High Resolution 3D Shape Texture from Multiple Videos

Vagia Tsiminaki, Jean-Sébastien Franco, Edmond Boyer

High Resolution 3D Shape Texture from Multiple Videos

Vagia Tsiminaki Jean-Sébastien Franco Edmond Boyer
Inria Grenoble Rhône-Alpes, LJK - Grenoble Universities, France
first.last@inria.fr

Abstract

We examine the problem of retrieving high resolution textures of objects observed in multiple videos under small object deformations. In the monocular case, the data redundancy necessary to reconstruct a high-resolution image stems from temporal accumulation. This has been vastly explored and is known as image super-resolution. On the other hand, a handful of methods have considered the texture of a static 3D object observed from several cameras, where the data redundancy is obtained through the different viewpoints. We introduce a unified framework to leverage both possibilities for the estimation of an object's high resolution texture. This framework uniformly deals with any related geometric variability introduced by the acquisition chain or by the evolution over time. To this goal we use 2D warps for all viewpoints and all temporal frames and a linear image formation model from texture to image space. Despite its simplicity, the method is able to successfully handle different views over space and time. As shown experimentally, it demonstrates the interest of temporal information to improve the texture quality. Additionally, we also show that our method outperforms state of the art multi-view super-resolution methods existing for the static case.

1. Introduction

Gathering appearance information of objects through multi-camera observations is a challenging problem, of particular interest for multi-view capture systems. In such systems, typically, a geometric model is reconstructed, tracked or refined to be as close as possible to reality. Adding an appearance or texture layer to this geometric information plays an essential part in the realism of the result, and is often more important than geometric detail to convey the object's visual aspect. Applications of this acquisition pipeline, such as broadcast, special effects or entertainment, among others, are very highly demanding in terms of quality. Yet, even with state of the art multi-camera studio equipment, simply reprojecting texture from any one of the high resolution video streams used in the acquisition process is not enough to guarantee good texture coverage and high quality renderings or close-ups. Because several such input video streams are available in this context, and in order to take advantage of all the information they carry, we naturally turn to the various sources of data redundancy to boost texture quality. Following 2D monocular super-resolution techniques that successfully regain details from low resolution images, we consider here a similar framework for multi-viewpoint videos.

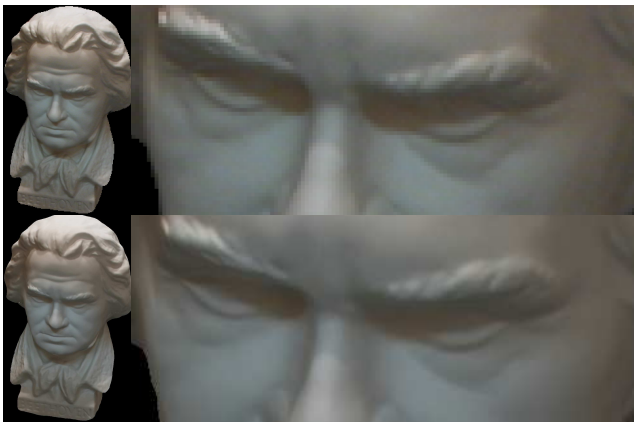

Figure 1. Input view 768×576 resolution with up-sampling by factor of three, BEETHOVEN dataset. Super-resolved 2304×1728 output of our algorithm rendered from identical viewpoint.

Such a framework is however significantly different from 2D super-resolution. First, dealing with multiple video streams is a different problem than using only one, where little parallax is usually assumed to occur. In a multi-view scenario, the intrinsic appearance of a single 3D object is only partially visible in each view, and observed only after being perspective projected, distorted by 3D geometry, and self-occluded. The 3D geometry itself is subject to reconstruction error and thus uncertain. Seamlessly blending and super-resolving the different input contributions into one single coherent texture space, while accounting for all such sources of variability is thus quite challenging. In fact it has only recently started to be addressed as such [10] for static objects.

But to fully exploit data redundancy, temporal accumulation of all views also needs to be examined. Not only is it an additional source of data, but interestingly temporal accumulation might make it possible to obtain high quality results with a sparser set of viewpoints than in the static case. This is not without its own source of difficulties. More often than not the subjects of interest are of arbitrarily deformable nature, such as human actors. This means that consistent temporal accumulation of texture data can only be done by realigning the relevant parts of the texture from one temporal frame to another, and accounting for sources of geometric variability. Fortunately, recent progress in non-rigid surface tracking methods [3] offer a path to resolve such issues, which we open with this work.

