247 triangulation approach is that it allows registration with exact landmark con-  
 248 straints on a coarse triangular representation. By contrast, the regular grid-based  
 249 registration can only be achieved on very dense rectangular grid domains in order  
 250 to reduce the numerical errors.

251 With the abovementioned advantage of our approach, we can freely add a set  
 252 of landmark points  $\mathcal{P}_{LM}$  to the set of sparse features  $\mathcal{P}$  extracted by the previous  
 253 procedure. In other words, the landmark points are now considered as a part of  
 254 our coarse triangulation vertices:

$$\mathcal{P} = \bigcup_i \bigcup_j \{p_{i,j}^1, p_{i,j}^2, m_{i,j}^1, m_{i,j}^2\} \cup \mathcal{P}_{LM}. \quad (22)$$

255 Then, the landmark-constrained registration of images can be computed by the

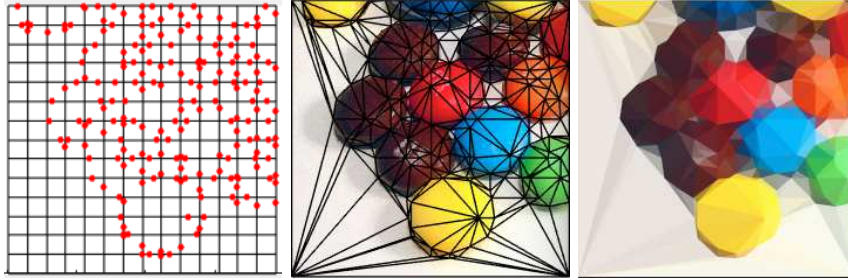


Figure 5: An illustration of computing a Delaunay triangulation on the extracted features. Left: the points obtained by the feature extraction step from Figure 3. Middle: a Delaunay triangulation on the feature points. Right: the triangulation with a color approximated on each triangle.

256 existing feature-matching techniques for triangular meshes. The existing feature  
 257 detection approaches such as [12] and [21] can be applied for obtaining the landmark  
 258 points.

### 259 3.1.6. Computing a Delaunay triangulation

260 In the final step, we construct a triangulation based on the set  $\mathcal{P}$  of feature  
 261 points. Among all triangulation schemes, the Delaunay triangulation method  
 262 is chosen since the triangles created by the Delaunay triangulations are more  
 263 regular. More specifically, if  $\alpha$  and  $\beta$  are two angles opposite to a common edge in  
 264 a Delaunay triangulation, then they must satisfy the inequality

$$\alpha + \beta \leq \pi. \quad (23)$$

265 **In other words, Delaunay triangulations always aim to minimize the formation**  
 266 **of sharp and irregular triangles.** Note that the regularity does not only enhance  
 267 the visual quality of the resulting triangulation but also lead to a more stable  
 268 approximation of the derivatives on the triangles when applying various registra-  
 269 tion schemes. Therefore, we compute a Delaunay triangulation on the set  $\mathcal{P}$  of  
 270 feature points for achieving the ultimate triangulation  $\mathcal{T}$ . An illustration of the  
 271 construction of the Delaunay triangulations is shown in Figure 5.

272 These 6 steps complete our *TRIM* algorithm as summarized in Algorithm 1.

---

**Algorithm 1:** Our proposed *TRIM* algorithm for triangulating images

---

**Input:** An image  $I$ , the desired number of image intensity levels  $l$  for segmentation, the desired number of feature points  $n$ , the sparse ratio  $p$ .

**Output:** A coarse triangulation  $\mathcal{T}$  that captures the main features of the image.

- 1 Subsample the input image  $I$  to a suitable size and denote the result by  $\tilde{I}$ ;
  - 2 Apply an unsharp masking on the subsampled image  $\tilde{I}$  and denote the result by  $\bar{I}$ ;
  - 3 Apply the PSO segmentation for compressing the color space of  $\bar{I}$  to  $l$  intensity levels, and extract boundaries  $\mathcal{B}$  of the segments;
  - 4 Extract a set of sparse feature points  $\mathcal{P}$  from the segment boundaries  $\mathcal{B}$  based on the parameters  $n$  and  $p$ ;
  - 5 Add a set of extra landmark points  $\mathcal{P}_{LM}$  to  $\mathcal{P}$  if necessary;
  - 6 Compute a Delaunay triangulation  $\mathcal{T}$  on the sparse feature points  $\mathcal{P}$ .
- 

273 It is noteworthy that our proposed *TRIM* algorithm significantly trims down  
 274 high resolution images without distorting their important geometric features.  
 275 Experimental results are shown in Section 4 to demonstrate the effectiveness of  
 276 the *TRIM* algorithm.

### 277 3.2. Stage 2 – Registration of two triangulated image surfaces

278 With the above triangulation algorithm for images, we can simplify the image  
 279 registration problem as a mapping problem of triangulated surfaces rather than  
 280 of sets of landmark points. Many conventional image registration approaches are  
 281 hindered by the long computational time and the accuracy of the initial maps.  
 282 With the new strategy, it is easy to obtain a highly efficient and reasonably accurate  
 283 registration of images. Our registration result can serve as a high quality initial  
 284 map for various algorithms.

285 To preserve angles and hence the local geometry of two surfaces, rather than  
 286 simply mapping two sets of points, conformal mappings may not exist due to  
 287 presence of landmark constraints. We turn to consider quasi-conformal mappings,  
 288 a type of mappings which is closely related to the conformal mappings. Mathe-  
 289 matically, a *quasi-conformal mapping*  $f : \mathbb{C} \rightarrow \mathbb{C}$  satisfies the Beltrami equation

290

$$\frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z} \quad (24)$$

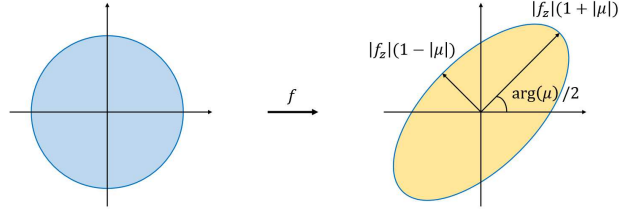


Figure 6: An illustration of quasi-conformal mappings. The maximal magnification and shrinkage are determined by the Beltrami coefficient  $\mu$  of the mappings.

