

Layered Scene Decomposition via the Occlusion-CRF

Supplementary material

Chen Liu¹ Pushmeet Kohli² Yasutaka Furukawa¹

¹Washington University in St. Louis

²Microsoft Research Redmond

1. Additional experimental results

All additional experimental results are in figure 1, figure 2, figure 3, and figure 4. See the captions for the details.

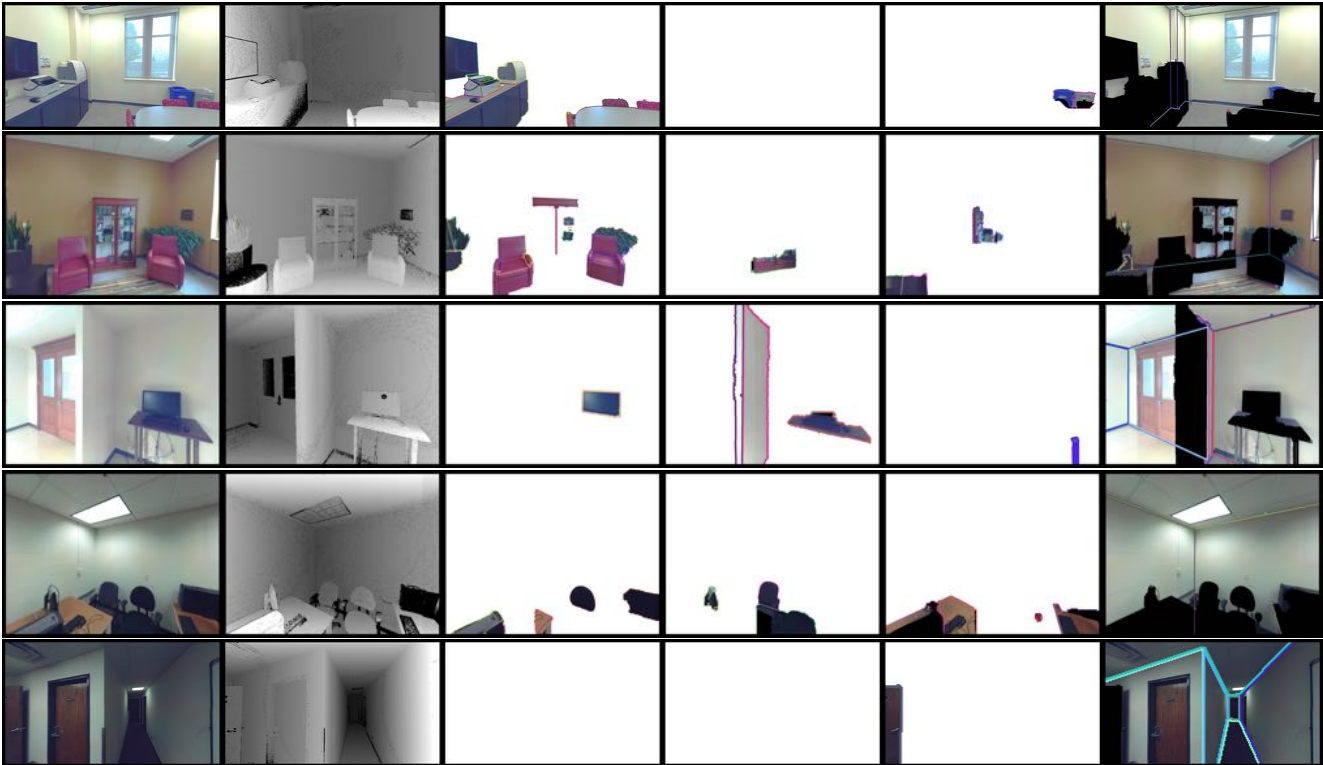


Figure 1. More experimental results on our datasets.



Figure 2. More experimental results on our datasets.