Overview. Generalizing existing multi-view appearance super-resolution work to the temporal domain requires a robust model of variability. As the appearance of subjects may drastically change over the long run, we focus in this paper on small non-rigid motions of the subject around a stable pose and observed appearance. We propose to deal with the largest non-rigid motion component using a surface-tracking method [3], and to compensate for any remaining geometry perturbations with a per-view, per-time frame warp. This per-view registration popularized in various rendering techniques [20, 8] has the large advantage of uniformly dealing with all sources of error, calibration, reconstruction, temporal misalignments and ghosting for our texture super-resolution, and is one of the major contributions of the paper. This paper is also the first, to our knowledge, to deal both with multiple viewpoints and temporal frames to build one common super-resolved texture, as opposed to [21] which enhance the input views directly, and [10] which only deals with the static multi-view aspect. Warping is done on an intermediate, high-resolution projected proxy of the model texture, where variability can be appropriately densely compensated (§3.1). We also expose a straightforward model and algorithm for this task, illustrated in Fig. 2. We notably show that some linear models [17] of the image formation can be generalized to the multi-view, multi-frame case (§3), as well as the monocular noise models (§4). We exhibit a two-stage iterative algorithm (§5), whose convergence is illustrated in experiments (§6). Our validation protocol also includes favorable comparison with the closest state of the art method [10], at the intersection of the validity domains of the methods (static, multi-view texture resolution case). Furthermore, we quantitatively demonstrate the convergence and temporal improvement of our method over using the same number of views in the static case.

2. Related Work

View-Dependent Texturing. Various strategies exist to retrieve and render the appearance of objects from input views and given a viewpoint, a geometric reconstruction being assumed available in general. One of the first proposed is to reproject and blend view contributions according to visibility and viewpoint-to-surface angle [7]. View-dependent techniques have been generalized to model and approximate the plenoptic function for the scene object, capturing view dependent shading effects [2] but this requires many dense views. Imperfect proxies and other geometric errors create rendering misalignments (ghosting), which various techniques correct with an additional image-space registration step [8], building a local basis of appearance variability [5], or refining the geometry proxy [19]. By nature, these methods are not targeted to capture intrinsic, view-independent texture properties and generally do not exploit viewpoint redundancy to super-resolve visual quality, nor do they easily extend to the time domain for deformable objects as proposed.

Multi-View Texture Estimation. To store intrinsic details of the acquired object and later render them, numerous methods build an image-based texture atlas to store appearance information, where each texel blends contributions from each view. Realignment is often proposed again to avoid ghosting [20, 15], but a second strategy exists which instead builds the texture as a mosaics of unique-view contributions, whose seam locations are optimized to minimize appearance change between fragments [14]. Interestingly, this strategy was extended to the temporal domain [13]. Only a handful of particularly relevant works examine how to super-resolve fine appearance detail from viewpoint redundancy at a single time frame [12, 10]. We propose an improved, unified model to deal with geometric variability due to reconstruction error and small deformation across time for multi-view super-resolution.

Video Super-Resolution While very few works exist concerning super-resolution techniques applied in a multi-view context, the problem has been extensively studied in the monocular case. The image formation model is well identified, as a geometric warping, blurring and subsampling process of the initial high-resolution image [1]. Two features of particular interest to us are that this model can be represented by a stack of linear transforms, and that Bayesian models have been developed to explicit the noise dependencies and priors over the target image and estimated warps [9, 17]. L1-norm based priors and total variation (TV-)minimization are increasingly popular [17] for their image restoration qualities. Notably, super-resolving multiple videos of a moving subject was examined in a performance capture context, but only for the input viewpoints [21]. Our model proposes temporal and multi-view super-resolution, yet super-resolves a single, intrinsic appearance map which can be re-used to render new viewpoints.

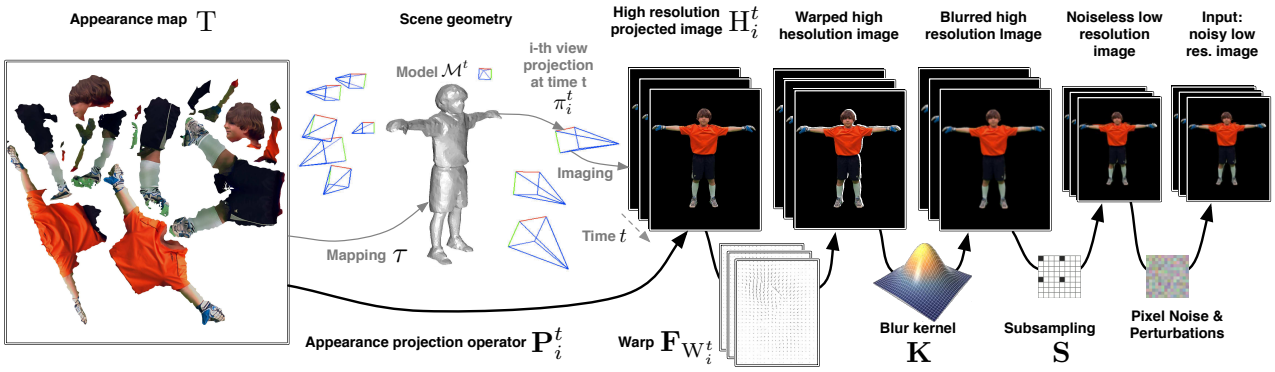

Figure 2. Summary of image formation model and problem notation.

