Stamping and Extracting Bump Maps

Rick Ramstetter
rick.ramstetter@gmail.com

Ravneet Singh

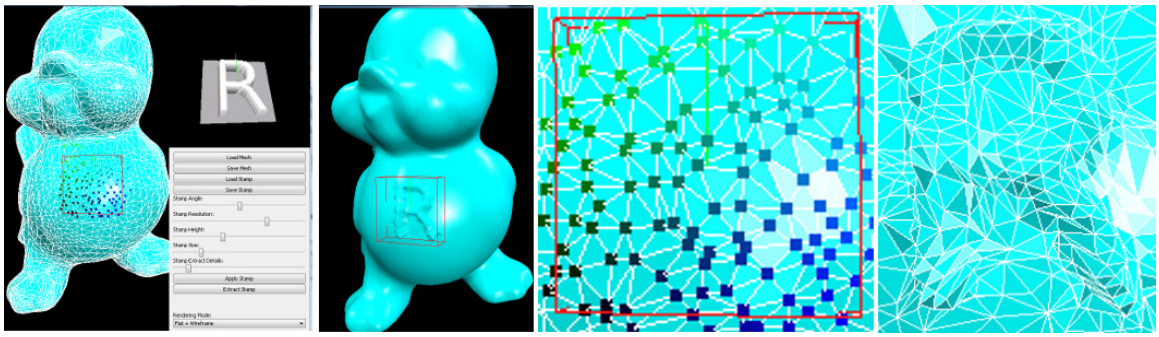


Figure 1: A. A familiar mesh and "R" stamp loaded into our interface. B. The same mesh with stamp applied. C. Close-up view of affected mesh area before stamping. D. Close-up view of affected mesh area after stamping.

1 Abstract

We have implemented a "copy and paste" styled interface for mesh editting. This interface allows the user to both a) input a grayscale bump-mapping image and "stamp" this image onto parts of a loaded mesh, and b) extract grayscale bump-mapping images from a given mesh. For stamp application, mesh faces are umpsampled via Sqrt-3 subdivision for high gradient areas of the bump-mapping image. Relevant vertices are then moved along their smoothed-surface normal to achieve their new, post-stamping position. For extraction, relevant areas of the mesh are interpolated via Moving Least Squares in order to meet the user's stamp size and resolution requirements.

2 Introduction

Image editing tools like Adobe Photoshop allow the user to "copy and paste" image details from a source to destination. This tool allows users to, for example, quickly (and often crudely) replace background noise or missed features in an image. Unfortunately, mesh editing tools such as Maya do not, by default, include such a "copy and paste" functionality. We have implemented such functionality in a manner that attempts to keep computation time to a minimum, thus allowing for a smooth and intuitive mesh editting session for end users. By upsampling and using interpolation where appropriate, we have created a robust tool for mesh "stamping".

A selection box approximation to an on-mesh area of interest is presented to the user (see Figure 1A, for example). By double clicking, this approximate selection box is mapped to a set of mesh vertices. These vertices will be either deformed (upon stamp application) or analyzed (upon stamp extraction). The vertices are mapped to the stamp's domain in a manner similar to that frequently used in texture mapping. During stamp application, mesh areas that map to a high frequency area of the stamp are upsampled using Square Root 3 Subdivision [Labsik and Greiner 2000]. During stamp extraction, Moving Least Squares [Nealen 2004] is used to interpolate areas of the stamp which map to lower resolution areas of the mesh.

2.1 Terminology

Terminology used throughout this paper:

- *Bump-mapping image*: the image loaded or extracted by a user, representative of a mesh deformation from the mesh's smoothed, base surface.
- *Stamp size*: the size of the stamp as is rendered on and mapped to the mesh.
- *Stamp resolution*: the size of the stamp's data representation. This can be thought of in image pixels.
- *Stamp extraction details*: a weighting value for our interpolation calculation. See section TODO.
- *Stamp harshness*: a multiplier for the bump-mapping image's deformation; a low magnitude bump-mapping image can be magnified with this value.
- U, V *mapping*: a mapping from either of the mesh to a 2-dimensional coordinate system (with $U, V \in 0, 1$), or a mapping from the bump-mapping image to the same domain.
- *Up-direction*: for some point on a rendered object, this is the projection of the viewer/camera vector onto the plane formed by that point and its surface normal.