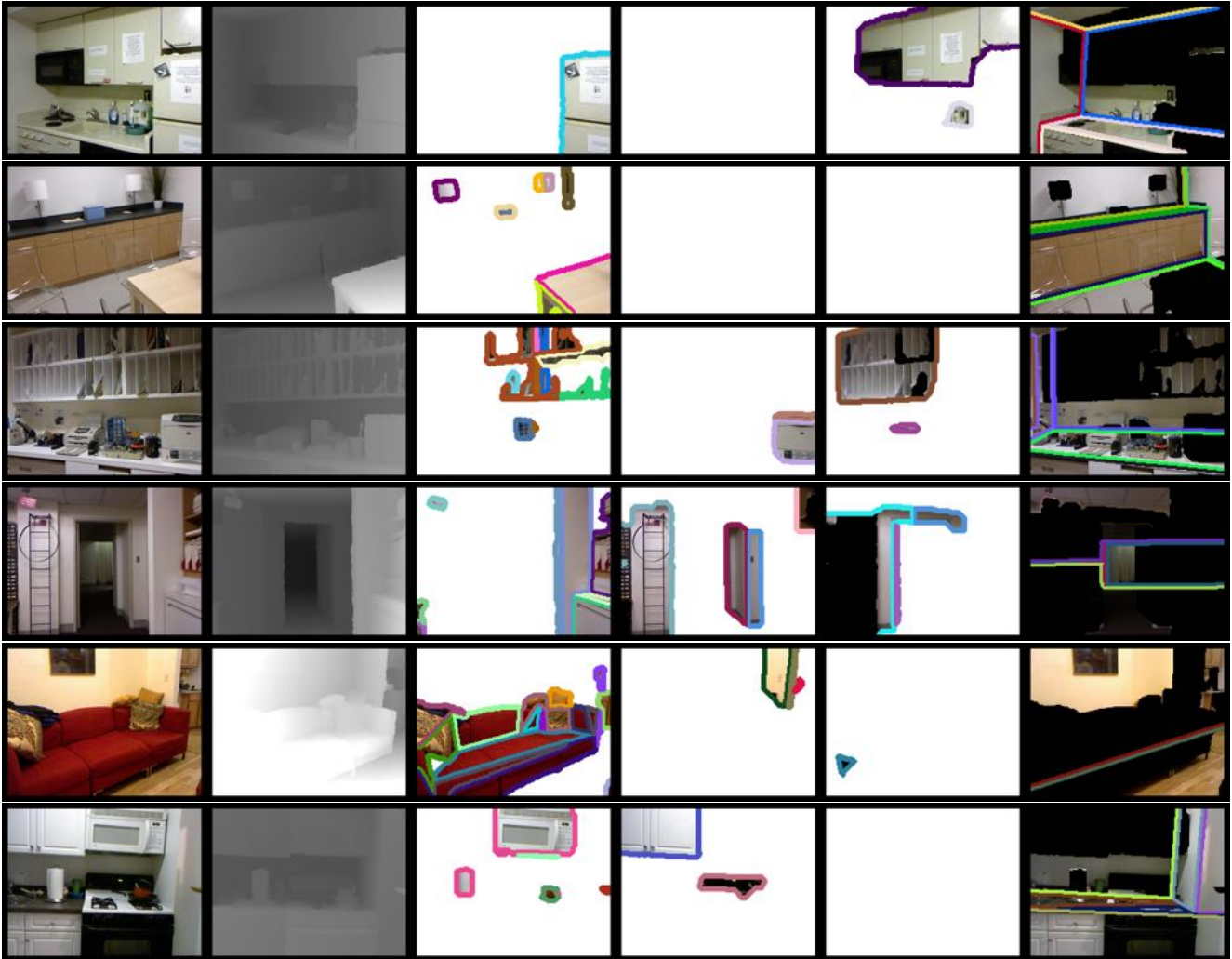


Figure 3. More experimental results on NYU V2 datasets.