3. Image Formation Model

Our goal is to estimate an appearance map T of an object of interest from a set of input color images $\{\mathrm{I}_i^t\}$, where $i \in \{1, \cdots, n_i\}$ is the camera number and $t \in \{1, \cdots, n_t\}$ the time. We assume a temporally coherent mesh model of the object, *i.e.* whose connectivity is time independent but of varying pose $\{\mathcal{M}^t\}$, obtained by tracking the surface tracking [3].

3.1. High Resolution Projection

We project the texture to a high resolution (HR) image $\{\mathrm{H}_i^t\}$ for each viewpoint $\{i, t\}$. Before reaching H_i^t, the texels of T undergo two geometric transformations.

Texture Mapping. For the appearance to be mapped to the mesh, a geometric mapping function must map each texel of T to the mesh surface. Thanks to fixed connectivity of the mesh across time, only one such function τ needs to be defined. Conformal mappings are preferred, because preservation of angles ensure low distortion during the transfer, such that the texel density of T is kept homogeneous on the 3D surface. Note that due to potential cuts and non-zero genus topology of the objects of interest, τ may not be continuous and may have a support region with several connected components (or charts) in the texture domain. To obtain τ, we use [18], which yields large charts with relatively few components, a useful feature for regularization and to avoid continuity artifacts.

Projection to High Resolution Image. We assume projections $\{\pi_i^t\}$ are known for each view i and time t. A texel at texture location x is mapped to a geometric point $\tau(x)$ on model $\mathcal{M}^t$, this point is then projected in view $\{i, t\}$ at point $\pi_i^t \circ \tau(x)$. This projection model is intended to provide a high resolution image space to be able to precisely compute a correction warp, which remaps the texture contributions with the matching content of input images I_i^t. In particular we do not model any optical blur here; rather for each HR pixel q we collect all texel contributions projecting within. Because calibration and 3D models are available, we can use GPU z-rendering to filter out non-visible texels [7]. Occasionally the density of projected texels is insufficient (*i.e.* in high curvature regions of the surface) for a pixel to receive any texture samples. In this situation we assume the underlying surface appearance perceived by this pixel is an interpolation of neighboring texels. For a uniform, continuous treatment of both cases, we combine all texel contributions falling in the vicinity of q by a spatial Gaussian weight with small variance σ_p^2, and normalize to one the sum of texel contribution weights for a pixel q. The continuity of this scheme ensures that no artificial discontinuity is created as a result of a discrepancy in the treatment of these cases. This insures that samples present at the pixel contribute overwhelmingly when present at the center of the pixel, and that the pixel is computed as a weighted sum of texels further away otherwise.

Note that, with this formulation, HR pixels are a linear combination of texels of T. Let $\mathbf{P}_i^t$ be the resulting sparse projection operator such that $\mathrm{H}_i^t = \mathbf{P}_i^t \mathrm{T}$, appropriately collecting the weights previously discussed after being mapped and projected in view $\{i, t\}$. Each $\mathbf{P}_i^t$ can be stored as a sparse matrix with $w_{\mathrm{H}_i^t} \times h_{\mathrm{H}_i^t}$ rows and $w_{\mathrm{T}} \times h_{\mathrm{T}}$ columns, respectively HR image resolution and the chosen texture resolution.

3.2. Inputs as Warped, Downsampled HR Images

To generate an input image from each HR image H_i^t, the HR image is first warped according to the different apparent sources of variability impacting the input image - calibration error, distortion, model geometry error - using a dense warp field W_i^t. This warp results in a linear operator over the HR image, which we note $\mathbf{F}_{\mathrm{W}_i^t}$. The image then traverses the optical system, where it is blurred and captured

by the CCD which performs light integration at every photosite. Following 2D super-resolution literature [1, 9] this is generally modeled using a Point Spread Function (PSF) with the form of a Gaussian blur kernel, followed by an image subsampling stage. Both operations can be written as linear operators, the image-wide blur operator $\mathbf{K}$ and subsampling operator $\mathbf{S}$, which are applied to the HR image to obtain a view's observed image $\mathrm{I}_i^t = \mathbf{SK}\mathrm{H}_i^t$. Remarkably, in its noiseless form, the full image formation model can thus be noted as a single, sparse linear operator $\mathbf{A}_i^t = \mathbf{SKF}_{\mathrm{W}_i^t}\mathbf{P}_i^t$ for each view $\{i,t\}$, with $w_{\mathrm{I}_i^t} \times h_{\mathrm{I}_i^t}$ rows and $w_{\mathrm{T}} \times h_{\mathrm{T}}$ columns, such that $\mathrm{I}_i^t = \mathbf{A}_i^t\mathrm{T}$ for each view $\{i,t\}$. This elegantly generalizes the linear formation models used in various 2D super-resolution models [17] to the 3D+t case.