3 Related Work

Our work was motivated in part by work on "Displaced Subdivision Surfaces" [Lee et al. 2000]. This work introduces a means for representing an arbitrary mesh by a lower resolution mesh and a displacement surface from that mesh. As has already been mentioned, Sqrt-3 subdivision [Labsik and Greiner 2000] is used for upsamping, and Moving Least Squares [Nealen 2004] for interpolation.

4 Methodology

In this section, the particular workings of our mesh stamping process will be elaborated on.

4.1 Selection Region

The displayed selection box is only an estimate of the region to be selected. This selection box must be mapped to an area of relevant vertices before mesh editting can occur. The mapping used

translates the coordinates of mesh vertices to a U, V domain, with $U, V \in 0, 1$. The bump-mapping image is equivalently mapped to the same U, V domain, providing us with a mapping from mesh coordinates to bump-map image pixels. A breadth first traversal of the mesh starting at the selection box's center point is used to label vertices relevant to the drawn estimating selection box (see section TODO).

The underlying surface normal at the center, clicked-on point of the selection is stored as the stamp's normal. This surface normal is calculated by running repeated smoothing operations on the mesh. These smoothing operations are performed once at mesh loading time. As such, the case that a user's stamping will alter the underlying surface normal of the mesh as this would require re-calculation of the smoothed mesh at every stamp placement operation, a non-trivial operation for larger meshes.

The selection box is oriented against the mesh using a stored "up" vector. This vector is calculated as the "up" direction of the rendering camera projected onto the plane described by the stamp's center point and the stamp's normal (see section TODO). [1]

4.2 U,V Mapping

A U, V mapping is used to translate from the mesh domain to the bump-mapping image domain. Similar to texture mapping methods, both the relevant mesh vertices and the bump-mapping image are mapped to the U, V domain. The point at the center of the user's selection (that is, where the user clicked) is mapped to $U, V = .5, .5$. Similarly, the center pixel of a bump-mapping image is mapped to $U, V = .5, .5$.

The bump-map image's U, V mapping is a factor of:

- The bump-map image's size.
- The current "stamp resolution," which affects the sampling rate from the bump-mapped image's actual data to it's loaded and used freqeuncy. This value can be used to downsample [2] the currently loaded bump-mapped image, or to set the target size of an extracted bump-mapping image.

The mesh's U, V mapping is a factor of the current 'stamp size" value, which affects the size of the on-mesh, approximating selection box to relevant vertices.

4.3 Prioritized BFS

A prioritized breadth first traversal of the mesh is used in calculating the U, V coordinates of selected vertices. This PBFS is prioritized by the distance of each selected vertex from the clicked, center vertex. For each vertex *Ve*, this distance *D* from the center point is calculated as:

$$Ve.D = Ve.Parent.D + ||Ve.UV - Ve.Parent.UV|| \quad (1)$$

Where *Ve.Parent* is the vertex that was last dequeued during the prioritized breadth first traversal. (To rephrase, *Ve.Parent* had *Ve* as an unvisited neighbor.) Intuitively, the breadth first traversal prefers vertices with lower *D* values. Thus, the PBFT can be thought of as tracking the distance travelled in U, V space from U, V=.5, .5 to arrive at some vertex, and calculating U, V coordinates in the process.

This PBFT arrangement performs both a mapping of a the selection box to vertices [3] and mapping of those vertices to U, V space.

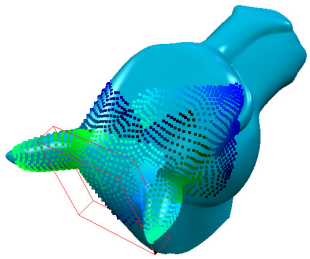


Figure 2: Failure to include the $P.Distance < 1$ PBFT stopping condition

4.3.1 Information stored

For each vertex visited during BFS traversal, the vertex's normal, its "up" direction, and its calculated U, V coordinates are stored. The *stored normal* is the normal for the vertex on the underlying, smoothed surface mesh. The U, V coordinates of *Ve* are calculated by projecting *Ve* onto the plane formed by *Ve.Parent* and *Ve.Parent.Normal*. Utilizing the known up direction of *Ve.Parent*, simple trigonometry yields the U, V coordinates of *Ve*. Similarly, the *"up" direction* of *Ve* is calculated by projecting the up direction of *Ve.Parent* onto the plane created by *Ve* and *Ve.Normal*.

