2006

# Evaluation and optimization of central vision compensation techniques

Ahmed M. El-Sherbeeny
*West Virginia University*

## Recommended Citation

El-Sherbeeny, Ahmed M., "Evaluation and optimization of central vision compensation techniques" (2006). *Graduate Theses, Dissertations, and Problem Reports*. 2742.
https://researchrepository.wvu.edu/etd/2742

# Evaluation and Optimization of Central Vision Compensation Techniques

Ahmed M. El-Sherbeeny

Dissertation submitted to the
College of Engineering and Mineral Resources
at
West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Mechanical Engineering

James E. Smith, Ph.D., Chair
Wade W. Huebsch, Ph.D.
Kenneth H. Means, Ph.D.
James V. Odom, Ph.D.
Gregory J. Thompson, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2006

Keywords: compensation, enhancement, eye, vision, warping

# ABSTRACT

## Evaluation and Optimization of Central Vision Compensation Techniques

**Ahmed M. El-Sherbeeny**

Non-costly, non-invasive, safe, and reliable electronic vision enhancement systems (EVES) and their methods have presented a huge medical and industrial demand in the early 21st century. Two unique, vision compensation and enhancement algorithms are reviewed and compared, qualitatively optimizing the view of a restricted (or truncated) image. The first is described as the convex or “fish-eye” technique, and the second is the “cartoon superimposition” or “Peli” technique (after the leading author for this research). The novelty in this dissertation is in presenting and analyzing both of these with a comparison to a novel technique, motivated by characterization of quality vision parameters (or the distribution of photoreceptors in the eye), in an attempt to account for and compensate reported viewing difficulties and low image quality measures associated with these two existing methods.

This “partial cartoon” technique is based on introducing the invisible image to the immediate left and right of the truncated image as a superimposed cartoon into respective sides of the truncated image, yet only on a partial basis as not to distract the central view of the image. It is generated and evaluated using *Matlab*® to warp sample grayscale images according to predefined parameters such as warping method, cartoon and other warping parameters, different grayscale values, as well as comparing both the static and movie modes. Warped images are quantitatively compared by evaluating the Root-Mean-Square Error (RMSE) and the Universal Image Quality Index (UIQI), both representing image distortion and quality measures of warped, as compared to original images for five different scenes; landscape, close-up, obstacle, text, and home (or low-illumination) views. Remapped images are also evaluated through surveys performed on 115 subjects, where improvement is assessed using measures of image detail and distortion.

It is finally concluded that the presented partial cartoon method exhibits superior image quality for all objective measures, as well as for a majority of subjective distortion measures. Justification is provided as to why the technique does not offer superior subjective detail measures. Further improvement is suggested, as well as additional techniques and research.

*to my discriminated, oppressed, beloved*

*brothers and sisters all around the world*

# Acknowledgements

I would like to express my appreciation to several groups of people on the academic, institutional, and personal levels. I would like to first, and foremost, show my deepest gratitude to my advisor and mentor Dr. Smith for believing in my abilities and never giving up hope; I wish you always the best, and most of all guidance. I would also like to thank Dr. Odom, who has been a great source of motivation, ideas, as well as professional and moral support and encouragement. Dr. Means, Dr. Thompson, and Dr. Huebsch are appreciated for serving on the AEC and having generously given their time and expertise to raise this work to a higher professional level.

I would like to show my appreciation for the following faculty in the MAE Department and CEMR, including, but not limited to, Dr. J. Prucz, Dr. E. Barbero, Dr. W. Myers, Dr. A. Noore, R. Dr. Hensel for her continued support, and Debbie Willis for exceptional administrative assistance. Dr. Karwowski has also offered some great mathematical and professional ideas for this dissertation. Thanks also go to Emily and Franz Pertl for their assistance. I appreciate also the comments and suggestions given by Dr. E. Peli during the brief encounter we had.

I must acknowledge as well my friends and colleagues. Especially, I express my gratitude to my dear friends Dr. H. Ammar for his relentless support on the personal and academic levels, Ali Gahtani for his sincere, unconditional friendship and assistance, Eyad Haj Said with assistance in *Matlab*®, Abdul-Rahman Al-Shammari for his assistance with the movie and other features in this research, and to Ayman Abaza with overall support.

Lastly, and most importantly, I owe the work accomplished here to my dear family, including my mother and father, brother and sister; I ask Allah to reward you for your continued support and for never forgetting me in your prayers. My beloved wife and two great sons have made all the trouble and late nights worthwhile.

# Table of Contents

# List of Tables

# List of Figures

# List of Symbols

| | |
|---|---|
| $(R, \Theta)$ | Polar position of any pixel in the original image |
| $(I, J)$ | Cartesian position of any pixel in the original image |
| $(r, \theta)$ | Polar position of any remapped pixel |
| $(i, j)$ | Cartesian position of any remapped pixel |
| $K$ | factor of proportionality |
| $k$ | remapping factor |
| $N$ | Number of pixels in an image |
| $x_n$ | Value of pixel in original image |
| $y_n$ | Value of pixel after remapping |
| RMSE | Root Mean Square Error |
| UIQI | Universal Image Quality Index |

# Introduction

The presented Ph.D. work falls within the growing field of Electronic Vision Enhancement Systems (**EVES**) that requires development in such areas as vision optimization as well as the application in different forms of image enhancement, such as selective area magnification and edge highlighting [58]. In simpler terms, EVES carry out **image warping**, a geometric process that involves transforming the dimensions and/or components of an image, in order to achieve correction (such as in low-vision applications), or to magnify a section(s) of the image (such as in remote sensing or surveillance), both providing an overall increase in the quality of the image, or **image enhancement** [56].

Image warping is closely related to **image compensation**; the latter is usually used to address geometrically deforming an image, which necessitates that aspects such as selective area magnification be compromised with the overall size, homogeneity, clarity, and illumination of the image. As a result, the EVES technology strives to achieve an application-dependent optimization, involving the factors governing geometric warping (or compensation) to produce the desired shape, versus the realistic appearance and minimized distortions in the output image. Consequently, this balance of variables, also referred to as **image optimization**, involves the challenge of dealing with both analytical (or geometric) and physical (or physiological) aspects of the **human perception** to images.

### *Image Optimization*

A geometric optimization is achieved through ensuring that the image warping is carried out with a **"good" match** between the original (or **source**) and final (or **target**) images. A good match is most strongly governed by how much of the source image is conserved in the warping process. A "bad" match, for example, involves warping a section or a group of pixels from the source image to a location that does not exist in the target image. Another discrepancy is the existence of **"holes"** or discontinuous regions in the target image resulting from the use of an improper warping function or parameter. Such discontinuities, however, are usually treated using filters. Filters are used to enhance the appearance and quality of the target image, such as brightness, contrast, color, contour, etc.

Geometric warping is also governed by the **distortion** of the target as opposed to the source image. Distortion involves the relationship between each pixel and its adjacent pixel in the target image, usually determined by scaling, translation, and rotation. The more each of these factors becomes significantly apparent in the target image, the more distorted it becomes. A preferred warping is thus conformal (or **conformal mapping**) if it involves conserving the angle between the adjacent pixels in the target and source images, usually imposing a uniqueness, reduced distortion, as well as a higher quality, more realistic image.

The human perception aspect is another very significant aspect affecting image warping. Although producing a distinct analytic function is much more difficult to attain physiologically, ophthalmologists have been able to find significant trends or models for the distribution of the retinal cells [30], determined by quality vision or "macular" cells. This spread is denoted as the spatial viewing resolution (or **spatial retinal density**), and humans have been found to have the highest spatial density of retinal cells (more than 5,000 cells/mm$^2$) within the first 10 mm or so of the retinal eccentricity (measured from the front or dead viewing center), as shown in Figure 1; this corresponds to the central viewing area of an image [47].

As a result, **central vision optimization** involves achieving both an optimized geometric warp having a good match with minimal distortion, as well as visibility within the region of the highest spatial resolution. Therefore, the objective of this dissertation is to examine this geometric-physiological –motivated– relationship through the optimization of the factors governing the geometric warping leading to a target image that can be best viewed by humans. This is achieved through applying and evaluating techniques used to maximize the viewing of an image by introducing as much of the peripheral sections into the central portions of an image.