4. Bayesian Generative Model

The linear model previously discussed describes how input pixels are obtained through warping and blending of texels in noiseless fashion. As in the 2D case, inverting the problem to estimate T and the warps W_i^t from I_i^t is ill-posed, non-convex, and noise ridden [1]. We thus introduce a noise model and priors for better problem conditioning, formulating the solution as a MAP estimation over T, and the warps $\{\mathrm{W}_i^t\}$ for all views and temporal frames $\{i,t\}$:

$$\{\hat{\mathrm{T}}, \{\hat{\mathrm{W}}_i^t\}\} = \arg\max_{\mathrm{T},\{\mathrm{W}_i^t\}} p(\mathrm{T}, \{\mathrm{W}_i^t\}|\{\mathrm{I}_i^t\}), \quad (1)$$

where the posterior is a product of prior and likelihood:

$$p(\mathrm{T}, \{\mathrm{W}_i^t\}|\{\mathrm{I}_i^t\}) = p(\mathrm{T}) \prod_{i,t} p(\mathrm{W}_i^t) \prod_{i,t} p(\mathrm{I}_i^t|\mathrm{W}_i^t, \mathrm{T}). \quad (2)$$

Prior Terms. To ensure sparsity of variations of the estimated texture and warp, we impose minimal Total Variation (TV) constraints on appearance image T and each W_i^t:

$$p(\mathrm{T}) = \frac{1}{Z_{\mathrm{T}}(\lambda)} e^{-\lambda\|\nabla\mathrm{T}\|}, \quad (3)$$

$$p(\mathrm{W}_i^t) = \frac{1}{Z_{\mathrm{W}}(\gamma)} e^{-\nu\left(\|\nabla u_i^t\| + \|\nabla v_i^t\|\right)}, \quad (4)$$

where ∇ is the gradient operator, $\|\nabla\mathrm{T}\| = \sum_q \|\nabla\mathrm{T}(q)\| = \sum_q (\|\mathrm{T}_x(q)\| + \|\mathrm{T}_y(q)\|)$, the sum over pixel index q of the L_1-norm over spatial image derivatives of $\mathrm{T}(q)$. The same definition holds for u_i^t and v_i^t, the x- and y- components of the warp W_i^t. $Z_{\mathrm{T}}(\lambda), Z_{\mathrm{W}}(\gamma)$ denote the normalization constants of both distributions.

The TV constraint ensures that T is treated as a natural image to be restored with sparse and preserved edges. However, a discontinuity between some neighboring object surface points can be created due to necessary cuts in the mesh unwrapping algorithm, leading some mapped texels to appear in different charts despite their proximity on the surface [13]. For such texels, we carefully compute gradients by computing the transform of axis directions as reprojected in the chart where surface neighbors were mapped [10, 11]. This minimizes discontinuities in treatment across chart boundaries in the estimation.

Data Term. Under the assumption that the noise is independent per pixel given the information about the texture, model and cameras, we impose a Gaussian prior for each frame $\{i,t\}$:

$$p(\mathrm{I}_i^t|\mathrm{W}_i^t, \mathrm{T}) = \frac{1}{Z(\mathrm{D}_i^t)} e^{-(\mathrm{I}_i^t - \mathbf{A}_i^t\mathrm{T})^\top \mathrm{D}_i^t (\mathrm{I}_i^t - \mathbf{A}_i^t\mathrm{T})}, \quad (5)$$

where D_i^t is a diagonal covariance matrix introduced to allow different noise characteristics per pixel q, and $Z(\mathrm{D}_i^t)$ a normalization function of D_i^t. In 2D super-resolution models, a single variance per input image is usually used, with the i.i.d. noise assumption [9]. However when acquiring appearance in the 3D case it is well known that contributions need to be modulated according to the angle θ_q between viewing vector and local surface normal [7]. This can be purposely identified in the generative model, where each diagonal element $\mathrm{d}(\theta_q)$ of D_i^t materializes the breadth of the underlying Gaussian predictive model and thus the confidence in the pixel. We set this value as a robust, conservative function of θ_q given in §6, which we assume fixed for the purpose of estimation, under small warp perturbations. Note that this is a valid assumption since visibility and grazing angles are generally stable, as we assume given the full poses of the model $\mathcal{M}^t$ for all frames.

5. Inference

Directly maximizing all variables in (1) is intrinsically hard and seldom done in the literature. We opt for a coordinate descent scheme, alternating between T and W_i^t.

Appearance Map. We maximize (1) by minimizing its negative log, dropping all terms independent of T:

$$\hat{\mathrm{T}} = \arg\min_{\mathrm{T}} \sum_{i,t} (\mathrm{I}_i^t - \mathbf{A}_i^t\mathrm{T})^\top \mathrm{D}_i^t (\mathrm{I}_i^t - \mathbf{A}_i^t\mathrm{T}) + \lambda\|\nabla\mathrm{T}\|, \quad (6)$$

where the data term develops to a weighted sum of per-pixel L_2-norms. Although not specifically using a robust data term norm here as opposed to some works, we nevertheless obtain excellent results enforcing robustness through the constant covariance matrix D_i^t, as will be shown. Optimizing a L_2 data term with a TV-regularizer has been specifically studied [4], yielding a family of forward-backward splitting solvers whose implementation are available off-the-shelf [6]. Let us note $f_d(\mathrm{T})$ and $f_{\mathrm{TV}}(\mathrm{T})$ the data and

the TV-term. Forward-backward splitting is an iterative algorithm for estimating T, alternating between computing a gradient update step and projection $\mathrm{prox}_{\gamma,f_{\mathrm{TV}}}$ which computes an implicit subgradient descent step for the TV-norm.

