Retargeting Vector Animation for Small Displays

Vidya Setlur
Northwestern University and Nokia Research Center
Xuejin Chen
USTC and Microsoft Research Asia
Yingqing Xu
Microsoft Research Asia
Bruce Gooch
Northwestern University

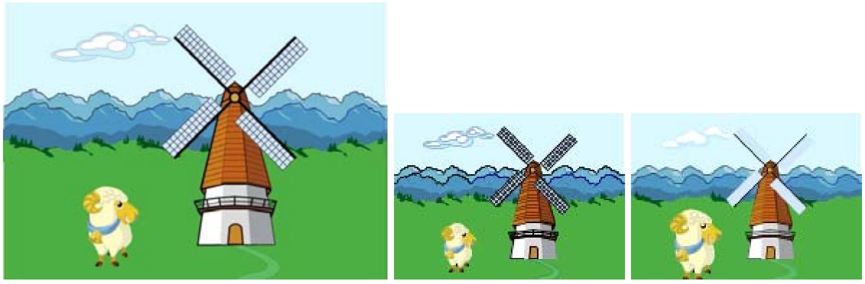


Figure 1: *Preserving the spatial detail in important objects from a source animation to a smaller sized animation.*

Abstract

We present a method that preserves the recognizability of key object interactions in a vector animation. The method allows an artist to author an animation once, and then output it to any display device. We specifically target mobile devices with small screen sizes. In order to adapt an animation, the author specifies an importance value for objects in the animation. The algorithm then identifies and categorizes the vector graphics objects that comprise the animation, leveraging the implicit relationship between extensible Markup Language (XML) and scalable vector graphics (SVG). Based on importance, the animation can then be automatically retargeted for any display using artistically motivated resizing and grouping algorithms that budget size and spatial detail for each object.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms;

Keywords: perception, animation, small displays, WWW applications, information visualization, non-photorealistic rendering, vector graphics, XML

1 Introduction

Advances in mobile devices and wireless telecommunication provide users with ubiquitous access to online information and services. However, user access and interaction are still quite restricted with regard to the display of imagery such as animations, diagrams, maps, and charts. There exists a growing need for the effective adaptation of imagery for small size displays. This work presents an algorithm for retargeting vector based animations while maintaining the recognizability of object interaction.

Effective display consists of understanding the communication goal of some form of imagery and then fitting that imagery to the display device in a manner that aids this goal. In most animation, the story is communicated to the viewer via the interaction of a few **key objects**. Remaining objects in the scene provide a context for this interaction, and are referred to as **contextual objects**. In order to achieve the communication goal of an animation on a mobile device, key object interactions must be displayed at both sufficient size and spatial detail for easy recognition. The contextual objects in the animation are less important. The premise of our method is that when the key objects are known, their features can be exaggerated in order to render their interaction more obvious.

Objects in vector graphics images and animation are typically uniformly scaled regardless of their importance. Therefore, we introduce a perceptually motivated algorithm that exploits the semantics of vector graphics data to guide the retargeting process. In addition, the algorithm redistributes spatial detail among the objects in the scaled animation based on importance.

1.1 Contributions

This work provides tools for artists to intuitively author and manipulate machine-readable forms of animation. We also demonstrate that these tools are useful for encoding multiple levels of detail in a single image to enhance the utility of mobile devices as information displays. The intellectual contribution of this work is the idea of importance tags that can be leveraged to make imagery dynamically adapt to a user's needs.

MUM 2005 Christchurch, New Zealand
ISBN 0-473-10658-2

2 Related Work

Prior work in human perception and computer graphics has established that it often becomes necessary to sacrifice detail in order to meet the computational demands of complex scenes [O'Sullivan and Dingliana 2001; Reitsma and Pollard 2003]. Level of detail (LOD) techniques for real-time rendering and other perception based computer graphics problems have become necessary for meeting real-time demands for most scenes of significant complexity and to adaptively modulate levels of detail in different parts of a simulation process. Visual artifacts that occur in areas with less amount of detail, may go unnoticed to an average viewer if these areas are perceptually less important for a given visual task. For example, the work of Chenney and Forsyth involved culling non-visible parts of the scene [Chenney and Forsyth 1997]. Carlson and Hodgins explored techniques for reducing the computational cost of simulating groups of creatures by using less accurate simulations of individuals when they are less important to the viewer or to the action in the virtual world [Carlson and Hodgins 1997]. Similarly, there has been work on reducing time complexity in geometrical models based on lower importance [Reddy 1997; Funkhouser and Séquin 1993]. These approaches allow speed-accuracy trade-offs to be optimized by exploiting a viewers inability to distinguish simplifications in less important parts of an image or animation. All these techniques demonstrate the idea that adaptive detail modulation can be more effective than uniformly reducing the complexity of the entire scene. Our work is inspired by this idea. While previous work deals with speed and time constraints for reducing complexity, we address the problem of adaptation based on a size constraint. Instead of uniformly scaling vector graphics animation, we apply adaptive detail modulation to emphasize important objects.

