COMPARATIVE STUDY OF STEREO ALGORITHMS FOR 3D FACE RECONSTRUCTION

Mark Chan, Patrice Delmas, Georgy Gimel'farb, Philippe Leclercq, Jiang Liu and Alexander Woodward

p.delmas@auckland.ac.nz
The University of Auckland, Dept. of Computer Science
Private Bag 92019 Auckland, New Zealand

ABSTRACT
This paper proposes to compare different sets of stereo algorithms. To generate 3D faces from a pair of images, first, algorithms theory is presented. The acquisition system setup and the creation of a face disparity map test image are explicated. The performances of the stereo-algorithms are then compared. Conclusion follows. Several stereo algorithms are compared by accuracy of 3D reconstruction of human face from a stereo pair of images. The accuracy is measured in terms of deviations of reconstructed surfaces from a prototype found by visual matching of corresponding nodes of a dense grid projected onto the faces. It is shown that by combining the most efficient but slow maximum-cut technique with fast dynamic programming, more accurate reconstruction results can be obtained.

1. INTRODUCTION

2. STEREO ALGORITHM DESCRIPTION

3. TECHNIQUES FOR STEREO RECONSTRUCTION

Because face reconstruction has to be performed in real time but closely approximate actual facial details, we compare stereo techniques that have either higher accuracy or faster performance in known experiments.

3.1. Minimum-cut stereo (MCS)

This approach performs a statistically optimal estimation of a disparity map for a 3D human face from a given stereo pair of images. The estimation is based on general local 2D coherence constraints [1], instead of the previous traditional 1D ordering ones. To perform the 2D optimization, the matching is formulated as a maximum-flow graph problem. The images and maps are described with a simple Markov random field (MRF) model taking into account the x-disparity of and the difference between grey values in each pair of corresponding points. Let R be the a planar lattice supporting both a disparity map and stereo images reduced to the lattice in accord to the map.

Each point i ∈ R is characterized by x-disparity di of and difference δi = δleft,i; δright,i between signals gleft,i and corresponding points in the left and right stereo images. The disparity map D = {di : i ∈ R} has to conform to visibility and smoothness restrictions given in terms of admissible disparity increments in the nearest neighbourhood N of each point. The difference image ∆ = {δi : i ∈ R} relates to the simplest matching criterion of almost equal corresponding signals in both stereo images. Pairwise 2D restrictions on neighbouring disparities and signal differences are specified by a joint Gibbs probability distribution [2]:

Pr(M, ∆) ∝ exp (−(Eres(M) + Esim(∆, M))) (1)

where Eres(M) and Esim(∆, M) denote Gibbs energies of pairwise interactions between the disparities and corresponding signals, respectively:

Eres(M) = ∑ i∈R ∑ a∈N Vres(di, di+a)
Esim(∆, M) = ∑ i∈R Vsim(di, δi)

and Vres(·, ·) and Vsim(·, ·) are Gibbs potentials that probabilistically weigh violations of map restrictions and signal deviations. The goal is to find a piecewise smooth x-disparity map that is consistent with the desired accuracy and minimises the total energy of Eq. (1). Epipolar surface profiles are stacked together to form a 3D matching cube. The disparity map of a desired surface is obtained by approximately solving the maximum-flow/minimum-cut problem on a graph linking discrete 3D points in the cube [2]. Comparisons of many stereo correspondence algorithms on a small database of four stereo pairs with the known ground truth [3] show that the minimum-cut energy minimisation is so far one of the best stereo algorithms. Let D and N = |R| be the total range of x-disparities and the total number of the lattice points, respectively. The worst case running time is O(N^2 D^2 log(N D)) but experiments in [2] suggested that the average running time is only about O(N^1.2 D^1.3)).

This algorithm meets with difficulties in selecting proper potential values for partially occluded points, matching
uniform image areas, and accounting for non-uniform photometric distortions of stereo images (in this latter case, minimum differences of the corresponding signals do not indicate accurate image matching).