$$\mathrm{T}_{n+1} = \mathrm{prox}_{\gamma,f_{\mathrm{TV}}}\left(\mathrm{T}_n - \gamma\nabla f_d(\mathrm{T}_n)\right), \tag{7}$$

where γ is a step-size parameter. Our re-weighted functional (6) only implies a modification of the gradient update with respect to the standard case, with $\nabla f_d(\mathrm{T}_n) = 2{\mathbf{A}_i^t}^\top \mathrm{D}_i^t(\mathbf{A}_i^t\,\mathrm{T}_n - \mathrm{I}_i^t)$.

Warp Estimates. We independently estimate each W_i^t for an input view $\{i,t\}$. Minimizing the negative log of (1), and dropping all terms independent of W_i^t yields:

$$\begin{aligned}\hat{\mathrm{W}}_i^t = &\arg\min_{\mathrm{W}_i^t} \nu\left(\|\nabla u_i^t\| + \|\nabla v_i^t\|\right) \\ &+ (\mathrm{I}_i^t - \mathbf{S}\mathbf{K}\mathbf{F}_{\mathrm{W}_i^t}\mathbf{P}_i^t\mathrm{T})^\top \mathrm{D}_i^t(\mathrm{I}_i^t - \mathbf{S}\mathbf{K}\mathbf{F}_{\mathrm{W}_i^t}\mathbf{P}_i^t\mathrm{T}),\end{aligned} \tag{8}$$

which can be interpreted as a modified optical flow equation with a TV-regularizer, where the data term is re-weighted by D_i^t. The intuition here is that the minimization favors the TV prior of sparse variation over the data term for untrustworthy pixels according to D_i^t, and puts more relative importance on trying to follow data on reliable pixels. We opt for a similar strategy to [17] for solving this equation, and initialize the estimation of W_i^t with the result of a standard optical flow method [16], applied between H_i^t and an upsampled I_i^t at each iteration.

6. Experiments

We exhibit results with a MATLAB prototype implementation, and run experiments on a 16-core 2.4 GHz PC with 32GB RAM[1]. Our current implementation is mainly mono-thread, with the exception of the optical flow which we launch in 10 separate threads. To initialize the algorithm, we first use a small C++/OpenGL program to render visibility maps from texture to image space, then initialize the texture map with a simple weighted average of visible inputs. The visibility maps are also used to generate each projection matrix operator $\mathbf{P}_i^t$. We use the Optical Flow package from Liu *et al.* [16] for per-iteration optical flow initialization, and the UNLocBOX package [6] for the texture re-estimation in the loop. We use a threshold on the relative norm of the objective function (6) as stopping criterion, and observe convergence in 30 to 70 iterations for a given λ. The execution time of the algorithm is in the range of 30 minutes to an hour per iteration depending on the dataset, number of views and number of frames. These are not a good indication of the final achievable performance as many enhancements are possible, including making the flow and image update estimations massively parallel on a GPU, better inter-time flow bootstraps as suggested by [17], more compact data-structures, C++ inner loop.

[1] See video results at `http://hal.inria.fr/hal-00977755`

Parameter values. We set the Gaussian variances with $\sigma_p = 0.25$ and $\sigma_k = 0.1$, respectively for the projection weight and PSF kernel $\mathbf{K}$, for all datasets. Although these could be optimized alongside other parameters, we observe low sensitivity to these parameters when set in the $[0.1, 1]$ range. Higher values introduce over-blurring, while lower values tend to reveal the underlying discretization of the texture map ($\sigma_p < 0.1$) or the input image ($\sigma_k < 0.1$). We also fix the convergence parameters to $\gamma = 0.05$ and $\lambda = 5.10^{-4}$ for all experiments, using a second and third round of iterations with $\lambda = 5.10^{-5}$ and $\lambda = 5.10^{-6}$ to down-weigh TV-regularization and thus reveal higher frequency detail. We set $\mathrm{d}(\theta_q) = \frac{1}{C}e^{-s\tan\theta_q}$ as a faster approximation of a normal distribution over the angles of the perceived surface, and use C to normalize these weight contributions to 1 among all pixels that see a common texel x to obtain homogeneous weights among pixels in the data term $\sum_{i,q=\tau\circ\pi_i^t(x)}\mathrm{d}(\theta_q) = 1$. We use $s = 7\pi/16$ over all experiments. This weight is more conservative than the $\cos\theta_q$ weight usually used for blending in multi-view texturing techniques [7], and yields improved results in our experiments, as it downgrades unreliable contributions from surface points at a grazing angle.