Existing work on intelligent adaptation of images and video for smaller displays, focusses on maintaining the recognizability of more important objects in the visual scene. Suh et al. proposed a technique for automatic image thumbnail cropping based on a visual attention model to detect interesting areas in the image [Suh et al. 2003]. This method however, crops only the most important region and does not retain the entire context of the visual scenario from the original image in the smaller sized image. The absence of contextual information, may not convey the entire visual story to the viewer. This may not be an issue for images containing a single subject, where the surrounding context is less influential in understanding the content of the image. On the other hand, for images containing multiple objects or for images where the entire visual context is necessary for performing visual tasks, thumbnail cropping may not be suitable. Chen et al. introduced an image adaptation technique that delivers the most important region to mobile devices [Chen et al. 2003]. The user can scroll between different pages of an image to view different important regions. Work by Wang et al. uses a sampling-based dynamic attention model to obtain and maintain the users attention on video streams [Wang et al. 2004]. The amount of visual data presented to the user is adjusted by uniformly zooming in and out of the visual scene based on user interest. Related to this work, Fan et al. introduced an approach that allows users to explicitly zoom into video frames while browsing on small displays [Fan et al. 2003; Liu et al. 2003]. Computational attention models tend to perform poorly on animation, because they depend strongly on image luminance and contrast, and often do not identify features important for a visual task [Ferwerda 2003]. Instead, our animation retargeting method allows users at the authoring level to assign object importance at the authoring level. Further, our system displays *all* content on the screen at once, and allows differential zooming to emphasize and de-emphasize information.

Rist and Brandmeier have explored automated adaptation mechanisms for transforming images to serve mobile devices using down-sampling and color reduction [Rist and Brandmeier 2002]. Martin has developed a system for adaptive delivery of 3D models in heterogeneous networked environments, enabling access by clients with diverse graphics capabilities [Martin 2000]. Marriott et al. address the client-side adaptation of documents to various viewing conditions, such as varying screen sizes, style preferences, and different device capabilities, by including one-way constraints into SVG [Marriott et al. 2002]. These constraints mainly manipulate document layout specifications by declaratively specifying the desired layout of the web document. There has also been research on map generalization [Agrawala and Stolte 2001; Neuffer et al. 2004; Visvalingam 1999]. Cartographic generalization is concerned with deriving small scale, less detained maps from larger scale maps. While their work is mainly concerned with automatic generalization techniques for static map images, we introduce a general framework to work for both static as well as dynamic imagery.

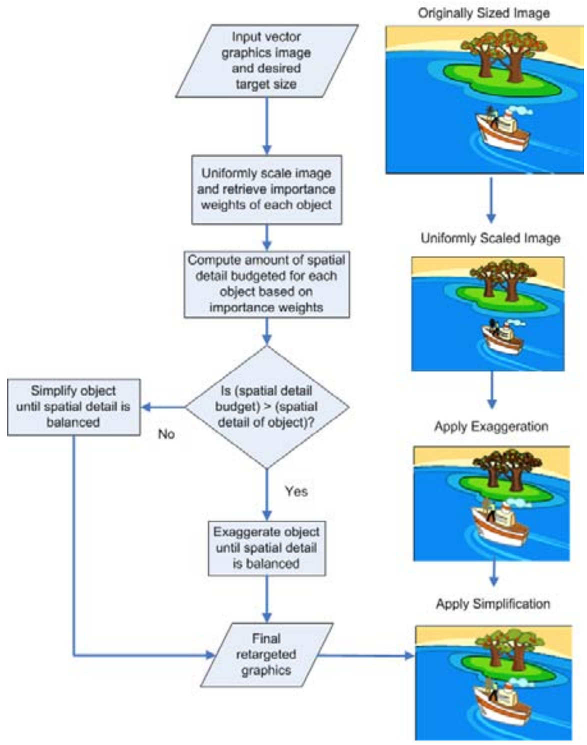


Figure 2: *Flowchart of the animation retargeting process. The goal of the algorithm is to increase the recognizability of the key objects, while simplifying contextual objects in order to maintain the net spatial detail of the retargeted animation. Here, the boat is exaggerated, while the tree is simplified.*

3 The Retargeting Process

This work presents an algorithm for retargeting vector based animations while maintaining the recognizability of object interactions. Our retargeting algorithm takes a target size and a vector animation or image as input. The XML format of the vector graphics structure is parsed to identify objects and their assigned importance. The importance parameter is an SVG tag set by the animation author, and is constrained to be $\in [0, 1]$. The animation is then resized using traditional graphics methods resulting in uniform scaling of all objects, regardless of importance. We use the term spatial detail to

Figure 4: *The relationship between spatial details and features present in an object. Here, the spatial detail of an object decreases as the number of its features reduce. (Left) Spatial detail* = 0.333754. *(Center) Spatial detail* = 0.258025. *(Right) Spatial detail* = 0.231496.

```
<svg width="700", height="500">
 ...
  <desc> Car Definition </desc>
  <g id="0",  importance = "0.5">
   <animateMotion path=" M42.45265 124.8342........."
     dur="4s" fill="restore"   begin="load" />
       <fill-color r="238", g="171", b="34", a="255" />
        <boundary-color r="61", g="57", b="49", a="255" />
           < path d = "M700.500000 499.507996, C700.5
            499.507996 C0.5 …./>
               <polyline class="Connect" points="100,200
               100,100" />
                 ...
  </g>

  <desc> Clouds Definition </desc>
  <g id="1",  importance = "0.3">
  <animate attributeName="x" from="265.7341 "
  to="294.9668 " dur="4s" fill="restore" begin="load" />
  <fill-color r="241", g="211", b="25", a="255" />
     <path d="M63.73145 261.89025 C89.32236 257.6145 ..
          <path d="M300,200 h-150 a150,150 0 1,0
                  150,-150 z" stroke="blue" />
             ...
    </g>
</svg>
```