291 where  $\mu$  (called the *Beltrami coefficient* of  $f$ ) is a complex-valued function with  
 292 sup norm less than 1. Intuitively, a conformal mapping maps infinitesimal circles  
 293 to infinitesimal circles, while a quasi-conformal mapping maps infinitesimal circles  
 294 to infinitesimal ellipses (see Figure 6). Readers are referred to [7] for more details.

295 In this work, we apply the quasi-conformal landmark registration (QCLR)  
 296 algorithm (designed for general surfaces in [17]) to our coarse triangulations of  
 297 images. More explicitly, to compute a registration mapping  $f : I_1 \rightarrow I_2$  between  
 298 two images  $I_1$  and  $I_2$  with prescribed point correspondences

$$p_i \longleftrightarrow q_i, i = 1, 2, \dots, n, \quad (25)$$

299 where  $\{p_i\}_{i=1}^n$  are a set of points on  $I_1$  and  $\{q_i\}_{i=1}^n$  are a set of points on  $I_2$ , we  
 300 first apply our proposed *TRIM* algorithm and obtain a coarse triangulation  $\mathcal{T}_1$   
 301 on  $I_1$ . Here, we include the feature points  $\{p_i\}_{i=1}^n$  in the generation of the coarse  
 302 triangulation, as described in the fifth step of the *TRIM* algorithm. Then, instead  
 303 of directly computing  $f$ , we can solve for a map  $\tilde{f} : \mathcal{T}_1 \rightarrow I_2$ . Since the problem  
 304 size is significantly reduced under the coarse triangulation, the computation for  $\tilde{f}$   
 305 is much more efficient than that for  $f$ .

306 The QCLR algorithm makes use of the penalty splitting method and minimizes

307

$$E_{LM}^{split}(\nu, \tilde{f}) = \int_{\mathcal{T}_1} |\nabla \nu|^2 + \alpha \int_{\mathcal{T}_1} |\nu|^2 + \rho \int_{\mathcal{T}_1} |\nu - \mu(\tilde{f})|^2 \quad (26)$$

308 subject to (i)  $\tilde{f}(p_i) = q_i$  for all  $i = 1, 2, \dots, n$  and (ii)  $\|\nu\|_\infty < 1$ . Further  
 309 alternating minimization of the energy  $E_{LM}^{split}$  over  $\nu$  and  $\tilde{f}$  is used. Specifically,  
 310 for computing  $\tilde{f}_n$  while fixing  $\nu_n$  and the landmark constraints, we apply the  
 311 linear Beltrami solver by Lui et al.[25]. For computing  $\nu_{n+1}$  while fixing  $\tilde{f}_n$ , by  
 312 considering the Euler-Lagrange equation, it suffices to solve

$$(-\Delta + 2\alpha I + 2\rho I)\nu_{n+1} = 2\rho\mu(\tilde{f}_n). \quad (27)$$

313 From  $\nu_{n+1}$ , one can compute the associated quasi-conformal mapping  $\tilde{f}_{n+1}$  and  
 314 then update  $\nu_{n+1}$  by

$$\nu_{n+1} \leftarrow \nu_{n+1} + t(\mu(\tilde{f}_{n+1}) - \nu_{n+1}) \quad (28)$$

315 for some small  $t$  to satisfy the landmark constraints (25).

316 After computing the quasi-conformal mapping  $\tilde{f}$  on the coarse triangulation,  
 317 we interpolate once to retrieve the fine details of the registration in the high  
 318 resolution. Since the triangulations created by our proposed *TRIM* algorithm  
 319 preserves the important geometric features and prominent straight lines of the  
 320 input image, the details of the registration results can be accurately interpolated.  
 321 Moreover, since the coarse triangulation largely simplifies the input image and  
 322 reduces the problem size, the computation is significantly accelerated.

323 The overall registration procedure is summarized in Algorithm 2. Experimental  
 324 results are illustrated in Section 4 to demonstrate the significance of our coarse  
 325 triangulation in the registration scheme.

#### 326 4. Experimental results

327 In this section, we demonstrate the effectiveness of our proposed triangulation  
 328 scheme. The algorithms are implemented using MATLAB. **The unsharp masking**  
 329 **step is done using MATLAB's `imsharpen`. The PSO segmentation is done using**

---

**Algorithm 2:** Feature-based registration via our proposed *TRIM* algorithm

---

**Input:** Two images or video frames  $I_1, I_2$  to be registered, with the prescribed feature correspondences.

**Output:** A feature-matching registration mapping  $f : I_1 \rightarrow I_2$ .

- 1 Compute a coarse triangulation  $\mathcal{T}_1$  of  $I_1$  using our proposed *TRIM* algorithm (Algorithm 1). Here, we include the prescribed feature points on  $I_1$  in the generation of the coarse triangulation  $\mathcal{T}_1$ ;
  - 2 Select landmark correspondences of the coarse triangulation  $\mathcal{T}_1$  and the target image  $I_2$ . Denote the landmark points on  $\mathcal{T}_1$  and  $I_2$  by  $\{p_i\}_{i=1}^n$  and  $\{q_i\}_{i=1}^n$  correspondingly;
  - 3 Compute a landmark based quasi-conformal mapping  $\tilde{f} : \mathcal{T}_1 \rightarrow \mathbb{C}$  by the QCLR algorithm in [17];
  - 4 Obtain  $f$  by  $\tilde{f}$  with a bilinear interpolation between  $\mathcal{T}_j$  and  $I_j$ .
- 



Figure 7: Several images and the triangulations by our *TRIM* algorithm. Top: the input images. Bottom: the resulting triangulations. The key features of the images are well represented in our triangulations, and the regions with similar color can be represented by coarse triangulations.