6.1. Static Multi-View Comparison

We compare our model with the latest state of the art multi-view texture super-resolution technique of Goldlücke *et al.* [10]. As the latter does not deal with temporal sequences, the comparison is performed on the common applicability domain, *i.e.* static images, as shown Fig. 3. The authors provide a public dataset for three objects BEETHOVEN, BUNNY and BIRD, and kindly provided additional data on request, so we could reproduce the experiment in the closest possible setup. This included a high resolution output of their algorithm for the viewpoint originally reported per dataset in [10], to which we compare our high resolution output. We use the same super-resolution ratio of $3\times$ the input resolution for the texture and high resolution image domains. Respectively 108, 52 and 52 calibrated viewpoints were originally used at resolution 768×576. We have used identical views, and also use identical 3D models except for the BIRD dataset, for which we observed large reconstruction and silhouette reprojection artifacts on the model provided. In fairness we thus only provide crops in regions where the 3D model geometry is not significantly different.

It can be generally observed that our outputs provide lower noise levels and artifacts. This is particularly visible in the BUNNY dataset, in the ear region and shadow

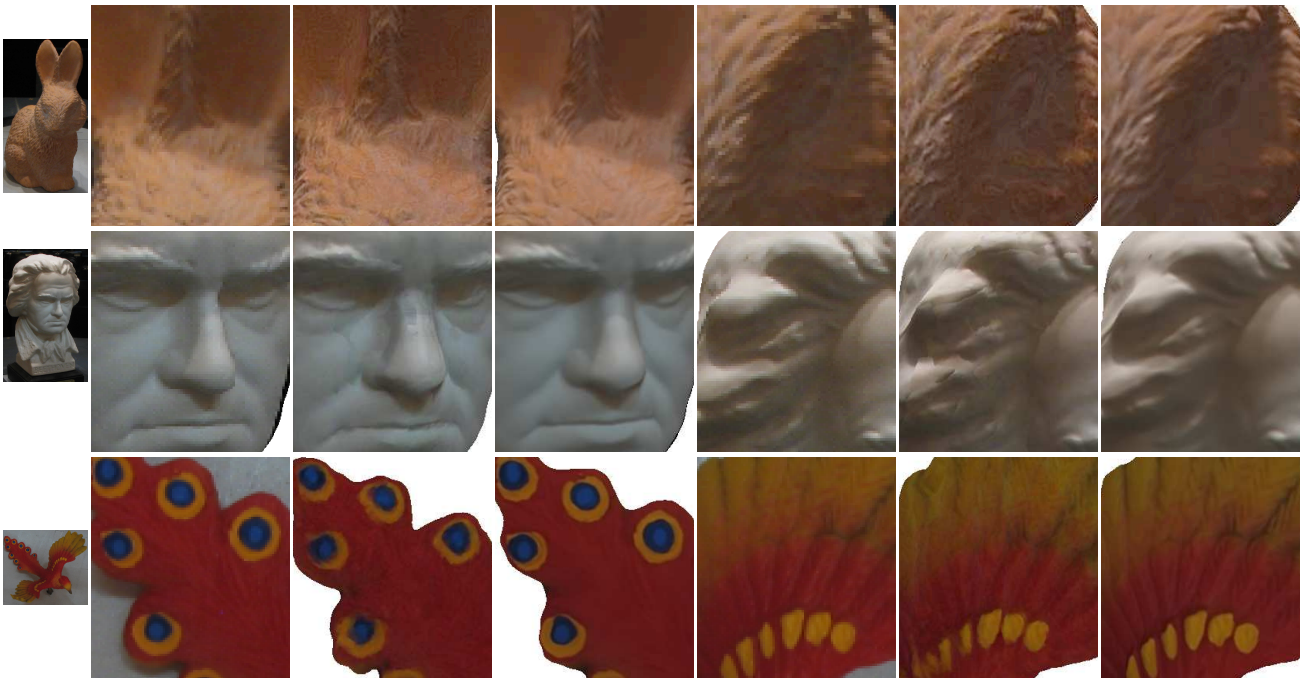

Figure 3. Comparison on BUNNY, BEETHOVEN and BIRD datasets. Left column: input images. Middle: output of [10]. Right: our algorithm. Best viewed magnified and in color.

region around the left eye. The BEETHOVEN exhibits some visibility difficulties due to the face geometry and presence of concave regions around the nose and hair, which generate artifacts for [10]. In contrast, our method is able to deal with these situations efficiently. A single texture domain cut is present on the nose but the discontinuity is barely visible thanks to the inter-chart terms we introduced. More accurate details and sharper pattern borders can be observed on the BIRD wing and tail, notably in the feather textures.

6.2. Temporal Superresolution Validation

To evaluate our approach on the temporal aspect, we introduce three synchronised datasets, GOALKEEPER, BACKPACK and ACTOR, in Fig. 4. The datasets were acquired with three different setups and camera models so as to maximize testing variability. All 3D models were obtained using silhouette-based reconstruction techniques and thus yield largely imperfect models. GOALKEEPER consists of 21 calibrated viewpoints at 1024×1024, which we downsample to 512×512 for the purpose of evaluation. BACKPACK consists of 15 viewpoints of a person, with resolution 1624×1224. ACTOR consists of 11 viewpoints in resolution 1920×1080. The ACTOR dataset is arguably the most difficult one, with lower views and higher noise levels both in the images and the reconstruction. We focus on small motions of the three subjects, and test the method for 2 to 7 frames. Significant improvements can be seen in the figure through temporal accumulation.