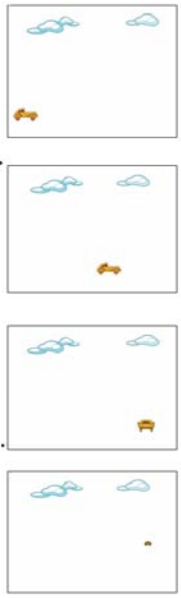


Figure 3: *Illustration of the one-one correspondence between vector graphics and its underlying XML structure.*

measure the feature density of objects. For example, a white sphere has less spatial detail than a soccer ball with the same dimensions. The overall spatial detail in the scene is redistributed, by exaggerating more important key objects and simplifying less important contextual objects. The amount of exaggeration or simplification is based on an objects' importance. Figure 2 illustrates the outline of the process. Section 6 provides a more detailed description of the algorithm.

4 The Vector Graphics Format

Our system extends the Scalable Vector Graphics (SVG) format. SVG was developed as an open standard grammar for vector graphics. SVG is written in XML, and can easily be extended using XML tags. We use SVG structural tags to define the building blocks of our vector graphics data format. These tags include the <svg> element, which is the top-level description of the SVG document, a group element <g>, which is a container element to group semantically related Bezier strokes into an object, the <path> element for rendering strokes as Bezier curves, and several kinds of <animate> elements to specify motion of objects.

4.1 Directed Acyclic Tree Representation

The SVG format conceptually consists of visual components that are modeled as nodes and links. Elements are rendered in the order in which they appear in the SVG document. Each object in the data format can be thought of as a canvas on which paint is applied. If objects are grouped together with a <g> tag, they are first rendered as a separate group canvas, then composited on the main canvas using the filters or alpha masks associated with the group. In other words, the SVG document can be viewed as a directed acyclic tree structure proceeding from the most abstract, coarsest shapes of the objects to the most refined details rendered on top of these abstract shapes. This property of SVG allows us to do a depth-first traversal of the nodes of the tree and manipulate the detail of any object by altering the structural definitions of that object. We observe that this framework is similar to several perceptually guided model and mesh simplification techniques [Floriani et al. 1997; Williams et al. 2003; Bolin and Meyer 1998]. SVG also tags objects throughout an animation sequence alleviating the issue of video segmentation. The motion of objects can be tracked through all frames of an animation by using <animate> tags. Figure 3 shows a fragment of the data format used for two objects in an animation.

4.2 Assigning importance tags to SVG objects

In order to redistribute spatial detail among objects we sort them based on importance. We provide the infrastructure for artists at the authoring level to assign importance values to objects in an animation. The artist or user has to annotate objects in a scene with importance tags that might become cumbersome as the number of objects in the scene increases. However, SVG has a number of open source GUI authoring tools [Ink n. d.; Sod n. d.], and the importance annotation functionality is incorporated as a plug-in to the GUI. This allows users to mouse click and annotate importance values more easily. Importance values are tagged per scene by adding them as attributes to the objects and propagated through the SVG data structure. The only constraint is that the importance value is $\in [0,1]$, with 0 indicating most simplified and 1 indicating least simplified. The process is analogous to using RGB boxes in Adobe Photoshop [Ado n. d.] to set a color, and then using sliders to fine tune the importance values. Once the importance tags are defined, the rest of the algorithm is completely automatic. The importance tags are hence defined at the authoring level and do not change with display size.