330 **the MATLAB Central function segmentation.** For solving the mentioned linear  
 331 systems, the backslash operator ( $\backslash$ ) in MATLAB is used. The test images are  
 332 courtesy of the RetargetMe dataset [28] and the Middlebury Stereo Datasets [29, 30].  
 333 The bird image is courtesy of the first author. All experiments are performed on a  
 334 PC with an Intel(R) Core(TM) i7-4500U CPU @1.80 GHz processor and 8.00 GB  
 335 RAM.

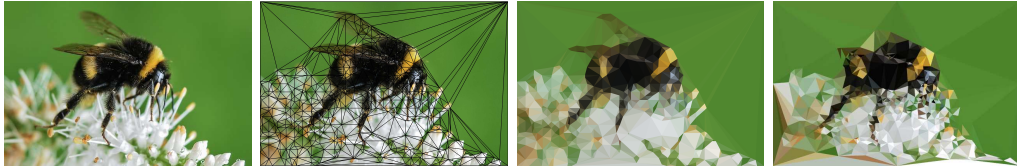


Figure 8: A bee image and the triangulations created by our *TRIM* algorithm and DMesh [35]. Left to right: The input image, the coarse triangulation created by *TRIM*, our *TRIM* coarse triangulation with a color approximated on each triangle, and the triangulation by DMesh [35].

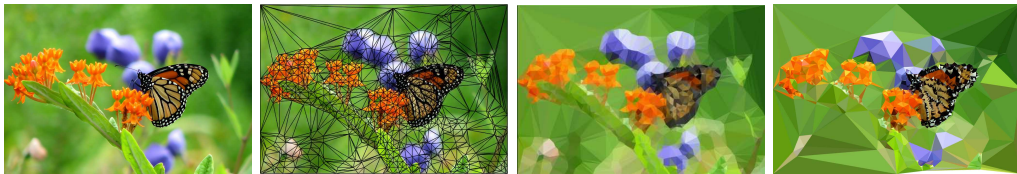


Figure 9: An butterfly image and the triangulations created by our *TRIM* algorithm and DMesh [35]. Left to right: The input image, the coarse triangulation created by *TRIM*, our *TRIM* coarse triangulation with a color approximated on each triangle, and the triangulation by DMesh [35].

#### 336 4.1. Performance of our proposed triangulation (Algorithm 1)

337 In this subsection, we demonstrate the effectiveness of our triangulation scheme  
338 by various examples.

339 Our proposed algorithm is highly content-aware. Specifically, regions with  
340 high similarities or changes in color on an image can be easily recognized. As a  
341 result, the triangulations created faithfully preserve the important features by a  
342 combination of coarse triangles with different sizes. Some of our triangulation  
343 results are illustrated in Figure 7. For better visualizations, we color the resulting  
344 triangulations by mean of the original colors of corresponding patches. In Figure 8,  
345 we apply our *TRIM* algorithm on a bee image. It can be observed that the regions of  
346 the green background can be effectively represented by coarser triangulations, while  
347 the region of the bee and flowers with apparent color differences is well detected  
348 and represented by a denser triangulation. Figure 9 shows another example of  
349 our triangulation result. The butterfly and the flowers are well represented in  
350 our triangulation result. The above examples demonstrate the effectiveness of  
351 our triangulation scheme for representing images in a simplified but accurate way.





Figure 10: Two more examples created by our *TRIM* algorithm and Dmesh [35]. Our coarse triangulations capture the important features and closely resemble the original images. Left: The input images. Middle: The triangulations by *TRIM*. Right: The triangulations by DMesh [35].

352 Some more triangulation examples created by our *TRIM* algorithm are shown  
 353 in Figure 10. Figure 11 shows some triangulation examples for noisy images. It  
 354 can be observed that our *TRIM* algorithm can effectively compute content-aware  
 355 coarse triangulations even for noisy images.

356 We have compared our algorithm with the DMesh triangulator [35] in Figure  
 357 8, Figure 9 and Figure 10. It can be observed that our triangulation scheme  
 358 outperforms DMesh [35] in terms of the triangulation quality. Our results can  
 359 better capture the important features of the images. Also, the results by DMesh  
 360 [35] may contain unwanted holes while our triangulation results are always perfect  
 361 rectangles. The comparisons reflect the advantage of our coarse triangulation  
 362 scheme.

363 To quantitatively compare the content-aware property of our *TRIM* method  
 364 and the DMesh method, we calculate the average absolute intensity difference  
 365  $\frac{1}{N} \|I_{\text{triangulated}} - I_{\text{original}}\|_1$  between the original image  $I_{\text{original}}$  (e.g. the left images  
 366 in Figure 10) and the triangulated image  $I_{\text{triangulated}}$  with piecewise constant color  
 367 for each method (e.g. the middle and the right images in Figure 10), where  $N$  is  
 368 the number of pixels of the image. Table 1 lists the statistics. It is noteworthy



Figure 11: Two triangulation examples by our *TRIM* algorithm for noisy images. Left to right: The noise-free images, the triangulations computed by *TRIM* based on the noise-free images, the noisy images, and The triangulations computed by *TRIM* based on the noisy images. Note that the important features of the images are preserved even for noisy images.

