Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps

Christopher Zach, David Gallup, Jan-Michael Frahm and Marc Niethammer
Department of Computer Science
University of North Carolina at Chapel Hill
Email: {cmzach,gallup,jmf,mn}@cs.unc.edu

August 26, 2008

Abstract

This work presents a real-time, data-parallel approach for global label assignment on regular grids. The labels are selected according to a Markov random field energy with a Potts prior term for binary interactions. We apply the proposed method to accelerate the clean-up step of a real-time dense stereo method based on plane sweeping with multiple sweeping directions, where the label set directly corresponds to the employed directions. In this setting the Potts smoothness model is suitable, since the set of labels does not possess an intrinsic metric or total order. The observed run-times are approximately 30 times faster than the ones obtained by graph cut approaches.

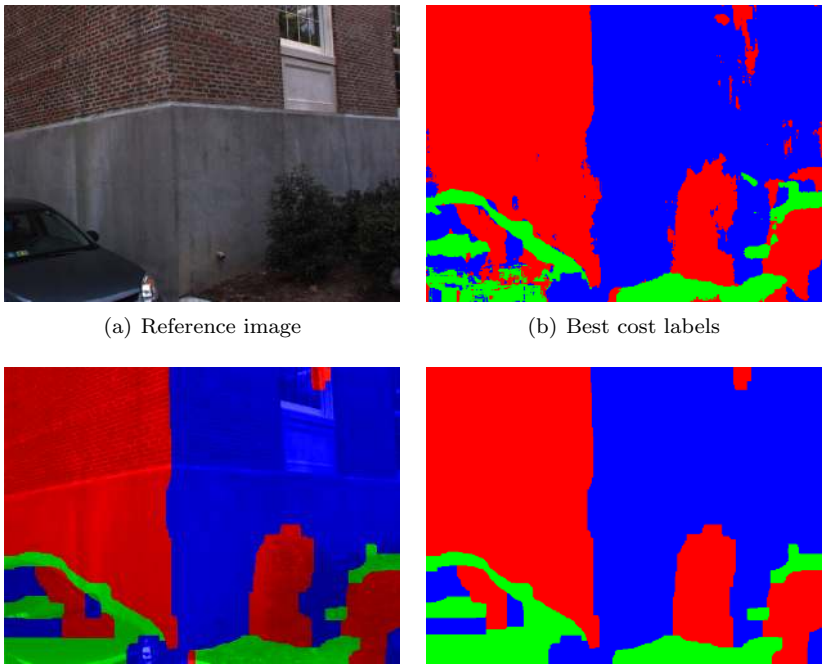
1 Introduction

The best results in stereo have come from global methods, however, these methods are still too computationally demanding in order to be used in real-time applications or other applications where processing time is a critical resource. Gallup et al. [11] present a real-time stereo method which uses plane-sweeping and local matching to quickly produce depthmaps for three surface-aligned sweeping directions. The final depthmap is a per-pixel selection from the three candidate depths, which is solved with regularization using a global energy. Although the problem involves only three labels (namely the employed sweeping direction), optimization using graph cuts still takes several seconds, making the final step unsuitable for real time. Thus, the authors recommend a local best cost selection scheme for real-time applications.

In this paper we present a real-time solution to the global labeling problem. By relaxing the energy to the continuous case, our method can compute the true global minimum, and since the computation is highly data-parallel, the solution can be computed efficiently using graphics hardware. The relationship between the Potts discontinuity model and total variation regularization enables the continuous formulation of the global labeling task.

We have applied our method to the three-label problem proposed for depth map clean-up in [11], with slight modifications to the energy formulation to facilitate efficient, data-parallel minimization. Despite these modifications, our results are comparable in quality to those presented in previous work, and the computation is orders of magnitude faster. Thus, our work enables higher quality global labeling results to be computed in real-time, which was not possible before. The core of our approach is not restricted to dense stereo computation, and can be applied on a varierty of labeling problems.

The combination of plane sweep stereo using multiple directions with global label assignment is interesting for the following reasons: (a) it allows the refinement and clean-up of the depth maps without the huge computational costs that come with other global stereo approaches. (b) Since the label set corresponds to dominant facade directions in urban scenes, the refined labels can be used to assist a subsequent semantic analysis of the captured geometry.



(a) Reference image (b) Best cost labels

(c) Graph cut result (from [11]) (d) Proposed method

Figure 1: Plane sweep with multiple directions results. (a) shows the reference image used to compute the best matching costs for multiple sweep directions. (b) shows the labels (represented by the three color channels) corresponding to the directions with the lowest matching costs. (c) depicts the cleaner label assignment using graph cuts as proposed in [11] (with enhanced colors, $\approx$ 2s runtime); and (d) displays our result, that is visually most similar to (c) ($\lambda = 60$, 58ms runtime). This figure is best viewed in color.

