Self-Occlusion and Disocclusion in Causal Video Object Segmentation Supplementary Material

Yanchao Yang[1], Ganesh Sundaramoorthi[2], and Stefano Soatto[1]
[1]University of California, Los Angeles, USA [2]King Abdullah University of Science & Technology (KAUST), Saudi Arabia
yyc8912@g.ucla.edu, ganesh.sundaramoorthi@kaust.edu.sa, soatto@ucla.edu

1. Extended Discussion

This section explores some subtleties and caveats that were not included in the main body for reasons of space limitations:

1. **Can regularization be used to determine motion in textureless regions? (L086)** By assuming spatial regularity and given that the region to which the textureless region belongs is known, motion can be assigned. However, in the problem that we wish to solve, it is not known a-priori to which region the textureless region belongs, and therefore, motion cannot be reliably assigned to the textureless region. Fig. 2 (in paper) shows a problem with a hypothesis test based on motion cues assigned in textureless regions.

2. **How does minimizing the energy lead to a MLE? (L197)** The energy E_{warp} comes from the following likelihood function:
$$p(x, I_t(x), I_{t+1}(w_i(x))|O_i, w_i) \propto \begin{cases} \exp\left[-\frac{(I_{t+1}(w_i(x))-I_t(x))^2}{2\sigma_n^2}\right] & x \in R_i \backslash O_i \\ \exp\beta & x \in O_i \end{cases},$$
where $I_{t+1}(w_i(x)) = I_t(x) + \eta(x)$, $\eta(x) \sim \mathcal{N}(0, \sigma_n^2)$ within $R_i \backslash O_i$, and we assume a uniform distribution within O_i. Assuming iid on the former distributions, we have
$$p(I_t, I_{t+1}|O_i, w_i) \propto \prod_{x \in R_i} p(x, I_t(x), I_{t+1}(w_i(x))|O_i, w_i),$$
and taking the minus log of above gives the energy, E_{warp}.

3. **Are Sobolev warps really diffeomorphisms? (L260)** Integrating a smooth (i.e., belonging to a Sobolev space) vector field produces a diffeomorphism [4]. The Sobolev warp is computed by integrating G_i, which is smooth within the region since it is the solution of the Poisson equation, and therefore smooth (i.e., belonging to an H^2 Sobolev space) [5]. Therefore, the Sobolev warp is a diffeomorphism from R_i to $w_i(R_i)$.

   The extension defined by solving the Laplace equation gives a continuous vector field on D, which is smooth in a small thickening of R_i by setup. By integrating the extended vector field over D as in Eqn. (8)-(9), we obtain a Lipchitz continuous warp on D whose inverse exists. The resulting warp is a diffeomorphism on a small thickening of R_i to a small thickening of $w_i(R_i)$, and Lipschitz on the entire domain D.

   The extension outside R_i is needed so that the gradient descent of the energy E_{seg} (Eqn. (25)) can be defined via a competition of regions. It is sufficient to extend the warp to a small thickening of R_i (via the Sobolev metric on a small thickening of R_i rather than the Dirchlet problem) so that the gradient descent of E_{seg} can be performed. The extension to all of D is simply performed for convenience in the implementation: one does not need to keep track of an evolving front.

4. **How are Sobolev warps "parameter independent"? (L287)** Algorithm 1 is formulated by the property of the Sobolev gradient shown in Eqn (5): that neither the translation nor the deformation component depends on α. This is because the PDE in Eqn. (7) is independent of α as is the translation Eqn (6). Taking $\alpha \to \infty$ implies that the warp evolves according to a translation that is independent of α. Thus, Algorithm 1 evolves the warp according to a translation until

the energy no longer decreases (when the translation is zero). At this point, the full gradient $G_i = \frac{1}{\alpha}\tilde{G}_i$ is a pure deformation. Evolving the warp does not depend on α since the factor $1/\alpha$ only changes the speed of evolution and not the final converged warp. Therefore, it suffices to choose $\alpha = 1$. After one step of the deformation, the translation step is repeated, and the whole process is iterated. This results in Algorithm 1 that is independent of α and decreases the energy E_{warp}.

