15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Texture Super-Resolution for 3D Reconstruction

Calum Burns, Aurélien Plyer, Frédéric Champagnat
ONERA - The French Aerospace Lab
Chemin de la Hunière, 91120 Palaiseau
calum.burns@onera.fr

Abstract

We describe a method for producing a high quality texture atlas for 3D models by fully exploiting the information contained in dense video sequences via super-resolution techniques. The intrinsic precision limitations of multi-view reconstruction techniques are analysed and overcome. Compared to similar methods that rely on camera pose correction and geometry refinement, our subpixel image registration technique directly corrects the registration error in the image domain is much more efficient. We illustrate our method by enhancing the texture resolution of a 3D model produced by a state of the art reconstruction software.

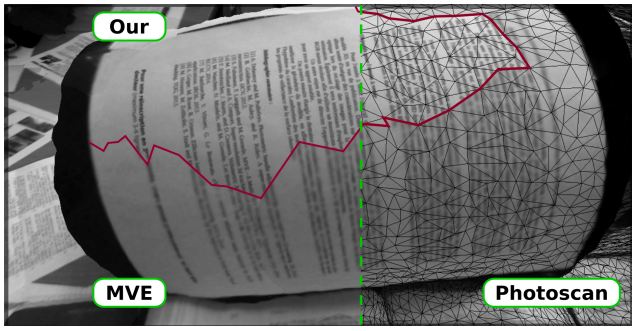

Figure 1. Sample of result of our TSR algorithm on a texture patch (in red), on a mesh built with Photoscan and textured with [12] from the MVE workflow.

1 Introduction

Research in 3D reconstruction has reached a decent level of maturity, indeed many commercial software products – such as those proposed by Agisoft and Pix4D – allow non-expert end users to generate quality large scale 3D models.

These reconstructions tend to rely on high quality sensors such as LIDAR and DSLR cameras mounted on a tripod that has to be moved around the scene. The acquisition protocols are impractical for large scenes with complex geometry, and make it hard for the operator to build a reliable 3D model. An interesting perspective for 3D reconstruction is video acquisition, particularly when the sensor is mounted on autonomous vehicles, such as drones [2]. Video acquisition also allows the operator to adjust scanning parameters, such as trajectory, in real-time by visualizing an initial approximate reconstruction of the scene. However, many state of the art reconstruction softwares are not designed to operate on dense video sequences, as the image-to-image combinatorial becomes computationnaly costly. The reconstruction can be estimated from a subset of the initial sequence, but in this case, much information remains unused.

We propose to use these extra images to enhance texture quality. Indeed, while often neglected, texture can be useful for a human operator wishing to detect fine details that are lost in the noise of the geometric reconstruction.

The main contribution of this paper is to exploit a full dense video sequence to augment the texture resolution of state of the art reconstruction software models, via video super-resolution (SR) methods [3, 4, 7].

1.1 Related work

To dissociate the texture resolution from the geometric resolution, many authors build a 2D texture space that parametrizes the 3D model and defines the intensity at each point of the surface. This is called a texture atlas. One way to build such an atlas is to use the input images [5][12], and thus the texture resolution is linked to the image resolution. The objective is to match each face of the polygon mesh with the input image best suited to provide texture information. In [12] the authors use a combinatorial optimization framework that chooses the best input image according to proximity, viewing angle and image quality, whilst also trying to agglomerate adjacent faces under the same label.

We aim to produce a texture atlas at the highest possible resolution. To this end we turn to video SR techniques [3, 4, 7]. The basic idea behind SR is to use the natural subpixel motion between low resolution (LR) frames in a video to increase the spatial resolution of the imaging process and produce a high resolution "SR image". Such SR effect can be obtained only if the inter-frame motion is estimated with sub-pixel accuracy. Extending these ideas to RGB-D sensors, Meilland and Comport [10] propose a visual SLAM technique that simultaneously super-resolves intensity image and depth-map based on a Kinect sensor.

Estimating high resolution textures for 3D reconstructed models is a non-trivial problem. Indeed, as stated by [6], "the geometric resolution of the model is usually well below the pixel resolution in a rendering". And despite the advances in multi-view reconstruction, the geometric precision is not sufficient to allow subpixel precision in the registration of the low resolution images.