369 that the average absolute intensity difference by *TRIM* is smaller than that by  
 370 DMesh by around 30% on average. This indicates that our *TRIM* algorithm is  
 371 more capable to produce content-aware triangulations.

Image	Size	Average intensity difference ( <i>TRIM</i> )	Average intensity difference (DMesh)
Bee	640 × 425	0.1455	0.2115
Bird	1224 × 1224	0.1842	0.2074
Butterfly	1024 × 700	0.1629	0.2647
Book	601 × 809	0.1446	0.2130
Baseball	410 × 399	0.1913	0.3554
Teddy	450 × 375	0.1505	0.2998
Pencil	615 × 410	0.2610	0.4443
Eagle	600 × 402	0.1618	0.1897

Table 1: The content-aware property of our *TRIM* algorithm and the DMesh method.

372 Then, we evaluate the efficiency of our triangulation scheme for various images.  
 373 Table 2 shows the detailed statistics. The relationship between the target coarse  
 374 triangulation size and the computational time is illustrated in Figure 12. Even for  
 375 high resolution images, the computational time for the triangulation is only around  
 376 10 seconds. It is noteworthy that our *TRIM* algorithm significantly compresses  
 377 the high resolution images as coarse triangulations with only several thousand

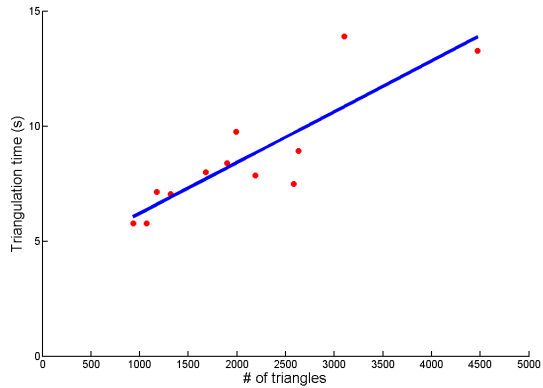


Figure 12: The relationship of the desired coarse triangulation size and the computational time of our proposed *TRIM* algorithm.

378 triangles.

Image	Size	Triangulation time (s)	# of triangles	Compression rate
Surfer	846 × 421	5.78	1043	0.1536%
Helicopter	720 × 405	5.78	1129	0.1989%
Bee	640 × 425	7.13	1075	0.2029%
Bird	1224 × 1224	7.04	1287	0.0859%
Butterfly	1024 × 700	8.00	1720	0.1232%
Book	601 × 809	8.38	1629	0.3350%
Baseball	410 × 399	7.85	2315	0.7201%
Teddy	450 × 375	7.48	2873	0.8652%
Pencil	615 × 410	8.93	2633	0.5838%
Tiger	2560 × 1600	13.91	3105	0.0414%
Eagle	600 × 402	13.27	1952	0.4299%

Table 2: Performance of our *TRIM* algorithm. The compression rate is  $\frac{\# \text{ of triangle nodes}}{\# \text{ of pixels}} \times 100\%$ .

379 It is noteworthy that the combination of the steps in our *TRIM* algorithm is  
 380 important for achieving a coarse triangulation. More specifically, if certain steps  
 381 in our algorithm are removed, the triangulation result will become unsatisfactory.  
 382 Figure 13 shows two examples of triangulations created by our entire *TRIM*  
 383 algorithm and by our algorithm with the segmentation step excluded. It can be  
 384 easily observed that without the segmentation step, the resulting triangulations  
 385 are extremely dense and hence undesirable for simplifying further computations.  
 386 By contrast, the number of triangles produced by our entire *TRIM* algorithm is

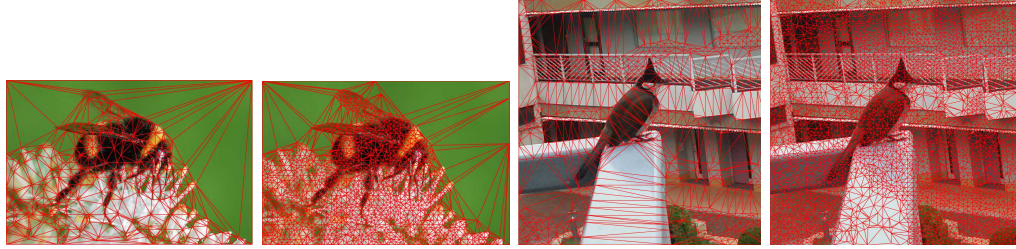


Figure 13: The triangulations created by our *TRIM* algorithm (left) and by the algorithm without the segmentation step (Right). The results show that the segmentation step is crucial for achieving a coarse triangulation. Number of triangles produced (left to right): 923, 3612, 1496, 8685.

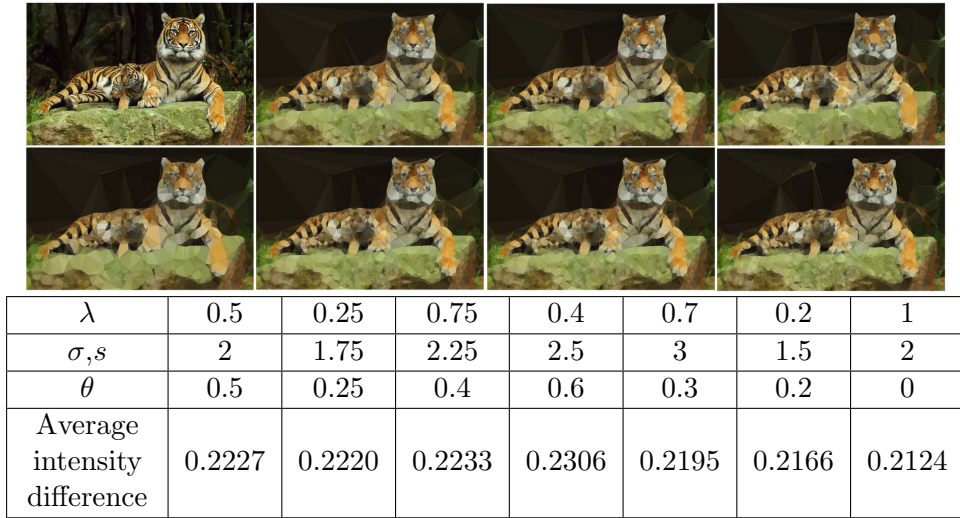


Figure 14: The triangulation results with different parameters ( $\lambda, \sigma, s, \theta$ ) in the unsharp masking step. (Top left: The original image. Top middle left to bottom right: results with different parameters.)

