$$E(\mathbf{u},\mathbf{v},\mathbf{g},\theta)=\sum_{t=1}^{T-1}\Big\{E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1})+\sum_{k=1}^{K}\lambda_a(E_{\text{space}}^{\text{flow}}(\mathbf{u}_{tk},\theta_{tk})+E_{\text{space}}^{\text{flow}}(\mathbf{v}_{tk},\theta_{tk}))$$
$$+\sum_{k=1}^{K-1}\lambda_b E_{\text{space}}^{\text{sup}}(\mathbf{g}_{tk})+\lambda_c E_{\text{time}}(\mathbf{g}_{tk},\mathbf{g}_{t+1,k},\mathbf{u}_{tk},\mathbf{v}_{tk})\Big\}+\sum_{k=1}^{K-1}\lambda_b E_{\text{space}}^{\text{sup}}(\mathbf{g}_{Tk}). \quad (1)$$

overall energy function is given by Eq. (1) in which $\mathbf{u}_{tk},\mathbf{v}_{tk}$ are flow fields for each of the K layers at time t, and $\mathbf{g}_{tk}$ are *binary* support functions for the first $K-1$ layers (in contrast to the smooth support functions in [27]). We also associate every layer with an affine motion field $\mathbf{u}_{\theta_{tk}},\mathbf{v}_{\theta_{tk}}$ parameterized by θ_{tk}; we motivate this choice shortly.

As shown in Figure 4, we can determine binary visibility masks $\mathbf{s}_{tk}$ for each layer from $\mathbf{g}_{tk}$ by sequentially multiplying the support functions and their complements.

$$s_{tk}^p=\begin{cases} g_{tk}^p\prod_{k'=1}^{k-1}(1-g_{tk'}^p), & 1\le k<K\\ \prod_{k'=1}^{K-1}(1-g_{tk'}^p), & k=K,\end{cases} \quad (2)$$

where $p=(i,j)$ denotes a pixel at frame t. The visibility masks provide a segmentation of the scene into layers. Given the visibility mask, the occlusion reasoning (data likelihood term) is the same as in [27] $E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1})=$

$$\sum_{k=1}^{K}\sum_p\Big(\rho_d(\mathbf{I}_t^p-\mathbf{I}_{t+1}^q)-\lambda_d\Big)s_{tk}^p s_{t+1,k}^q, \quad (3)$$

where $q=(i+u_{tk}^p,j+v_{tk}^p)$ denotes the corresponding pixel at frame $t+1$, and ρ is a robust penalty function. Temporal consistency of the support functions, as aligned by the inferred flow field, is encouraged by an Ising MRF:

$$E_{\text{time}}(\mathbf{g}_{tk},\mathbf{g}_{t+1,k},\mathbf{u}_{tk},\mathbf{v}_{tk})=\sum_p(1-\delta(g_{tk}^p,g_{t+1,k}^q)).(4)$$

For non-integer flow vectors, sub-pixel interpolation introduces high-order temporal terms. We round these flow vectors to obtain an approximation with only pairwise terms. As shown in Figure 4, these temporally consistent support functions ensure the layer structures persist over time.

We capture the spatial coherence of the binary support functions by a conditional Ising MRF with weights determined by image color differences:

$$E_{\text{space}}^{\text{sup}}(\mathbf{g}_{tk})=\sum_p\sum_{q\in\mathcal{N}_p}w_q^p(1-\delta(g_{tk}^p,g_{tk}^q)). \quad (5)$$

Here the weight is defined as in the continuous formulation [27]. These spatially coherent support functions ensure the scene segmentation is spatially coherent and respects the local image evidence. Note that the visibility term in Eq. (2) implies high-order interaction terms among several layer-specific spatio-temporal Ising MRFs.

We model the motion of each layer by a pairwise MRF with a unary term. The energy term is $E_{\text{space}}^{\text{flow}}(\mathbf{u}_{tk},\theta_{tk})=$

$$\sum_p\sum_{q\in\mathcal{N}_p}\rho_{\text{mrf}}\,(u_{tk}^p-u_{tk}^q)+\lambda_{\text{aff}}\sum_p\rho_{\text{aff}}(u_{tk}^p-u_{\theta_{tk}}^p), \quad (6)$$

where the unary term encourages the flow field of each layer to be close to its affine flow (with weight λ_{aff}), and the affine motion is computed as in [27]. Note that this semiparametric model still allows deviation from the affine motion and is more flexible than parametric models. In automatically determining the number of layers, there is an important balance between Equations (5) and (6): the former penalizes support discontinuities, while the latter favors additional layers so that each layer's flow is closer to affine.

3.2. "Cooperative" Discrete Optimization Moves

Optimization of Equation (1) is challenging. A common strategy is to alternate the optimization of the support functions and the flow fields for each individual layer. Unfortunately, this approach is susceptible to local optima (see Figure 2). We thus develop optimization moves that can simultaneously change the flow fields and segmentation.

The standard moves of graph cuts are not directly applicable to the discrete model, because of the high-order interaction terms in the data term. We therefore define a set of "cooperative" moves that can *a)* change a group of pixels to be visible at a particular layer while also selecting their flow fields; *b)* change a group of pixels to be visible at a particular layer; *c)* select the flow fields of a particular layer from a candidate set. Each move solves a binary problem via the QPBO algorithm [10, 14], where the auxiliary binary variable, $\mathbf{b}$, encodes the states of several model variables. Next we explain the most complicated simultaneous segmentation and flow move and provide the details of other moves in the **Supplemental Material**.

Simultaneous segmentation and flow move. Sometimes a region may be assigned to a wrong layer with the correct motion. To escape this local optimum, we typically need to simultaneously change the segmentation and flow fields.

Consider a pixel p in frame t, for which layer k' is currently visible. We define a binary decision variable b_t^p such