5. **What are "forward" and "backward" warps w_i^f, w_i^b? (L385)** Note that w_i^f is the warp that maps the region from t to $t+1$ (warps back I_{t+1} to match I_t in region i) and w_i^b is the warp that maps the region from t to $t-1$ (i.e., warps I_{t-1} to match I_t). Note
$$\text{Res}_i^b(x) = (I_{t-1}(w_i^b(x)) - I_t(x))^2,$$
which was a typo in Eqn. (14). Computing w_i^b is done using Algorithm 1, but using I_{t-1} and I_t.

6. **How does the Motion ambiguity function (maf) work? (L398)** Motion cues are not used in the data cost if $\text{maf}(x)$ is 1 and used if 0. The maf is 1 if $\underline{\sigma}(x)$ is small (x is in a textureless region or is close to a textureless region in another region), or if the current residuals are all large (in which case the motion is unreliable). The threshold on $\underline{\sigma}(x)$ is inversely proportional to r' since the large balls have more samples and thus noise is expected to be mitigated by the averaging (in the computation of $\underline{\sigma}(x)$) and thus, we have a more stringent criteria since noise is mitigated. Fewer samples (when the ball size is small) implies that noise effects may not mitigated, and thus we would like a less stringent criteria.

7. **The "complementarity" of motion and appearance cues is used by many prior works.** While many approaches use some kind of *combination* of appearance and motion, in our approach the two *complement* each other, in the sense that one dominates when the other does not. This is unlike general linear combinations where both cues are weighted independently. We have found this approach to work better than similar ones where cues are weighted independently (and typically with fixed, not data-dependent weights) in the final energy. See Fig. 1 in this manuscript for additional experiments.

2. Computational Cost Analysis and Speed-Ups

Our implementation of the proposed algorithm can be greatly improved in terms of computational efficiency: Since 90% of the processing time is devoted to computation of the warps, our algorithm can benefit from a number of obvious speedups. These are for the most part customary in the field of numerical PDEs, and would not affect the overall performance of our proposed algorithm, other than its speed. For these reasons, they have not been implemented, but are listed below for reference:

1. **Solving PDE on Thickened Regions** We have implemented our algorithm by solving PDE's for $\tilde{G}$ and the extension (Eqn. (7)) on the entire domain D. As discussed in Remark 5, this is unnecessary, and the PDE only needs to be solved in a small thickening of R_i, which would speed-up computations, but require more sophisticated coding.

2. **Multigrid Solver** The PDE in Eqn (7) was performed using conjugate gradient, which has a convergence rate $N^{3/2}$ where N is the number of pixels in the image. Multigrid solvers have linear convergence rate, significantly speeding up the computation. Note that although the PDE to be solved is on a irregular domain, there are many methods that tailor the original multigrid algorithm to such an irregular domain (e.g., see [9] and references therein), and still retain the speed-ups.

3. **Parallelization of Region Warps** The warps w_i of each regions are independent of each other, and each warp computation can be parallelized. This involves a more sophisticated implementation that is beyond our scope here.

4. **Transport PDE Replaced by Semi-Lagrangian Method** The transport PDE in Eqn (9) (Line 279) was implemented with a standard upwinding scheme, which has a limited step size determined by the CFL conditions. Larger time steps are possible and have proven to lead to significant speed-ups with a semi-Lagrangian method [11, 2, 3] This would lead to an even more significant speed increase in our method since each iteration involves the solution of a PDE (Eqn. (7)). By increasing the time step, fewer calls to the PDE solver for Eqn. (7) are needed.

5. **Speed-Up of Backward Warp Computation** At frame t, the backward warp (see line 385) to $t-1$ is re-computed starting at the identity map. However, from time $t-1$, we have already computed the forward warp to t (and its inverse). We may use these inverses to initialize the backward warp at time t, which would save considerable time.

6. **Shape-Similarity Function Updates** Currently, after each iteration of lines 9-12 of Algorithm 2, the shape similarity term S_i is computed, and since this requires a ball integral at each x, this can be expensive. An easy speed-up is to only update S_i only after every 50 iterations or so.