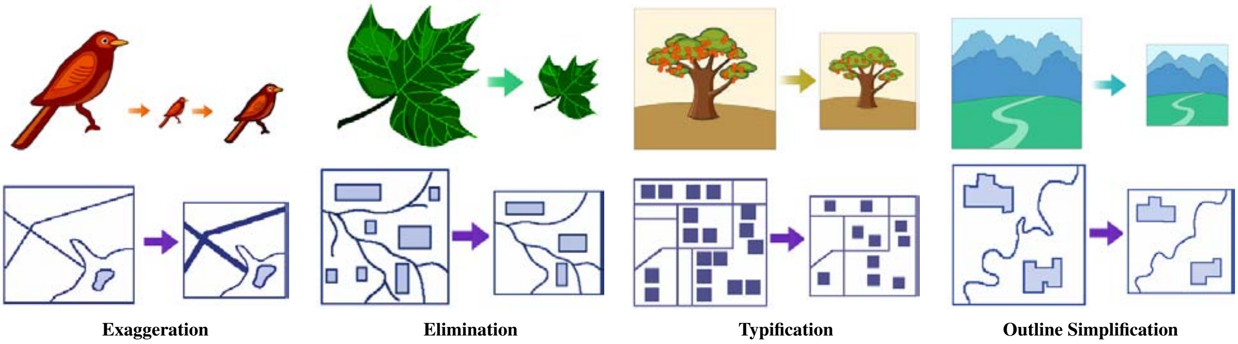


Figure 5: *Illustrating artistic rules for distributing spatial detail.*

5 Computing Spatial Detail

In order to perform differential zooming of objects in the scene, it is necessary to compute spatial detail of each object and to be able to redistribute this quantity based on importance. The spatial detail indicates how rapidly luminance is changing in the neighborhood of a given pixel. Figure 4 demonstrates that the features of the buildings' windows become simplified as the spatial detail decreases. The computational measure of this property is well studied particularly for texture analysis and retrieval applications. We did experiment with all the texture features described in [Amadasun and King 1989], including variance, but spatial detail or 'busyness textural property' best worked for our purpose. The Neighborhood Gray-Tone Difference Matrix *NGTDM* is a perceptual description of spatial detail for an image in terms of changes in intensity and dynamic range per unit area. The *NGTDM* is a matrix, in which the *i*th entry is the summation of the differences between the luminance value of all pixels in the image with the luminance value of the pixels in a neighborhood of pixel *i*.
We use YUV color space to compute the gray value for each pixel, which is equal to $(0.257 \times R) + (0.504 \times G) + (0.098 \times B) + 16$.

Let $f(k,l)$ be the luminance of the pixel at (k,l). We then find the average luminance over a neighborhood centered at, but excluding (k,l).

$$\overline{A}_i = \overline{A}(k,l) = \frac{1}{W-1}\left[\sum_{m=-d}^{d}\sum_{n=-d}^{d} f(k+m,l+n)\right]$$

where d specifies the neighborhood size, $W = (2d+1)^2$, and $(m,n) \neq (0,0)$.

Then the *i*th entry in the *NGTDM* is defined as

$$s(i) = \begin{cases} \sum \left|i - \overline{A}_i\right|, \forall\, i \in N,\ if\ N_i \neq 0 \\ 0,\ otherwise \end{cases}$$

where N_i is the set of all pixels having gray tone i (except in the peripheral regions of width d).

We then use the NGTDM to obtain the following computational measure for spatial detail after [Amadasun and King 1989].

$$\text{Spatial detail} = \frac{\sum_{i=0}^{G_h} p_i s(i)}{\sum_{i=0}^{G_h}\sum_{j=0}^{G_h}\left|ip_i - jp_j\right|} \qquad p_i \neq 0, p_j \neq 0$$

where G_h is the highest gray-tone value present in the image. The numerator is a measure of the spatial rate of change in intensity, while the denominator is a summation of the magnitude of differences between luminance values. Each value is weighted by the probability of occurrence. For an $N \times N$ image, p_i is the probability of occurrence of gray-tone value i, and is given by $p_i = N_i/n^2$, where $n = N - 2d$, and N_i is the set of all pixels having gray tone i (except in the peripheral regions of width d). Spatial detail is computed for a given target display size. Also, if an object changes size'or color during the course of the animation, spatial detail is recomputed for the changed object.

6 Spatial Detail Distribution

The goal of the retargeting process is to preserve the recognizability of the interactions between key objects after the animation is resized. While vector graphics animations are resolution independent, key object interactions may not be recognizable at all sizes due to artifacts introduced by uniform scaling. In order to automate the process of retargeting animations, we draw inspiration from a collection of perceptually based artistic techniques. These techniques facilitate differential resizing instead of a uniform scaling. Artistic techniques often involve de-emphasizing context objects, and increasing the detail in key objects [Kowalski et al. 2001; Johnston and Thomas 1995; Markosian et al. 2000; Lansdown and Schofield 1995; Winkenbach and Salesin 1994; Meier 1996]. Similarly, generalization is a process used by cartographers [Agrawala and Stolte 2001; Board 1978; MacEachren 1995] to reduce the scale and complexity of imagery while maintaining detail in important elements. The following rules are automatically applied to the object nodes in the SVG representation of the animation based on the importance value of the object. The rules can be classified based on whether they emphasize or de-emphasize objects.

The redistribution of spatial detail in the retargeted image is a simple budget allocation method based on the importance value of individual objects. The most important object is budgeted the largest amount of the total spatial detail available for the image, while the least important object is budgeted the least amount. The importance value of an object is constrained by definition to be $\in [0,1]$, and the importance values of all objects are then normalized. An object cannot be made more detailed than the original or more simplified than its basic outline. Additional constraints that may affect redistributing of spatial detail in the scene are derived from display configurations, and the bounds of human visual acuity. These constraints may be dictated by the physical limitations of display devices such as the size and resolution of display monitors, the min-

(a) (a) Original Animation (b) (b) Scaled Animation (c) (c) Object Enhancement (d) (d) Object Generalization

Figure 6: *Objects are enhanced or generalized based on spatial budget distribution. Here, the boat is enlarged and the tree detail is simplified to satisfy the spatial budget constraint. However, even though the spatial budget constraint requires the lake to be exaggerated, its bounding area is as large as the image and remains unchanged.*

imum size and width of objects that can be displayed or the minimum spacing between objects that avoids symbol collision or overlap. The following spatial detail redistribution algorithm computes a spacial detail constraint for every object to emphasize particular objects and to clarify by removing visual clutter:

1. Resize original vector graphics image or animation to desired target size. All objects are uniformly scaled.
2. Look up the *Importance Value* of each object.
3. Normalize the spatial detail value by dividing the *Original Spatial Detail* of each object by its corresponding *Bounding Area*. We call this the object's *Unit Spatial Detail*.
4. Add the *Unit Spatial Detail* values of all objects to obtain the *Total Unit Spatial Detail*.
5. Compute the *Weighted Unit Spatial Detail* for each object, which is the object's *Importance Value* $\times$ *Total Unit Spatial Detail*.
6. Compute *Spatial Detail Constraint* allocated for each object, which is the *Weighted Unit Spatial Detail* $\times$ *Bounding Area* of object.
7. If (*Original Spatial Detail* of object $<$ *Spatial Detail Constraint* of object), Then apply Key Object Enhancement until *Original Spatial Detail* of object $\geq$ *Spatial Detail Constraint* of object. However, when the retarget size is very small, there may not be enough space to exaggerate the size of the object. In such cases, the size of the objects remains the same as in the uniformly scaled image.
8. Else if (*Original Spatial Detail* of object $>$ *Spatial Detail Constraint* of object), Then apply Context Object Generalization until *Original Spatial Detail* of object $\leq$ *Spatial Detail Constraint* of object.

6.1 Key Object Enhancement

Key object enhancement consists of both size and line exaggeration rules. These rules are applied to *increase* the spatial detail and visibility of the object after the vector animation or image is uniformly scaled down. Our system increases the object's size to satisfy the spatial detail constraint. If the object is just a line stroke, such as routes in informational images, our system then applies line exaggeration, by increasing the line weight. Figure 5 shows both line and size exaggeration.

6.2 Context Object Generalization

Generalization is a process of making entity classes less specific by suppressing characteristics that describe the class. These rules are applied when the spatial detail of the object needs to be *reduced*, after uniform scaling of the vector animation or image. Starting from leaf nodes of the SVG tree, regions in objects are eliminated based on the spatial detail constraints.

1. **Elimination**: The process selectively removes regions inside objects that are too small to be presented in the retargeted image. Beginning from the leaf nodes of the SVG tree, that represent the smallest lines and regions in an object, primitives are iteratively eliminated until the spatial detail constraint for the object is satisfied at the new target size. Figure 5 shows elimination applied to the veins of a leaf.
2. **Typification**: Typification is the reduction of feature density and level of detail while maintaining the representative distribution pattern of the original feature group. Typification is a form of elimination constrained to apply to multiple similar objects. Our system applies typification based on object similarity. Computing object similarity is a difficult pattern recognition problem. We use the heuristic of tree isomorphism within the SVG data format to compute a measure of spatial similarity. Each region of the object is represented as a node in the tree. Nested regions form leaves of the node. A tree with a single node (the root) is isomorphic only to a tree with a single node that has approximately the same associated properties. Two trees with roots A and B, none of which is a single-node tree, are isomorphic if and only if the associated properties at the roots are identical and there is a one-to-one correspondence between the subtrees of A and of B. This method works well on objects that are semantically grouped and in the same orientation. Figure 5 shows typification for removing apples from a tree.
3. **Outline Simplification**: Often the control points of the Bezier curves, representing ink lines at object boundaries become too close together resulting in noisy outline. Outline simplification reduces the number of control points to relax the Bezier curve. We use a vertex reduction technique, which is a simple and fast $O(n)$ algorithm. In vertex

reduction, successive vertices that are clustered too closely are reduced to a single vertex. In our system, control points with minimum separation are simplified iteratively until the spatial detail constraint is reached. In Figure 5 the silhouettes of the mountains are simplified using the vertex reduction rule. Anti-aliasing could also be applied in conjunction with outline simplification to minimize the occurrence of scaling effects in the outlines of objects.

While retargeting animation containing textual objects, certain measures could be taken for greater legibility: using a thinner font, and readjusting text to prevent overlap during object enhancement.

Applying the spatial distribution algorithm to Figure 6, we can compute the following values for each object. The input importance values $\in [0, 1]$ for Object 1, Object 2, and Object 3 are 0.1, 0.8, and 0.7 respectively.

Object			
Normalized Importance	0.0625	0.5	0.4375
Original Spatial Detail	0.2	1.02	0.77
Unit Spatial Detail (e-005)	0.86	16.1	24.86
Weighted Unit Spatial Detail (e-005)	2.61	20.87	18.27
Spatial Detail Constraint	0.62	1.32	0.6
New Spatial Detail	0.2	1.32	0.6

Figure 7: *Intermediate values calculating during the spatial detail budgeting process. The goal is to make the new spatial detail of each object as close as possible to its Spatial Detail Constraint. However for object 1, the spatial detail cannot be increased as its area is equal to that of the retargeted area.*

Notice that the first object is constrained by the animation's bounding size and cannot be exaggerated further, and so it spatial detail (0.2) remains unchanged as shown in Figure 6a. The spatial detail constraint (1.32) of the second object is satisfied by applying exaggeration. The increase in size is shown in Figure 6c. The spatial detail of the third object reduces to the budgeted spatial detail (0.6) by applying typification. Figure 6d shows that typification removes apples from the tree.

