Reduction of Animated Models for Embedded Devices

Jiri Danihelka, Lukas Kencl, Jiri Zara
Czech Technical University in Prague, Faculty of Electrical Engineering
{danihjir, kencl, zara}@fel.cvut.cz

ABSTRACT

We present a new supplementary method for reduction of animated 3D polygonal models. The method is applicable mainly in animation of human faces and it is based on intelligent merging of visemes represented by key polygonal meshes. It is useful for devices with limited CPU and memory resources like mobile phones or other embedded devices. Using this approach we can reduce operation memory needs and time to load the model from storage. We describe the algorithm for viseme merging and we prove that our method is optimal for selected metrics. Finally we validate method performance on an example and compare with the case when only traditional methods for 3D models reduction are used.

Keywords: animation, model, reduction, viseme

1 INTRODUCTION

Modern technology devices like personal computers and mobile phones are becoming more and more powerful and complicated. Many people have difficulties controlling miscellaneous computer systems and applications [17]. Computer graphics and designers of computer programs look for new kinds of interfaces to control still more complex computer programs. Talking-head interface seems to be a promising alternative to traditional menu/windows/icons interface for sophisticated applications. Such interface has proven to be useful as a virtual news reader [1], blog enhancement [11] and in many other cases.

So far talking-head interface has been applied mostly on desktop PCs. However, recent small electronic equipment, such as mobile phones, pocket computers and embedded devices possess enough CPU power to offer the talking-head interface as well.

Current smartphones and pocket computers usually have 128MB or 256 MB of RAM. Most of this memory is occupied by the operating system(OS) itself or by OS extensions like HTC TouchFLO or Samsung TouchWiz (formerly pocket computers had only 16 or 32 MB of operation memory, but the OS was stored in read-only memory rather than in RAM). The lack of memory is a bottleneck for animations computed by interpolation of polygonal meshes, because it requires a lot of possibly large polygonal meshes loaded in memory.

To achieve the lowest memory requirements, we have decided to reduce both the amount of polygons in the mesh and the number of key meshes (see figure 1). We propose a dissimilarity metric to detect similar models and a technique to merge them. We prove that our merging technique is optimal for the given dissimilarity metric.

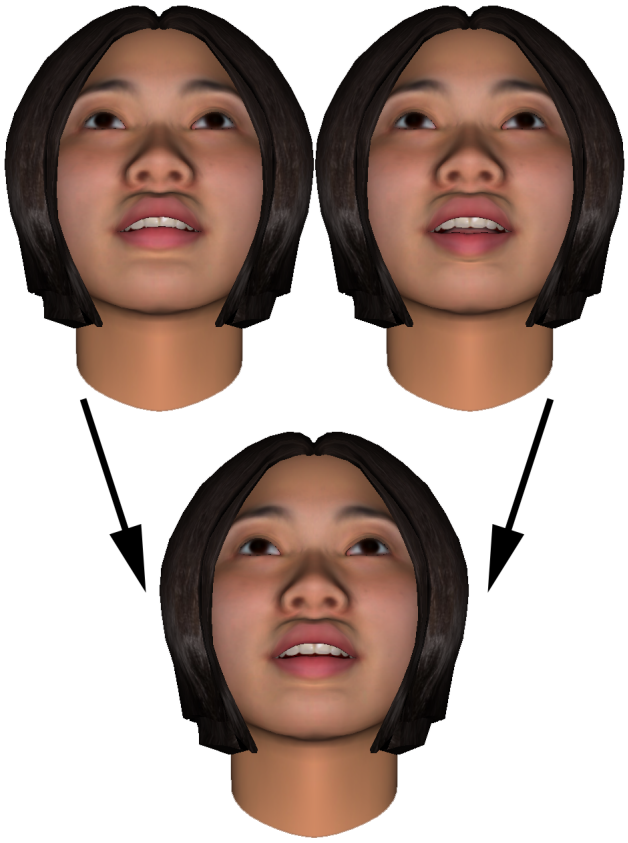

Figure 1: A talking head keyframe model articulating the phoneme "f" (left) is similar to a keyframe model articulating the phoneme "th" (right). Our algorithm detects such similarity and replaces both models with one merged model (down).