387 significantly reduced. The examples highlight the importance of our proposed  
 388 combination of steps in the *TRIM* algorithm for content-aware coarse triangulation.

389 We also analyze the sensitivity of the triangulated images to the parameters  
 390 in the unsharp masking step. Figure 14 shows several triangulation results with  
 391 different choice of  $(\lambda, \sigma, s, \theta)$ . It can be observed that the triangulation results are  
 392 robust to the parameters.



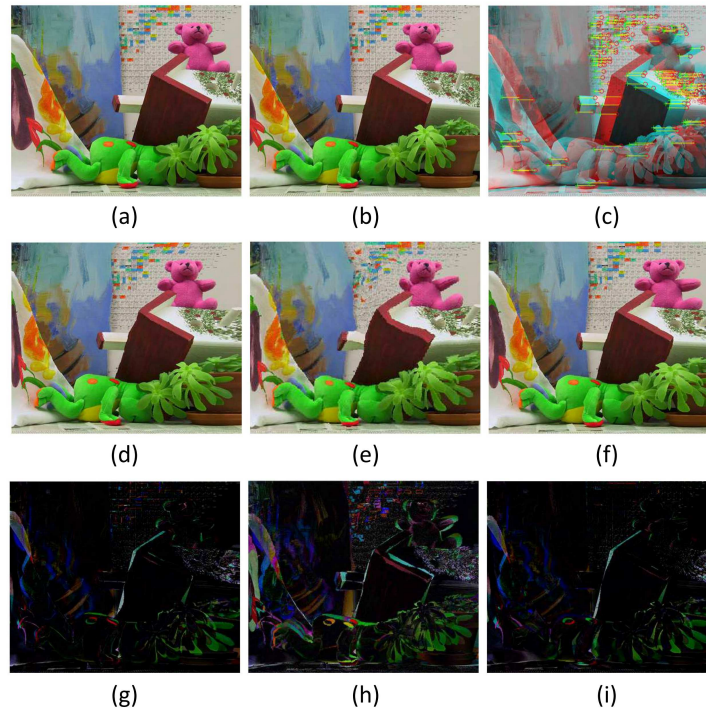


Figure 15: Stereo landmark registration of two images of doll using different algorithms. (a): The source image. (b): The target image. (c): The prescribed feature correspondences. (d): The registration result by the dense grid-based approach (4 pixels per grid). (e): The registration result via DMesh [35]. (f): The registration result by our *TRIM*-based method. (g): The intensity difference after the registration by the dense grid-based approach. (h): The intensity difference after the registration via DMesh [35]. (i): The intensity difference after the registration by our *TRIM*-based method.

#### 393 4.2. Registration of two triangulated image surfaces (Algorithm 2)

394 In this subsection, we demonstrate the effectiveness of our proposed triangulation-  
 395 based method for landmark-based image registration. In our experiments, the  
 396 feature points on the images are extracted using the Harris–Stephens algorithm [12]  
 397 as landmark constraints. The landmark extraction is fully automatic. More specif-  
 398 ically, we use the MATLAB functions `detectHarrisFeatures`, `extractFeatures`  
 399 and `matchFeatures` on the images. For the teddy example, 132 landmark pairs  
 400 are generated using the above procedure. For the cones example, 162 landmark  
 401 pairs are generated.

402 For simplifying the image registration problems, one conventional approach

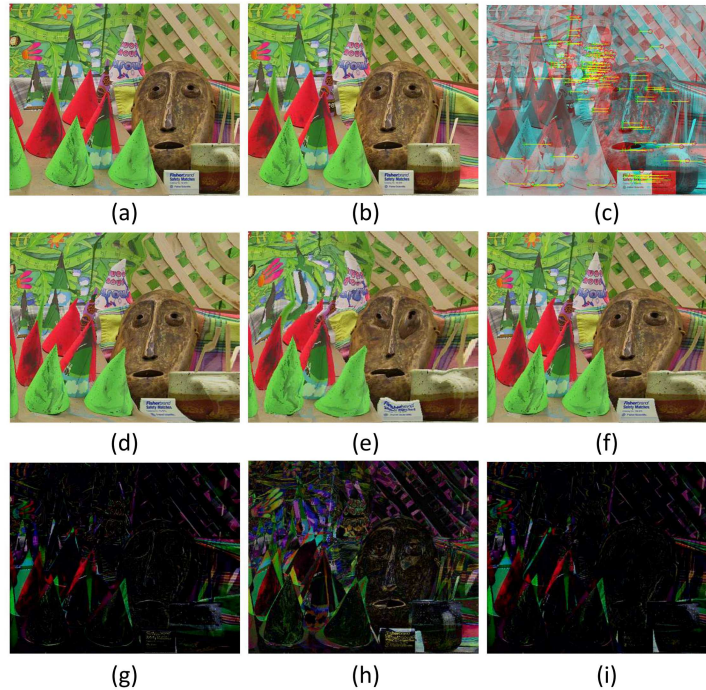


Figure 16: Stereo landmark registration of two images of cones using different algorithms. Please refer to Figure 15 for the description of the images.