7 Informational Images

The retargeting framework for animation may be extended to informational images as well (Figures 9c, 9d, 9e). Informational images are an abstraction, or generalization, of physical reality, and their effectiveness as a communication medium is strongly influenced by the nature of the spatial data, the form and structure of representation, the intended purpose, the experience of the viewer, and the context and time in which the images are viewed [Buttenfield and McMaster 1991].

The retargeting process needs to exploit the artists' intentions for each entity in the information image to create a representation consistent with the knowledge conveyed by the original image. Determining the knowledge to be conveyed to the viewer, often involves a high level semantic understanding of the context of the visual task. Converting such a high level semantic ontology of information into a computational form is often a non-trivial problem. Informational image systems such as MapQuest [Map n. d.] and Google Maps [Goo n. d.] work around this problem by applying differential zooming based on where the user clicks on the map. Although our system has a similar goal as these systems, the difference in our approach is that differential levels of detail are applied to each object in the scene based on importance tags in the underlying XML structure of the graphics data. Our work may be extended to location based services by using Global Positioning Systems (GPS) to guide the annotation of objects with importance tags. Here, the contribution of this work is the methods for increasing and reducing complexity in the image, resulting in differential zooming.

8 Results and Discussion

Our results demonstrate that the animation retargeting method performs reasonably well on vector graphics and images, where semantically important objects are rendered with greater clarity, while unimportant objects maintain the context of the animation or the information conveyed to the user.

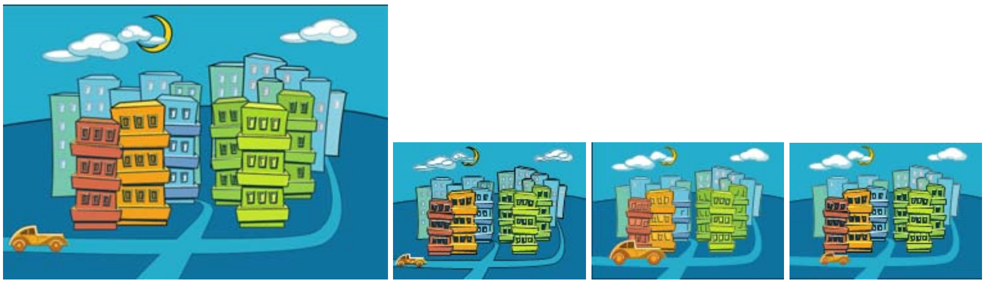
We ran the algorithm on an Intel(R) Xeon(TM) CPU 3.06GHz processor with 2GB RAM. The memory requirement for running the algorithm is 29+8MB. The run-time performance is as follows:

Windmill example (Figure 1): 1817ms
Boat example (Figure 6): 2459ms
House example (Figure 8): 1927ms
Frog example (Figure 10a): 1394ms
Eiffel tower example (Figure 10b): 1942ms
Map1 example (Figure 10c): 2255ms
Map2 example (Figure 10d): 1426ms
Map3 example (Figure 10e): 752ms

While the importance values are an effective way of designating key objects in an animation clip, these parameters often need to be tuned by the animation author for a given display. Figure 8 shows the variation in retargeted results depending on which object is more important. In addition, unless the artist specifically groups objects with implicit visual relationships these relationships may be destroyed by the retargeting process.

In the case of animations involving temporally consistent objects, we apply object transformations to the entire scene rather than on a frame-by-frame basis. This is because SVG provides the advantage of declarative animation rather than frame based animation. However, for objects temporally varying in size and/or color, spatial detail needs to be calculated at each new instance of change in object state. The process could get more complex particularly when key objects become context objects and vice-versa. The author may then have to annotate importance tags to every new state of the vector object.

The exaggeration and generalization rules that we use may have a non-linear effect on computed value of spatial detail. This can result in a noisy version of an animation at small target sizes. This effect becomes more evident as the target animation or image size becomes very small. The semantic grouping of objects also affects the performance of the algorithm. For example in Figure 6, the island is grouped with the water. Since this object cannot be further exaggerated, both the island and water remain the same size as in the uniformly scaled animation. However, ungrouping the water



(a) (a) Original Animation (b) (b) Scaled Animation (c) (c) Retargeted 1 (d) (d) Retargeted 2

Figure 8: *Importance can influence which objects are enhanced and simplified. (c) Importance for car* = 0.9*, houses in background* = 0.2*, houses in front* = 0.2*, sky with moon and clouds* = 0.0*. (d) Importance for car* = 0.2*, houses in background* = 0.2*, house in front* = 0.7*, sky with moon and clouds* = 0.0*.*

and island, allows the island to become exaggerated although the water object will remain the same.

Object integrity depends on how the SVG animation is laid out. For example, consider the car in Figure 8. If the car is rendered on the same layer as the road, then the car and road have an explicit relationship that will be maintained by the object transformations. However, if the car is authored as a new layer on top of the road, then the car and road have only an implicit relationship and it may become semantically difficult to ascertain whether the relationship should be maintained by the object transformations.

Survey: We conducted a web survey, asking users to provide feedback about the retargeted vector graphics results. We used the survey described by Agrawala and Stolte [Agrawala and Stolte 2001] as a basis for ours. 272 people took the survey on animation and 183 people volunteered for the informational images' survey. The survey gave us useful feedback about relative judgments with regard to vector animation and vector informational images. 76.1% of the participants said that the retargeted animation was more effective than the uniformly scaled animation, and 85.8% of the participants thought that the retargeted informational images were more effective than their uniformly scaled counterparts. 56.25% of the participants said that they would use retargeted animation rather than scaled, 26.47% would use retargeted animation along with scaled, and only 17.28% would not use retargeted animation. 69.4% of participants said that they would use retargeted informational images instead of scaled images, 22.96% would use retargeted informational images along with scaled images, and 7.65% would not use retargeted informational images.