4.3.2 Stopping conditions

It has already been said that a prioritized breadth first traversal of the mesh is used in calculating the set of selected vertices and their U, V coordinates. In doing so, a breadth first traversal stopping condition of $U, V > 0, 1$ or $U, V > 1, 0$ is used. Unfortunately, another stopping condition is required to ensure the search halts; if the PBFS has travelled distance $D > 1$ to arrive at some point, that point is not queued for traversal (one is the maximal 2-dimensional Manhattan distance for a point from U, V = .5, .5). See Figure TODO for a visual example of failing to utilize this stopping condition.

4.4 Stamp application

In applying a stamp, areas of the mesh which map to U, V areas with high gradient are upsampled. This is done using Sqrt-3 [Labsik and Greiner 2000] subdivision. After upsampling, vertices are shifted along their surface normal as prescribed by the bump-mapping image. The "harshness" of the deformation is controlled by the end user.

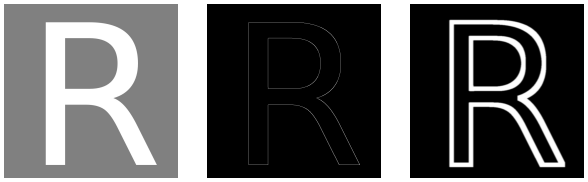
4.4.1 Mesh Upsampling

A gradient map for a bump-map image is created when it is loaded or extracted. A naive implementation of our work would use this simple gradient as a criterion for mesh upsampled (or face splitting). Unfortunately, this leads to tesselation and vertex valence issues on the mesh. In our implementation, the gradient is dilated then blurred. After blurring, high magnitude areas of the original gradient map are run through a process of contrast enhancement. In doing this, the number of faces to be upsampled has been increasedd. In this implementation, even faces *next to* high gradient

[1] As a reminder, only the normal of the selected point cannot be used to orient the box because the plane formed by the selected point and surface normal is not necessarily perpendicular to the camera's view vector.

[2] At present, the "stamp resolution" value will, if set too high, cause aliasing of the loaded loaded bump-mapped image. Aliasing introduces a range of problems, and ideally would be avoided. It is possible, though not implemented, for our interface to recognize the maximum frequency of a bump-mapped image, and adjust the maximum value of the "stamp resolution" slider accordingly

[3] Vertices with U, V coordinates outside $0, 0$ through $1, 1$ are not traversed and thus not selected.



(a) Bump map) (b) Gradient (c) New Gradient

Figure 3: An example bump map, its original gradient, and its dilated, smoothed, and tesselation friendly gradient

areas of the bump-mapping image are upsampled. For an illustration, see Figure TODO.

It is assumed that it is preferable to have a higher count of well tesselated faces than a lower count of poorly tesselated faces. Thus, introducing extra faces is not an issue. In the case that a lower count of poorly tesselated faces is preferable, the afforementioned naive implementation is suitable.

4.5 Stamp extraction

A stamp is extracted by first mapping a selected vertex to it's U, V domain coordinates. Then, the bump-mapping image value at this coordinate is set to:

$$BMI(U,V) = ||Mesh.Vertex - SmMesh.Vertex|| \quad (2)$$

Where SmMesh is the smoothed, surface mesh. The bump mapping image value is set to the magnitude of the deformation from the surface mesh to the smoothed surface mesh; the bump mapping image is a mapped scalar displacement field.

4.5.1 Moving Least Squares Interpolation

Moving least squares interpolation is used for (frequent) cases when the mesh does not have enough data in some selection to fill the bump mapping image. To illustrate, a selection size of 128 by 128 vertices would be needed to fill a 128 by 128 pixel bump-mapping-image. Assuming a restrained selection size, most meshes do not have nor require this level of detail. Thus, Moving Least Squares interpolation is used.