There are several difficulties in designing an experiment to quantify this improvement, such as the absence of ground truth data in texture space for real datasets. Synthetic datasets are less than ideal for image restoration and super-resolution problems: a significant conclusion can only be achieved if the different sources of variability are correctly introduced and simulated: sensor noise, calibration error, local reconstruction errors, specularities, temporal misalignments. Instead, we focus here on showing the temporal improvement by running our algorithm on a $2\times$ downsampled version of the GOALKEEPER dataset, and comparing our reprojected result with the higher resolution inputs using the mean squared error metric (MSE). Fig. 5 shows the result of this experiment, with convergence curves from one frame (static case) to three frames, and MSE's evaluated on the 21 input views. Several observations can be made from these curves. First, they illustrate convergence of the iterations toward the high resolution ground truth. Second the temporal improvement leveraged by our algorithm is validated in two forms: acceleration of the rate of convergence using more temporal frames, and improvement of the final result quality over using only one temporal frame.

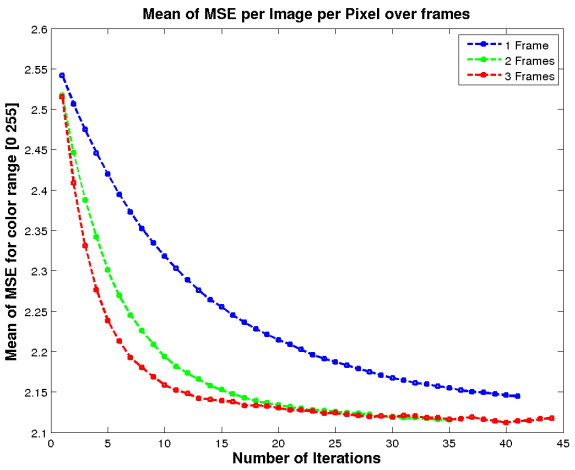

Figure 5. Results from GOALKEEPER dataset. We computed the mean value over frames of the Mean Square Error between our output and high resolution ground truth image. The resolution of input images is 512×512 and of the super-resolved output images is 1024×1024. We use a time step of 2 in experiments, corresponding to an acquisition frequency of 15Hz.

7. Discussion

We have presented a novel method to retrieve a single, coherent texture from several viewpoints and temporal frames of a deformable subject. The noiseless formation model introduced is linear from texture space to image space, and noise and regularization are achieved using a Bayesian framework. We have demonstrated the usefulness of this approach with respect to state of the art, and quantified the convergence and temporal improvement. The method opens several interesting research possibilities. First, more of the parameters and variability could be automatically learned, such as the projection parameters and regularization weight. The framework proposed enables this, with adapted convergence algorithms. Second, the trade-off between using more views or more temporal frames could be further explored to understand how each modality contributes to the result. Third, longer term resilience could be explored as an extension of this model.

Acknowledgement

This work was funded by the Seventh Framework Programme EU project RE@CT (grant agreement no. 288369).

References

[1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE PAMI*, 24(9):1167–1183, Sept. 2002.

[2] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, p. 425–432, 2001.

[3] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. In *ECCV*, vol. 6314, p. 326–339, 2010.

[4] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, Jan. 2004.

[5] D. Cobzas and M. Jägersand. Tracking and rendering using dynamic textures on geometric structure from motion. In *ECCV*, vol. 2351, p. 415–432, 2002.

[6] P. L. Combettes and J.-C. Pesquet. Proximal Splitting Methods in Signal Processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, p. 185–212. 2011.

[7] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH*, p. 11–20, 1996.

[8] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating Textures. *Comp. Graph. Forum*, 27(2):409–418, Apr. 2008.

[9] R. Fransens, C. Strecha, and L. V. Gool. Optical flow based super-resolution: A probabilistic approach. *CVIU*, 106(1):106–115, 2007.

[10] B. Goldluecke, M. Aubry, K. Kolev, and D. Cremers. A super-resolution framework for high-accuracy multiview reconstruction. *IJCV*, 2013.

[11] B. Goldluecke and D. Cremers. A superresolution framework for high-accuracy multiview reconstruction. In *Pattern Recognition (Proc. DAGM)*, 2009.

[12] B. Goldluecke and D. Cremers. Superresolution texture maps for multiview reconstruction. *ICCV*, p. 1677–1684, Sept. 2009.

[13] Z. Janko and J.-P. Pons. Spatio-temporal image-based texture atlases for dynamic 3-D models. In *IEEE 3DIM*, p. 1646–1653, Oct. 2009.

[14] V. S. Lempitsky and D. V. Ivanov. Seamless mosaicing of image-based texture maps. In *CVPR*, 2007.

[15] H. P. A. Lensch, W. Heidrich, and H.-P. Seidel. A silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63(4):245–262, 2001.

[16] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, May 2009.

[17] C. Liu and D. Sun. A Bayesian approach to adaptive video super resolution. *CVPR*, p. 209–216, June 2011.