403 is to make use of coarse regular grids followed by interpolation. It is natural  
 404 to ask whether our proposed coarse triangulation-based method produces better  
 405 results. In Figure 15, we consider a stereo registration problem of two scenes. With  
 406 the prescribed feature correspondences, we compute the feature-endowed stereo  
 407 registration via the conventional grid-based approach, the DMesh triangulation  
 408 approach [35] and our proposed *TRIM* method. For the grid-based approach  
 409 and the DMesh triangulation approach [35], we take the mesh vertices nearest to  
 410 the prescribed feature points on the source image as source landmarks. For our  
 411 proposed *TRIM* method, as the landmark vertices are automatically embedded  
 412 in the content-aware coarse triangulation, the source landmarks are exactly the  
 413 feature points detected by the method in [12].

414 It can be observed that our triangulation-based approach produces a much more  
 415 natural and accurate registration result when compared with both the grid-based

416 approach and the DMesh triangulation approach. In particular, sharp features  
 417 such as edges are well preserved using our proposed method. By contrast, the  
 418 edges are seriously distorted in the other two methods. In addition, the geometry  
 419 of the background in the scenes are well retained via our *TRIM* method but not  
 420 the other two methods. The higher accuracy of the registration result by our  
 421 approach can also be visualized by the intensity difference plots. Our triangulation-  
 422 based approach results in an intensity difference plot with more dark regions  
 423 than the other two approaches. The advantage of our method over the other two  
 424 methods is attributed to the geometry preserving feature of our *TRIM* algorithm,  
 425 in the sense that the triangulations created by *TRIM* are more able to fit into  
 426 complex features and have more flexibilities in size than regular grids. Also, the  
 427 triangulations created by DMesh [35] do not capture the geometric features and  
 428 hence the registration results are unsatisfactory. They reflect the significance of  
 429 our content-aware *TRIM* triangulation scheme in computing image registration.  
 430 Another example is illustrated in Figure 16. Again, it can be easily observed that  
 431 our proposed *TRIM* triangulation approach leads to a more accurate registration  
 432 result.

433 To highlight the improvement in the efficiency by our proposed *TRIM* algorithm,  
 434 Table 3 records the computational time and the error of the registration via the  
 435 conventional grid-based approach and our *TRIM* triangulation-based approach. It  
 436 is noteworthy that our proposed coarse triangulation-based method significantly  
 437 reduces the computational time by over 85% on average when compared with the  
 438 traditional regular grid-based approach. To quantitatively assess the quality of  
 439 the registration results, we define the matching accuracy by

$$A = \frac{\# \text{ pixels for which } \|\text{final intensity} - \text{original intensity}\|_1 \text{ is less than } \epsilon}{\text{Total \# of pixels}} \times 100\%. \quad (29)$$

440 The threshold  $\epsilon$  is set to be 0.2 in our experiments. Our triangulation-based

441 method produces registration results with the matching accuracy higher than that  
 442 of the regular grid-based method by 6% on average. The experimental results  
 443 reflect the advantages of our *TRIM* content-aware coarse triangulations for image  
 444 registration.

Images	Size	Registration				Time saving rate
		Via regular grids		Via <i>TRIM</i>		
		Time (s)	Matching accuracy (%)	Time (s)	Matching accuracy (%)	
Teddy	450 × 375	102.3	59.5	13.8	70.7	86.5103%
Cones	450 × 375	108.7	51.3	28.2	61.2	74.0570%
Cloth	1252 × 1110	931.0	70.7	36.0	75.4	96.1332%
Books	1390 × 1110	1204.5	59.0	51.0	63.0	95.7659%
Dolls	1390 × 1110	94.3	62.3	11.0	62.3	88.3351%

Table 3: The performance of feature-based image registration via our proposed *TRIM* coarse triangulation method and the ordinary coarse grids. Here, the time saving rate is defined by  $\frac{\text{Registration time via regular grids} - \text{Registration time via } TRIM}{\text{Registration time via regular grids}} \times 100\%$ .

445 We further compare our *TRIM*-based registration method with two other state-  
 446 of-the-art image registration methods, namely the Large Displacement Optical  
 447 Flow (LDOF) [4] and the Diffeomorphic Log-Demons [20]. Table 4 lists the  
 448 performance of the methods. It is noteworthy that our method is significantly  
 449 faster than the two other methods, with at least comparable and sometimes better  
 450 matching accuracy.

Images	Size	<i>TRIM</i>		LDOF		Spectral Log-Demons	
		Time (s)	Matching accuracy (%)	Time (s)	Matching accuracy (%)	Time (s)	Matching accuracy (%)
Aloe	222 × 257	2.8	91.1	12.1	86.4	18.2	94.6
Computer	444 × 532	4.0	69.9	51.4	70.0	7.9	41.3
Laundry	444 × 537	4.1	73.2	52.4	75.7	9.02	50.8
Dwarves	777 × 973	7.9	80.5	311.8	82.5	36.7	50.8
Art	1390 × 1110	13.5	84.8	1110.6	87.4	242.9	77.9
Bowling2	1110 × 1330	20.9	90.1	1581.9	86.1	22.0	57.6

Table 4: Comparison between our *TRIM*-based image registration, the Large Displacement Optical Flow (LDOF) [4] and the Spectral Log-Demons [20].