2 Related Work

In this section we focus on previous work related to global label assignment. We refer to [21, 5] for an overview and evaluation of stereo methods.

Binary labeling problems incorporating a matching cost term and a spatial smoothness prior can be solved using network flow approaches [14]. Since the primary tool is the construction of an appropriate directed graph and determining the minimum cut, a whole class of methods based on this principle is usually referred as graph cut methods. Label assignment with more than two labels can be approximately addressed by a sequence of binary labeling methods, e.g. α-expansion and α-β-swaps [3]. A lot of work has been done recently to address some shortcomings of graph cuts for multi label problems (e.g. [16, 13, 15]). Exact solutions for labeling problems with linear discontinuity costs [2] and convex pairwise interactions [12] can be obtained by suitable graph constructions.

Graph cut based approaches are mostly sequential algorithms, and all attempts to accelerate these methods by highly parallel computations, in particular on GPUs, have had limited success so far. Another method to optimize Markov random fields is loopy belief propagation based on message passing [10, 22, 9]. The updates executed in message passing methods can be performed in parallel and are therefore highly suitable for GPU implementation. The major problem with loopy belief propagation is, that inference (i.e. optimal label assignment) is only exact for tree graphs, where the message passing algorithm is an extension of dynamic programming. Belief propagation in loopy graphs (e.g. image grids with 4-neighborhoods) is only a heuristic optimization procedure. Although loopy belief propagation often works well in practice, divergent and oscillating behaviour may be observed.

Consequently, continuous methods for global labeling problems, that provide at least global optimal results for convex constraints replacing non-convex ones, are appealing for an accelerated GPU implementation. In [19] a variational method is proposed to find the global optimum of labeling problems with total variation (TV) regularization (in particular for dense depth estimation from stereo images). This approach is only applicable if the set of labels has a natural distance metric, since the employed TV regularization is equivalent to a linear discontinuity cost model for the label values. In our application the set of labels (major sweep directions) have no natural metric, and a different solution is required. In the following we derive a continuous method for global labeling with the Potts discontinuity model.

3 Global Label Assignment and the Potts Model

In this section we propose a novel label assignment approach based on continuous energy minimization. Typically, global label assignment maps pixel locations to labels, $\Lambda : \Omega \to \mathcal{L}$, and it is formulated as a energy minimization problem. Here, Ω denotes the typically rectangular image domain, and $\mathcal{L} = \{1, \dots, L\}$ is the set of labels. The underlying energy functional accumulates the data cost for selecting label l at pixel x, $D(x, l)$, and a spatial smoothness cost to regularize the resulting assignment. Thus, the goal is to find the minimizer of

$$E(\Lambda) = \lambda \int_\Omega D(x, \Lambda(x))\, dx + V(\Lambda), \tag{1}$$

where λ controls the importance of the data fidelity on the overall energy. For many computer vision problems typical choices for $V(\Lambda)$ are the homogeneous regularization, $V(\Lambda) = \int_\Omega \|\nabla\Lambda\|^2\, dx$, and the total variation, $V(\Lambda) = \int_\Omega \|\nabla\Lambda\|\, dx$. Since in our application the labels representing plane directions have no natural order, these gradient based regularization terms are not applicable.

If we focus on the data energy $\int_\Omega D\big(x, \Lambda(x)\big)\, dx$ and ignore the smoothness cost for now, determining the optimal data energy is just a point-wise minimization problem,

$$\min_{l \in \mathcal{L}} c_{x,l}, \tag{2}$$

where we substitute $c_{x,l} = D(x, l)$. By interpreting the optimization task above as a (rather trivial) linear program, we can analyze its dual problem:

$$\begin{aligned} \min_{u_{x,l}} \sum_{l \in \mathcal{L}} c_{x,l} u_{x,l} \quad \text{s.t.} \\ u_{x,l} \geq 0 \\ \sum_{l \in \mathcal{L}} u_{x,l} = 1. \end{aligned} \tag{3}$$

The interpretation of the unknowns $u_{x,l}$ is, that $u_{x,l} \in [0, 1]$ is the continuous version of the indicator function $\chi(\Lambda(x) = l)$. Although $u_{x,l}$ is not enforced to be binary by the constraints, the dual linear program will result in a unique binary solution if all data costs are distinct for a pixel, i.e. $c_{x,l} \neq c_{x,l'}$ for $l \neq l'$. Otherwise a set of equally optimal assignments for $u_{x,l}$ is obtained, but an optimal binary solution $u^*_{x,l}$ can be determined by setting exactly one non-zero $u_{x,l}$ to 1, and all other variables to 0.

