A Theory Details

In this appendix, we give a complete exposition and proof of Lemma 1 and use it to prove Theorem 2 from Section 3. We also discuss a subtlety regarding the size of stable blocks, and show that adding perturbations to the node costs seems necessary to prove Lemma 1.

A.1 Proofs of Lemma 1 and Theorem 2

We now more formally develop the connection between the block dual (5) and block stability. To begin, the *pairwise dual* of the LP (3) is given by:

$$\max_\eta P(\eta) = \max_\eta \sum_{u\in V} \min_i (\theta_u(i) + \sum_v \eta_{uv}(i)) + \sum_{uv\in E} \min_{i,j}(\theta_{uv}(i,j) - \eta_{uv}(i) - \eta_{vu}(j)) \quad (10)$$

This can be derived by introducing Lagrange multipliers η on the two consistency constraints for each edge $(u,v)\in E$ and each $i \in L$:

$$\sum_i x_{uv}(i,j) = x_v(j)\ \forall j$$
$$\sum_j x_{uv}(i,j) = x_u(i)\ \forall i$$

A dual point η is said to be *locally decodable* at a node u if the cost terms

$$\theta_u(i) + \sum_{v:uv\in E} \eta_{uv}(i)$$

have a unique minimizing label i. This dual P has the following useful properties for studying persistency of the LP (3):

Property 1 (Strong Duality). *A solution η^* to the maximization (4) has $P(\eta^*) = Q(x)$, where x is a solution to the pairwise LP (3). Here $Q(x)$ is the objective function of (3); this is identical to Q from (2) when x is integral.*

Property 2 (Complementary Slackness, Sontag et al. (2012) Theorem 1.2). *If x is a primal solution to the pairwise LP (3) and there exists a dual solution η^* that is locally decodable at node u to label i, then $x_u(i) = 1$. That is, if the dual solution η^* is locally decodable at node u, the primal solution x is not fractional at node u.*

Property 3 (Strict Complementary Slackness, Sontag et al. (2012) Theorem 1.3). *If the LP (3) has a unique, integral solution x, there exists a dual solution η^* to (4) that is locally decodable to x.*

In particular, Property 2 says that to prove the primal LP is persistent at a vertex u, we need only exhibit a dual solution η^* to (4) that is locally decodable at u to $g(u)$, where g is an integer MAP solution. Properties 1 and 3 will be useful for proving results about a different Lagrangian dual that relaxes fewer constraints, which we study now.

Given a partition $V = (S_1,\dots S_B)$ (henceforth a "block decomposition"), we may consider relaxing fewer consistency constraints than (4) does, to form a *block dual*.

$$\max_\delta B(\delta) := \max_\delta \sum_b \min_{x^b} \left(\sum_{u\in S_b}\sum_{i\in L} \left(\theta_u(i) + \sum_{v:(u,v)\in E_\partial} \delta_{uv}(i)\right) x_u^b(i) + \sum_{uv\in E_{S_b}} \sum_{i,j} \theta_{uv}(i,j) x_{uv}^b(i,j) \right) + \sum_{uv\in E_\partial} \min_{i,j}(\theta_{uv}(i,j) - \delta_{uv}(i) - \delta_{vu}(j)) \quad (11)$$

subject to the following constraints for all $b \in \{1,\dots,B\}$:

$$\begin{aligned} &\sum_i x_u^b(i) = 1, && \forall u \in S_b,\ \forall i\in L\\ &x_u^b(i) \ge 0, && \forall u\in S_b\ \forall i \in L.\\ &\textstyle\sum_j x_{uv}^b(i,j) = x_u^b(i) && \forall (u,v)\in E_{S_b},\ \forall i\in L,\\ &\textstyle\sum_i x_{uv}^b(i,j) = x_v^b(j) && \forall (u,v)\in E_{S_b},\ \forall j\in L. \end{aligned} \quad (12)$$