2 RELATED WORK

Traditional methods for polygonal reduction are sufficiently covered in [10] and [15]. Specific aspects about geometric rendering and model reduction on mobile phones and embedded devices were presented by Pulli et al. [16].

An interesting way for speeding up morphing animation on embedded devices was proposed by Berner [5]. It is based on optimization strategies by omitting less important polygonal meshes during the animation.

In our research we aim to develop software compatible with the Xface animation framework [2, 3] that is open-source and widely used in academia. There are also more advanced animation frameworks that use skeleton-muscle [18] animation model instead of MPEG-4 standard. The best known of them is Greta [13]. A method of anatomical musculature modeling to achieve realistic and real-time figure animation was proposed by Zuo Li et al. [12].

However none of the works above focuses on reducing the number of visemes (as our work does).

3 FACE ANIMATION PRINCIPLES

3.1 Phonemes and visemes

When using face animation in talking-head applications, we have to consider both visual and audio effects. They are described by visemes and phonemes. A phoneme is an element of spoken language similarly like a letter is an element of written language. A viseme is an element of facial animation. It describes the particular facial position when pronouncing a phoneme. Usually one phoneme corresponds to one viseme, but sometimes multiple phonemes share the same viseme. This happens when facial position of two or more phonemes differs only by position of non-displayed body parts like vocal cords or a tongue.

The frequencies of occurrence of phonemes and visemes depend on spoken language, there are also differencies e.g. between frequencies in British and American English. English has 40 different phonemes.

For our algorithm we need to know the frequencies of phonemes and visemes. The frequencies of phonemes can be determined by converting a long text (at least several pages) using a phonetic transcription software and then by counting the phoneme frequencies in the transcribed text. Such process is usually part of text-to-speech-engine pre-processing of text input for voice synthesis. There is also a free transcription engine available together with typical frequencies of American English phonemes [6]. Having the frequencies of phonemes one can determine the frequencies of visemes using phoneme-to-viseme mapping function.

For our experiments we use the FaceGen facial editor [19] to generate human head visemes. This editor generates 16 different visemes.

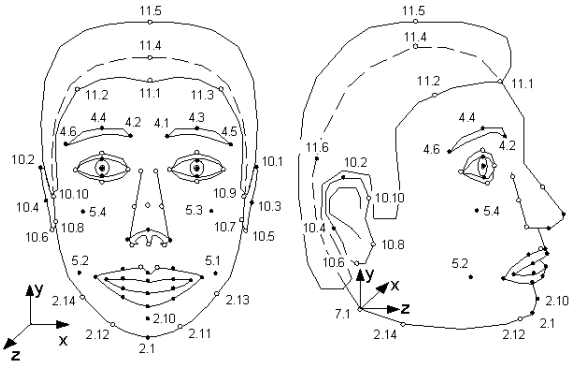

Figure 2: A subset of feature points (FP) defined in MPEG-4 facial animation standard [8]

3.2 MPEG-4 animation

The most widely accepted standard for human face animation is the ISO standard MPEG-4 released by the Moving Pictures Experts Group in 1999 [7, 8].

In this standard 84 feature points (FPs) are specified on human face (see figure 2). The facial animation is controlled by 68 parameters called Facial Animation Parameters (FAPs).

The MPEG-4 standard allows two ways of facial animation. The first one manipulates the feature points individually and can achieve various range of facial expressions. The second one is based on interpolating between two keyframe models. This interpolation can be done either linearly or with cubic interpolation function.

In this paper we focus on the keyframe facial animation. This approach is less CPU intensive and the visual results of this animation are sufficient for mobile phones and embedded devices.

4 DEFINITIONS

4.1 Polygonal model

For purposes of this paper, the polygonal model is a triplet (V, E, P) of vertices V, edges E, and polygons P. To avoid rendering problems with general polygons after geometric transformations, we triangulate all polygons in advance.

Fully triangulated models allow us use a specific metric for model comparison (see section 4.3). They also fit very well into commonly used graphics libraries for mobile phones and embedded devices like OpenGL ES (OpenGL for Embedded Systems) [9] which are optimized for processing triangles only.