In the following we will use the notation u_l for the indicator function of label l, i.e. $u_l(x) = u_{x,l}$. Further, let u_x denote the vector $(u_{x,1}, \dots, u_{x,L})$ at location x, i.e. $u_x(l) = u_{x,l}$. Thus, u_l is a particular slice of the volume $\Omega \times \mathcal{L} \to \mathbb{R}$, and u_x is a specific column in the label direction. These notations are also used for other mappings with domain $\Omega \times \mathcal{L}$.

The Potts Model The Potts model for smoothness priors is appropriate if the numeric value of a label has no particar meaningful interpretation, e.g. when labels have a purely symbolic character. In the Potts model the smoothness cost is zero, if neighboring locations are assigned

with the same label. Otherwise a constant penalty (independent of the actual values of the labels) is added to the overall energy. More formally,

$$P_\Lambda(x) = \begin{cases} 0 & \text{if } \|\nabla\Lambda(x)\| = 0 \\ 1 & \text{otherwise,} \end{cases} \tag{4}$$

where we assign a unit cost for discontinuities. Note, that $\nabla\Lambda$ is understood on a discrete grid, i.e. as finite differences. Further, we can employ the L^2 (Euclidean), L^∞ (maximum norm) or the L^1 norm for $\|\nabla\Lambda\|$ (resulting in different preferred orientations induced by the regularization).

In the following, we denote the spatial gradient of u by $\nabla_x u$, i.e. for 2-dimensional image domains we have $\nabla_x u = (\partial u/\partial x, \partial u/\partial y)^T$. We can rewrite the accumulated Potts discontinuity cost $\int_\Omega P_\Lambda dx$ in terms of u:

Proposition 1 *If $u_{x,l} \in \{0,1\}$ is the binary indicator function, $u_{x,l} = \chi(\Lambda(x) = l)$, then*

$$\int_\Omega P_\Lambda dx = \frac{1}{2} \sum_l \int_\Omega \|\nabla_x u_l\| dx. \tag{5}$$

(Recall that $u_l(x) = u_{x,l}$.)

Proof: By the coarea formula for functions of bounded variation it is known that (for $l \in \mathcal{L}$)

$$\int_\Omega \|\nabla_x u_l\| dx = \mathrm{Per}(\partial U_l), \tag{6}$$

where $U_l = \{x \in \Omega : u_{x,l} = 1\}$ is the set induced by the indicator function u_l. Note that ∂U_l is exactly the set of jumps involving label l. Since for every $x \in \Omega$ exactly one $u_{x,l}$ is set (i.e. has value 1), every discontinuity in the label assignment (e.g. switching from l to l') contributes to two such perimeters — namely $\mathrm{Per}(\partial U_l)$ and $\mathrm{Per}(\partial U_{l'})$. Hence the right hand side of Eq. 5 is twice the number of discontinuities of Λ, i.e. $2 \times \int P_\Lambda dx$. □

A Continuous Formulation The results from the previous paragraphs can be combined to obtain a continuous formulation for the labeling problem with relaxed constraints. Merging the linear program in Eq. 3 with the relation from Proposition 1 leads to the following energy to minimize over u:

$$E_1(u) = \int_\Omega \left(\sum_l \|\nabla_x u_l\| + \lambda \sum_l c_{x,l} u_{x,l} \right) dx, \tag{7}$$

with the constraints $u_{x,l} \geq 0$ and $\sum_l u_{x,l} = 1$. Since Eq. 7 is difficult to optimize directly, we decouple the regularization and data term by introducing an additional function v linked to u by a quadratic approximation force [1, 4]:

$$\begin{aligned} E_2(u;v) = \int_\Omega \Bigg(& \sum_l \|\nabla_x u_l\| \\ & + \sum_l \frac{1}{2\theta} (u_{x,l} - v_{x,l})^2 \\ & + \lambda \sum_l c_{x,l} v_{x,l} \Bigg) dx, \end{aligned} \tag{8}$$

subject to $v_{x,l} \geq 0$ and $\sum_l v_{x,l} = 1$. θ is a parameter that controls the influence of the squared distance between u and v in Eq. 8. This technique of "quadratic relaxation" allows the combination of TV-based smoothness costs with arbitrary data terms, and the utilization of well-established and efficient methods for TV-regularization. If θ is set to a very small number, then u and v are very close respective approximations, but the speed of convergence is substatially reduced. Typical choices for θ are 1/10 and 1/20. Now the (convex) optimization task can be solved by alternating steps minimizing either u for constant v or vice versa.