This is simply a more general version of the dual (5), written for an arbitrary partition $V = (S_1,\dots,S_B)$. Here the consistency constraints are only relaxed for edges in E_∂ (boundary edges, which go from one block to another). The dual subproblems in the first term of (11) are LPs on each block, where the node costs of boundary vertices are modified by the block dual variables δ. For any δ, we can define the *reparametrized* costs θ_u^δ as

$$\theta_u^\delta(i) = \begin{cases} \theta_u(i) + \sum_{v:(u,v)\in E_\partial} \delta_{uv}(i) & \exists (u,v)\in E_\partial\\ \theta_u(i) & \text{otherwise}\end{cases},$$

so the block dual objective can also be written as

$$B(\delta) = \sum_b \min_{x^b} \left(\sum_{u\in S_b}\sum_{i\in L} \theta_u^\delta(i) x_u^b(i) + \sum_{uv\in E_{S_b}} \sum_{i,j} \theta_{uv}(i,j) x_{uv}^b(i,j) \right) + \sum_{uv\in E_\partial} \min_{i,j}(\theta_{uv}(i,j) - \delta_{uv}(i) - \delta_{vu}(j))$$

When there is only one block, equal to V, the block dual is equivalent to the primal LP (3). When every vertex is in its own block, the block dual is equivalent to the pairwise dual (4).

The following propositions allow us to convert between solutions of the pairwise dual (4) and the generalized block dual (11).

Proposition 1. *Let η^* be a solution to* (4). *Let δ^* be the restriction of η^* to the domain of B; that is, δ^*_{uv} is defined only for pairs uv, vu such that $(u,v) \in E_\partial$ or $(v,u) \in E_\partial$:*

$$\delta^*_{uv}(i) = \eta^*_{uv}(i) \quad (u,v) \in E_\partial \text{ or } (v,u) \in E_\partial$$

Then δ^* is a a solution to (11).

This proposition gives a simple method for converting a solution to pairwise dual P to a solution to the block dual B: simply restrict it to the domain of B. As we explain in Appendix B, this allows us to avoid ever solving the block dual directly; we simply solve the pairwise dual once, and can then easily form a block dual solution for any set of blocks.

Proof. It is clear that δ^* defined in this way is dual-feasible (there are no constraints on the δ's). We show that $B(\delta^*) \geq P(\eta^*)$. Let x be a primal LP solution. Because $B(\delta) \leq Q(x)$ for any dual-feasible δ (this is easy to verify), and $P(\eta^*) = Q(x)$ (Property 1), this implies $B(\delta^*) = Q(x)$. δ^* must then be a solution for the block dual B. Note that this proof also implies strong duality for the block dual.

To see that $B(\delta^*) \geq P(\eta^*)$, one could observe intuitively that B is strictly more constrained than P unless every vertex is its own block; since the subproblems are all minimization problems, the optimal objective of B will be higher. More formally, consider two adjacent nodes a and b in the pairwise dual P. The terms corresponding to a in b in P can be written as:

$$\min_{x_a} \sum_i \left(\theta_a(i) + \eta^*_{ab}(i) + \sum_{c:N(a)\setminus\{b\}} \eta^*_{ac}(i) \right) x_a(i)$$
$$+ \min_{x_b} \sum_i \left(\theta_b(i) + \eta^*_{ba}(i) + \sum_{c:N(b)\setminus\{a\}} \eta^*_{bc}(i) \right) x_b(i)$$
$$+ \min_{x_{ab}} \sum_{i,j} \left(\theta_{ab}(i,j) - \eta^*_{ab}(i) - \eta^*_{ba}(j)\right) x_{ab}(i,j),$$

where $N(u)$ is the set of vertices adjacent to u. The x terms written here do not appear in (4) because the minimum choice at a single vertex u can clearly be chosen by $x_u(i) = 1$ for a label i that minimizes the reparametrized potential, but we have left them in for convenience (under the constraint that $\sum_i x_u(i) = 1$).