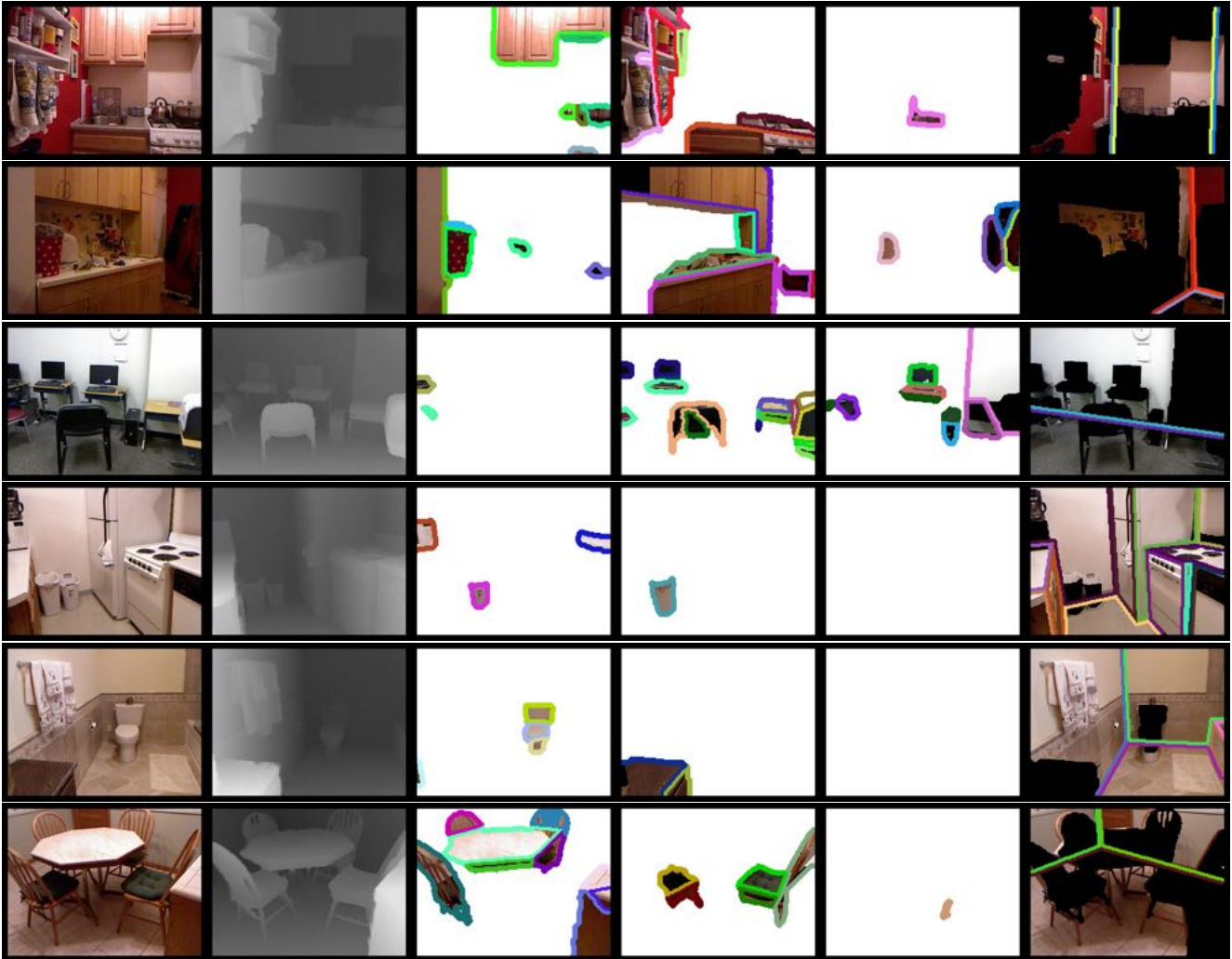


Figure 4. More experimental results on NYU V2 datasets.

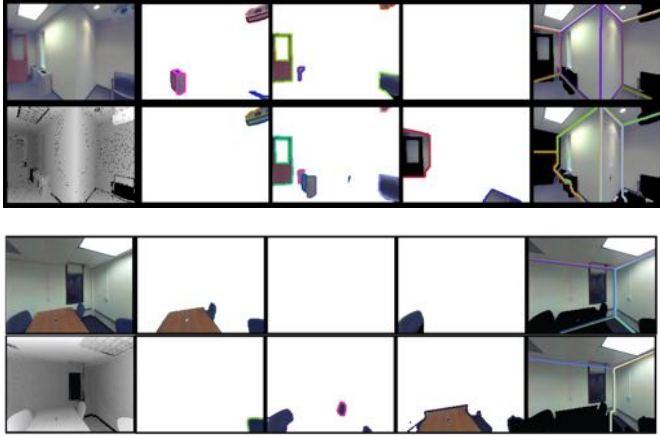


Figure 5. In the main paper, we have compared our Fusion Space (FS) algorithm against Fusion Move (FM) algorithm based on some statistics. This figure shows some qualitative evaluations on the the two methods. The first column shows the input image and the depth image. The remaining columns compares FS (top) and FM (bottom) methods.



Figure 6. Typical failure cases: (1) When there are many objects in a scene, our method might produce many small surfaces which fail to represent objects in a meaningful way. (2) Our method is capable of capturing most background structures accurately but may fail when the background is complex and contains windows, opening door or large free space. (3) Our method sometimes fails to capture small objects accurately, for example, the bottle on the table and the foot of the chair. (4) In some cases, our method produces many unnecessary surfaces due to the lack of balance between the data term and the MDL term.