3.2. Symmetric dynamic programming stereo (SDPS)

This more traditional algorithm sequentially estimates a collection of continuous 1D epipolar profiles of a 2D disparity map of a surface by maximising the likelihood of each individual profile. Regularisation with respect to partial occlusions along the continuous profile is based on the Markov chain models of epipolar profiles and image signals along each profile [4]. The algorithm accounts for non-uniform photometric distortions of stereo images by mutually adapting the corresponding signals. The adaptation is based on sequential estimation of the signals for each binocularly visible point along the profile variant by averaging the corresponding signals in the stereo images. The estimated signals are then adapted to each image by changing the corresponding increments to within an allowable range.

Comparing to the MCS, the matching cube is analysed in a row-by-row mode without preserving inter-row restrictions. This simplification results in non-smooth surfaces having arbitrary disparity increments between the adjacent profiles. But the surface itself is obtained much faster, because the total running time is \( O(ND) \). Additional drawbacks of the SDPS relate to difficulties of choosing a proper Markov model of partially occluded profile points. Figure 1 exemplifies typical face reconstruction results with the MCS and SDPS. The most accurate face surface is obtained after refining the MCS output with the SDPS.

3.3. Medioni

The Medioni et al algorithm [5] takes a volumetric approach to calculating a disparity map of a 3D surface. A correct disparity map corresponds to a maximal surface in the u-v-d volume, where \( u \) and \( v \) are pixel co-ordinates, \( d \) is the disparity. The disparity surface is extracted by first locating seed points in the volume (correspondences with a high likelihood of being correct matches), and propagating outward from these positions, thus tracing out a surface in the volume. An upper threshold is used to discriminate these seed points. Propagation utilizes the disparity gradient limit \([X2]\) which states that the rate of change in disparity is less than the rate of change in pixel position. Therefore only directly adjacent neighbours along a scan line are analyzed. A lower threshold is be used to reject poor matches at the propagation stage. Normalized cross-correlation is used as the similarity measure. The initial step for the algorithm requires the location of seed points. A multi-resolution approach is utilized where a mipmap of the original stereo pair is created. At the lowest resolution, the left image is divided into buckets and pixels inside are randomly selected for seed points suitability. Next, as a result of surface tracing, a low resolution disparity map is obtained. The disparities from this map are inherited as seed points for each higher resolution, until the full scale disparity map is constructed. As a consequence this multi-resolution approach aims to globally constrain the resulting disparities to the same surface within the u-v-d volume. By propagating from initial seed points a level of local coherence is obtained. The propagation algorithm is as follows: the propagation pseudo code of the algorithm presented in the paper goes here

3.3.1. Correlation-Based Algorithms

Three standard correlation based variants are presented: two from Faugeras et al. [6] and a third one which was simplified so that it was capable of an efficient (small and fast) hardware implementation: the sum of absolute differences (SAD).

Note that the Census algorithm has been considered with the correlation algorithms because computation of the Hamming distance on the computed Census transform vectors is similar to a correlation function.

3.3.2. Notation

\( I(P) \) is the intensity at point, \( P \); \( I_L(P) \) is an intensity in the left image and \( I_R(P) \) in the right one. Correlation functions are evaluated over a ‘window’ of neighboring pixels in each image. A window, \( w(P, r) \), is defined by its centre, \( P \), and its radius, \( r \). A radius, \( r \) implies a square window of \( (2 \cdot r + 1) \times (2 \cdot r + 1) \) pixels.

Figure 2 illustrates the correlation process: for a point \( P \) on the reference image (left for instance), all correlations with a window in the right image for the whole disparity range are computed and the best value is chosen.
The first correlation cost, \( C(P) \), is the normalised intensity difference:

\[
C(P) = \frac{\sum_{P' \in w(P,j)} (I_L(P') - I_R(P'))^2}{\sum_{P' \in w(P,j)} I_L(P')^2 \cdot \sum_{P' \in w(P,j)} I_R(P')^2}
\]  

(2)

3.3.3 Corr1: Normalised Square of Differences Correlation (C1 in Faugeras et al. [6])

The second cost is a normalised multiplicative correlation function:

\[
C(P) = \frac{\sum_{P' \in w(P,j)} I_L(P') \cdot I_R(P')}{\sqrt{\sum_{P' \in w(P,j)} I_L(P')^2} \cdot \sqrt{\sum_{P' \in w(P,j)} I_R(P')^2}}
\]  