By the convexity of min, the value of the objective above is at most

$$\min_{x_a, x_b, x_{ab}} \sum_i \left(\theta_a(i) + \eta^*_{ab}(i) + \sum_{c:N(a)\setminus\{b\}} \eta^*_{ac}(i) \right) x_a(i)$$
$$+ \sum_i \left(\theta_b(i) + \eta^*_{ba}(i) + \sum_{c:N(b)\setminus\{a\}} \eta^*_{bc}(i) \right) x_b(i)$$
$$+ \sum_{i,j} \left(\theta_{ab}(i,j) - \eta^*_{ab}(i) - \eta^*_{ba}(j)\right) x_{ab}(i,j),$$

Adding a new constraint to this minimization problem can only increase the objective value, so the value of the objective above is at most the value of:

$$\min_{x_a, x_b, x_{ab}} \sum_i \left(\theta_a(i) + \sum_{c:N(a)\setminus\{b\}} \eta^*_{ac}(i) \right) x_a(i)$$
$$+ \sum_i \left(\theta_b(i) + \sum_{c:N(b)\setminus\{a\}} \eta^*_{bc}(i) \right) x_b(i)$$
$$+ \sum_{i,j} \theta_{ab}(i,j) x_{ab}(i,j)$$

subject to the constraints $\sum_j x_{ab}(i,j) = x_a(i)$ for all i and $\sum_i x_{ab}(i,j) = x_b(j)$ for all j. Now the vertices a and b have been combined into a block. One can continue in this way, enforcing consistency constraints within blocks, until arriving at:

$$\sum_{u \in V} \min_i (\theta_u(i) + \sum_v \eta^*_{uv}(i)) + \sum_{uv \in E} \min_{i,j} (\theta_{uv}(i,j)$$
$$- \eta^*_{uv}(i) - \eta^*_{vu}(j)) \leq$$
$$\sum_b \min_{x^b} \left(\sum_{u \in S_b} \sum_{i \in L} \left(\theta_u(i) + \sum_v \eta^*_{uv}(i) \right) x^b_u(i) \right.$$
$$\left. + \sum_{uv \in E_b} \sum_{i,j} \theta_{uv}(i,j) x^b_{uv}(i,j) \right)$$
$$+ \sum_{uv \in E_\partial} \min_{i,j} (\theta_{uv}(i,j) - \eta^*_{uv}(i) - \eta^*_{vu}(j)),$$

where the minimizations over x^b on the right-hand-side are subject to the constraints (12). The left-hand side is $P(\eta^*)$. The expression on the right hand side is precisely the objective of $B(\delta^*)$, since we defined δ^* as the restriction of η^* to edges in E_∂. This completes the proof. □

Corollary 2 (Strong duality for block dual)**.** *If x is a primal solution and δ^* is a solution to the block dual, $B(\delta^*) = Q(x)$.*

So we are able to easily convert between a pairwise dual solution and a solution to the block dual. This will prove convenient for two reasons: there are many efficient pairwise dual solvers, so we can quickly find η^*. Additionally, we can solve the pairwise dual once and convert the solution η^* into solutions δ^* to the block dual for *any* block decomposition without having to recompute a solution. As we mentioned above, this will allow us to quickly test different block decompositions.

The following proposition allows us to convert a solution to the block dual to a pairwise dual solution.

Proposition 2. *Let δ^* be a solution to the block dual* (11). *Recall that each subproblem of the block dual is an LP of the same form as* (3). *So we can consider the pairwise dual P* defined on this subproblem. *For block b, let η^b be a solution to the pairwise dual defined on that block's (reparametrized) subproblem. That is,*

$$\eta^b = \max_{\eta} \sum_{u\in S_b} \min_i \left(\theta_u(i) + \sum_{v:uv\in E_{S_b}} \eta_{uv}(i) + \sum_{v:uv\in E_\partial} \delta^*_{uv}(i)\right) + \sum_{uv\in E_{S_b}} \min_{i,j}(\theta_{uv}(i,j) - \eta_{uv}(i) - \eta_{vu}(j))$$

Then the point η^* defined as

$$\eta^*_{uv}(i) = \begin{cases} \eta^b_{uv}(i) & (u,v)\in E_{S_b} \text{ or } (v,u)\in E_{S_b} \\ \delta^*_{uv}(i) & (u,v)\in E_\partial \text{ or } (v,u)\in E_\partial \end{cases}$$

is a solution to (4).