Few works are devoted to texture super resolution for 3D models. Goldluecke *et. al.* [6] tackle the problem by simultaneously optimizing the super resolved texture, the camera parameters and a surface displacement map in a global framework. Their results are impressive, but the computational cost is high : 2h for a dataset of around 50 images on a 10cm large 3D model.

Maier *et. al.*[9] use super resolved keyframes to build high quality textures. The authors attribute a value to each pixel of their super resolved keyframes by looking

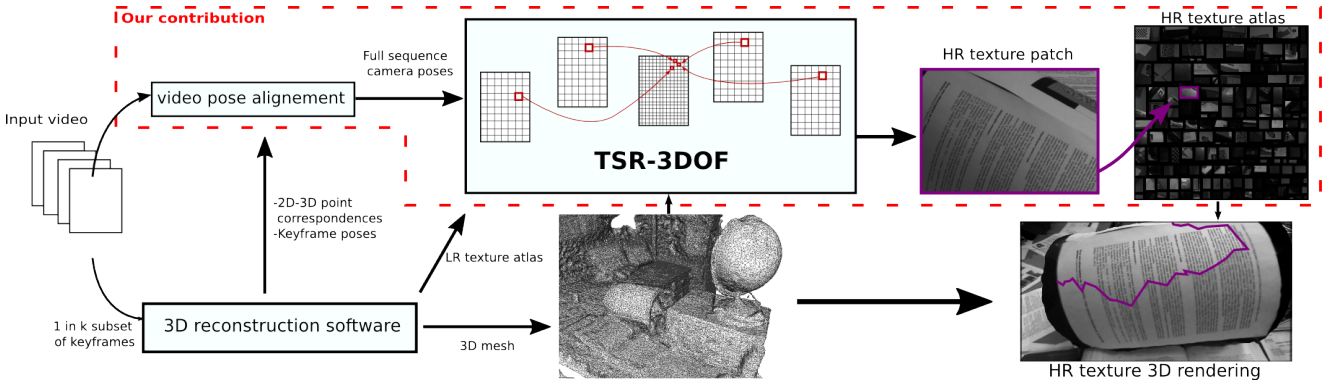

Figure 2. Pipeline overview

up all the corresponding pixel values in each neighboring LR image and computing a weighted median over these observations. This look-up procedure is based solely on the geometry of the 3D model and camera poses and is subject to the same geometric inaccuracies that affect multi-view reconstruction methods, the median filtering is expected to mitigate this imprecision.

1.2 Contribution and paper overview

In this paper we describe an original texture super-resolution (TSR) method for 3D multi-view reconstruction. Compared to previous approaches, we overcome the intrinsic precision limitations of 3D reconstruction algorithms by correcting for the image registration errors directly in the image domain, rather than in the 3D domain as in [6]. Correcting registration errors is performed in a cost efficient manner using a fast optical flow algorithm [11], this is in contrast with the costly displacement maps described in [6] or refined depth maps as in [9]. We demonstrate this SR method on a monocular video sequence by enhancing the texture of a state of the art 3D reconstruction software. In section 2, we describe the main blocs of the TSR reconstruction pipeline which produces a 3D mesh of the observed scene with a SR texture atlas built from the input video sequence. Section 3 discusses the results of the full TSR pipeline.

2 Texture super-resolution pipeline

The pipeline for producing a 3D model with a SR texture atlas from a monocular video is sketched in Fig. 2. We describe its essential parts in this section.

2.1 Baseline geometric and texture models

Our method builds upon a Multi-view stereo reconstruction software (MVSRS) such as Photoscan or MVE [12].

Such an MVSRS provides a 3D mesh model of the scene from a subset of video frames, the so-called *keyframes*. Temporal subsampling of the original video sequence is required because MVSRS do not scale well on dense videos.

The MVSRS also provides keyframe poses, that will be used in the building of *atlas textures*, it essentially consists of finding for each triangle face of the 3D mesh the label of the keyframe which is best suited for texture information. We use the Markov random field (MRF) energy formulation of [12] that seeks to promote the views with best image quality while preserving identical labeling of neighboring faces so as to minimize seems between atlas patches.