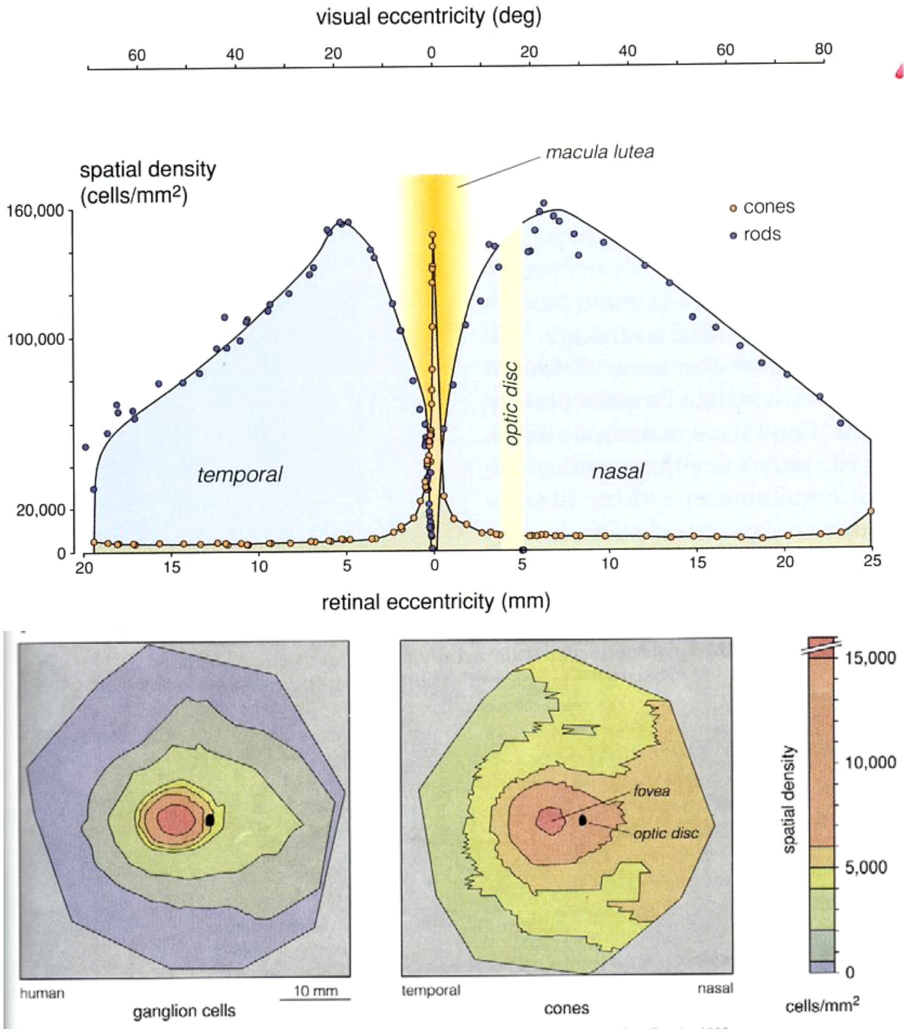


**Figure 1 Spatial density distribution in human retinal cells; top: chart shows how most of the human spatial density lies within a few degrees of the visual eccentricity (highlighted area); bottom: high-quality vision is demonstrated for human ganglion cells and cones concentrated in the red, foveal cells [47, pp. 198 and 203, respectively].**

### *Overview of Dissertation*

In order to better understand the concept of central vision optimization the reader is introduced to the basic fundamentals, types, applications (both medical and industrial), as well as recent advances in image warping and compensation. Subsequently, the dissertation evaluates the applications of various image compensation techniques in central vision optimization.

Two techniques are presented that demonstrate how the central image can be preserved while showing as much of the surrounding (or invisible) image, (1) the **convex image warping scheme** [2,33], and (2) the **cartoon superimposition scheme**, or "Peli" technique, after the leading author for this research [7,36,42]. The convex scheme is an analytical warping technique that is created by projecting the 2-D image onto a convex 3-D surface, and projecting the surface back to a smaller 2-D plane. The analytical formulation is based on the spatial retinal distribution. In doing so, the target image is smaller in size, its central portion is almost conserved in size (with minimal distortion), while the peripheral image (of less visual significance) has a gradually minimizing shape.

The cartoon superimposition technique involves two images, the source image, truncated down solely to its central, undeformed portion, as well as a contour superimposition of the image periphery onto the truncated image. Thus, the basic idea regarding both methods is that of fitting a larger size image into a reduced size (either for low-vision purposes or for offering selective area magnification). However, the convex scheme involves warping the entire image, whereas the cartoon superimposition or "Peli" method retains much of the source image while offering a hint of the surrounding.

Furthermore, in addition to the two image compensation techniques mentioned above, a novel technique is presented by the author of this work, (3) the **partial cartoon superimposition** (or, for short, partial cartoon) technique, where only the invisible image to the left and right of the truncated image is superimposed on the respective sides of the truncated image, but only on a partial basis as not to distract the central view of the image. This technique can be viewed in a sense to evaluate and combine both the analytical features of the cartoon technique as well as the physiological attributes of the described human spatial resolution.

The dissertation, thus, aims at evaluating and comparing each of these image compensation techniques, which are presented in detail in the *Methodology* section. This comparison is achieved both analytically using the Root-Mean-Square Error (RMSE) and the Universal Image Quality Index (UIQI) measures, as well as subjectively, using 115 surveyed volunteers. This information is also used to assess the suitability and superiority of each method over the other, and the environment or application where each may be applicable. Table 1 presents a summarizing research matrix of these variables, which will be revisited in the *Conclusions* section.

**Table 1 Overview of research objectives, including image compensation methods presented in this dissertation, and various theoretical, analytical, and experimental variables and parameters to be treated among each method; this table will later be addressed in the *Conclusions* section.**

| Comparison | Convex Method | "Peli" Method | Partial Cartoon Method |
|---|---|---|---|
| ***Basis/Grounds of Technology*** | ? | ? | ? |
| ***Objective Results*** | ? | ? | ? |
| ***Subjective Results*** | ? | ? | ? |
| ***Customizability/ Variability*** | ? | ? | ? |
| ***Applications*** | ? | ? | ? |

# Literature Review

In order to better understand the means used to select, generate, and optimize image warping and compensation, the different methods and classifications involved with image warping are first outlined, as well as methods used for choosing and assessing the appropriate warp, including measuring distortion. This also provides an important basis both to understand the terminology involved with image warping, as well as an introduction to the adopted techniques in the *Methodology* section. Literature reviewed in this section has been acquired from different journal articles, textbook material, and recent inventions.

Image warping techniques are first classified into parametric and non-parametric types. It has been shown that parametric types (especially perspective mapping) produce smoother target images. Conformal mapping is also explained, and it is clarified as to which of the parametric functions best suits the conformality conditions. Applications of image warping in medicine and industry are then demonstrated, including low-vision, tissue-modeling, surveillance, mapping, image projection, etc. Finally, a highlight is provided for recent technologies and publications that discuss different image warping applications, many of which have motivated the research presented in this dissertation.

### *Image Warping Techniques*

Image warping is also referred to as image remapping, image morphing, and spatial image transformation. It involves a 2-D geometric transformation that maps pixel positions in one image plane to positions in another plane [17,29].

A warping (as shown in Figure 2) is a pair of two-dimensional functions, *u(x, y)* and *v(x, y)*, which map a source position *s(x ,y)* in one image, where *x* denotes the column number and *y* denotes the row number, to a target position *t(u, v)* in another image. A forward warping or transformation function T is thus one that relates the source and target coordinates in the following manner, $t_{u,v}$ = T($s_{x,y}$), whereas an inverse transformation, which means mapping the target image into the source image, is described by $s_{x,\mathrm{y}}$ = $\mathrm{T}^{-1}$($t_{\mathrm{u},v}$) [11].

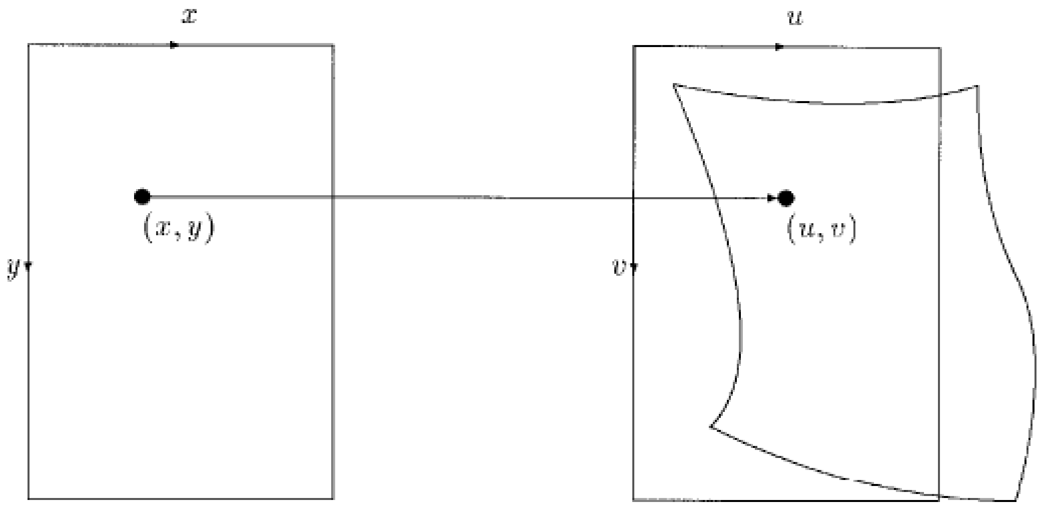


**Figure 2 Representation of image warping using a single-pixel image [17, p. 156].**

Inverse mapping is preferred over forward mapping for a number of reasons [11], including creating a warped image with a guarantee that all warped pixels are mirrored into an existing location in the new image. There have been many approaches to finding an appropriate warp, but a common theme is that achieved by a tradeoff between the matching and smoothness of the warp, as explained earlier.