4.2 Interpolable set of models

We call polygon models interpolable if they differ only in coordinates of their vertices. Interpolable models have the same topology and the same number of vertices, edges and polygons. There must also be given a

bijection function that matches the corresponding vertices/edges/polygons.

4.3 Polygonal model dissimilarity

We define the polygonal model dissimilarity as a metric (distance function) ρ for two interpolable models.

$$\rho(A,B) := \sum_{k=1}^{\|V\|} w(v_k)\|v_{A,k} - v_{B,k}\|^2 \quad (1)$$

where
A and B are the polygonal models.
$w(v)$ is the weight of the vertex v. It represent an importancy of the vertex in the model. The author of the model can set higher weights for vertices important for human perception.

For models with unspecified weights, we have considered two general metrics:

$$\rho_1(A,B) := \sum_{k=1}^{\|V\|} \left\| v_{A,k} - v_{B,k} \right\|^2 \quad (2)$$

$$\rho_2(A,B) := \sum_{k=1}^{\|V\|} S(v_N) \left\| v_{A,k} - v_{B,k} \right\|^2 \quad (3)$$

where
$S(v_{N,k})$ is a sum of surfaces of triangles incident with vertex $v_{N,k}$. Since the triangle surface may differ for individual visemes, we work with polygon surfaces in the neutral expression of the model $N = (V_N, E_N, P_N)$.

The first metric assumes that more important areas are tessellated more densely. The weight of a face part is given by a number of its vertices.

The second metric can be used if each part of the model surface is equally important for the animation. If we use this metric it is necessary to split all polygons to triangles first as mentioned in section 4.1. We have proven that both metrics give the same results if applied in our reduction algorithm. Thus the real implementation can utilize the first and more simple metric only.

4.4 Dissimilarity for sets of polygonal models

Let $\mathbb{A} = \{A_1, A_2, \ldots, A_n\}$, $\mathbb{B} = \{B_1, B_2, \ldots, B_m\}$ are two sets of polygonal models that represents visemes. Let $f(A_1), f(A_2), \ldots, f(A_n)$ are frequencies of visemes in $\mathbb{A}$. If we have a dissimilarity metric for polygonal models $\rho(A,B)$, we can define dissimilarity for two sets of polygonal models $\rho_f(\mathbb{A},\mathbb{B})$ as:

$$\rho_f(\mathbb{A},\mathbb{B}) = \sum_{i=1}^{n} f(A_i) \min_{j=1\ldots m} \rho(A_i, B_j) \quad (4)$$

It is the sum of distances from each model from $\mathbb{A}$ to its most similar models in $\mathbb{B}$. Note that dissimilarity function for sets of polygonal models is not a metric because it is not symmetrical.

4.5 Problem definition

We describe an algorithm for the following problem:
Input:
Set of polygonal models $\mathbb{A} = \{A_1, A_2, \ldots, A_n\}$. These models represent visemes of a human face that have frequencies $f(A_1), f(A_2), \ldots, f(A_n)$. An integer number m; $m < n$
Task:
Find a set of new polygonal models with m elements $\mathbb{B} = \{B_1, B_2, \ldots, B_m\}$ that is the most similar to $\mathbb{A}$. ($\rho_f(\mathbb{A},\mathbb{B})$ is minimal for all such sets of polygonal models)

5 FINDING OPTIMAL SOLUTION

The solution for the problem is described in two steps: Firstly, we describe how to solve the extreme case when $m = \|\mathbb{B}\| = 1$. Then we describe the solution for arbitrary value of $\|\mathbb{B}\|$.