3. Implementation details and discretizations

3.1. Poisson Equation for the Sobolev Gradient

We show how to discretize Eqn (7), the Poisson equation. The discretization of the Laplacian is

$$-\Delta\tilde{G}(x) = -\sum_{y\sim x}\tilde{G}(y) - \tilde{G}(x) = \tilde{F}(x), \tag{1}$$

where $\tilde{F} = F_i - \text{avg}(F_i)$, and $y \sim x$ indicates that y is a 4-neighbor of x. Discretizing the boundary condition $\nabla\tilde{G}(x)\cdot N = \tilde{G}(y) - \tilde{G}(x) = 0$, when $y \sim x$, $y \notin w_i(R_i)$, and substituting it above, we have that

$$-\sum_{y\sim x, y\in w_i(R_i)}\tilde{G}(y) - \tilde{G}(x) = \tilde{F}(x), \tag{2}$$

which can now be fed into any iterative solver (e.g., conjugate gradient or multigrid).

3.2. Discretization of the Transport Equation

We now describe the discretization of Eqn. (9) in the Sobolev gradient descent. Let $\phi^{\tau,0} : D \to \mathbb{R}$ denote the backward warp at time τ. Then Eqn. (9) is discretized using an up-winding difference scheme as:

$$\phi^{\tau_{i+1},0}(x) = \phi^{\tau_i,0}(x) + \Delta t\left(G_1^{\tau_i}(x)D_{x_1}[\phi^{\tau_i,0}, G_1^{\tau_i}, x] + G_2^{\tau_i}(x)D_{x_2}[\phi^{\tau_i,0}, G_2^{\tau_i}, x]\right) \tag{3}$$

where $\Delta t > 0$ is the time step,

$$D_{x_j}[\phi_{\tau_i}, G^j_{\tau_i}, x] = \begin{cases} D^+_{x_j}\phi_{\tau_i}(x) & \text{if } G_j^{\tau_i}(x) < 0 \\ D^-_{x_j}\phi_{\tau_i}(x) & \text{if } G_j^{\tau_i}(x) \geq 0 \end{cases} \tag{4}$$

where $D^+_{x_j}$ ($D^-_{x_j}$) denotes the forward (backward, resp.) difference with respect to the j^{th} coordinate, and $G^\tau(x) = (G_1^\tau(x), G_2^\tau(x))$ denotes the x and y components of the Sobolev gradient. The time step is chosen as

$$\Delta t < 0.5/\max_{x\in S, j=1,2}|G_j^{\tau_i}(x)|.$$

4. Additional Comparison to Direct Combination of Motion and Appearance

We provide more comparison on the examples shown in Fig. 3 in the paper. In particular, we show that our complementary data term (Eqn. (18)) in which the motion or appearance cues are chosen based on the motion ambiguity function, maf, is necessary. To this end, we add additional comparison to direct combination of motion and appearance cues in our method, where the maf is replaced with a constant weight (not spatially varying). See Fig. 1. The result for the optimal constant weight (the manually tuned weight with best segmentation performance) is shown and compared to using the maf. This shows that a direct linear combination of motion and appearance cues does not lead to as accurate segmentation as using the complementary motion and appearance achieved through the maf (our approach).

5. Additional Visual Comparison on FBMS-59

In Fig. 2, we provide additional visual comparison of our results to current state-of-the-art on FBMS-59 to extend Fig. 7 in the paper.

6. Additional Comparisons

We report further comparisons not included in the main paper for reasons of space limitations. We compare our method to [13] on the dataset introduced there. Note that [13] solves disocclusions with appearance cues and the method only applies to a single region. We also compare to Adobe After Effects 2013 (based on[1]) as well as [6]. Table 1 shows quantitative results, and Fig. 3 shows an example result, where disocclusions that have different appearance than the covisible region are present (e.g., the hand is disoccluded and the covisible region including the shirt sleves have different appearance).