## Warping Optimization Factors

The choice of warp is a compromise between a smooth distortion and one which achieves a good match. Smoothness can be ensured by assuming a parametric form for the warp, such as the affine transformation (**parametric warping**) which is described in the next section, or by penalizing roughness (**non-parametric warping**). Depending on the application, matching might be specified by points which must be brought into alignment, by local measures of correlation between images, or by the coincidence of edges [17].

Ensuring a good correlation (or warping) between the original and warped image is emphasized through optimizing image informational content as well as improving image quality. Image content of information is stressed through physiological properties of the human visual system (HVS), e.g. by correlation to the maximum visual eccentricity of the retinal spatial density, as described in the introduction.

### Image Distortion

Image quality is best described through an evaluation of image distortion. Distortion is a rather general term used to describe optical aberration. They include coma (images blurred due to non-axial ray points), astigmatism (ray points without a focus), chromatic aberrations (due to varying refractive wavelengths), spherical aberrations (due to varying refraction at the thin margins versus the thick central portions of the sphere), curvature of field (due to an image not being projected onto a plane parallel to the lens), and geometric distortion, where magnification varies between the center and the periphery of the image. Distortion due to greater relative central magnification is referred to as barrel-formed distortion (like a convex lens effect), while the opposite is called pin-cushion-formed (or concave) distortion [8].

Evaluation of distortions is based on quantifying some deviation measure of the warped versus the original image. Such measures are classified into pixel difference-based, correlation-based, edge-based, spectral-based, context-based, and human visual system-based measures [3]. An example of a pixel difference-based method is the mean square error (MSE) or more commonly known as the root mean square error (or **RMSE**).

The RMSE is defined as the square root of the mean of the squared difference among each respective pixel value in the original and warped image [50]. The RMSE is considered one of the most commonly used methods of evaluating distortion (or image quality), since it is nonadaptive (i.e. has general applicability and practicality) [4,53], offers computational simplicity [3], and provides a clear indication of error minimization [38]. It is, thus, the distortion measure selected in this dissertation, and will be elaborated upon in the methodology. The Universal Image Quality Index (**UIQI**), is another considered general image quality index [52,54].

## Parametric Warping Methods [17]

Figure 3 shows a hierarchical summary of the most frequently encountered parametric transformations, where each arrow shows progression from a more specific warping method to a more general one. The figure shows the different parametric forms involving translation (motion of the image), scaling (magnification or demagnification of the image), and shearing (horizontal or vertical motion of one part of the image relative to another).

After a brief definition and explanation of each method's form and applications, the reader is presented with a description of the method having the best applicability for central vision optimization, including lowest distortion and highest conformality. It is noted, as indicated by Glasbey and Mardia, that the choice of image warping technique should be as specific as possible to suit the desired application [17]. Therefore, emphasis is given in the following subsections to parametric mapping, including the procrustes, affine, perspective, bilinear, and polynomial transformation, due to their higher relevance (especially the perspective transformation) to central vision optimization. For completion, this is followed by a brief mention of non-parametric mapping transformations, as well as conformal and texture mapping.

### Procrustes Transformation

The Procrustes transformation is used if there is some change in magnification (or scaling) between the images, along with translation, and a rotation of $\theta$ degrees. This is referred to as a pose in computer vision. A value of $c = 1$ corresponds to no change in magnification, whereas $c > 1$ is an enlargement, and $c < 1$ is a shrinkage.

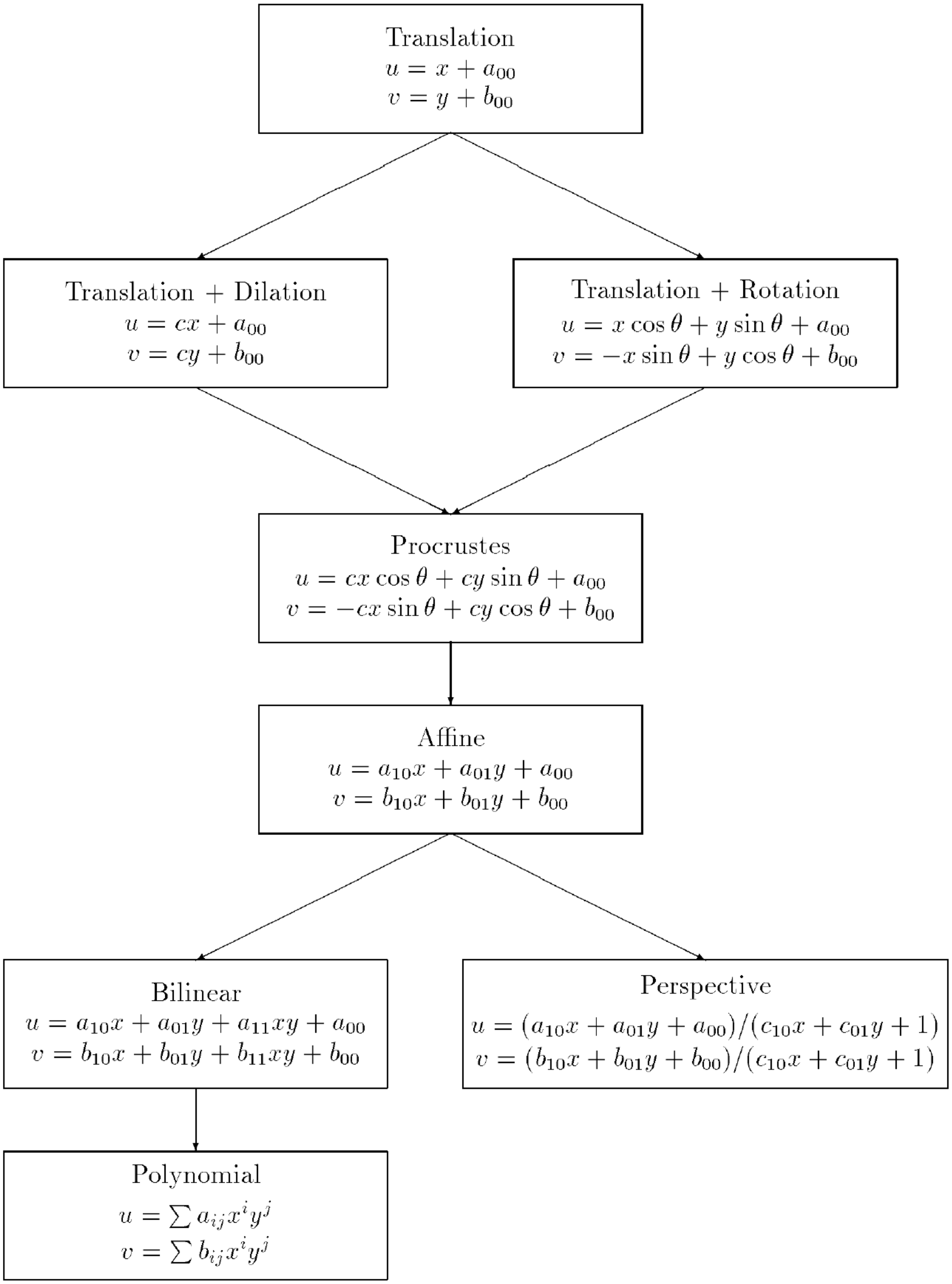

Translation
$u = x + a_{00}$
$v = y + b_{00}$
Translation + Dilation
$u = cx + a_{00}$
$v = cy + b_{00}$
Translation + Rotation
$u = x\cos\theta + y\sin\theta + a_{00}$
$v = -x\sin\theta + y\cos\theta + b_{00}$
Procrustes
$u = cx\cos\theta + cy\sin\theta + a_{00}$
$v = -cx\sin\theta + cy\cos\theta + b_{00}$
Affine
$u = a_{10}x + a_{01}y + a_{00}$
$v = b_{10}x + b_{01}y + b_{00}$
Bilinear
$u = a_{10}x + a_{01}y + a_{11}xy + a_{00}$
$v = b_{10}x + b_{01}y + b_{11}xy + b_{00}$
Perspective
$u = (a_{10}x + a_{01}y + a_{00})/(c_{10}x + c_{01}y + 1)$
$v = (b_{10}x + b_{01}y + b_{00})/(c_{10}x + c_{01}y + 1)$
Polynomial
$u = \sum a_{ij}x^iy^j$
$v = \sum b_{ij}x^iy^j$

**Figure 3 Display of different parametric transformations, from specific to general; this serves in the choice of the appropriate transformation, which should be as specific as possible [17, p. 157].**

### Affine Transformation

The affine transformation is a generalization of the Procrustes transformation, allowing different stretching along rows and columns of an image and shearing. It is the most general linear transformation. Thus, an orthogonal pair of directions in the *x*-*y* image remains orthogonal in the *u*-*v* image, and the transformation either stretches or shrinks in these two directions.