(3)

3.3.4 Corr2: Normalised Multiplicative Correlation (C2 in Faugeras et al. [6])

The third cost simply sums absolute differences without normalisation. This is naturally much faster and has a simple hardware implementation:

\[
C(P) = \sum_{P' \in w(P,j)} |I_R(P') - I_L(P')|
\]  

(4)

A variant of the SAD cost is the Sum of Squared Differences (SSD):

\[
C(P) = \sum_{P' \in w(P,j)} (I_R(P') - I_L(P'))^2
\]

For hardware implementations, SAD is easier to implement as it only needs a substraction, a comparison, and a possible sign change; SSD needs a substraction and a multiplication. In hardware, a parallel-array multiplier requires \( O(n^2) \) space \(^1\) and a propagation delay approximately twice that of a substraction which requires \( O(n) \) space, see Hamacher as a reference in [7].

3.3.5 Sum of absolute differences (SAD)

The algorithms also looks for intensity gradients to find depth discontinuities. An intensity gradient is defined as a minimum intensity jump over a minimum number of pixels. For instance a gradient of at least 5 grey levels over 3 pixels. A pixel \( x \) in the left scanline is said to lie to the left of an intensity gradient (\( \kappa_L \)) or reward for a match (\( \kappa_r \)) and \( d(x_i, y_i) \) is the dissimilarity measure.

The dissimilarity measure is used to decide whether \( I_L(x_i) \) and \( I_R(y_i) \) are images of the same scene point. It is defined as:

\[
d(x_i, y_i) = \min(d(x_i, y_i, I_L, I_R), \tilde{d}(y_i, x_i, I_R, I_L))
\]

where:

\[
\tilde{d}(x_i, y_i, I_L, I_R) = \min_{y_i - \frac{n}{2} \leq y_i \leq y_i + \frac{n}{2}} |I_L(x_i) - I_R(y_i)|
\]

where \( I_R(y) \) is the linearly interpolated greyscale value.

\(^1\)1 Bit serial multipliers can be built but are very slow - they trade speed for space.
The computed costs are then saved in an array from which the lowest cost path is inferred. The array has the length of a scanline and the height of the maximum possible disparity. Figure 5 shows a diagram of the cost array, where the cost is calculated for each \((x,y)\) cell, where \(x\) is the pixel position on the current epipolar line in the left image and \(y\) is the pixel position on the same line in the right image. The black cells represent cells that do not need to be computed: when the maximum disparity is known \((3\) in the example implies \(\delta = y - x \leq 3\)).

For efficient storage, the array in figure 5 can be transformed to a dense array (see figure 4) indexed by \((\delta, y)\) where \(y\) is the pixel position on the current epipolar line in the reference image and \(\delta\) ranges over all possible disparities.

The output of the cost array is a match sequence: an exemple is given in figure 6, which shows the matching points for a same line for the left and the right images. Occlusions are marked by the white dots: when a match occurs, a line links pixels from left and right images: the disparity is proportional to the slope of this line.

4. EXPERIMENTAL SETUP AND BENCHMARK DEFINITION

4.1. Stereo bench setup-Patrice

Mark any photo of the stereo system in your thesis or papers we can use? Else can I have an exact description of the camera used, distances etc... Plus soft used to control the Panasonic cameras?

4.2. Benchmark creation

4.2.1. landmarks-patrice

I need images of the manekin and test subject with and without a grid placed in shared-vision. For each image where points were clicked I need how many points were clicked. It would be great to have a face (manekin and a test subject) reconstructed with texture mapped for display purpose.

4.2.2. interpolation: from sparse to dense disparity map

Both alex and mark please write whatever you did to generate the benchmark disparity map on that section.

- bicubic or related-alex-mark
- model-based-mark-alex (still used ?)

5. EXPERIMENTAL RESULTS - COMPARISON

We need results from the manikin and at least one human for each algorithm. Results mean disparity map to show. Then curve of disparity discrepancy between benchmark and algo results. Plus some statistics. Send me any excel file of the results and see with georgy and philippe as well. Try all to use the same excel file definition so we can merge them easily.
6. CONCLUSION

When we will get there we will have a good reason to celebrate...

7. REFERENCES