9 Conclusion

In this paper we have introduced an algorithm for automatically retargeting vector based animation and informational images to small display sizes. We introduce a framework for users to annotate objects in an animation with importance values, and a system for applying differential resizing and simplification to objects in the scene. By allowing objects to be annotated with importance tags, detail in the vector graphics imagery can be distributed accordingly. This work potentially has applicability for automatic cartography, location-based services, and game interfaces using vector graphics.

10 Acknowledgements

We would like to thank the reviewers for their valuable suggestions and feedback. Thanks to Amod Setlur, Amy Gooch and the rest of the Northwestern University graphics group for their comments and critical review. A special thanks also goes out to the artists at Microsoft Research Asia who designed the input material for our system. This material is based upon work supported by the National Science Foundation under grant numbers IIS-0097456 IIS-0416284. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

http://www.adobe.com. Adobe Photoshop.

AGRAWALA, M., AND STOLTE, C. 2001. Rendering effective route maps: improving usability through generalization. In *SIGGRAPH 2001: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM Press, 241–249.

AMADASUN, M., AND KING, R. 1989. Textural features corresponding to textual properties. In *IEEE Trans. Sys. Man, Cybern.*, vol. 19, 1264–1274.

BOARD, C. 1978. Map reading tasks appropriate in experimental studies in cartographic communication. *Canadian Cartographer 15*, 1, 1–12.

BOLIN, M. R., AND MEYER, G. W. 1998. A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, 299–309.

BUTTENFIELD, B., AND MCMASTER, R. 1991. *Map Generalization: Making Rules for Knowledge Representation*. Longman, London.

CARLSON, D. A., AND HODGINS, J. K. 1997. Simulation levels of detail for real-time animation. In *Proceedings of the conference on Graphics interface '97*, Canadian Information Processing Society, 1–8.

CHEN, L.-Q., XIE, X., MA, W.-Y., AND ZHOU, H.-Q. 2003. Image adaptation based on attention model for small-form factor devices. In *Proceeding of the 9th International Conference on Multimedia Modeling*, IEEE, 421–439.

CHENNEY, S., AND FORSYTH, D. 1997. View-dependent culling of dynamic systems in virtual environments. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM Press, 55–58.

FAN, X., XIE, X., ZHOU, H.-Q., AND MA, W.-Y. 2003. Looking into video frames on small displays. In *Proceedings of the eleventh ACM international conference on Multimedia*, ACM Press, 247–250.

FERWERDA, J. 2003. Three varieties of realism in computer graphics. In *Proceedings SPIE Human Vision and Electronic Imaging*.

FLORIANI, L. D., MAGILLO, P., AND PUPPO, E. 1997. Building and traversing a surface at variable resolution. In *Proceedings of the 8th conference on Visualization '97*, IEEE Computer Society Press, 103–ff.

FUNKHOUSER, T. A., AND SÉQUIN, C. H. 1993. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, ACM Press, 247–254.

http://maps.google.com/. Dynamic Interactive Maps.

http://www.inkscape.org/. Open Source Linux/Windows Scalable Vector Graphics Editor.

JOHNSTON, O., AND THOMAS, F. 1995. *The Illusion of Life: Disney Animation*. Disney Editions.

KOWALSKI, M. A., HUGHES, J. F., RUBIN, C. B., AND OHYA, J. 2001. User-guided composition effects for art-based rendering. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, ACM Press, 99–102.

LANSDOWN, J., AND SCHOFIELD, S. 1995. expressive rendering: A review of nonphotorealistic techniques. In *IEEE Computer Graphics and Applications*, vol. 15, 29–37.

LIU, H., XIE, X., MA, W.-Y., AND ZHANG, H.-J. 2003. Automatic browsing of large pictures on mobile devices. In *Proceedings of the eleventh ACM international conference on Multimedia*, ACM Press, 148–155.

MACEACHREN, A. M. 1995. *How Maps Work*. The Guilford Press.

http://www.mapquest.com/. Consumer-focussed Interactive Mapping Website.

MARKOSIAN, L., MEIER, B. J., KOWALSKI, M. A., HOLDEN, L. S., NORTHRUP, J. D., AND HUGHES, J. F. 2000. Art-based rendering with continuous levels of detail. In *NPAR 2000: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, ACM Press, 59–66.

MARRIOTT, K., MEYER, B., AND TARDIF, L. 2002. Fast and efficient client-side adaptivity for svg. In *Proceedings of the eleventh international conference on World Wide Web*, ACM Press, 496–507.

MARTIN, I. M. 2000. Arte: an adpative rendering and transmission environment for 3d graphics. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, ACM Press, New York, NY, USA, 413–415.

MEIER, B. J. 1996. Painterly rendering for animation. In *SIGGRAPH 1996: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM Press, 477–484.