2. Data term details

Our data term consists of the following four terms:

$$E_{data}(\mathcal{F}) = \sum_{p \in \mathcal{I}} \lambda_{depth} E_{depth}(f_p) + \lambda_{norm} E_{norm}(f_p) + \lambda_{color} E_{color}(f_p) + E_{order}(f_p).$$

We provide the details of the first three terms in this supplementary document. For the first depth term, we compute the difference between the model depth at the first non-empty layer and the input depth along the model surface normal. Let us denote this depth difference as d , then $E_{depth}(f_p)$ is set to $1 - \exp(-\frac{d^2}{2\sigma_d^2})$. $\sigma_d = 0.1m$ and $\lambda_{depth} = 2000$ is used throughout the experiments, while the unit of the depth difference is a meter. If the first non-visible vertex is at the background most layer, we tolerate a small amount of fitting error (0.05m) and use $\max(d - 0.05, 0)$ as the depth deviation to prefer smooth surfaces in the background.

The second normal term is simply the angle between the model normal and the input normal. $\lambda_{normal} = 200$.

The color term measures the likelihood of observing a pixel color given a color model for the segment. For each segment (when generated in a proposal), we train a mixture of Gaussian models with two clusters. To alleviate the effects of lighting and exposure variations, we convert RGB color into the HSV space and ignore the V channel. The color term is a truncated linear function of the negative log-likelihood with the truncation at 20. $\lambda_{color} = 10$ is used.

3. More proposal designing schemes

Single surface expansion proposal: This proposal seeks to expand a surface to more accurately explain input data, or inpaint invisible surfaces to minimize regularization penalties. Given a current surface segment, we have two mechanisms to specify the solution space. For each pair of a surface and a mechanism, we update the model. We would like to merge the solution space and update the model in much fewer steps, but the convergence becomes poor in our experiments. Given a surface $s \in \mathcal{S}$ in the model, the first mechanism simply allows this surface to be used in all the pixels in all the layers. The second mechanism seeks to expand this surfaces behind other surfaces in the same layer, while moving other surfaces to frontal layers. We allow other surfaces in the same layer to be at the same pixel locations in frontal layers, while allowing s to be at all the pixels in this layer.

Backward merging proposal: The first background hull proposal may fail in getting the proper background structure, and this proposal is specifically designed to improve the background layer by merging more surfaces. For every surface that is not in the background layer, we allow it at

the same pixels in the background layer, while excluding “impossible pixels” where the surface depth is in front of the input depth or behind the current background layer. We grow the region as before.

Structure expansion proposal: One limitation of the single surface expansion is that the proposal expands only one surface. This proposal seeks to expand two surfaces simultaneously, and is more powerful in recovering structures of large objects such as tables and chairs that consist of more than one surface. This proposal is conducted only in the middle layers, where large objects likely reside in.

We exhaustively pick a pair of surfaces in a middle layer that share a common boundary and are orthogonal to each other with a tolerance of 20° . We compute the score of each combination by using the metric used in the smooth hull proposal. We randomly pick a candidate pair with a probability proportional to the score. We grow each surface as before. We allow pixels in the layer to be either the current one or the predicted structure. We also allow current labels to be in frontal layers to move objects to the front as in the background hull proposal.

4. Proposal designing for Fusion Move method

We design proposals for the Fusion Move method (FM) in the same manner to make a fair comparison. The key difference is that for FM approach, only one new label is allowed for each pixel. So here we explain the modification required for each proposal.

Surface adding proposal: For surface adding proposal, we allow each surface to grow to cover more pixels in order to better find the extent of a surface. While for FM, a surface can not grow to a pixel if it is already occupied by another surface. So the extent of growth is limited.

background hull proposal: We first extract background hull in the same way. The difference is that for our FS approach, we can allow other surfaces to appear in any frontal layer or to be removed, but for FM approach, we cannot afford this because only one new label is allowed. So we always move other surface one layer front. This difference is critical as the lack of freedom makes frontal surfaces reluctant for moving.

Segment refitting proposal: For our FS approach, we can refit multiple new surfaces for one surface at the same time. And in our layer representation, different layers could choose either a refitted surface or the original one independently. While for FM approach, as only one new label is allowed, we refit one new surface for each surface and a pixel has to choose either using new surfaces for all layers or using original surfaces for all layers. The growth of surfaces is also limited as in segment adding proposal.

Single surface expansion proposal: For FM approach, there is only one mechanism allowed for single surface expansion proposal because of the label space limitation. That is the chosen surface is expanded in its layer and other surfaces in the same layer is moved one layer front.

Layer swap proposal: This proposal is limited by FM approach. Now each surface has only one new layer to choose. For surfaces in the first layer, they have the second layer as their option. For other surfaces, they could choose to be moved one layer front. Also, a pixel has to choose either moving all surfaces or not moving any surfaces.

Backward merging proposal: The limitation of FM has little impact for this proposal as surfaces could choose to be moved to the background layer or not. The difference is that the growth of surfaces in the background layer is limited as in segment adding proposal.