Optimizing E_2 with respect to u We can omit the constant data fidelity term depending only on v in 8. Thus, the task is to solve

$$\min_u \int_\Omega \Big(\sum_l \|\nabla_x u_l\| + \sum_l \frac{1}{2\theta}(u_{x,l} - v_{x,l})^2 \Big) dx. \tag{9}$$

This decomposes into L independent image denoising problems (for $l \in \mathcal{L}$):

$$E_l^{ROF} = \min_{u_l} \int_\Omega \Big(\|\nabla_x u_l\| + \frac{1}{2\theta}(u_{x,l} - v_{x,l})^2 \Big) dx, \tag{10}$$

which is known as the Rudin-Osher-Fatemi (ROF) model [20] and can be solved efficiently by a gradient descent procedure [6, 7]. Here we only briefly sketch the procedure proposed in [7]: $\|\nabla_x u_l\|$ can be rewritten as $\max_{p_l : \|p_l\| \le 1} \langle p_l, \nabla_x u_l \rangle$, thereby introducing the dual vector-valued function p_l, which essentially removes the non-linearity induced by $\|\nabla_x u_l\|$. Eq. 10 reads then as

$$\min_{u_l} \max_{\|p_l\| \le 1} \int_\Omega \left(\langle p_l, \nabla_x u_l \rangle + \frac{1}{2\theta}(u_{x,l} - v_{x,l})^2 \right) dx. \tag{11}$$

Computing the functional derivatives of Eq. 11 with respect to the unknowns u_l and p_l yields the following gradient descent/reprojection equations for p_l:

$$\begin{aligned} \tilde{p}_l^{(t+1)} &= p_l^{(t)} + \tau \nabla_x u_l \\ p_l^{(t+1)} &= \pi_B(\tilde{p}_l^{(t+1)}), \end{aligned} \tag{12}$$

where $\tau < \theta/4$ is the timestep, and $\pi_B(\cdot)$ denotes the projection into the unit ball $B = \{x : \|x\| \le 1\}$. The corresponding values of u_l can be determined by

$$u_{x,l} = v_{x,l} + \theta\big(\nabla \cdot p_l(x)\big). \tag{13}$$

It turns out that the finite difference implementation of ∇u_l and $\nabla \cdot p_l$ must be dual (in the sense of linear operators), e.g. if ∇u_l is approximated by forward differences, $\nabla \cdot p_l$ is based on backward differences.

We did not specify the exact norm $\|\cdot\|$ appearing in the equations above. The Euclidean norm $\|\cdot\|_2$ does not prefer certain directions and is the appropriate choice for many applications. We observed, that the L^1 norm, $\|x\|_1 = \sum |x_i|$, applied on ∇u_l gives visually more appealing results. Note that using $\|\nabla u_l\|_1$ results in preference of horizontal and vertical directions along the discontinuities. Further, $\|\nabla u_l\|_1$ translates to using the maximum norm on the dual variables p_l, since

$$\|y\|_1 = \max_{\|p\|_\infty \le 1} \langle p, y \rangle \tag{14}$$

for every $y \in \mathbb{R}^n$. Further, the unit ball $B = \{x : \|x\|_\infty \le 1\}$ is just the unit square, and $\pi_B(\tilde{p}_l)$ is obtained by clamping the components of $\tilde{p}_l$ to the range $[-1, 1]$.

Optimizing E_2 with respect to v This task decouples into separate subproblems for every pixel x, since the $v_{x,l}$ do not spatially interact with neighboring values. Hence, we are facing the following (data-parallel) minimization problems for every position x:

$$\min_{v_{x,l}} \sum_l \left(\frac{1}{2\theta}(u_{x,l} - v_{x,l})^2 + \lambda c_{x,l} v_{x,l} \right) \quad \text{s.t.} \quad \sum_{l \in \mathcal{L}} v_{x,l} = 1,\ v_{x,l} \ge 0. \tag{15}$$

We can rewrite Eq. 15 to obtain the equivalent formulation:

$$\min_{v_{x,l}} \sum_l \big((u_{x,l} - \lambda\theta c_{x,l}) - v_{x,l}\big)^2 \quad \text{s.t.} \tag{16}$$
$$\sum_{l\in\mathcal{L}} v_{x,l} = 1,\ v_{x,l} \geq 0,$$

which means, that the vector $v_x = (v_{x,1}, \ldots, v_{x,L})$ is the closest point (in terms of the Euclidean distance) to the vector $u_x - \lambda\theta c_x$ on the canonical simplex, i.e. the projection $\pi_S(\cdot)$ on the unit simplex. This problem is well-known in the literature, and a particular simple algorithm is an active set method based on successive projections and corrections [17]: Since the set I of inactive