### Perspective Transformation

Perspective transformation arises if a planar object is viewed from a fixed point in space, making it useful in various industrial applications. It is a non-linear transformation requiring eight scaling ($a_{10}$, $a_{01}$, $b_{10}$, $b_{01}$, $c_{10}$, and $c_{01}$) and translation ($a_{00}$ and $b_{00}$) parameters for 2-D warping (see Figure 3 for complete formulation). Perspective transformation is the most general transformation which maps straight lines at all orientations to straight lines, as do the previously considered transformations. In addition, it preserves conic sections, including circles, ellipses, parabolas and hyperbolas.

Furthermore, the perspective transformation is **functionally invertible**, i.e. the inverse transformation $(u, v) \rightarrow (x, y)$ has the same functional form, and it is (therefore) arbitrary as to which image is chosen to be mapped on to the other one. Therefore, the transformation is guaranteed to be **bijective**, i.e. it is impossible for folding to occur, where two points in the *x*-*y* image are mapped to the same point in the *u*-*v* image. This is the only parametric transformation that possesses this property [17]. The concept of forward and inverse mapping is further illustrated in the following paragraph.

As shown in Figure 4, grid lines converge in the target image for forward perspective transformation, while the grid lines converge in the source image for an inverse transformation. The latter technique offers the advantage that it uses interpolation to regenerate the intensity of target points from nearby source pixels. Forward transformation, however, uses the more complex process of numerical integration to generate the intensity of a target pixel, since mapped points do not lie on distinct pixel locations. An inverse transformation offers the added advantage of avoiding holes, since it results in uniquely mapped target pixels, as well as a unique interpolation kernel, which guarantees a fixed neighborhood of source pixels [11]. Note, this discussion is the basis for the “fish-eye” or convex technique presented in the methodology.

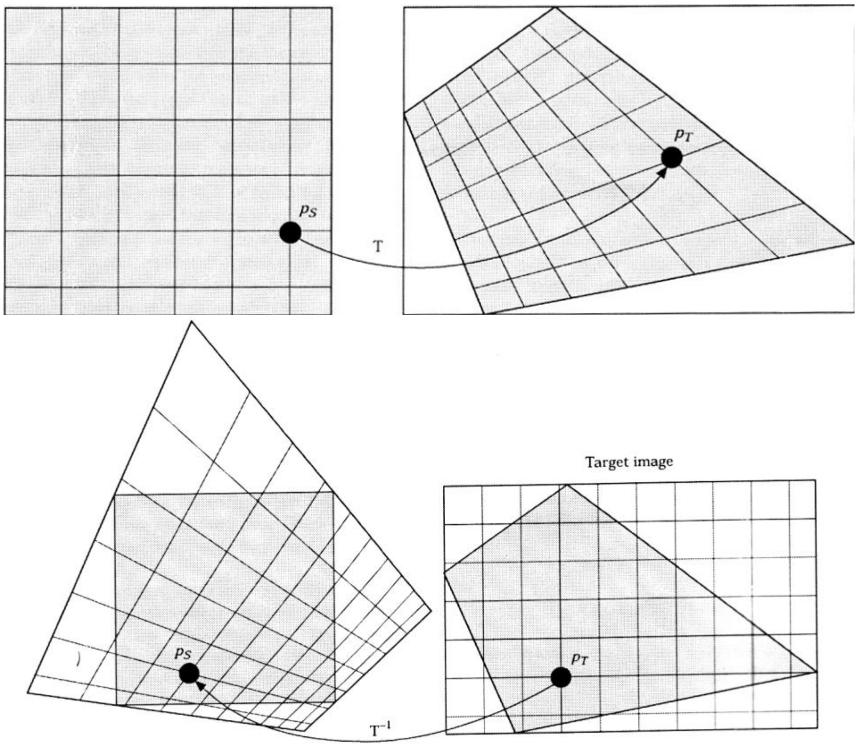


**Figure 4 Forward (top) and inverse (bottom) perspective transformation between source (left) and target (right) images; gray areas represent valid image points [11, p. 418].**

## Bilinear Transformation

Bilinear transformation is another eight-parameter generalization of the affine transformation, but with different properties. Straight lines in three particular directions are preserved, including lines parallel to either x- or y-axes. The transformation is, thus, not rotationally invariant, i.e. it has the disadvantage that if both images are rotated, the warping transformation between them changes. Also, this transformation, and those to follow, is not guaranteed to be bijective.

## Polynomial and Other Transformations

Polynomial transformations of order p include quadratic, biquadratic, cubic and bicubic. Polynomials of third and higher order are used in the registration of remotely sensed images, as well as matching landmarks for spiral structures [17].

### Non-Parametric Warping Methods

Parametric transformations do not perform good matching in the presence of local distortions. Piecewise affine transformations, for example, offer an alternative to polynomials in generalizing affine transformations. As no smoothness constraints are considered, such transformations can therefore be very rough, i.e. in situations where the image need not be smooth or continuous [17].

### Conformal and Texture Mapping

The literature review has already covered some important aspects of warping, like bijectivity, smoothness, etc. This section examines conformal mapping, its meaning, significance, and its application in image warping as demonstrated in recent literature describing texture (or 3-D) mapping, which will also be treated in the section regarding applications of image warping. The importance of these aspects is emphasized in the *Methodology* section, as the appropriate methods are consequently selected for comparison and experimentation.

Conformal mapping is a unique warping since it preserves the local angles between pixels (after warping), and is accordingly referred to also as an angle-preserving transformation. Figure 5 shows an example of a quartoid, a conformal analytic function given by $f(x,y)=\frac{(x^2+y^2)^2}{\alpha^3}$, where $\alpha$ = 4 [24]. Mathematically, conformal mappings have additional properties, including:

- A complex analytic (or parametric) function f(z) is conformal at any point where it has a non-zero derivative,
- A conformal mapping is locally isotropic, i.e. an infinitesimal area element is magnified equally in all directions,
- A conformal mapping preserves infinitesimal angles,
- The real and imaginary parts of the map function are harmonic conjugate functions, i.e. they intersect orthogonally, and finally,
- The Riemann mapping theorem guarantees the existence and uniqueness of conformal mappings between regions [14, 39].

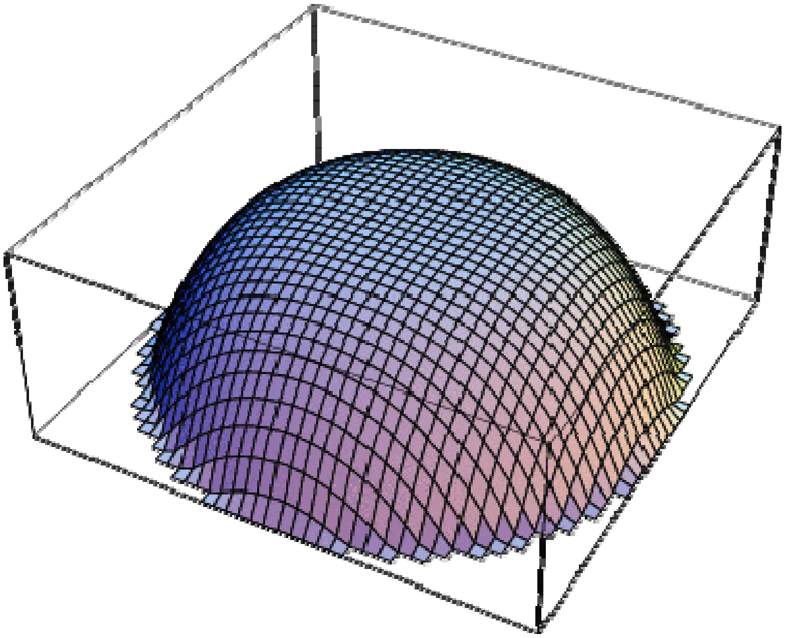


**Figure 5 Quartoid: an example of a conformal analytic function [24].**

The original idea behind texture mapping and texture synthesis was initially that of enhancing an image by adding 2-D visual effects or a wrapping to it [51]. Recent applications include the ability to generate conformal, bijective spatial transformations of a 2-D image convex polygon and some types of curves (e.g. circles) [13]. Challenges include the proper choice of routines to produce the appropriate texture, geometrically mapping texture spaces into objects, filtering the image, and finally producing an image of chosen quality with minimal distortion. Texture mappings also offer the advantage that they can be combined with non-conformal mappings, like perspective mappings (similar to the process proposed by Juday et al. [26] and Amerijckx et al. [2]).

Heckbert in his master's thesis describes the "ideal, space variant antialiasing filter" used for parametrization and filtering necessary to remove aliasing during texture mapping [21]. Aliasing is the improper representation of an image that occurs due to insufficient sampling along a space axis. Such a problem is solved using appropriate low-pass filters which act to smooth the appearance of the image.