2.2 Texture super-resolution

We proceed with the remaining parts of our pipeline. Using an atlas, texture resolution is independent of the geometric resolution. We can thus enhance texture resolution without modifying our 3D mesh. Our goal is now to super-resolve keyframes using all the frames of the original video. For this purpose we need to register with subpixel accuracy each frame to its neighboring keyframe. Then the information from neighboring frames is fused to augment the resolution of the keyframe.

2.3 Image registration

Each pixel of each neighboring image needs to be registered to the closest keyframe with sub-pixel accuracy. This could be performed using an optical flow algorithm, but we can achieve better robustness and texture resolution using the depth cue provided by the 3D mesh. The first step is to estimate the pose of the remaining frames

2.3.1 Full sequence pose estimation

MVSRS provides 2D/3D correspondences between 2D keyframe features and 3D points on the mesh. We propagate these correspondences to neighboring frames with a KLT feature tracker [1]. Then 2D/3D correspondences feed a an iterative PnP algorithm based on Levenberg-Marquard optimization to estimate the neighboring camera pose. This process is made robust to outliers with fundamental matrix filtering of the tracked 2D points and RANSAC filtering of the 2D/3D correspondences.

2.3.2 Subpixel registration

Based on camera pose, each pixel from the neighboring image is projected onto the 3D mesh, then by backtracking the 3D point into the keyframe, we obtain the corresponding 2D point in keyframe image coordinates. However, this initial registration cannot be done with sufficient precisiondue to the piecewise affine surface approximation implied by 3D mesh. This approximation may lead to a 1 or 2 pixel registration error. Thus

camera pose and 3D mesh are unable to provide the registration accuracy required for SR.

In order to provide the refined pixel-to-pixel registration required by SR, Goldluecke *et. al.* [6] correct the error in the 3D domain, we find it is more efficient to correct the error directly in the image domain using optical flow estimation. We compute approximate displacements using the 3D model and camera pose, these displacements initialize an optical flow estimation between the neighboring image and the texture keyframe. To be able to handle large image datasets, we choose eFolki [11] for optical flow estimation. This multiresolution dense Lucas-Kanade algorithm favors computational efficiency whilst maintaining a good level of precision. To further improve the robustness of our optical flow refinement, we implement a "forward-backward" check as in [11]: we also estimate motion from the keyframe to the neighboring images, the sum of the forward and backward pixel motion should be null. In practice we set a threshold and eliminate the contribution of image pixels whose "forward-backward" motion sum exceeds the threshold.

2.4 Super-resolution

Once we have a subpixel registration for each image we construct the SR image by inversion of a direct image formation model (IFM). This IFM relates each LR image I_k^{LR} to the SR image I^{HR}. A general IFM is described in [3] as :

$$I_k^{LR} = DBW_k I^{HR} \quad (1)$$

Where B is the blur matrix, which accounts for the system's Point Spread Function (PSF), modeled here as a Gaussian function. W_k is the warp matrix that models pixel motion for image registration and D the decimation matrix, defined by the SR factor L.

In this work, we wish to implement a cost effective SR algorithm, in order to process large image databases. For this purpose, we use an approximation of the above IFM model as proposed in [4]:

$$I_k^{LR} = DW_k BI^{HR} \quad (2)$$

Indeed the underlying approximation that $BW_k = W_k B$ is only mathematically valid under translation motion, but Hardie *et. al.* [7] showed that in practice this model can be extended to more general motions. The main advantage is that the decimation matrix D is directly applied to W_k, which means that the inversion of this model only requires inter-frame motion in LR geometry. We can directly use our refined registration estimation. This avoids high memory requirements for storing HR warps and costly interpolation of our LR warps. Furthermore, as described in [4], this formulation allows us to separate the inversion of the IFM into two sub-tasks : a Shift&Add step that produces a blurred HR image followed by a deconvolution step.

The Shift&Add (S&A) step works as a vote strategy. Each LR pixel has been registered to HR texture image grid. We truncate the motion to the closest HR pixel. This first approximation produces a registration error that can be reduced by taking a large SR factor. Each HR pixel is assigned the mean value of all the LR pixels that have voted on it's coordinates. Some HR pixels will have received no votes, in which case their value is 0. We call this intermediate HR image the "shift-and-mean image" [4] that we denote $\widehat{I}_{HR}$.