5.1 Case $\|\mathbb{B}\| = m = 1$

We have to find such a set of polygonal models $\mathbb{B} = (B)$ with one element for which the expression in equation (4) is minimal.

$$\mathbb{B} = \arg\min_{\mathbb{B};\|\mathbb{B}\|=1} (\rho_f(\mathbb{A},\mathbb{B})) \quad (5)$$

We the definition of the dissimilarity for sets (see equation (4)):

$$\mathbb{B} = \arg\min_{\mathbb{B};\|\mathbb{B}\|=1} (\sum_{i=1}^{n} f(A_i) \min_{j=1\ldots m} \rho(A_i, B_j)) \quad (6)$$

Because $m = 1$ we can leave out the second minimum.

$$B = \arg\min_{B} (\sum_{i=1}^{n} f(A_i)\rho(A_i, B)) \quad (7)$$

Now we use the definition of model dissimilarity metric (see equation (1)).

$$B = \arg\min_{B} (\sum_{i=1}^{n} f(A_i) \sum_{k=1}^{\|V\|} w(v_k)\|v_{A_i,k} - v_{B,k}\|^2) \quad (8)$$

We swap the summations.

$$B = \arg\min_{B} (\sum_{k=1}^{\|V\|} \sum_{i=1}^{n} f(A_i) w(v_k)\|v_{A_i,k} - v_{B,k}\|^2) \quad (9)$$

Since the vertices of model B are mutually independent, we can calculate each of them individually.

$$V_{B,k} = \arg\min_{V_{B,k}} (\sum_{i=1}^{n} f(A_i) w(v_k)\|v_{A_i,k} - v_{B,k}\|^2) \quad (10)$$

The vertex weight $w(v_k)$ remains constant for individual vertex. Thus it does not affect the arg min expression. We can leave it out.

$$V_{B,k} = \arg\min_{V_{B,k}}(\sum_{i=1}^{n} f(A_i)\|v_{A_i,k} - v_{B,k}\|^2) \quad (11)$$

We use the definition of the Euclidian distance. $v_{A_i,k} = [x_{A_i,k}, y_{A_i,k}, z_{A_i,k}]$, $v_{B,k} = [x_{B,k}, y_{B,k}, z_{B,k}]$

$$\begin{aligned} V_{B,k} \quad = \quad & \arg\min_{[x_{B,k},y_{B,k},z_{B,k}]} \sum_{i=1}^{n} f(A_i)(x_{A_i,k} - x_{B,k})^2 + \quad (12) \\ & + f(A_i)(y_{A_i,k} - y_{B,k})^2 + f(A_i)(z_{A_i,k} - z_{B,k})^2 \end{aligned}$$

We can determine individual coordinates separately, because they are independent on each other. Let us consider the x-coordinate only:

$$x_{B,k} = \arg\min_{x_{B,k}} \sum_{i=1}^{n} f(A_i)(x_{A_i,k} - x_{B,k})^2 \quad (13)$$

We expand the expression.

$$x_{B,k} = \arg\min_{x_{B,k}} \sum_{i=1}^{n} f(A_i)(x_{A_i,k}^2 - 2x_{A_i,k}x_{B,k} + x_{B,k}^2) \quad (14)$$

In order to find the minimum, we find where the derivation is equal to 0.

$$0 = \frac{\partial}{\partial x_{B,k}} \sum_{i=1}^{n} f(A_i)(x_{A_i,k}^2 - 2x_{A_i,k}x_{B,k} + x_{B,k}^2) \quad (15)$$

After the derivation we get:

$$0 = \sum_{i=1}^{n} f(A_i)(-2x_{A_i,k} + 2x_{B,k}) \quad (16)$$

The second derivation is equal to $2\sum_{i=1}^{n} f(A_i)$. This is greater than 0 because all of the frequencies are positive. Thus this is a minimum. We express the $x_{B,k}$.

$$x_{B,k} = \frac{\sum_{i=1}^{n} f(A_i)x_{A_i,k}}{\sum_{i=1}^{n} f(A_i)} \quad (17)$$

We express the vertex $v_{B,k}$:

$$v_{B,k} = \frac{\sum_{i=1}^{n} f(A_i)v_{A_i,k}}{\sum_{i=1}^{n} f(A_i)} \quad (18)$$

We finally express the model B:

$$B = \frac{\sum_{i=1}^{n} f(A_i)A_i}{\sum_{i=1}^{n} f(A_i)} \quad (19)$$

5.2 Case $\|\mathbb{B}\| = m > 1$

We have to find such a set of polygonal models $\mathbb{B} = (B_1, B_2, \ldots, B_m)$ with m elements for which the expression in formula 4 is minimal.