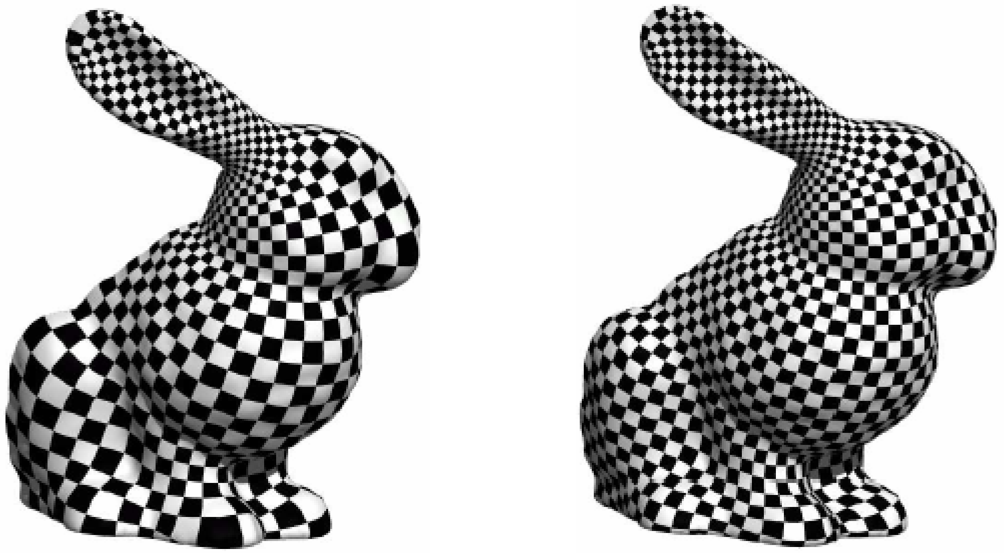


**Figure 6 Two applications of texture mapping in enhancing the view detail of an object; left: conformal texture mapping based on "global parameterization;" right: texture mapping using "quasi-isometric parameterization" [51, p. 801].**

Recent advances in texture mapping include the concept of conformal parametrization by Wang et al. (as displayed in Figure 6), which involves creating and optimizing texture mapping algorithms for surfaces with arbitrary (including non-trivial or complex) shapes [19,25,51]. The authors describe how this method simplifies 3-D surface texture synthesis into that of a 2-D image synthesis. Other contributions of this method involve the combination of preserved orthogonality and size of the mapping.

## *Applications of Image Warping*

Image warping arises in many image analysis problems. Primary applications include remote sensing, medical imaging, as well as computer graphics [11]. This dissertation is concerned with the application of warping in vision enhancement, which includes fields like low-vision, and surveillance. Other examples include removing optical distortions introduced by a camera or a particular viewing perspective, to register an image with a map or template, or to align two or more images [17]. For example, matching is important in reconstructing a three-dimensional shape from a series of two-dimensional sections, as discussed in the following sections.

### Medical Applications

There are various applications of image warping in medicine. This section highlights recent techniques used in improving low vision. Other applications include improving X-rays and tissue imaging.

### **Image Warping and Low-Vision**

As documented in eight reviewed publications, Loshin and Juday conducted research (at NASA, Johnson Space Center), creating warped, fish-eye images designed to compensate for loss of peripheral vision (e.g. due to Retinitis Pigmentosa or RP) [26,28,33], although later research seemed to focus almost exclusively on central vision losses (e.g. Age-Related Macular Degeneration) and improving reading skills for such patients [22,27,34,35]. The authors also considered integrating their “programmable remapper” into a video display [12].

Figure 7 shows several warps for a scene of a kitchen that is not all viewable due to the RP disease, as well as different warping schemes generated by the authors through which the same image could be modified so that the central image suffers minimal reduction in size while the periphery is also visible. Further analysis to this research is discussed in the *Methodology* section. The authors, however, do not discuss results of any experimentation performed in this research, including approaches that may have been taken to analytically or experimentally evaluate the warped images, a process undertaken in this dissertation.

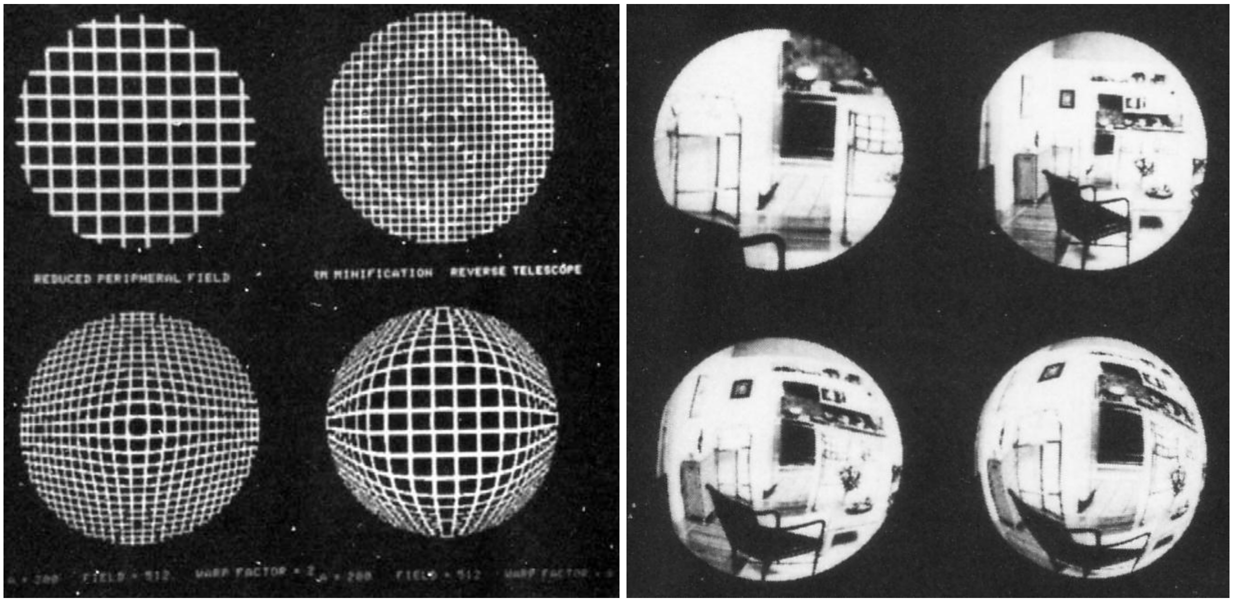


**Figure 7 Image as seen by a Retinitis Pigmentosa patient and several image-fitting algorithms (or meshes) to compensate for loss of peripheral vision; left: Juday-Loshin warp diagrams; right: corresponding images as seen by RP patient [33, pp. 394-395].**

Amerijckx et al. used a similar, more recent technique for central image magnification via the use of two CCD cameras which cluster images onto the healthy part of RP and AMD visual fields [2]. The authors describe the implementation of this technique into a microchip. However, as with Juday and Loshin, there is no reference to experimentation that may have taken place to test the efficiency of the device or technique. Furthermore, the authors comment that the warped image (as shown in Figure 8) is highly distorted, although they also note that they expect patients with RP could gradually accommodate their vision to interpret such warped images [37].

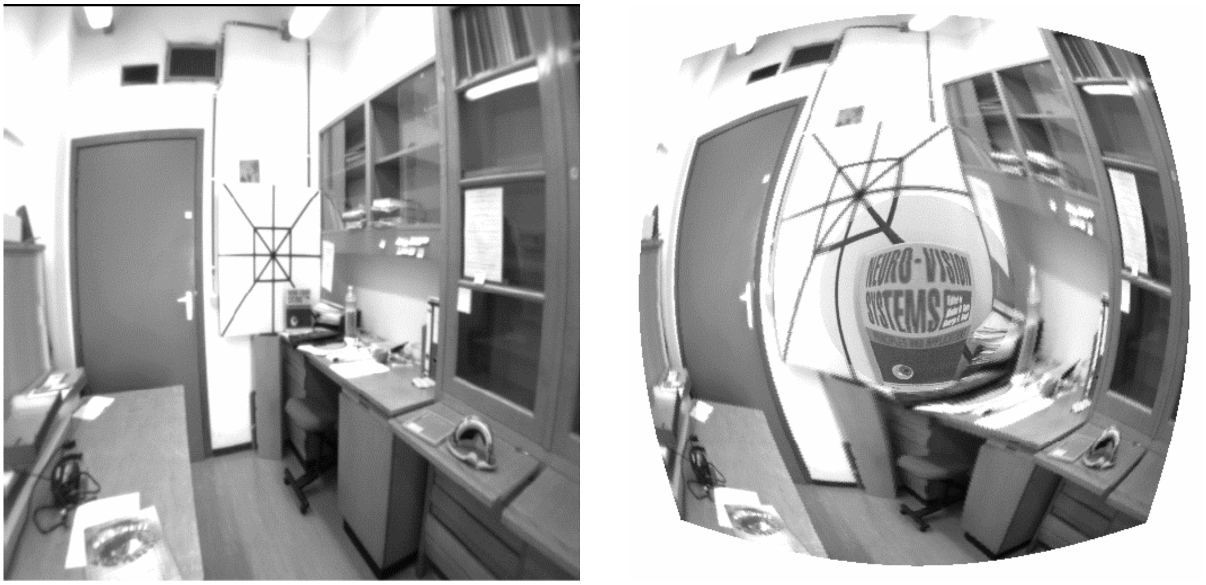


**Figure 8 Amerijckx et al. technique for central vision magnification with a remapping algorithm similar to that of Juday-Loshin but using two CCD cameras; left: original image; right: warped image, with highly enlarged central image, yet with also a highly distorted overall image [2, p. 381].**