Given a solution δ^* to the block dual, we use Proposition 2 to extend it to a solution to the pairwise dual defined on the full instance; combining δ^* with pairwise dual solutions on the subproblems induced by δ^* and the block decomposition gives an optimal η^*.

Proof. This is immediate from strong duality of the pairwise dual and the block dual (Property 1 and Corollary 2, respectively). □

With this proposition, we are finally ready to prove Lemma 1.

Proof of Lemma 1. We are given a Potts instance (G,θ,w,L). Let δ^* be a solution to (11) with $S_1 = S$ and $S_2 = V\setminus S$. We know the sub-instance

$$((S,E_S),\theta^{\delta^*}|_S, w|_{E_S}, L)$$

is $(2,1)$-stable. Let g_S be the exact solution to the instance $((S,E_S),\theta^{\delta^*}|_S, w|_{E_S}, L)$. If g is the exact solution for (G,θ,w,L), g_S may or may not be the same as $g|_S$. For this Lemma, they need not be equal, and we just work with g_S. Because of the $(2,1)$-stability, Theorem 1 implies that g_S is the unique solution to the following LP:

$$\begin{aligned}
\min_{x^S} \quad & \sum_{u\in V}\sum_{i\in L}\theta_u^{\delta^*}(i)x_u^S(i) + \sum_{uv\in E}\sum_{i,j}\theta_{uv}(i,j)x^S_{uv}(i,j) \\
\text{s.t.} \quad & \sum_i x_u^S(i) = 1, && \forall u\in V,\ \forall i\in L \\
& \textstyle\sum_j x^S_{uv}(i,j) = x^S_u(i) && \forall (u,v)\in E,\ \forall i\in L, \\
& \textstyle\sum_i x^S_{uv}(i,j) = x^S_v(j) && \forall (u,v)\in E,\ \forall j\in L, \\
& x^S_u(i) \ge 0, && \forall u\in V,\ i\in L. \\
& x^S_{uv}(i,j)\ge 0, && \forall (u,v)\in E,\ i,\ j\in L.
\end{aligned}$$

This LP is simply the pairwise LP (3) defined on $((S,E_S),\theta^{\delta^*}|_S, w|_{E_S}, L)$. Strict complementary slackness (Property 3) implies that the pairwise dual problem defined on $((S,E_S),\theta^{\delta^*}|_S, w|_{E_S}, L)$ has a solution η^S that is locally decodable to g_S. That is, there is some η_S with

$$\eta^S = \max_{\eta} \sum_{u\in S} \min_i \left(\theta_u(i) + \sum_{v:uv\in E_S} \eta_{uv}(i) + \sum_{v:uv\in E_\partial} \delta^*_{uv}(i)\right) + \sum_{uv\in E_S} \min_{i,j}(\theta_{uv}(i,j) - \eta_{uv}(i) - \eta_{vu}(j))$$

and for all $u\in S$,

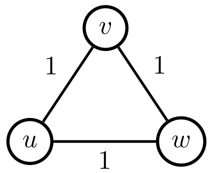
$$\arg\min_i \left(\theta_u(i) + \sum_{v:uv\in E_S}\eta^S_{uv}(i) + \sum_{v:uv\in E_\partial}\delta^*_{uv}(i)\right) = \{g_S(u)\}.$$

In other words, $g_S(u)$ is the unique minimizer of the modified node costs at $u\in S$. By Proposition 1, we can extend η^S and δ^* to a solution η^* to the pairwise dual (4) defined on (G,θ,w,L). This extended solution is locally decodable to g_S on S by construction. If x is a solution to the primal LP (3) defined on (G,θ,w,L), complementary slackness (Property 2) implies that $x_u(g_S(u)) = 1$ for all $u\in S$. That is, the LP solution x is equal to g_S on S. □

Nothing special was used about the block decomposition $(S, V\setminus S)$, and indeed Lemma 1 also holds for an arbitrary decomposition $(S_1,\dots S_B)$; if the instance restricted to a block S_b is $(2,1)$-stable after its node costs are perturbed by a solution δ^* to the block dual (11), the primal LP is equal on S_b to the exact solution of that restricted instance.