The processing time for the stamp extraction process is a factor of the user's set level of "detail extraction." This value controls the MLS Weighting Value for bump-map images (*d* in the following equation). Specifically, the *d* parameter tells the MLS calculation how large the radius of neighbors to be included in the calculation of a given point's MLS value is. Setting a small *d* value will cause aliasing problems while extracting the stamp, while setting too large of a *d* value will cause the stamp to miss mesh details.

$$MLSWeight = \exp^{\frac{||p-v||^2}{d}} \quad (3)$$

4.6 Performance Analysis

The performance of the mesh stamping and extraction processes are quick enough to allow for an intuitive and smooth end user experience.

4.6.1 Stamp Application

As can be intuitively expected, the time taken to apply a stamp scales with the average frequency of that stamp. Stamps with many areas of high gradient magnitude require more upsampling operations to be applied, whereas a stamp with no details is applied instantaneously. The processing time for a select few stamp applications can be found in Figure 2. A key point to notice from Figure 2 is that all mesh stamping takes less than 3 seconds, even for maximal stamp-to-mesh resolution.

Stamp Application Times				
Mesh	Resolution	Start vertices	End vertices	Time
A	1/64 (default)	94	459	1/2 s
A	1/256 (max)	98	1474	2 s
B	1/64 (default)	125	337	1/2 s
B	1/256 (max)	112	2263	2 1/2 s
C	1/64 (default)	56	334	1/2 s
C	1/128 (max)	61	1382	2 s

Figure 4: Stamp resolution, start vertices, end vertices, and processing time for the A) Armadillo Back, B) "R" and C) Armadillo Leg stamps. All were applied to the familiar bird ("Tweety") mesh. The Vertex counts are representative of the number of vertices inside the approximating selection box, and not the total mesh vertices. The increase in vertices is due to mesh upsampling.

Stamp Application Times			
MLS Weight *d*	Resolution	Vertices	Time
100	64x64	1313	2 s
250	64x64	2864	3 s
5	256x256	2864	56 s
15	256x256	2864	56 s
30	256x256	2864	53 s
75	256x256	2864	54 s

Figure 5: MLS Weight *d* value, vertices selected, destination stamp resolution, and computing time required.

4.6.2 Stamp Extraction

The processing time for the stamp extraction process is a factor of the user's set level of "detail extraction." This value controls the MLS Weighting Value for bump-map images that

4.7 Future Work

More advanced means of U, V mapping are possible. See, for example, TODO.

Currently, though our means of mesh upsampling (see section TODO) does gradient smoothing to aid in the creation of well tesselated deformations, good mesh tesselation is not necessarily preserved. In our trials, at most approximately 15% of faces affected by our mesh stamping process were upsampled in a manner resulting in poor tesselation. This is in large part due to our current means of U, V mapping.

Similarly, our mesh upsampling does not necessarily respect the well known Nyquist rate. As a reminder, a binary decision is made to upsample a mesh face based on the mapped gradient for that face. Though the positive criterion for this decision is quite low, it should be noted that a low gradient image does not neccesarily correlate with a low frequency image. For example, if an image alternates from 0 to 100 frequently in one area, and alternates from 55 to 56 frequently in another area, the image gradient will like not represent the true image frequency in the second area (as the magnitude of change in the first area is taken into consideration). This problem can be formulated as either an improper calculation of image gradient or an improper application of image gradient.

5 Tools Used

Where possible, we utilize Intel's OpenCV library for our image loading and processing operations. OpenMesh is used for our mesh storage. Earlier revisions of the project used Newmat for the solution of small linear systems. The QGLViewer project's widget is used for our OpenGL interface, and TrollTech's Qt is used for GUI generation.

5.1 Conclusion

We have provided a methodology and software application for mesh stamping. Though both have flaws, we believe that they strike a reasonable balance between required computational power and

References

LABSIK, U., AND GREINER, G. 2000. Interpolatory sqrt3-subdivision. *Computer Graphics Forum 19*, 3, 131–138.

LEE, A., MORETON, H., AND HOPPE, H. 2000. Displaced subdivision surfaces. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 85–94.

NEALEN, A., 2004. "an as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation", May.