Algorithm 1 Update procedure to minimize E_2 with respect to v

$v_{x,l} \leftarrow u_{x,l} - \lambda\theta c_{x,l}$, $I = \{1, \ldots, L\}$
repeat
  {Projection onto plane}
  $\bar{v} = \left(\sum_{l\in I} v_{x,l}\right)/|I|$
  $\forall l \in I : v_{x,l} \leftarrow v_{x,l} - \bar{v} + 1/|I|$
  {Enforce inequality constraints}
  $I \leftarrow I \setminus \{l : v_{x,l} < 0\}$
  $\forall l \notin I : v_{x,l} \leftarrow 0$
until $\sum_l v_{x,l} = 1$

inequality constraints ($v_{x,l} \geq 0$) is reduced by at least one element in every iteration (if the current solution is still infeasible), the algorithm requires at most L iterations. Since the label set $\mathcal{L}$ is very small in our application, we can manually unroll the loop to obtain coherent parallel execution on GPUs (see Section 5).

Discussion The Potts model for discontinuities formulated in terms of assigned labels is not convex, but our formulation in (Eqs. 7 and 8) based on soft indicator variables is. The constraints of the original problem, $u_{x,l} \in \{0, 1\}$, are not convex and are essentially replaced by the bounds $u_{x,l} \in [0, 1]$, hence $u_{x,l}$ can attain fractional values. Exactly this modification makes the problem convex and allows a global optimal solution to be determined efficiently. But this means, that assigning labels according to $\Lambda(x) = \arg\max_l u_{x,l}$ does not necessarily return a global optimum of the original discrete problem. In certain cases the relaxed continuous formulation provides a global optimum for the discrete problem, e.g. TV-L^1 denoising of binary input images results in global optimal solutions after thresholding for allmost all threshold values [8, 18]. No such result is known for minimizing Eq. 7, hence the obtained discrete label assignment will be a (usually very strong) solution. In practice we observed that the assigned $u_{x,l}$ is binary for allmost all pixels (after convergence). Note, that the iterated graph cut approach used in [11] only returns a strong solution as well.

4 Application to Stereo with Multiple Sweeping Directions

Efficient solutions to the global labeling problem are of particular importance in large scale 3D reconstruction from video. Due to the enormity of video data, processing time is a major concern. Other applications such as mission planning, change detection, and robot navigation require results in a timely fashion, if not immediately. Capturing even a small city from ground requires literally millions of frames of video. For practical use, processing time must be comparable to the capture time, thus real-time is an important goal. Note, that this requirement excludes many global or semi-global approaches using the full disparity range as potential labels.

In that regard, Gallup et al. [11] present a real-time stereo tailored (but not limited) to broad planar surfaces such as those found in urban environments. The fastest stereo methods are local,

which compute matching in windows in the image. In urban scenes acquired from street-level, ground surfaces such as streets and sidewalks, and facade surfaces between buildings, are often viewed at steep angles and thus appear highly slanted in the image. Such surfaces pose a problem for window-based matching: while the center pixel is in correspondence, other pixels in the window, especially outer pixels, are not, which can lead to mismatches in the final result. Preferably the window should be aligned to the surface in 3D, in which case the correct match will feature all pixels in correspondence.

Performing local stereo with cost windows aligned with an exhaustive set of surface normals is not feasible, hence only promising sweeping directions are retained. In urban environments the ground surface normal and two orthogonal facade normals are dominant, therefore three directions are sufficient for city modeling. The main sweep directions can be determined from vanishing points or from sparse reconstructions obtained by visual odometry methods.

Once the surface normals are found, a local best-cost plane-sweep is performed for each direction. This produces one depthmap for each sweeping direction. The final depthmap can be obtained simply by selecting per-pixel the depth with minimal matching cost. However, matching scores are often somewhat noisy, which leads to errors in the selection. Hence, regularization with spatial smoothness priors is inevitable.

The minimization method presented in this paper can solve the labeling task orders of magnitude faster than graph-cuts (which require a few seconds), making the high quality method possible in real-time. In [11] two types of smoothness penalties are proposed: compatibility between labels, and smoothness between depths ("integrability cost"). We found that the integrability penalty has a minor contribution to the results, and it is difficult to optimize efficiently.

Figure 1 shows that our formulation produces nearly identical results to those obtained using the more complex graph cut formulation proposed in [11]. The run-time for the minimization step is 58ms, and in addition to 30ms for the plane-sweeps, the overall processing rate is about 11 Hz. The observed runtime for the graph cut implementation on the same image data is 2s, i.e. approximately 34 times slower.