It is clear from Lemma 1 that if the solutions g_S to the restricted instances are equal to $g|_S$ (the exact solution to the full problem, restricted to S), the primal LP x is persistent on S (this is formalized in Corollary 1). This is why Theorem 2 requires that the restricted instance is stable with solution $g|_S$.

Proof of Theorem 2. Note that a block dual solution δ^* is a valid ϵ^*-bounded perturbation of θ by the choice



Node	Costs		
u	∞	0	ε
v	0	∞	ε
w	ε	0	∞

Figure 4: Instance where each node belongs to a block that is (∞, ∞)-stable when the node costs are *not* perturbed. The LP solution is fractional everywhere.

of ϵ^* and Definition 4. Because we have assumed in the statement of the theorem that the solution g_S to the restricted instance is equal to the restricted solution $g|_S$, the result follows directly from Lemma 1. □

A.2 Do we need dual variables?

A simpler definition for block stability would be that a block S is stable if the instance

$$((S, E_S), \theta|_S, w|_{E_S}, L)$$

is $(2,1)$-stable. Unfortunately, this is not enough to guarantee persistency. Consider the counterexample in Figure 4.

The optimal integer solution g labels u and w with label 2, and v with label 1, for a total objective of 2. The optimal LP solutions assigns weight 0.5 to each label with non-infinite cost, for a total objective of $\frac{3}{2}(1+\varepsilon) < 2$ for any $\varepsilon < \frac{1}{3}$. Define the block decomposition $S_1 = \{u\}$, $S_2 = \{v\}$, $S_3 = \{w\}$. Note that each block has a unique optimal solution given by the minimum-cost label, and that these labels match the ones assigned in the combined optimal solution g. Every vertex in this instance therefore belongs to an (∞, ∞)-stable block, according to the simpler definition, but the LP is not persistent anywhere. It is relatively straightforward to check that this instance does not satisfy Definition 5 or the conditions of Lemma 1.

A.3 Stable block size

Assume the pairwise dual solution η^* is locally decodable on vertex u to the label $g(u)$, where g is the exact solution. Then the reparametrized node costs $\theta'_u(i) = \theta_u(i) + \sum_{v \in N(u)} \eta^*_{uv}(i)$ have a unique minimizing label i. Now consider solving the block dual (11) when $S_u = \{u\}$ is a block with just one vertex, u. Around block S_u, $\delta^*_{uv}(i) = \eta^*_{uv}(i)$ is a solution to the block dual (see Proposition 1). But this means that S_u is a (∞, ∞)-stable block with the modified node costs (there are no edges to perturb, and the node costs have a unique minimizer). In this way, it is trivial to give a stable block decomposition any time the LP (3) is persistent on a node u—simply add u to its own block. However, it is not possible *a priori* to find stable blocks of size greater than one, and we show in Section 5 that many such blocks exist in practice. These practical instances therefore exhibit structure that is more special than persistency: large stable blocks are not to be expected from persistency alone, and their existence implies persistency.

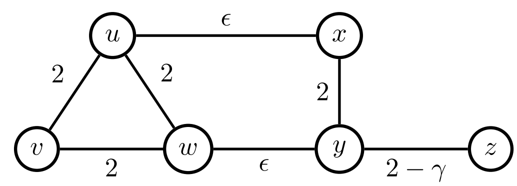


Figure 5: Potts model instance with both stable and tree structure.

Node	Costs		
	1	2	3
u	0	0	2
v	0	∞	∞
w	0	0	2
x	2	0	2
y	2	0	2
z	0	1	1

(a) Original node costs θ

Node	Opt. Label
u	1
v	1
w	1
x	2
y	2
z	2

(b) Exact solution g

Figure 6: Details for the instance in Figure 5. The strictly positive values ϵ and γ are both taken sufficiently small.