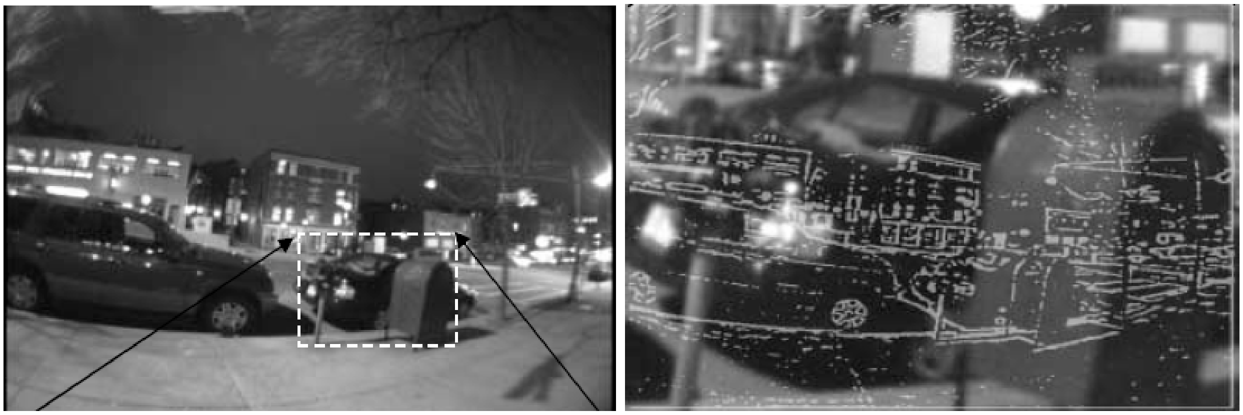


**Figure 9 Peli et al. augmented-view concept; left: full view of a street scene taken at night, where the white rectangle represents the size of the field of the display (e.g. that of an RP patient or other minimal view application); right: edge-contour image of the same scene (on the left) superimposed on the natural view (shown here as the car and mail box in the edge-contour image) [7, p. 297].**

Another ongoing research effort is the Peli et al. augmented vision, see-through HMD (helmet-mounted display) which superimposes a low-resolution, cartoon (or contour) image of the surroundings over the patients' natural view [36,42]. This technique is demonstrated in Figure 9, whereby a person with a tunnel vision or reduced field of vision would only be able to see the highlighted car and mailbox, etc. The technique allows the subject to see an edge-contour view of the entire image superimposed onto the subject's restricted view [7]. The technique additionally allows for a night vision feature. However, the technique is more suited to outdoors and dynamic applications, where only a hint of the general view is desired.

## Image Warping and Histology

Recent interest has developed in producing 3-D images of body tissues, as the case with the study by Miller et al., which describes a new imaging technique used to portray both the structure as well as mechanical properties of oculomotor tissue [40]. The authors note that a tradeoff must be produced between increasing the spatial resolution of the warped image (as compared to the original image) versus the degraded realism of the image. Image warping is also useful in correcting distortions caused by thin sectioning and subsequent histological processing, as shown in Figure 10.

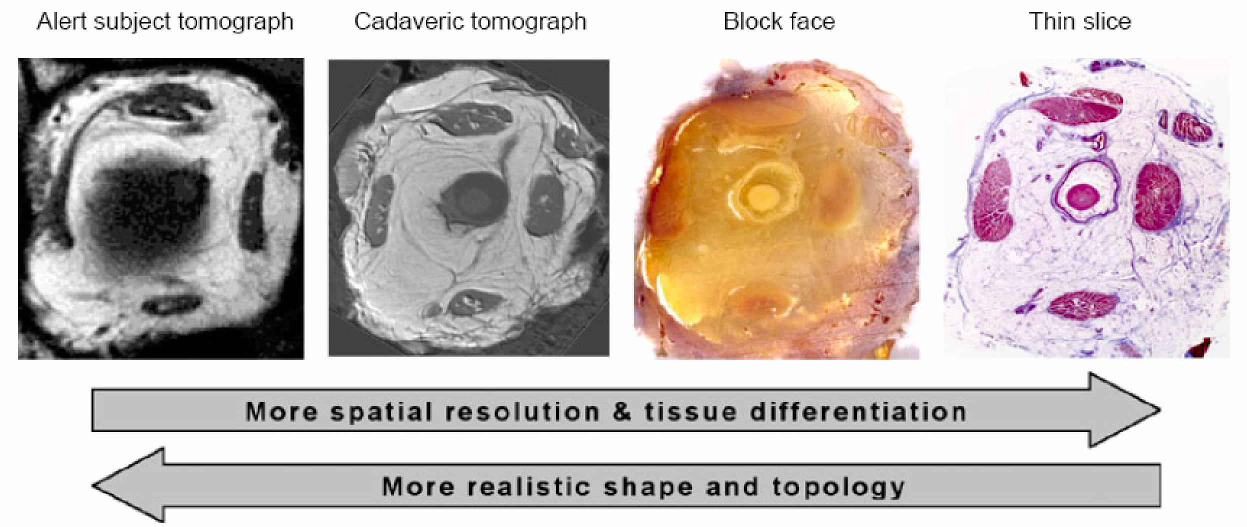

Alert subject tomograph
Cadaveric tomograph
Block face
Thin slice
More spatial resolution & tissue differentiation
More realistic shape and topology

**Figure 10 Warping in medical imaging, represented as increasing spatial resolution of tissue image at the expense of decrease in the realistic appearance of the image [40, p. 243].**

### Industrial Applications

Apart from medical applications, image warping has served in a host of other imaging applications, including surveillance, navigation, image projection, and shape recovery. Lie and

Toet describe how a fisheye warping (as that described with Juday-Loshin) can be used to optimize the view of pilots in various surveillance and navigation applications [31].

The importance of surveillance is described in applications where monitoring of a large field of view is required, along with the simultaneous focus onto a small region of high priority. Such regions of interest are better displayed onto the general view using conformal fisheye warping described by the authors. A similar warping procedure is used in navigation where a relative position of a location on a map may need to be emphasized (or the "local layout"), while retaining the general map view (the "global layout").

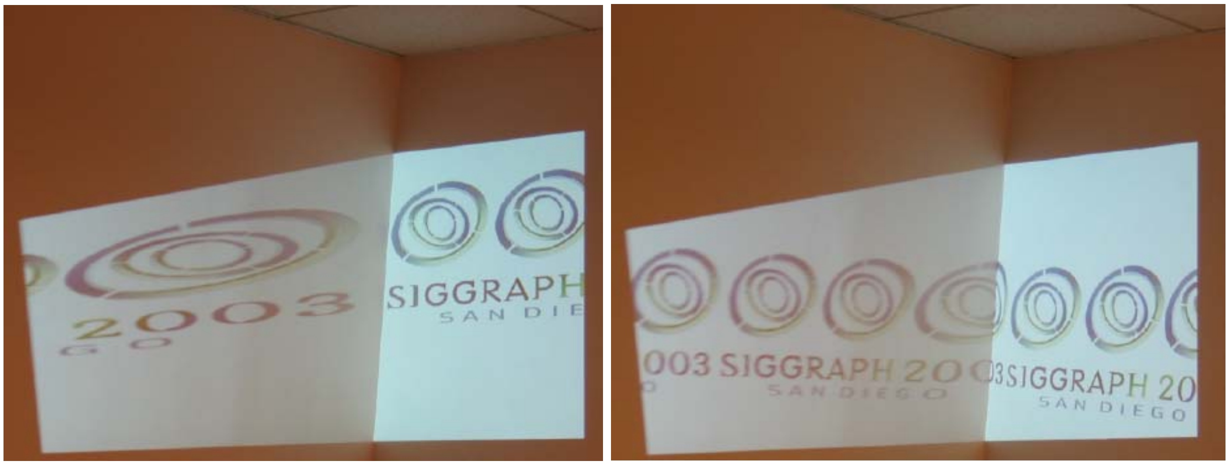


**Figure 11 The use of image warping in correcting distortions produced by projecting images from an LED source onto a wall; left: skew image produced by uncorrected projection; right: same projection with correction algorithm described by Raskar et al. [46, p. 811].**

Raskar et al. describe a new technique for "adaptive" projection, which acts to enhance images and correct skews and distortions of projections on various 3-D surfaces using conformal texture mapping, an example of which is shown in Figure 11. The authors describe shape-adaptive display variations (through horizontal and vertical constraints), object-adaptive display (through the augmentation of objects), planar display, and curved display (using a cluster of projectors) [46].

Warping is also applied in the field of face recognition and recovery, a feature of particular importance in identification and security applications. As demonstrated in Figure 12, warping algorithms can be applied to an existing 2-D image to attempt the recovery of a target subject (or 3-D profile) when compared to a profile or prototype of an existing shape, shade, or intensity [60].