5 Implementation

This section provides more details on our GPU-accelerated algorithm for the continuous formulation of global label assignment. Although the CUDA programming paradigm is currently considered as the state-of-the art approach for efficient GPU programming, we still employ the OpenGL API and Cg shading language for the following reasons: (a) it allows the implementation to be executed on a substantially larger range of graphics hardware from different vendors and on older GPU generations as well. Note that in contrast to the scalar G80 architecture from NVidia, the current generation of GPUs from AMD/ATI still use vectorized processing units. In particular, the shader-based specification of Algorithm 1 conveniently makes use of intrinsic vector operations. (b) CUDA is usually only substantially faster than shader based methods if the proposed shared memory programming model can be exploited, which is only the case to a limited extent for the approach presented in this work.

Shader-based Implementation Since the number of required labels for our applications is very small (three labels in our setting), we can represent the data cost volume $c_{x,l}$, the soft indicator functions u_l and the respective dual variables p_l by regular 2D textures with the respective number of color channels. In practice, it is sufficient to represent u_l and p_l by 16-bit floating point components, hence the required memory bandwidth can be reduced (which results in improved runtimes). Since p_x is then comprised by 6 components (x and y components for every label l), p_x can be represented by two textures. Updating p_x in a single pass requires the ability to render into multiple targets. Alternatively, the two 16-bit components of $p_{x,l}$ for a specific label can be packed into one 32-bit floating point value on NVidia GPUs. Thus, the complete set of values for p_x can be encoded in one multi-channel floating point texture.

The alternating optimization step Eq. 12 (and Eq. 13) and Algorithm 1 corresponds directly to a pair of shader programs, which are outlined in Eq. 17–20:

$$1a. \quad v_x \leftarrow \pi_S\left(u_x^{(t)} - \lambda\theta c_x\right) \tag{17}$$

$$1b. \quad u_x^{(t+1)} \leftarrow v_x + \theta\,\nabla\cdot p_x^{(t)} \tag{18}$$

$$2a. \quad \tilde{p}_x \leftarrow p_x^{(t)} + \frac{\tau}{\theta}\nabla u_x^{(t+1)} \tag{19}$$

$$2b. \quad p_x^{(t+1)} \leftarrow \max\bigl(-1, \min(1, \tilde{p}_x)\bigr), \tag{20}$$

where the min and max operators in 2b. are understood as component-wise application on the input vector. $\tilde{p}_x$ is a temporary variable local to the update step. The gradient and the divergence are computed by finite differencing neighboring pixels.

The projection $\pi_S(\cdot)$ on the canonical simplex is achieved by unrolling the loop in Algorithm 1. Essentially, the following Cg source fragment is repeated L times:

```
cardI = (I.x + I.y + I.z);
v = v + (1.0 - dot(I, v)) / cardI;
I = (v < 0) ? half3(0) : I;
v = max(v, float3(0));
```

The binary vector `I` represents the set of inactive constraints. The first line moves `v` on the respective plane by subtracting a modified mean over all non-zero elements. The second and third line determine the active inequality constraints and forces `v` to be non-negative.

Coarse-to-fine Strategy Since incorporating a global smoothness prior allows pixels to communicate over the entire image, convergence to stable results can be slow. Figure 4 illustrates an example, where the initally assigned labels are revised again (in the lower left corner of the image). Figure 4(a) shows the obtained result after 300 iterations without a multi-scale approach, and Figure 4(b) depicts the labeling after 1000 iterations. There is still no clear assignment in the indicated region, although most of the image appears already converged. In order to accelerate the procedure we employ a multi-scale approach similar to the one proposed in [9] (see also Figure 2). The positive influence of a coarse-to-fine method on the convergence rate can be seen in Figure 4(c), where 4 levels with half the resolution of the previous one are used. The obtained result after 100 iterations on the base level is virtually identical to the fully converged result (Figure 4(d)).

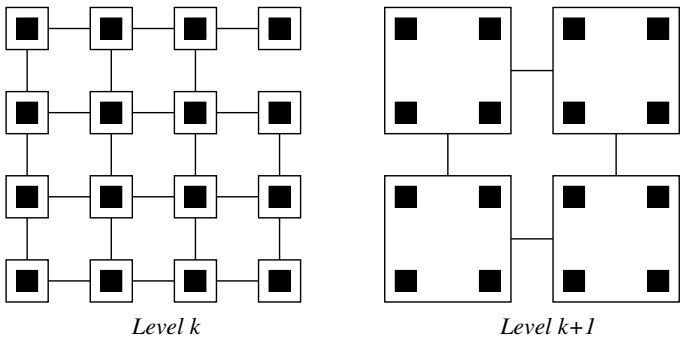


Figure 2: Illustration of the multi-scale approach. A group of 2×2 pixels is merged in the next level.