A.4 Combining stability with other structure

Consider the instance in Figure 5. The tables in Figure 6 give the original node costs θ and the exact solution g for this instance. The objective of g is $1 + 2\epsilon$. The pairwise LP (3) is persistent on this instance. How can we explain that? The instance is not $(2,1)$-stable: when the weight between y and z is multiplied by $\frac{1}{2}$, the optimal label for z switches from 2 to 1. However, if we take ϵ to be very small, the blocks $S = \{u, v, w\}$ and $T = \{x, y, z\}$ seem loosely coupled, and the strong node costs and connections in S suggest it might have some stable structure. Unfortunately, the block T is not stable for the same reason that the overall instance is not stable. However, this block is a tree!

It is fairly straightforward to verify that δ^* given by

$$\delta^*_{ux} = (\epsilon, 0, 0) \quad \delta^*_{xu} = (-\epsilon, 0, 0)$$
$$\delta^*_{wy} = (\epsilon, 0, 0) \quad \delta^*_{yw} = (-\epsilon, 0, 0)$$

is a solution to the block dual with blocks $\{S, T\}$. Indeed, Figure 7 shows the node costs θ^{δ^*} updated by this solution. If we solve the LP on each modified

Node	Costs		
	1	2	3
u	ϵ	0	1
v	0	∞	∞
w	ϵ	0	1
x	$2-\epsilon$	0	2
y	$2-\epsilon$	0	2
z	0	1	1

Figure 7: Updated node costs θ^{δ^*}

block, ignoring the edges between S and T, we get an objective of 2ϵ for S and an objective of 1 for T. Because this matches the objective of the original exact solution g, we know in this case that δ^* must be optimal for the block dual. It can then be shown that the modified block

$$((S, E_S), \theta^{\delta^*}|_S, w|_{E_S}, L)$$

is $(2, 1)$-stable: when all the weights of edges in E_S become 1 instead of 2, the solution is still to label u and w with label 1 for sufficiently small ϕ and ϵ. Similarly, the block

$$((T, E_T), \theta^{\delta^*}|_T, w|_{E_T}, L)$$

is a tree with a unique integer solution; because the pairwise LP relaxation is tight on trees (Wainwright and Jordan, 2008), this implies by Property 3 that there is a pairwise dual solution to this restricted instance that is locally decodable. Put together, these two results explain the persistency of the pairwise LP relaxation on the *full* instance by applying different structure at the sub-instance level.

B Experimental Details

In this appendix, we provide more details and additional discussion regarding the algorithms and experiments in Sections 4 and 5.

B.1 Explaining Algorithm 2

We briefly give more details on the steps of Algorithm 2. One key point is that we can efficiently compute block dual solutions with very little extra computation per outer iteration of the algorithm. We effectively only need to solve a dual problem once; we can then easily generate block dual solutions for any block decomposition for all subsequent iterations. In practice, we simply find a pairwise dual solution η^* using the MPLP algorithm (Globerson and Jaakkola, 2008), then use Proposition 1 to convert it to a solution of the generalized block dual (11) for a given decomposition.

Algorithm 3: `BlockStable`(g, β, γ) (optimized)

Given g, create blocks $(S_1^1, \ldots, S_k^1, S_*^1)$ with (9).
Initialize $K^1 = |L|$.
Find a solution η^* to (4).
for $t \in \{1, \ldots, M\}$ **do**
  Initialize $S_*^{t+1} = \emptyset$.
  Compute δ^* for $(S_1^t, \ldots S_{K^t}^t, S_*^t)$ using η^* and Proposition 1.
  Form $\mathcal{I} = ((V, E \setminus E_\partial), \theta^{\delta^*}, w|_{E \setminus E_\partial}, L)$ using δ^* and (7).
  Set $(f_1, \ldots f_{K^t}, f_*) =$ `CheckStable`(g, β, γ) run on instance $\mathcal{I}$.
  for $b \in \{1, \ldots, K^t, *\}$ **do**
    Compute $V_\Delta = \{u \in S_b^t | f_b(u) \neq g(u)\}$.
    Set $S_b^{t+1} = S_b^t \setminus V_\Delta$
    Set $S_*^{t+1} = S_*^{t+1} \cup V_\Delta$.
    if $b = *$ **then**
      Set $R = S_*^t \setminus V_\Delta$.
      Let $(S_{K^t+1}^{t+1}, \ldots, S_{K^t+p+1}^{t+1}) =$ `BFS`(R) be the p connected components in R that get the same label from g.
      Set $K^{t+1} = K^t + p$.
    end
  end
end