Figure 1. *Rotating around an object. This experiment shows that the motion ambiguity function is necessary. The top rows show the results when a constant weight (and tuned to achieve the optimal result) is chosen between the motion cues and appearance cues in our data cost, f_i. No weight can be chosen to obtain as accurate results as our method (bottom rows), which uses the motion ambiguity function.*

	ours	[13]	[6]	Adobe Effects
Library	0.9579	**0.9654**	0.8926	0.9193
Fish	**0.9852**	0.9792	0.9239	0.9513
Skater	**0.9272**	0.9086	0.8884	0.6993
Lady	**0.9699**	0.9508	0.2986	0.8243
Station	**0.9445**	0.9216	0.5367	0.8258
Hobbit	**0.9807**	0.9335	0.7312	0.5884
Marple	**0.9217**	0.9186	0.6942	0.8013

Table 1. *Comparison of methods on a dataset used in [13]. All methods use manual annotation in the first frame. Evaluation in terms of F-measure (higher is better).*

7. Spawning New Regions

We illustrate an additional feature of our algorithm that was not discussed in the paper due to space. We illustrate that new regions are automatically spawned. This is important since new objects may come into sight or may begin to move after being stationary (and thus are not detected in the initialization). This is accomplished by adding a case to Step 6 in Algorithm 2 in the paper. Any part of the disocclusion with all $H_i(x)$ large, i.e., $x \in \mathbf{D}$ such that $\max_i H_i(x) > \tau$ (where τ is a threshold) is assigned to a new region. Fig. 4 shows some examples to illustrate this feature of our algorithm.

References

[1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics (TOG)*, 28(3):70, 2009. 3

[2] M. Beg, M. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *IJCV*, 61(2):139–157, 2005. 2

[3] D. R. Durran. Semi-lagrangian methods. In *Numerical Methods for Fluid Dynamics*, pages 357–391. Springer, 2010. 2

[4] D. G. Ebin and J. Marsden. Groups of diffeomorphisms and the motion of an incompressible fluid. *The Annals of Mathematics*, 92(1):102–163, 1970. 1

[5] L. C. Evans. Partial differential equations. graduate studies in mathematics. *American mathematical society*, 2, 1998. 1

[6] J. Fan, X. Shen, and Y. Wu. Scribble tracker: a matting-based approach for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1633–1644, August 2012. 3, 4, 5