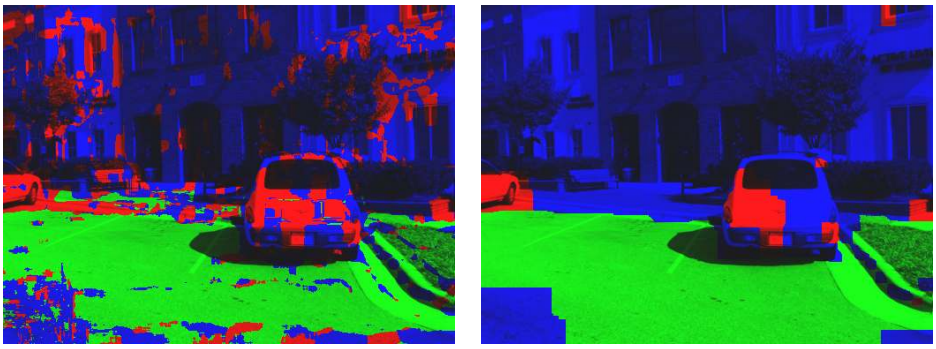
Some considerations about the data costs on coarser levels are required. Figure 5(a) shows the result on the full resolution base level with $\lambda = 50$. In order to benefit from the multi-scale approach, a suitable initialization from previous levels in the pyramid is required. Figure 5(b)

shows the desired result at level 2 (quarter resolution), but neither averaging nor accumulation of data costs from the finer resolution levels provide the intended result, if λ is fixed for all levels (Figure 5(c) and (d)).

Assume for now that all pixels in a 2×2 block as indicated in Figure 2 have the same data cost $c_{x,l}$ in level k, and we use the average cost in the next level $k+1$. Then the overall contribution of the data fidelity term, $\int \sum_l c_{x,l} u_{x,l}$, to the combined energy E_1 on level $k+1$ is one quarter of the data energy at level k. But the set of discontinuities contributing to the smoothness energy is only reduced by a factor of two, since they are one-dimensional level curves. Hence, the correct value of $\lambda^{(k+1)}$ at level $k+1$ is $2\lambda^{(k)}$, with $\lambda^{(0)} = \lambda$, the given data weight.

6 Results

In this section we provide timing and visual results for our method. The utilized PC hardware is equipped with a NVidia Geforce 8800 Ultra GPU and a 3 GHz CPU. Run-times are measured under a Linux OS using current OpenGL drivers. One issue with GPU-based iterative methods is the stopping criterion, since this usually involves an expensive reduction operation e.g. to compute the current energy or the maximal update of the unknowns. We empirically found out that 150 iterations on each level using the coarse-to-fine approach yields to (visually) converged results. The observed run-times for global label assignment are approximately 60ms for 512×384 images, and 45ms for 384×288 pixels.



(a) Best cost labels (b) Global assignment

Figure 3: Local and global label assignment. (a) shows a lot of noise in the plane directions (labels) selected by taking the minimal matching costs. (b) shows the much more consistent labeling result. In terms of the dominant plane orientations the car in the foreground is a highly ambiguous object, resulting in the non-uniform label assignment (which is less important for non-planar objects).

Figure 3 and Figure 6 illustrate the obtained labelings with local (best-cost) assignment and the global approach. Incorporating a smoothness prior does not only result in cleaner label maps, but reduces the noise in the 3D model as shown in Figure 6(c) and (d). The ability of global optimization to reduce the noise in the final model is limited by early determining a small set of possible depth values for every pixel. The refined labels provide a significantly improved segmentation of the scene at low computational costs.

7 Conclusion

In this work we introduced a data-parallel approach to solve Markov random fields on regular grids with a Potts smoothness prior. Using modern GPUs the observed performance is more than 30 times faster than a graph cut based approach. One suitable application demonstrated in this work is the postprocessing and clean-up step for depth maps obtained by real-time stereo methods.

In this work we restricted ourselves to a uniform weighting between data costs and smoothness priors. Future work needs to explore the applicability of weighted TV-norms, that yield to generalized Potts discontinuity models. Note that in this setting a slightly extended version of Proposition 1 still holds. Additionally, incorporating the refined label assignment into a subsequent semantinc analysis procedure for urban environments is left as future work.

Acknowledgments: We gratefully acknowledge support from NVidia Corporation.

References

[1] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition–modeling, algorithms, and parameter selection. *Int. Journal of Computer Vision*, 67(1):111–136, 2006.

[2] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–655, 1998.

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

[4] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Fast Global Minimization of the Active Contour/Snake Model. *Journal of Mathematical Imaging and Vision*, 2007.

[5] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[6] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1–2):89–97, 2004.

[7] A. Chambolle. Total variation minimization and a class of binary MRF models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 136–152, 2006.

[8] T. F. Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2004.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 261–268, 2004.