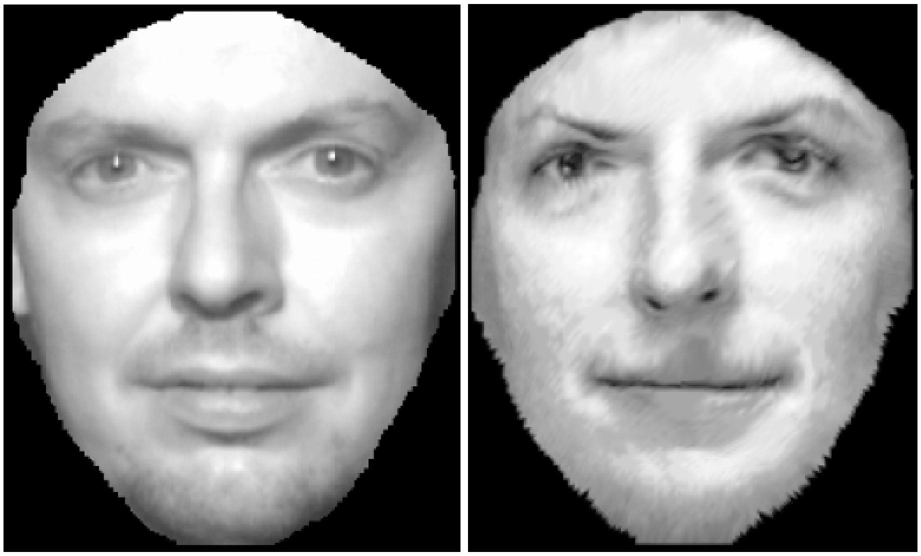


**Figure 12 The application of image warping in shape recovery and recognition [60, p. 18].**

A similar application is that of image-based rendering (IBR), a term usually used to refer to different views or viewpoints of an image created by combining several 2-D images (using image warping) in order to reconstruct (i.e. a 3-D image) or estimate certain features of an image, e.g. the depth of the image [23]. The technique consists of three steps, “a preprocess for edge-pixel extraction, inverse warping from the primary image, and hole filling from the remaining reference images” [59]. A Ph.D. dissertation by Agrawal involves a novel, non-linear warping scheme for reconstructing 3-D images from two or more planar views, e.g. from three views in a right-angled configuration. He explains that one of his methods involves using spheres to perform camera calibration [1].

### *Recent Image Warping Technologies*

This section is dedicated to describing some recent technologies and studies involving image warping. A patent by Georgiev describes a computer-controlled system that performs a conforming mapping of one enclosed-contour area into a second enclosed-contour area [16]. The steps involved include generating the first source area contour using a Bezier curve, modifying the contour shape to match that of the second area, generating an analytic function to match this process, and subsequently using this function to warp the source image into the destination image.

A master's thesis by Lee also uses a Radial Transform, forward mapping to map images of arbitrary shapes onto each other [29]. Of particular interest is also a relevant Ph.D. dissertation by Gao where the author creates a warping between two given 2-D images using only a few "anchor" points, a process referred to as "work minimization" [15]. The aim of this research is to produce a tradeoff between minimizing the processing time (by minimizing warping computations) and optimizing the quality of the warping. Another Ph.D. dissertation by Wolberg proposes several new techniques (like the separable image warping technique) that avoid many limitations, visual artifacts, distortions, and computational difficulties encountered in several of the parametric and non-parametric transformations previously discussed [57].

Another technology uses a camera to directly produce a warped image from an input or captured image. The warping algorithm is encoded onto a card in the camera, which reads and maps each pixel from the input image onto a corresponding location in the warped image [49]. A patent by Hamza consists of a system where pixels from an input image are assigned intensity values and weights, representing the horizontal and vertical dimensions in an output lattice for each pixel. The weight and intensities are used to map each input pixel to the output lattice in a single processing pass [20].

Popescu et al. describe another technology which involves the sampling of a reference image in its 3-D space into several tiles, each tile defined by a corresponding set of image samples. The relationship between each image sample is determined “based on the relative curvature of a surface of the corresponding tile at the sample coordinates” [44].

Sample warping is also adopted by Lohmeyer et al., who invented a warping system that increases the sampling rate of a sampled image above the Nyquist rate. The Nyquist rate is defined as the minimum sampling rate (in samples per second) needed to avoid aliasing [5], which was described in the section on *Conformal Mapping*. Lohmeyer also applies filtering to the image to avoid distortions that occur during the warping process. The “upsampled” image is then warped using a bilinear interpolator. Finally, the upsampled, warped image is “down-sampled” to the same resolution as the input image to produce the warped image [32]. An earlier patent also describes a similar process, achieving real-time warping in one or two dimensions using Nyquist bandlimiting [18].

### *Literature Review Summary*

This section has been designed with the intention that the reader can appreciate the basic concepts and definitions involved with image warping, including various image remapping factors (such as image content and quality) and image distortion factors (such as RMSE and UIQI). Also, various classifications of image warping were covered including parametric, non-parametric, and conformal transformations, with various applications, technologies (e.g. industrial, low-vision, and surveillance), and recent studies showing where each technique may serve, including enhancing central vision.

Subsequently, the greater goal is to show the reader that there are currently two technologies or methods targeted by researchers aimed to maximize central vision. The first (convex or "fish-eye") method involves shaping a remapped projection of the image to fit a limited field of view, retaining the details of the image, at the cost of distorting the image, as described by Juday et al. [26] and Amerijckx et al. [2]. Peli et al. [42] demonstrate an alternative method which involves superimposing a generated cartoon or contour of the world image *completely* onto the truncated field of view, at the drawback of high contrast and dominance of the contour image over the truncated image.

The remaining sections are aimed at describing a third technique generated by the author of this dissertation (the partial cartoon technique), motivated by these techniques (the convex and "Peli" techniques) as well as properties of quality photoreceptors (as described in the *Introduction*) to superimpose a *partial* cartoon into the truncated image, while clearing the central view, attempting to avoid the limitations of the these two techniques.

# Text of Investigation

The aim of this research, as outlined earlier, is to investigate, compare, and evaluate the application of various image compensation techniques in central vision optimization, as well as to introduce a new image compensation technique, the “partial cartoon superimposition” technique. This investigation is achieved through an objective and a subjective incorporation of the proposed techniques to test their efficiency and significance.

### *Research Rationale*

The literature surveyed in this dissertation serves to explain and portray both the value and diversity of applications in which image warping is applied. However, no known study has gone into evaluating and comparing these various techniques, specifically those concerned with central image optimization, which has been shown to serve in a myriad of applications, like low-vision, surveillance, mapping, etc. Research on compensating a minified field of view (**local FOV**) for that of a larger, **global FOV**, has worked to achieve a warped image possessing a magnified, clearer view of the image, with minimal distortion.

This has motivated the direction of the dissertation, which focuses on evaluating the only two major, distinct techniques (as highlighted in the *Literature Review* section) for enlarging and enhancing central vision. The first technique (as that described by Juday et al. [26] and Amerijckx et al. [2]) is that of enlarging the central portion of the image, while fitting the remaining portion of the original image in the remaining (or **truncated image**) space, which is referred to as the **convex projection technique** from this point on in the dissertation. It possesses the unique property that it is based on the perspective warping algorithm, which will be shown, to result in a warp of least discontinuities and distortions. The other technique (proposed by Peli et al. [42]) involves leaving the truncated image undistorted while introducing a contour of the surrounding into the truncated image. The latter technique is referred to as the **"Peli" technique** from this point forward.

The **partial cartoon** technique is a novel, adapted version of the "Peli" technique, where instead of introducing the complete image into the truncated image, only the invisible image to the immediate left and right of the truncated image is overlaid on the respective sides of the truncated image, as described below. Furthermore, the two cartoons are introduced only partially, leaving the central portion of the original truncated image clear for maximizing the central (quality) view.

### *Research Objectives*

The contribution of this research, therefore, lies in determining the means of optimizing central vision through a numerical (or objective, using programming) and experimental (or subjective, with human subjects) comparison and evaluation of the three above-mentioned techniques, namely, the convex, "Peli," and partial cartoon methods. In the process, the dissertation compares and evaluates image enhancement as governed by the following general-to-specific related variables and criteria, including:

- Viewing a deformed image as compared to an undeformed (or truncated) image.
- Determining which of the three compensation/warping methods creates a better physical tradeoff (through image smoothness and distortion measures) between image enhancement and minimum distortion, the convex projection technique, the "Peli" technique, and the partial cartoon superimposition technique.
- The variability of the warping for different images and scenes; this work investigates a landscape scenery, as well as those of a close-up, text, obstacle, and home scenes.

- Determining the optimum central viewing factors for each method. For the convex method, the factors to be considered include the magnification and proportionality factors. Higher emphasis is given to the two cartoon methods; factors of evaluation and comparison include the **cartoon factor** (for the partial cartoon method only), as well as the gray shade of the superimposed cartoon (in both methods).
- Comparison of static versus live images (for both deformed and undeformed images). The purpose is to simulate the warping schemes under real-life circumstances. This is achieved by displaying the same images while passing a dot across the screen and through introducing a gradually-growing area inside the image.
- Subject response to the warping processes described above, as well as comparing objective to subjective results. 115 volunteers (mostly students) have been selected for experimentation.