The final step in reconstructing the SR image is deblurring. The following quadratic functional is minimized using conjugate gradient method:

$$E_{deconvQ}(I_{HR}) = \|\widehat{I}_{HR} - BI_{HR}\|_W^2 + \lambda\|\nabla I_{HR}\|^2 \quad (3)$$

Where W is a diagonal weight matrix, each diagonal entry of this matrix is the count of LR pixels that have been averaged in the HR pixel corresponding to this entry. This way, HR pixels of the shift-and-mean image are given more weight if they result from averaging of more pixels.

3 Results and discussion

For our experiments we scanned a 2m x 1m desk on which many textured objects are placed. The camera has a 5.5mm focal length lens working at $f/2.8$ mounted on a Bayer 1/18" e2v detector. For algorithmic simplicity we use monochromatic images. To do so we subsample the input images from 1600×1200 to 800×600 so as to obtain half of the green channel of the Bayer matrix. Our dataset consists of around 3200 images, 1 in 20 images of the full sequence is selected for geometric reconstruction. We use Agisoft's "Photoscan" reconstruction software to produce a 3D mesh of the scene.

The SR results shown here are obtained from 30 fused neighboring images. The SR factor L is set to 6. For the deblurring we use a Gaussian kernel with standard deviation $\sigma = 0.7$ LR pixels, and set the regularization parameter to $\lambda = 0.1$. These parameters where chosen empirically.

Figure 4 presents the 3D mesh and highlights two particular zones: a curved bin with text pasted on it, and a world globe. In figure 3 we compare the SR keyframe method described in[9] and our SR keyframe method with 3 image registrations techniques : registration with mesh and camera poses only ("TSR-3D"), registration with optical flow only ("TSR-OF") and refined registration as described in section 2.3 ("TSR-3DOF"). The LR interpolated images are shown at the far left. We first note that TSR-3DOF outperforms [9] and TSR-3D which both rely solely on 3D information for the image registration. This demonstrates the importance of our optical flow refinement. On the textured bin, TSR-3DOF and TSR-OF perform similarly, but TSR-3DOF outperforms TSR-OF on the world globe. This is further demonstrated in Fig. 5 which shows the number of image pixels correctly registered into the keyframe for each method, according to the forward-backward check (see section 2.3.2). Indeed we see that for the textured bin, the number of correctly registered pixels is similar. But for the world globe, initializing optical flow estimation with 3D information enables the registration of much more pixels. We believe that the optical flow alone fails due to the pronounced 3D aspect of the globe and to the specular reflections on it's surface. Finally, we note that TSR-3DOF is significantly more detailed than the interpolated LR image. On a single texture patch, the runtime of our image registration process is around 3mn for fusion of 30 images, and the deblurring step takes around 3.5mn for a 1200×800 portion of HR keyframe. Our texture SR is thus faster than methods