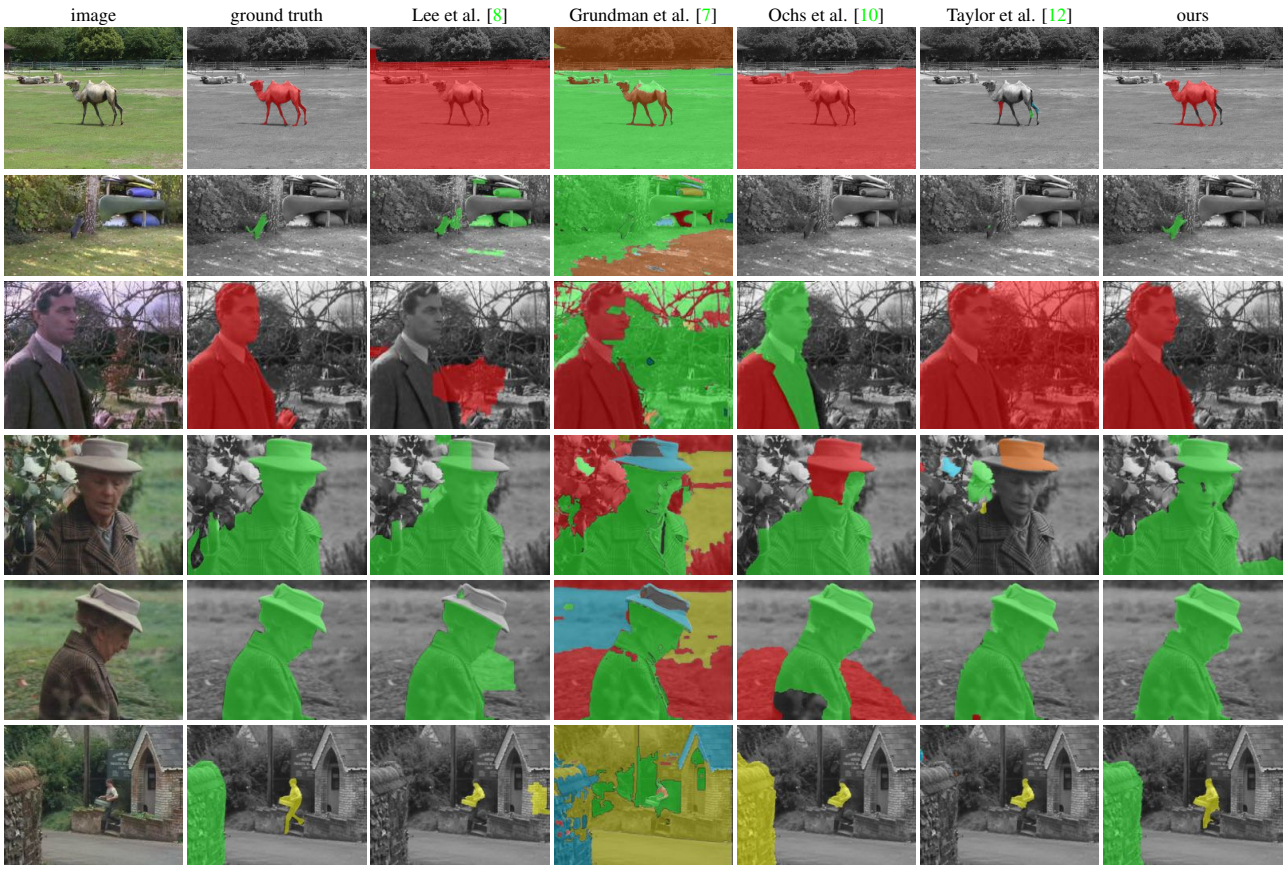

Figure 2. *Additional Visual Results on FBMS-59.*

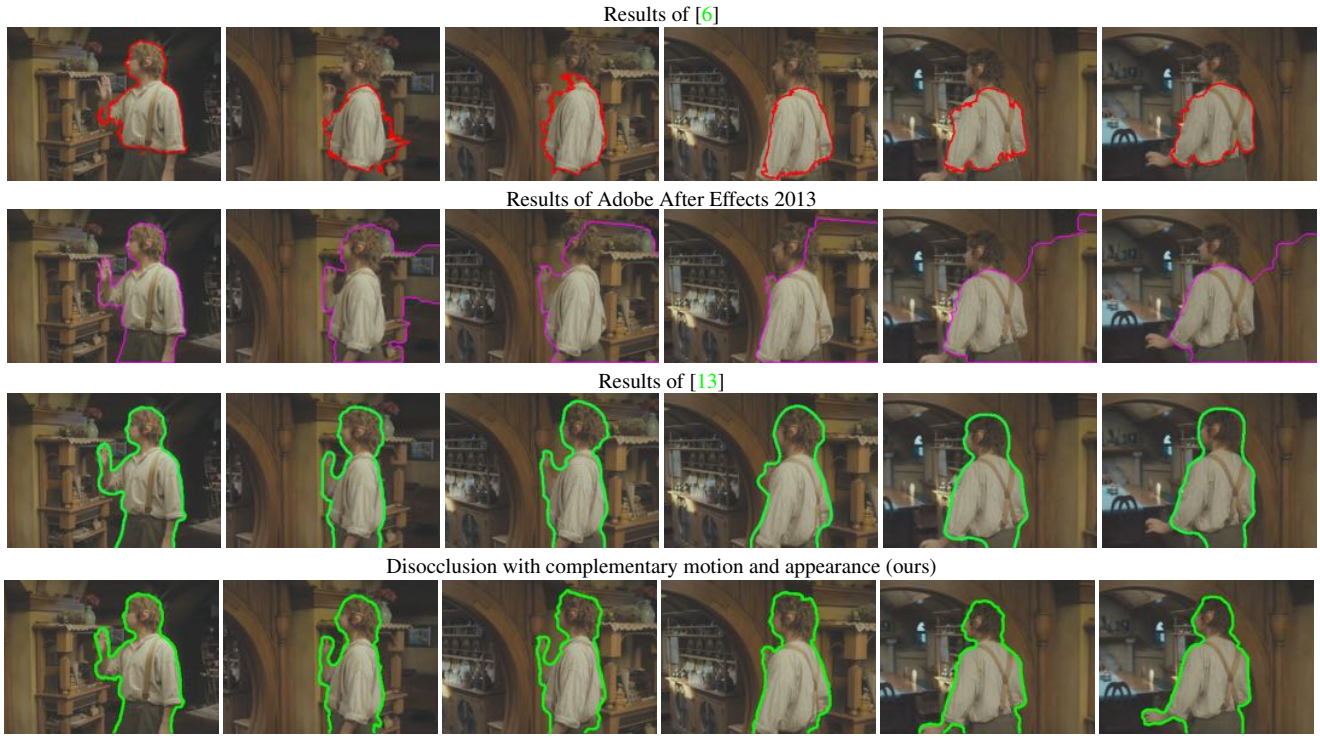

Figure 3. *Sample visual result in the dataset for Table 1. Notice that only our method is able to accurately capture and classify the disocclusion of different appearance than the covisible region (the left hand).*