451 Besides, we study the stability of the *TRIM*-based registration result with  
 452 respect to the feature points detected. Figure 17 shows the results with different



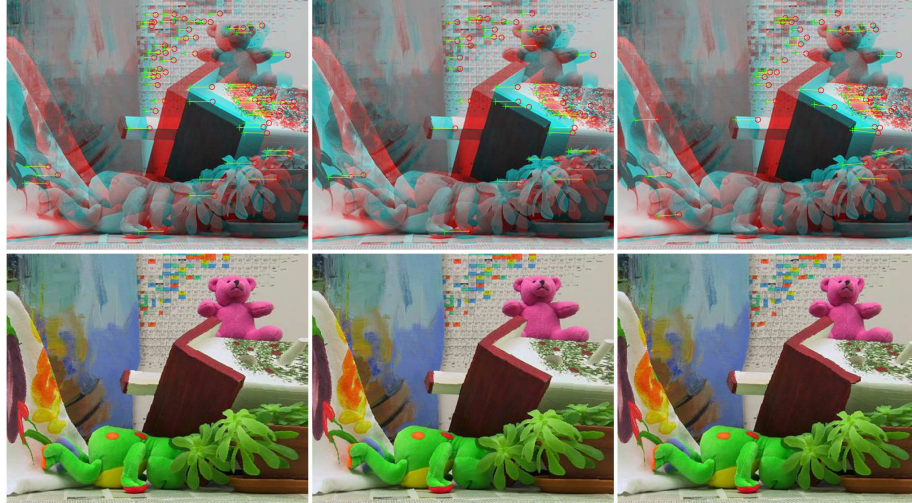


Figure 17: Different landmark correspondences and the *TRIM*-based registration results for the teddy example in Figure 15. Left: Only  $1/2$  of the landmark pairs in Figure 15 are randomly selected for computing the registration. Middle: Only  $1/3$  of the landmark pairs are used. Right: Only  $1/3$  of the landmark pairs are used, with 2% random noise added to the target landmark locations. The matching accuracies are respectively 70.5%, 70.4%, 67.2%, which are very close to the original result (70.7%).

453 *feature correspondences, including a change in the number of landmark pairs and a*  
 454 *change in the target landmark position. From the resulting triangulated images and*  
 455 *the statistics on the matching accuracy, it can be observed that the deformation is*  
 456 *stable with respect to the choice of the feature points.*

## 457 5. Conclusion and future work

458 In this paper, we have proposed a new image registration algorithm (Algorithm  
 459 2), which operates on content-aware coarse triangulations to aid registration of  
 460 high resolution images. The obtained algorithm is computationally efficient and  
 461 capable to achieve a highly accurate result while resembling the original image. It  
 462 has two stages with stage 1 obtaining content-aware coarse triangulations and stage  
 463 2 registering two triangulated surfaces. Both algorithms can be used as standalone  
 464 methods: Algorithm 1 for extracting main features of images (compression) and  
 465 Algorithm 2 for registering two surfaces (quality mapping).

466 Our proposed method is advantageous for a large variety of registration ap-  
 467 plications with a significant improvement of the computational efficiency and  
 468 registration accuracy. Our proposed method can also serve as an effective ini-  
 469 tialization for other registration algorithms. In the future, we aim to extend our  
 470 proposed algorithm to high dimensions.

## 471 References

- 472 [1] F. L. Bookstein, *The measurement of biological shape and shape change*. Springer-Verlag: Lecture  
 473 Notes in Biomathematics 24 (1978) 1–191.
- 474 [2] F. L. Bookstein, *Morphometric tools for landmark data*. Cambridge University Press, Cambridge  
 475 (1991).
- 476 [3] F. L. Bookstein, *Linear methods for nonlinear maps: Procrustes fits, thin-plate splines, and the*  
 477 *biometric analysis of shape variability*. Brain Warping, Academic Press, London (1999) 157–181.
- 478 [4] T. Brox and J. Malik, *Large displacement optical flow: descriptor matching in variational motion*  
 479 *estimation*. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(3) (2011) 500–513.
- 480 [5] P. T. Choi, K. C. Lam, and L. M. Lui, *FLASH: Fast landmark aligned spherical harmonic parame-*  
 481 *terization for genus-0 closed brain surfaces*. *SIAM J. Imaging Sci.* 8 (1)(2015) 67–94.
- 482 [6] W. R. Crum, T. Hartkens, and D. L. G. Hill, *Non-rigid image registration: theory and practice*. *Br.*  
 483 *J. Radiol.* 77 (2004) 140–153.
- 484 [7] F. Gardiner and N. Lakic, *Quasiconformal Teichmüller theory*. Mathematical Surveys and Mono-  
 485 graphs 76 (2000) American Mathematics Society.
- 486 [8] J. C. Gee, D. R. Haynor, M. Reivich, and R. Bajcsy, *Finite element approach to warping of brain*  
 487 *images*. *Proceedings of SPIE* (1994) 327–337.
- 488 [9] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, *An efficient method for*  
 489 *segmentation of images based on fractional calculus and natural selection*. *Expert. Syst. Appl.* 39  
 490 (16)(2012) 12407–12417.
- 491 [10] J. Glaunès, M. Vaillant, and M. I. Miller, *Landmark matching via large deformation diffeomorphisms*  
 492 *on the sphere*. *J. Math. Imaging Vis.* 20 (1)(2004) 179–200.
- 493 [11] J. Glaunès, A. Qiu, M. I. Miller, and L. Younes, *Large deformation diffeomorphic metric curve*  
 494 *mapping*. *Int. J. Comput. Vis.* 80 (3)(2008) 317–336.