NEUFFER, D., BELL, M., AND WOODSFORD, P. 2004. Cartographic generalisation on the basis of model generalisation. *Symposium Praktische Kartographie Konigslutter*.

O'SULLIVAN, C., AND DINGLIANA, J. 2001. Collisions and perception. *ACM Trans. Graph. 20*, 3, 151–168.

REDDY, M. 1997. *Perceptually Modulated Level of Detail for Virtual Environments*. PhD thesis, University of Edinburgh.

REITSMA, P. S. A., AND POLLARD, N. S. 2003. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. *ACM Trans. Graph. 22*, 3, 537–542.

RIST, T., AND BRANDMEIER, P. 2002. Customizing graphics for tiny displays of mobile devices. *Personal Ubiquitous Computing 6*, 4, 260–268.

http://www.sodipodi.com/. Open Source Linux/Windows Vector-based Drawing Program.

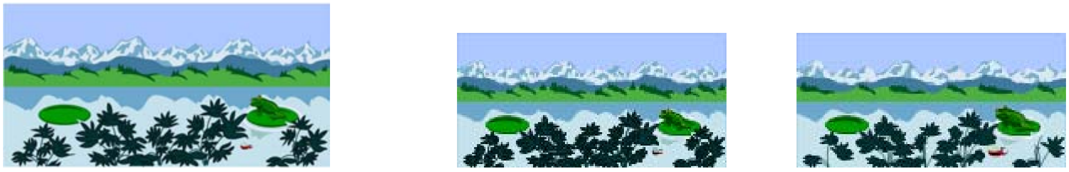
SUH, B., LING, H., BEDERSON, B. B., AND JACOBS, D. W. 2003. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, 11–99.

VISVALINGAM, M. 1999. Aspects of line generalisation. *3rd Workshop of ICA Working Group on Map Generalisation*.

WANG, J., REINDERS, M. J., LAGENDIJK, R. L., LINDENBERG, J., AND KANKANHALLI, M. S. 2004. Video content representation on tiny devices. In *Proc. Of IEEE Conference on Multimedia and Expo*.

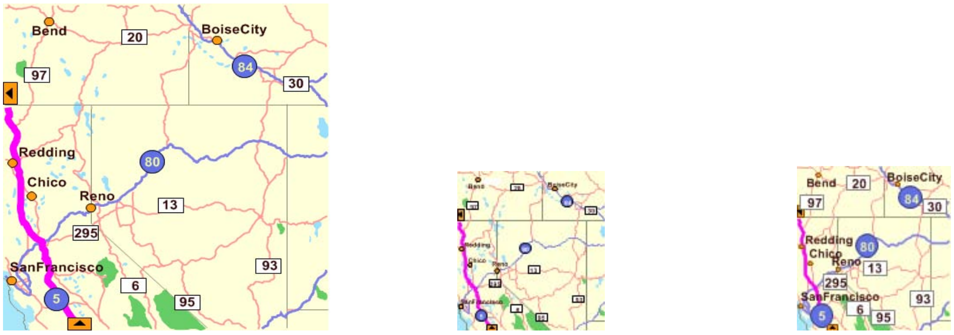
WILLIAMS, N., LUEBKE, D., COHEN, J. D., KELLEY, M., AND SCHUBERT, B. 2003. Perceptually guided simplification of lit, textured meshes. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, ACM Press, 113–121.

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *SIGGRAPH 1994: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, 91–100.

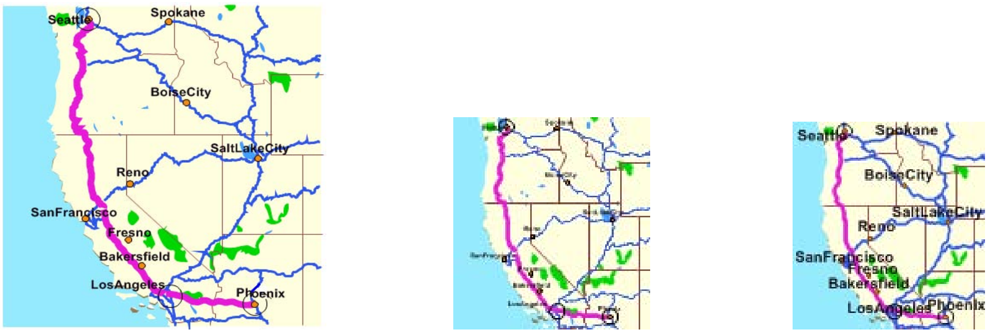


a) The importance are mountain = 0.2, grass = 0.2, pond = 0.4, frog = 0.6, fly = 0.8. The fly and frog are more emphasized.

b) The importance are sky = 0.2, grass = 0.4, tower = 0.6, plane=0.8. The plane is more emphasized.



c) The importance are background (land, water) = 0.2, route lines = 0.9, city labels = 0.9, speckles = 0.2.



d) The importance are background (land, water) = 0.2, route lines = 1.0, city labels = 1.0, speckles = 0.3.

e) The importance are green background = 0.2, sun and clouds = 0.4, temperature labels = 0.8.

Original image **Scaled image** **Retargeted image**

Figure 9: *A comparison between uniform scaling and our automatic animation retargeting method. The original images have been shown smaller than their actual sizes due to space constraints. The images and animation are provided in the supplementary materials.*