$$\mathbb{B} = \arg\min_{\mathbb{B};\|\mathbb{B}\|=m} (\rho_f(\mathbb{A}, \mathbb{B})) \quad (20)$$

We use a dynamic programming approach:
Let $minDis[\mathbb{T}, p]$ is an array of real numbers indexed by a subset $\mathbb{T} \subset \mathbb{A}$ and an integer $p \in \{1 \ldots m\}$ defined as:

$$minDis[\mathbb{T}, p] := \min_{\mathbb{U};\|\mathbb{U}\|=p} (\rho_f(\mathbb{T}, \mathbb{U})) \quad (21)$$

This array represents the distance for all subsets of $\mathbb{A}$ to its optimal reductions of size p. If we are able to fill the array, we can find the answer to our problem in the field $minDis[\mathbb{A}, m]$. We describe an algorithm to fill the array $minDis[\mathbb{T}, p]$ with values. For $p = 1$ we can use the equation (19).

$$minDis[\mathbb{T}, 1] = \rho_f(\mathbb{T}, \{\frac{\sum_{i=1}^{n} f(T_i)T_i}{\sum_{i=1}^{n} f(T_i)}\}) \quad (22)$$

Now we can increase the value of p step-by-step and compute the values of remaining fields of the array $minDis$. We try to find a subset $\mathbb{V} \subset \mathbb{T}$ that is reduced to a single mesh during the optimal reduction. The reduction is optimal if the sum of reduction of $\mathbb{V}$ to one mesh and reduction of $\mathbb{T} \setminus \mathbb{V}$ to $p-1$ meshes is minimal.

$$minDis[\mathbb{T}, p] = \min_{\mathbb{V} \subset \mathbb{T}} (minDis[\mathbb{V}, 1] + minDis[\mathbb{T} \setminus \mathbb{V}, p-1]) \quad (23)$$

Using the algorithm above we can compute the dissimilarity during the optimal reduction. We can find the set $\mathbb{B}$ itself easily by making notes about the performed reductions (found sets $\mathbb{V}$) during the algorithm.

The time complexity of the algorithm is $O(n2^n\|V\| + 4^n m)$. The spacial complexity of the algorithm is $O(n\|V\| + 2^n m)$. The algorithm is exponential to n. It is not a principal drawback because the values of n and m are small (e.g. $n = 16$, $m = 10$) and we use this reduction only once for each set of models.

6 IMPLEMENTATION

We have implemented the algorithm in Java. For our measurement we used a computer with Intel Core Duo processor T8300 2.4GHz with 2 GB of RAM. (Our implementation is single thread only.) We measured the time needed to reduce 16 visemes to 10 visemes. Each of these visemes was represented by a polygonal model with 3000 triangles. Initial reductions for the case $p = 1$ took 2 minutes and 43 seconds. Dynamic programming reductions for the case $p > 1$ took 2 minutes and 23 seconds. Input/output operations took 12 seconds. The total time was 5 minutes and 18 seconds.

input A
input $f(A_1), f(A_2) \ldots f(A_n)$
input m
for $\mathbb{T} \subset \mathbb{A}$ **do**
    minDis[$\mathbb{T}$, 1] := $\rho_f(\mathbb{T}, \frac{\sum_{i=1}^{n} f(T_i)T_i}{\sum_{i=1}^{n} f(T_i)})$
for $p := 2$ **to** m **do**
    for $\mathbb{T} \subset \mathbb{A}$ **do**
        $currentMinDistance$ **:=** ∞
        for $\mathbb{V} \subset \mathbb{T}$ **do**
            $distance := minDis[\mathbb{V}, 1]$ **+**
                $minDis[\mathbb{T} \setminus \mathbb{V}, p-1]$
            if $distance < currentMinDistance$ **then**
                $currentMinDistance := distance$
        $minDis[\mathbb{T}, p] := currentMinDistance$
output $minDis[\mathbb{A}, m]$

Algorithm 1: Algorithm for optimal mesh reduction

We use VRML (Virtual Reality Markup Language) as our input and output format for polygonal meshes. The output from our application is compatible with XFaceEd face editor proposed by Balci in [3].