- 495 [12] C. Harris and M. Stephens, *A combined corner and edge detector*. Proceedings of the 4th Alvey  
496 Vision Conference (1988) 147–151.
- 497 [13] H. J. Johnson, G. E. Christensen, *Consistent landmark and intensity-based image registration*. IEEE  
498 Trans. Med. Imag. 21 (5)(2002) 450–461.
- 499 [14] S. C. Joshi and M. I. Miller, *Landmark matching via large deformation diffeomorphisms*. IEEE  
500 Trans. Image Process. 9 (8)(2010) 1357–1370.
- 501 [15] P. Kaufmann, O. Wang, A. Sorkine-Hornung, O. Sorkine-Hornung, A. Smolic, and M. Gross, *Finite  
502 Element Image Warping*. Computer Graphics Forum, 32 (2pt1)(2013) 31–39.
- 503 [16] A. Klein, J. Andersson, B.A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen,  
504 D. L. Collins, J. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. Thompson, T.  
505 Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey, *Evaluation of 14 nonlinear deformation  
506 algorithms applied to human brain MRI registration*. NeuroImage 46 (3)(2009) 786–802.
- 507 [17] K. C. Lam and L. M. Lui, *Landmark and intensity based registration with large deformations via  
508 quasi-conformal maps*. SIAM J. Imaging Sci. 7 (4)(2014) 2364–2392.
- 509 [18] B. Lehner, G. Umlauf, B. Hamann, *Image compression using data-dependent triangulations*. Inter-  
510 national Symposium on Visualization and Computer Graphics (2007) 351–362.
- 511 [19] B. Lehner, G. Umlauf, B. Hamann, *Video compression using data-dependent triangulations*. Com-  
512 puter Graphics and Visualization (2008) 244–248.
- 513 [20] H. Lombaert, L. Grady, X. Pennec, N. Ayache, and F. Chriet, *Spectral log-demons: Diffeomorphic  
514 image registration with very large deformations*. Int. J. Comput. Vis. 107(3) (2014) 254–271.
- 515 [21] D. Lowe, *Distinctive image features from scale-invariant keypoints*. Int. J. Comput. Vis. 60 (2)(2004)  
516 91–110.
- 517 [22] L. M. Lui, Y. Wang, T. F. Chan, and P. M. Thompson, *Landmark constrained genus zero surface  
518 conformal mapping and its application to brain mapping research*. Appl. Numer. Math. 57 (5–7)(2007)  
519 847–858.
- 520 [23] L. M. Lui, S. Thiruvankadam, Y. Wang, P. M. Thompson, and T. F. Chan, *Optimized conformal  
521 surface registration with shape-based landmark matching*. SIAM J. Imaging Sci. 3 (1)(2010) 52–78.
- 522 [24] L. M. Lui, K. C. Lam, S. T. Yau, and X. Gu, *Teichmüller mapping (T-map) and its applications to  
523 landmark matching registration*. SIAM J. Imaging Sci. 7 (1)(2014) 391–426.
- 524 [25] L. M. Lui, K. C. Lam, T. W. Wong, and X. Gu, *Texture map and video compression using Beltrami  
525 representation*. SIAM J. Imaging Sci. 6 (4)(2013) 1880–1902.

- 526 [26] T. W. Meng, G. P.-T. Choi, and L. M. Lui, *TEMPO: Feature-endowed Teichmüller extremal map-*  
527 *plings of point clouds*. SIAM J. Imaging Sci. 9 (4)(2016) 1922–1962.
- 528 [27] A. Polesel, G. Ramponi, and V. J. Mathews, *Image enhancement via adaptive unsharp masking*.  
529 IEEE Trans. Image Process. 9 (3)(2000) 505–510.
- 530 [28] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, *A comparative study of image retargeting*.  
531 ACM Trans. Graph. (SIGGRAPH Asia 2010) 29 (5)(2010) 160:1–160:10.
- 532 [29] D. Scharstein and R. Szeliski, *High-accuracy stereo depth maps using structured light*. IEEE Com-  
533 puter Society Conference on Computer Vision and Pattern Recognition (CVPR) 1 (2003) 195–202.
- 534 [30] D. Scharstein and C. Pal, *Learning conditional random fields for stereo*. IEEE Computer Society  
535 Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 1–8.
- 536 [31] J. R. Shewchuk, *Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator*.  
537 Applied Computational Geometry: Towards Geometric Engineering 1148 (1996) 203–222.
- 538 [32] R. Shi, W. Zeng, Z. Su, H. Damasio, Z. Lu, Y. Wang, S. T. Yau, and X. Gu, *Hyperbolic harmonic*  
539 *mapping for constrained brain surface registration*. IEEE Conference on Computer Vision and Pattern  
540 Recognition (CVPR) (2013) 2531–2538.
- 541 [33] Y. Wang, L. M. Lui, X. Gu, K. M. Hayashi, T. F. Chan, A. W. Toga, P. M. Thompson, and S. T.  
542 Yau, *Brain surface conformal parameterization using Riemann surface structure*. IEEE Trans. Med.  
543 Imag. 26 (6) (2007) 853–865.
- 544 [34] Y. Wang, L. M. Lui, T. F. Chan, and P. M. Thompson, *Optimization of brain conformal mapping*  
545 *with landmarks*. Med. Image Comput. Comput. Assist. Interv. (MICCAI) II (2005) 675–683.
- 546 [35] D. Y. H. Yun, *DMesh triangulation image generator*. (2013) <http://dmesh.thedof.com/>
- 547 [36] W. Zeng and X. D. Gu, *Registration for 3D surfaces with large deformations using quasi-conformal*  
548 *curvature flow*. IEEE Conference on Computer and Pattern Recognition (CVPR) (2011) 2457–2464.
- 549 [37] W. Zeng, L. M. Lui, and X. Gu, *Surface registration by optimization in constrained diffeomorphism*  
550 *space*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 4169–4176.
- 551 [38] J. Zhang, K. Chen, and B. Yu, *An efficient numerical method for mean curvature-based image*  
552 *registration model*. East Asian Journal on Applied Mathematics 7(01) (2017) 125–142.
- 553 [39] B. Zitova and J. Flusser, *Image registration methods: A survey*. Image. Vision Comput. 21 (11)  
554 (2003) 977–1000.