[10] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int. Journal of Computer Vision*, 40(1):25–47, 2000.

[11] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[12] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(10):1333–1336, 2003.

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568–1583, 2006.

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

[15] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[16] N. Komodakis and G. Tziritas. A new framework for approximate labeling via graph cuts. In *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[17] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of α^n. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.

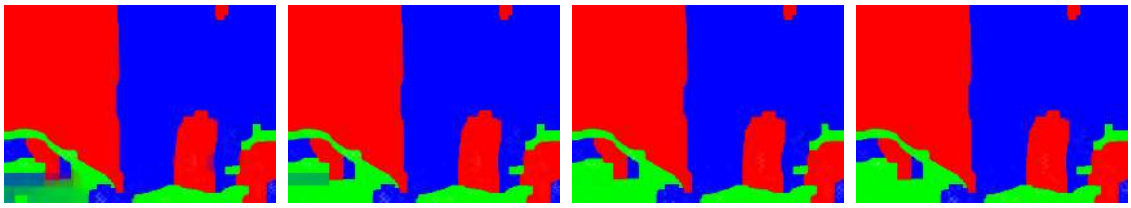
[18] M. Nikolova, S. Esedoglu, and T. F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[19] T. Pock, T. Schoenemann, D. Cremers, and H. Bischof. A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision (ECCV)*, 2008. to appear.

[20] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[21] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47(1-3):7–42, 2002.

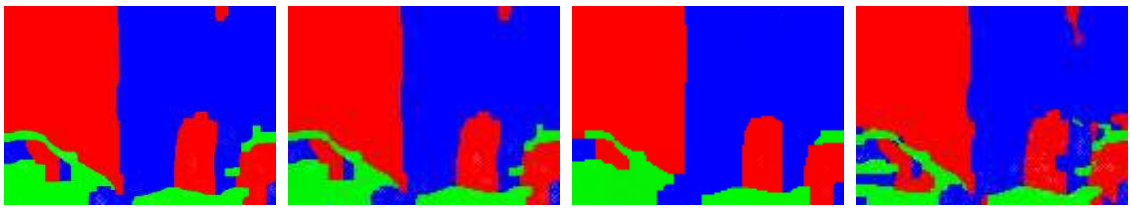
[22] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *IEEE International Conference on Computer Vision (ICCV)*, pages 900–907, 2003.



(a) Level 0, 300 iterations (b) Level 0, 1000 iterations (c) Level 4–0, 100 iterations (d) Converged result

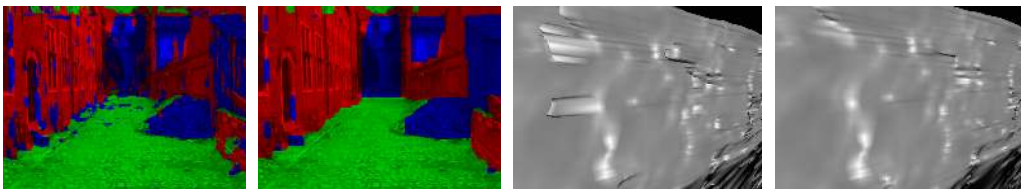
(e) Level 0, 300 iterations (f) Level 0, 1000 iterations (g) Level 4–0, 100 iterations (h) Converged result

Figure 4: A coarse-to-fine approach speeds up convergence. Top row: $u_{x,l}$ obtained after the specified number of iterations without (a–b) and with using multiple scales (c). (d) shows the converged result after 5000 iterations. Bottom row: The obtained labels by selecting $\arg\max_l u_{x,l}$ at every pixel. Although the labeling in (f) is similar to the final result (h), (b) indicates that the u_l still do not induce a clear decision in the lower left region. This figure is best viewed in color.



(a) Full resolution (b) Level 2, correct cost scaling (c) Level 2, cost averaging (d) Level 2, cost accumulation

Figure 5: A coarse-to-fine approach requires the correct scaling of the cost values. (a) shows the result on the full image resolution (512×384); (b) shows the result on quarter resolution (128×96) with correct downsampling of the data term; in (c) the downscaled costs are the means of the costs at the previous level, and yield to oversmoothed results; and in (d) the cost are added, which leads to less regularized label assignments. This figure is best viewed in color.



(a) Best cost labels (b) Global assignment (c) Best cost 3D model (d) Model with global assignment

Figure 6: The Begijnhof sequence (courtesy of Marc Pollefeys). (a) and (b) show the best cost and global labeling results, respectively. Global labeling almost perfectly results in a semantic segmentation of the ground (green), fronto-parallel parts (blue) and orthogonal facades (red). (c) and (d) depict the lit, but untextured facade in the left portion of the image without and with global labeling. This figure is best viewed in color.