Additionally, we can avoid the expensive component of the inner loop of the algorithm (solving `CheckStable` for each block b). To parallelize `CheckStable` without any additional work, we modify the node costs of each block using the solution δ^* to the generalized block dual, then remove all the edges in E_∂. We can then solve the ILP (8) used in `CheckStable` with one "objective constraint" for each block. The objective function of (8) decomposes across blocks once E_∂ is removed. This approach avoids the overhead of explicitly forming and solving the ILP (8) for each block, which is especially helpful as the number of blocks grows large. These optimizations are summarized in Algorithm 3.

B.2 Object Segmentation

Setup: Markov Random Field

We use the formulation of Alahari et al. (2010). The graph G is a grid with one vertex for each pixel in the original image; the edges connect adjacent pixels. In this model, the node costs θ are set according to Shotton et al. (2006); they are based on the location of the pixel in the image, the color values at that pixel, and the local shape and texture of the image. Shotton et al. (2006) learn these functions using a boosting method. The edge weights $w(u, v)$ are set as:

$$w(u, v) = \lambda_1 + \lambda_2 \exp\left(\frac{-g(u, v)^2}{2\sigma^2}\right) \frac{1}{\text{dist}(u, v)},$$

where $g(u, v)$ is the RGB value difference between pixel locations u and v, and $\text{dist}(u, v)$ is the spatial distance between those pixel locations. We follow Alahari et al. (2010) and set $\lambda_1 = 5$, $\lambda_2 = 100$, and $\sigma = 5$.[3] This setup yields an instance of a Potts model (UNIFORM METRIC LABELING), so we can proceed with our algorithms. Many vertices of the object segmentation instances appear to belong to large stable blocks. Unlike with stereo vision, we were able to use the full instances in our experiments, which, as we observed in Section 5, could contribute to the quality of our results for segmentation. Each instance has 68,160 nodes and either five or eight labels. The LP is persistent on 100% of the nodes for all three instances.

B.3 Stereo Vision

Setup: Markov Random Field

To begin, we let the graph G be a grid graph where each node corresponds to a pixel in L. We then need to set the costs $\theta_u(i)$ for each u, i, and the weights $w(u, v)$ for each edge (u, v) in the grid. This is where the domain knowledge enters the problem. For a pixel u, we set its cost $\theta_u(i)$ for disparity i as:

$$\theta_u(i) = \left(I_L(u) - I_R(u - i)\right)^2. \tag{13}$$

Here I_L and I_R are the pixel intensity functions for the images L and R, respectively, and the notation $u - i$ shifts a pixel location u by i pixels to the left. That is, if u corresponds to location (h, w), $u-i$ corresponds to location $(h, w-i)$. If the difference (13) is high, then it is unlikely that pixel u actually moved i pixels between the two images. On the other hand, if this difference is low, disparity i is a plausible choice for pixel u. In our experiments, we use a small correction to (13) that accounts for image sampling (Birchfield and Tomasi, 1998); this correction is also used by Boykov et al. (2001) and Tappen and Freeman (2003).

We can set the weights using a similar intuition. If u and v are neighboring pixels and $I_L(u)$ is similar to $I_L(v)$, then u and v probably belong to the same object, so they should probably get the same disparity label. In this case, the weight between them should be high. On the other hand, if $I_L(u)$ is very different from $I_L(v)$, u and v may not belong to the same object, so they should have a low weight—they may move different amounts between the two images. To this end, we set

$$w(u, v) = \begin{cases} P \times s & |I_L(u) - I_L(v)| < T \\ s & \text{otherwise.} \end{cases}$$

In our experiments, we follow Tappen and Freeman (2003) and set $s = 50$, $P = 2$, $T = 4$. This setup gives us a Potts model instance (G, θ, w, L).

[3] We use pre-built object segmentation models from the OpenGM Benchmark that are based on the models of (Alahari et al., 2010): http://hciweb2.iwr.uni-heidelberg.de/opengm/index.php?l0=benchmark