Structure expansion proposal: Similar with background hull proposal, we can expand a structure as we did for our FS approach, but we can only move other surfaces in the same layer one layer front, which severely affects the efficiency of this proposal.

In summary, for FM approach all these proposals are limited by the fact that only one new label is allowed for each pixel and become less effective for our layer decomposition task. As Table 1 in our main paper shows, for FM approach, the energy stays at a high state and cannot be further decreased by any proposal even though a much lower state is reachable by our FS approach.

5. Background hull construction algorithm

Given a set of planar surface segments in an image, we compute a “background hull” by their combinations.

Here each surface contains information such as 3D geometry parameters and the set of pixels it covers on image. So we can calculate the depth of pixel p if it is assigned to surface f , which we denote as d_p^f . We denote the set of image pixels f covers as \mathcal{P}^f . See Alg. 1 for the algorithm.

Note that this is a greedy algorithm based on a simple heuristic. The heuristic is that the convexity between two surfaces should be kept. The way we determine convexity is to extend each surface and check occlusion in the image region a surface covers. See Fig. 7. If two surfaces f^1 and f^2 form a convex structure, then the extended part of f^1 will occlude f^2 in the region covered by f^2 . Same for f^2 . While if two surfaces f^1 and f^2 form a concave structure, then their extended part is occluded by the other surface. So we determine convexity between two surfaces by counting how many pixels a surface covers are occluded by or occluding another surface’s extended part. Once we determine convexity between any pair of surfaces, for each pixel in \mathcal{R} , we iterate over all surfaces in \mathcal{S} . Each time, we

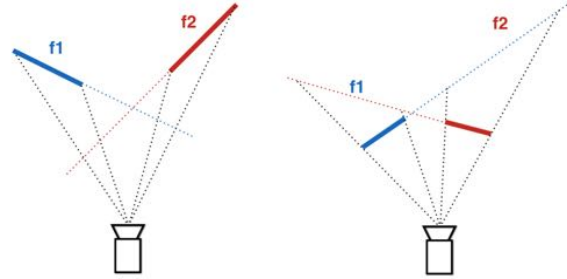


Figure 7. Left: Two surfaces are convex. The extended part of one surface is occluding the other surface. Right: Two surfaces are concave. The extended part of one surface is occluded by the other surface.

choose either the corresponding surface or the currently selected surface based on convexity between them. Note that this algorithm might not find the proper background hull in some extreme cases, but it works well practically in our experiment. And our FS method is robust against wrong background hulls. This algorithm is also used for structure construction applied in structure expansion proposal.

Algorithm 1: Background Hull Construction

input : A set of surfaces \mathcal{S} and the set of pixels \mathcal{R} in ROI

output: An assignment of \mathcal{S} to \mathcal{R}

```
for  $f^1$  in  $\mathcal{S}$  do
  for  $f^2$  in  $\mathcal{S} \setminus f^1$  do
     $Convexity[f^1, f^2] \leftarrow$ 
     $calcConvexity(f^1, f^2);$ 
  end
end
for  $p$  in  $\mathcal{R}$  do
   $f^h \leftarrow empty$  for  $f$  in  $\mathcal{S}$  do
    if  $f^h = empty$  then
       $f^h \leftarrow f;$ 
    else
      if  $Convexity f, f^h = true$  then
        if  $d_p^f > d_p^{f^h}$  then
           $f^h \leftarrow f;$ 
        end
      else
        if  $d_p^f < d_p^{f^h}$  then
           $f^h \leftarrow f;$ 
        end
      end
    end
  end
   $H[p] \leftarrow f^h$ 
end
return  $H;$ 
Func  $calcConvexity(f^1, f^2)$ 
   $num\_occluding\_pixels \leftarrow 0$ 
   $num\_occluded\_pixels \leftarrow 0$  for  $p$  in  $\mathcal{P}^{\infty}$  do
    if  $d_p^{f^2} < d_p^{f^1}$  then
       $num\_occluding\_pixels \leftarrow$ 
       $num\_occluding\_pixels + 1$ 
    else
       $num\_occluded\_pixels \leftarrow$ 
       $num\_occluded\_pixels + 1$ 
    end
  end
  for  $p$  in  $\mathcal{P}^{\epsilon}$  do
    if  $d_p^{f^1} < d_p^{f^2}$  then
       $num\_occluding\_pixels \leftarrow$ 
       $num\_occluding\_pixels + 1$ 
    else
       $num\_occluded\_pixels \leftarrow$ 
       $num\_occluded\_pixels + 1$ 
    end
  end
  return  $num\_occluding\_pixels ;$ 
   $num\_occluded\_pixels$ 
```