[18] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogom yakov. Abf++ : Fast and robust angle based flattening. *ACM Transactions on Graphics*, Apr 2005.

[19] T. Takai, A. Hilton, and T. Mastuyama. Harmonised texture mapping. In *3DPVT*, 2010.

[20] C. Theobalt, N. Ahmed, H. P. A. Lensch, M. A. Magnor, and H.-P. Seidel. Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Trans. Vis. Comput. Graph.*, 13(4):663–674, 2007.

[21] T. Tung. Simultaneous super-resolution and 3D video using graph-cuts. *CVPR*, p. 1–8, June 2008.

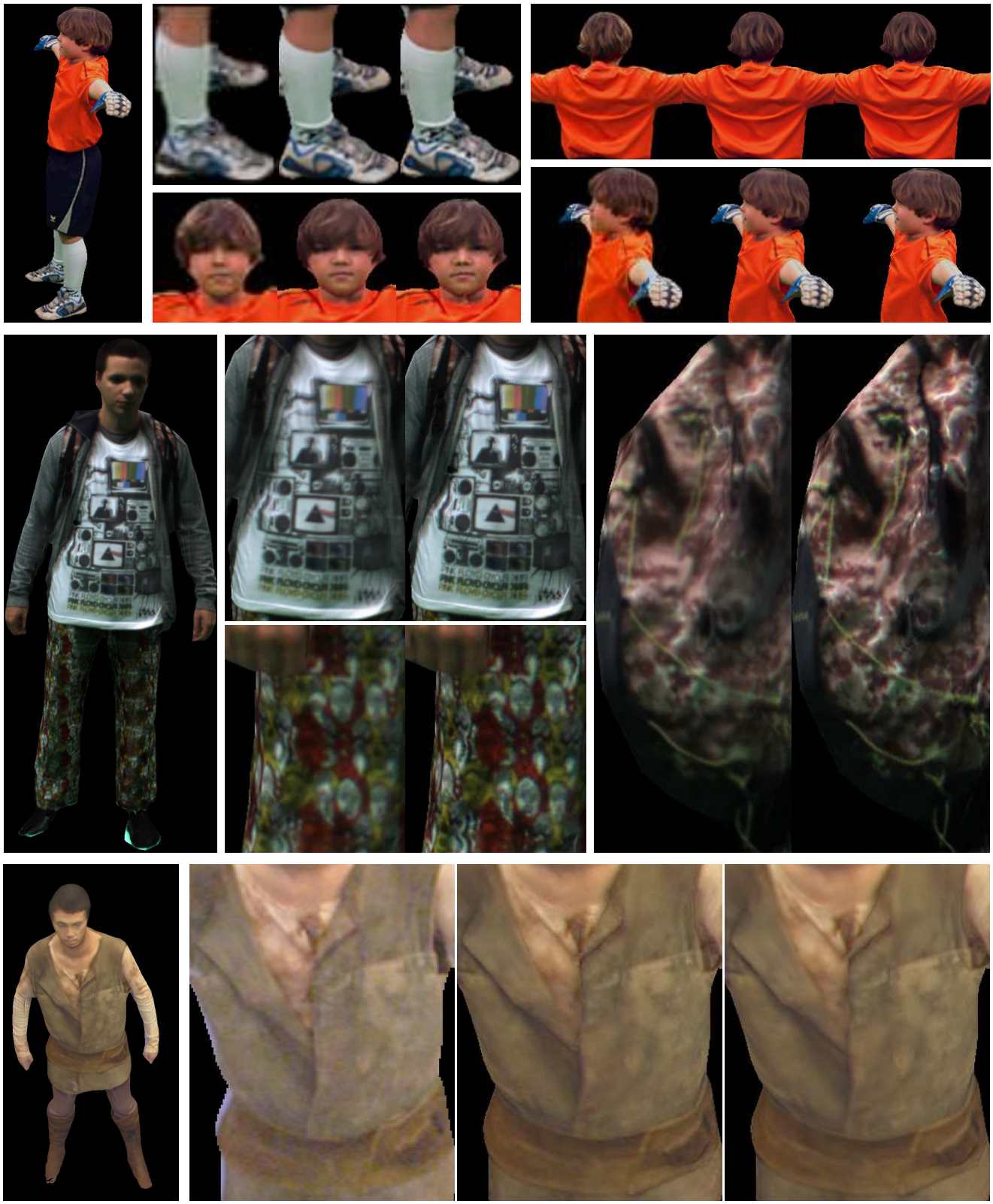

Figure 4. This figure illustrates various temporal improvements and detail enhancements obtained with various acquired datasets, comparing different convergence using one or several temporal frames. Top: GOALKEEPER dataset. Left: output of Frame 3. Input is compared to Frame 1 and Frame 3 for each close-up. Middle: BACKPACK dataset. Input on left, Frame 1 and Frame 2 comparisons for close-ups. Details are revived on the backpack, T-shirt and pants. Bottom: ACTOR; left to right: full result with three frames, close-up comparison between input, against Frame 1 and Frame 3. Best viewed magnified and in color.