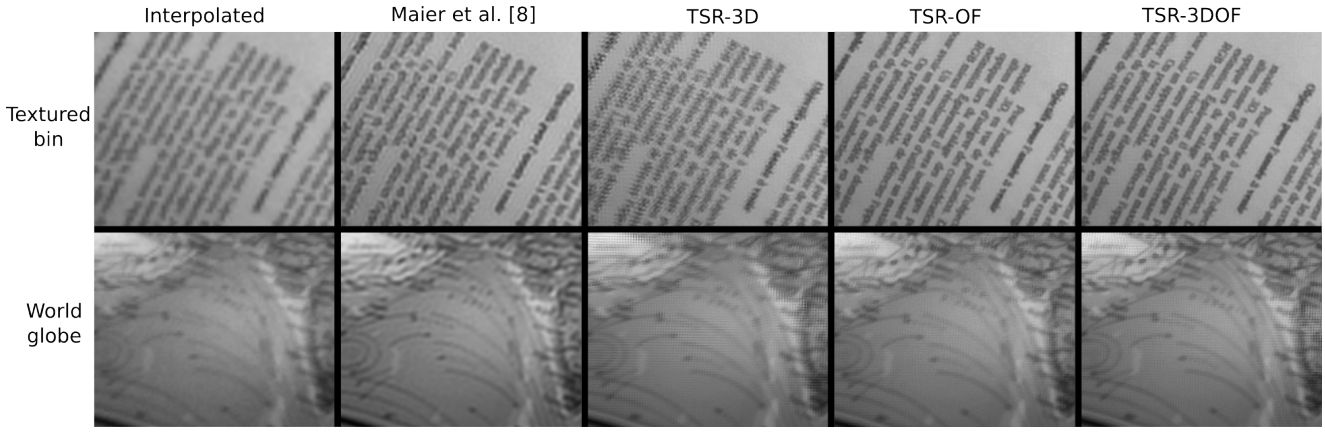

Figure 3. SR keyframes on portion of textured curved bin (top) and world globe (bottom). From left to right : bilinear interpolated LR image, SR fusion by Maier et al. [9], SR with 3D only registration, SR with optical flow only registration, SR with our 3D refined by optical flow method.

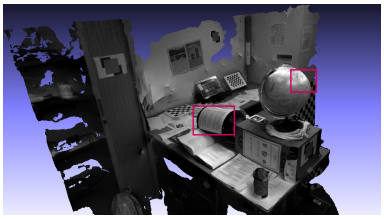

Figure 4. 3D mesh reconstructed with Photoscan

such as [6]. Furthermore we use non optimized Python code, runtimes can be greatly improved by using GPU implementations of optical flow and deconvolution such as those proposed in [11].

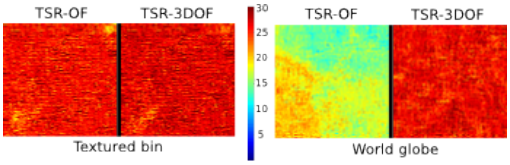

Figure 5. Number of image pixels correctly registered into the keyframe.

4 Conclusion

In this paper we propose an original TSR method that enhances the texture resolution of 3D models produced by current state of the art reconstruction softwares. To do so we fully exploit the information contained in a dense monocular video sequence. Previous approaches to TSR rely on refining the 3D geometry of the reconstruction to overcome the intrinsic precision limitations of multiview reconstruction algorithms. This re-optimization can become very computationally costly if one wishes to reach a precision that satisfies the requirements of current SR techniques. We show that the required accuracy for SR can be obtained by correcting the initial geometric registration error in a cost efficient manner directly in the image domain with an optical flow algorithm. In future work we wish to quantify the gain in resolution induced by our method by generalizing such measures as defined in [8] to the context of 3D reconstruction. We also wish to bypass the limitations induced by specularities by integrating illumination models that may allow us to estimate high resolution albedo and normal textures.

References

[1] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. Intel Corporation, Microprocessor Research Labs, 2000.

[2] S. Daftry, C. Hoppe, and H. Bischof. Building with drones: Accurate 3D facade reconstruction using mavs. CoRR, abs/1502.07019, 2015.

[3] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. IEEE Transactions on Image Processing, 6(12):1646–1658, 1997.

[4] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. Image Processing, IEEE Transactions on, 10(8):1187–1193, August 2001.

[5] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. Seamless Montage for Texturing Models. Computer Graphics Forum, 2010.

[6] B. Goldluecke, M. Aubry, K. Kolev, and D. Cremers. A super-resolution framework for high-accuracy multiview reconstruction. 2014.

[7] R. C. Hardie and K. J. Barnard. Fast super-resolution using an adaptive Wiener filter with robustness to local motion. Opt. Express, 20(19):21053–21073, Sep 2012.

[8] S. Landeau. Evaluation of super-resolution imager with binary fractal test target. volume 9249, pages 924909–924909–16, 2014.

[9] R. Maier, J. Stueckler, and D. Cremers. Super-resolution keyframe fusion for 3D modeling with high-quality textures. In International Conference on 3D Vision (3DV), 2015.

[10] M. Meilland and A. Comport. Super-resolution 3D tracking and mapping. ICRA, 2013.

[11] A. Plyer, G. Le Besnerais, and F. Champagnat. Massively parallel Lucas Kanade optical flow for real-time video applications. Journal of Real-Time Image Processing, 2014.

[12] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! Large-scale texturing of 3D reconstructions. In ECCV 2014, pages 836–850. Springer International Publishing, 2014.