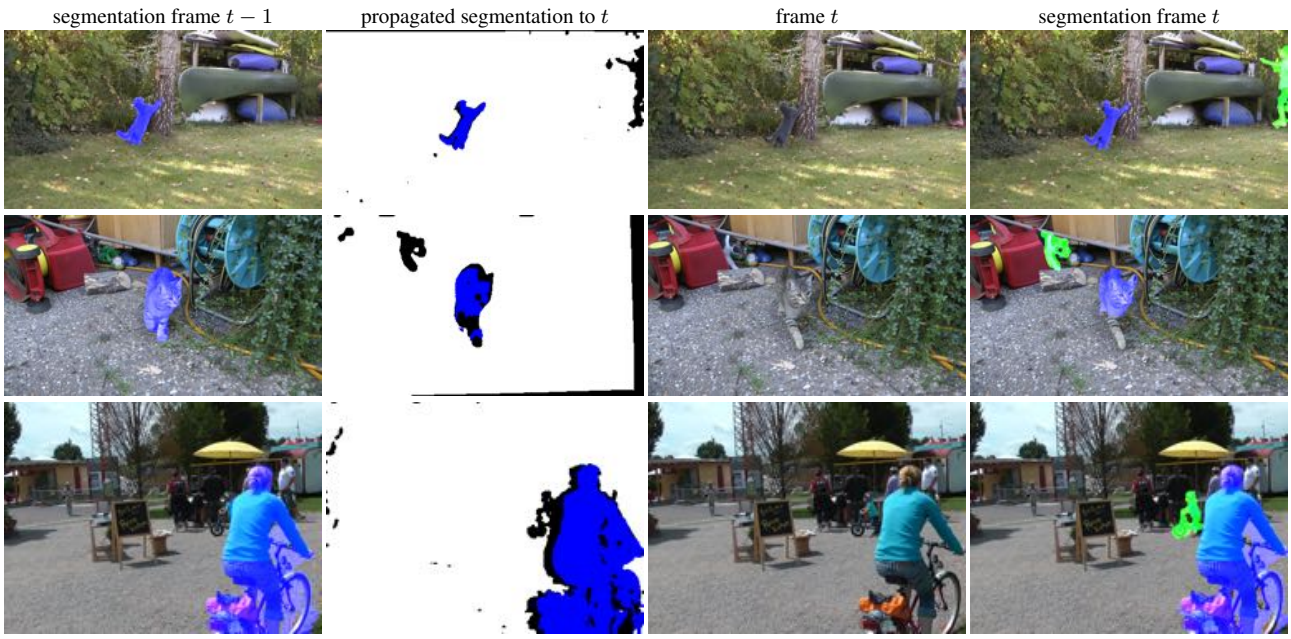

Figure 4. *Spawning new regions. Disoccluded regions (black) that do not fit existing regions in terms of motion or appearance (if motion is unreliable according to maf) automatically spawn a new region. Spawned regions are shown in green masks. This allows the number of regions to vary across frames.*

[7] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148. IEEE, 2010. 5

[8] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002. IEEE, 2011. 5

[9] A. McAdams, E. Sifakis, and J. Teran. A parallel multigrid poisson solver for fluids simulation on large grids. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 65–74. Eurographics Association, 2010. 2

[10] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 36(6):1187–1200, 2014. 5

[11] A. Staniforth and J. Côté. Semi-lagrangian integration schemes for atmospheric models-a review. *Monthly weather review*, 119(9):2206–2223, 1991. 2

[12] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*. IEEE, 2015. 5

[13] Y. Yang and G. Sundaramoorthi. Shape tracking with occlusions via coarse-to-fine region-based sobolev descent. *PAMI*, 37(5):1053–1066, 2015. 3, 4, 5