7 PERFORMANCE VALIDATION

We have compared animation of a head with unreduced set of 16 visemes and the same head with reduced set of 10 visemes. We used a textured head model with 3000 triangles exported from FaceGen [19] for our measurements and Windows Mobile phone HTC Touch Pro with OpenGL ES[9] support. An application with unreduced model required 18 seconds for startup, an application with the reduced model required only 8 seconds for startup. The speed of the model animation was 5.4 FPS for the unreduced and 12.2 FPS for the reduced version. The unreduced version was likely slowed down by memory swapping. The animation of the reduced version appeared much more smooth.

8 CONCLUSION AND FUTURE WORK

The presented method primary focusses on the head animation but it is general enough for use in other animation techniques using polygonal mesh interpolation (e.g. body, animals). In our work, we intend to investigate further reduction techniques as part of our ongoing effort of designing an open platform for development of talking-head applications on mobile phones (using the XFace framework developed by Balci [2, 4]).

ACKNOWLEDGEMENTS

This research has been partially supported by the MSMT under the research program MSM 6840770014, the research program LC-06008 (Center for Computer Graphics) and by Vodafone Foundation Czech Republic.

REFERENCES

[1] Marc Alexa, Uwe Berner, Michael Hellenschmidt, and Thomas Rieger. An animation system for user interface agents. In *Proceedings of WSCG 2001*, 2001.

[2] Koray Balci. Xface: Mpeg-4 based open source toolkit for 3d facial animation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 399–402, New York, NY, USA, 2004. ACM.

[3] Koray Balci. Xfaceed: authoring tool for embodied conversational agents. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 208–213, New York, NY, USA, 2005. ACM.

[4] Koray Balci, Elena Not, Massimo Zancanaro, and Fabio Pianesi. Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. In *ACM Multimedia*, September 2007.

[5] Uwe Berner. Optimized face animation with morph-targets. *Journal of WSCG 2004*, 12, 2004.

[6] Foreignword.
English-Truespel (USA Accent) Text Conversion Tool.
http://www.foreignword.com/dictionary/truespel/transpel.htm.

[7] ISO/IEC 14496-1:1999. *Information technology – Coding of audio-visual objects – Part 1: Systems*. ISO, Geneva, Switzerland.

[8] ISO/IEC 14496-2:1999. *Information technology – Coding of audio-visual objects – Part 2: Visual*. ISO, Geneva, Switzerland.

[9] Khronos Groups. OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. http://www.khronos.org/opengles/.

[10] Mike Krus, Patrick Bourdot, Françoise Guisnel, and Gullaume Thibault. Levels of detail & polygonal simplification. *Crossroads*, 3(4):13–19, 1997.

[11] Ladislav Kunc, Pavel Slavik, and Jan Kleindienst. Talking head as life blog. In *Text, Speech and Dialogue*, Lecture Notes in Computer Science, pages 365–372, 2008.

[12] Zuo Li, LI Jin-tao, and Wang Zhao-qi. Anatomical human musculature modeling for real-time deformation. *Journal of WSCG 2003*, 11, 2003.

[13] Radoslaw Niewiadomski, Elisabetta Bevacqua, Maurizio Mancini, and Catherine Pelachaud. Greta: an interactive expressive eca system. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1399–1400, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[14] Igor S. Pandzic and Robert Forchheimer, editors. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[15] W. Pasman and F. W. Jansen. Scheduling level of detail with guaranteed quality and cost. In *Web3D '02: Proceedings of the seventh international conference on 3D Web technology*, pages 43–51, New York, NY, USA, 2002. ACM.

[16] Kari Pulli, Jani Vaarala, Ville Miettinen, Robert Simpson, Tomi Aarnio, and Mark Callow. The mobile 3d ecosystem. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, page 1, New York, NY, USA, 2007. ACM.

[17] Thomas Rieger. Avatar gestures. *Journal of WSCG 2003*, 11:379–386, 2003.

[18] Eftychios Sifakis, Andrew Selle, Avram Robinson-Mosher, and Ronald Fedkiw. Simulating speech with a physics-based facial muscle model. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 261–270, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[19] Singular Inversion. FaceGen. www.facegen.com.