The following sections are dedicated to describing the theoretical and analytical background, including the presented research scheme as well as governing equations and parameters. This is followed by a description of the coding features used to achieve each of the image variation features discussed above. Subsequently, the data collection survey, experimental features, as well as the results and the discussion of results are presented.

## *Methodology*

Figure 13 and Figure 14 schematically demonstrate the adopted process for each of the respective convex projection and two cartoon superimposition techniques that have been adopted in this dissertation. The diagrams have been adopted from research on image compensation developed earlier by the author [9,10]. The research publications examine the relevance of selective area magnification on stressing different parts of an image based on spatial retinal density, which could serve in a host of different applications, such as low-vision, surveillance, mapping, robotics, and gaming, etc.

## Convex Projection Technique

For the convex technique (Figure 13), the magnification factors (remapping and proportionality) result in the enlargement of the central part of the image, along with a "minification" of the remainder of the image, also known as the "fish-eye" effect. The governing factors and equation are further discussed in the *Formulation* section. The area is thus remapped to the chosen shape according to the remapping equation, pixel by pixel. Finally, the image is resized down to the viewable, truncated, or target field of view.

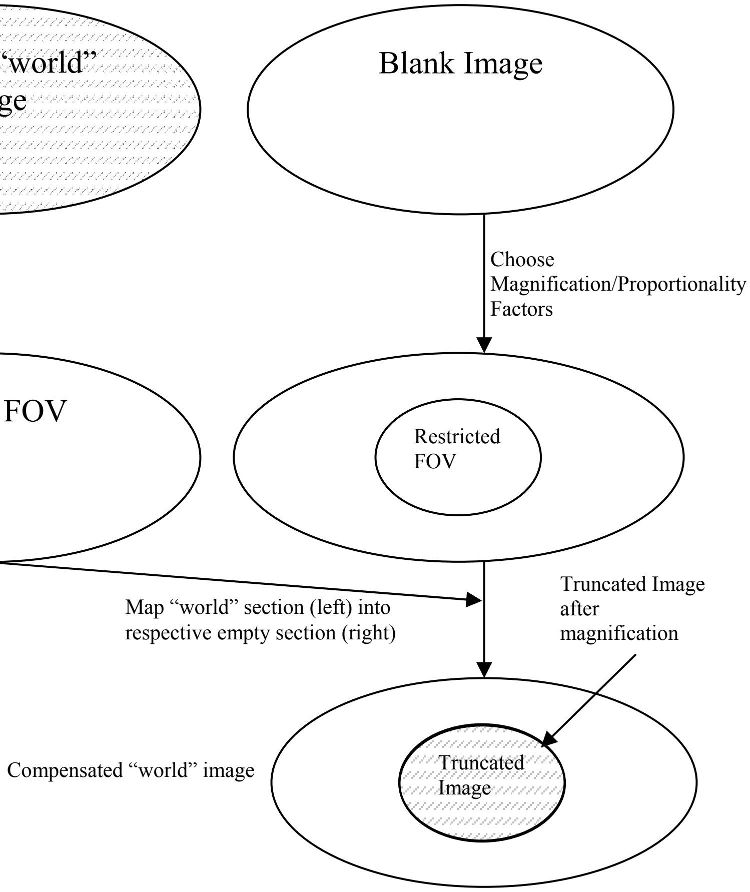

Original "world" image
Blank Image
Choose Magnification/Proportionality Factors
Global FOV
Restricted FOV
Map "world" section (left) into respective empty section (right)
Truncated Image after magnification
Compensated "world" image
Truncated Image

**Figure 13 Research matrix showing major steps of image compensation for the convex projection technique.**

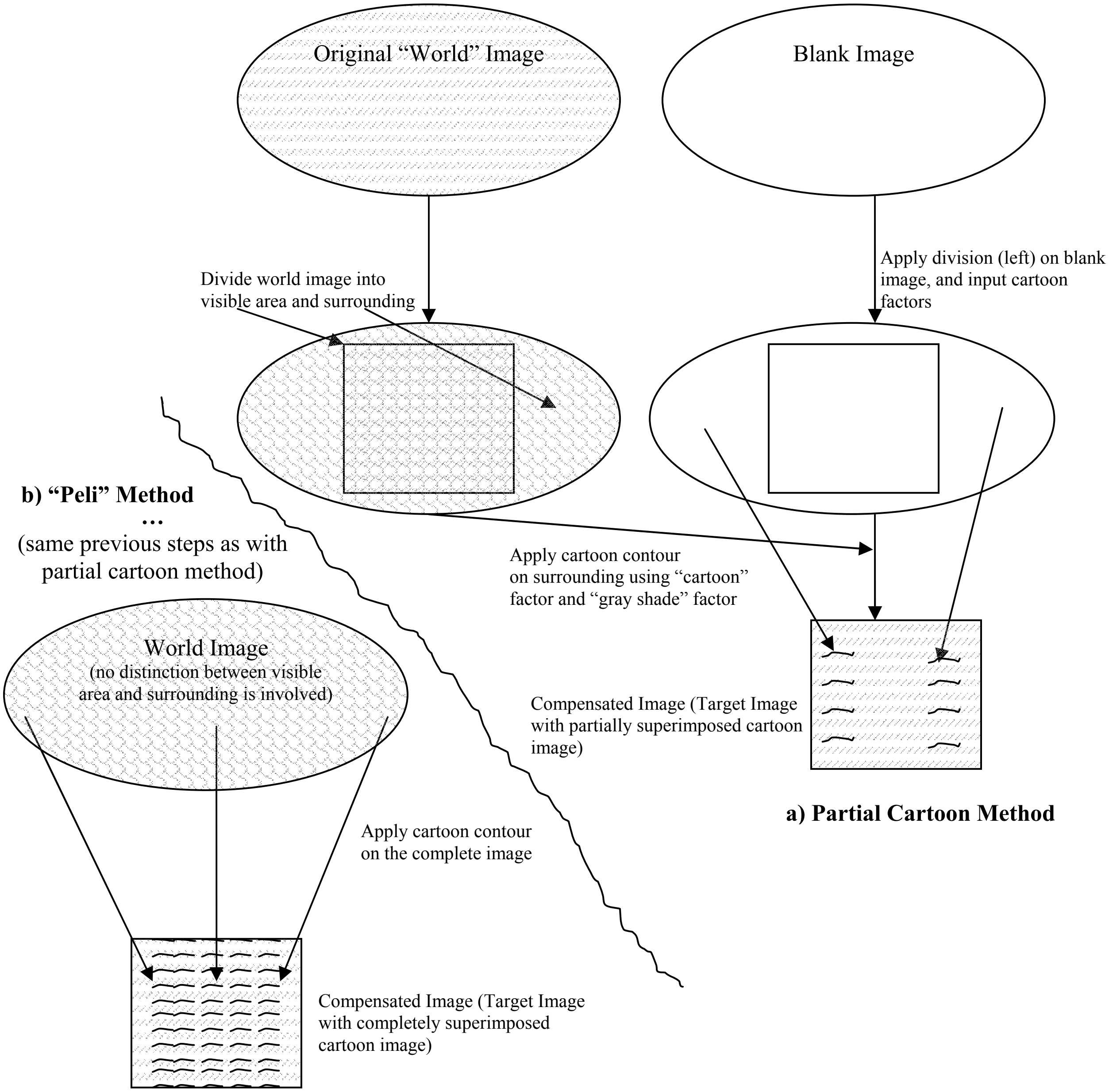

Original "World" Image
Blank Image
Divide world image into visible area and surrounding
Apply division (left) on blank image, and input cartoon factors
b) "Peli" Method
…
(same previous steps as with partial cartoon method)
Apply cartoon contour on surrounding using "cartoon" factor and "gray shade" factor
World Image
(no distinction between visible area and surrounding is involved)
Compensated Image (Target Image with partially superimposed cartoon image)
a) Partial Cartoon Method
Apply cartoon contour on the complete image
Compensated Image (Target Image with completely superimposed cartoon image)

**Figure 14 Research matrix showing major steps of image compensation for the partial cartoon and "Peli" methods; top-right (a) partial cartoon method; only immediate, invisible, left and right of the truncated image is resized and superimposed (as a cartoon), while central view of the final image is uncompromised; (b) "Peli" method; note the reduced size of cartoon and over-shadowed central view in the final image.**

## "Peli" and Partial Cartoon Techniques

As for the superimposition technique, as shown in Figure 14, the original or "world" image is first subdivided into predetermined target and surrounding areas, according to the visibility range of the subject or application. The surroundings are then converted into a contour image using appropriate parameters (to be discussed in the section on coding). The contour image is then superimposed, either partially (as in Figure 14-a, the partial cartoon method), or completely (as in Figure 14-b, the “Peli” method,) onto the target area.

Unlike the cartoon or “Peli” method described in the *Literature Review* section, the partial cartoon technique presented in this dissertation involves introducing only the surrounding section of the image (or invisible FOV) into the local (or visible) FOV. This eliminates the redundancy of overlaying contour pixels from the already visible local FOV (refer to Figure 9).

Furthermore, the surrounding image is introduced only on a partial basis (as shown in the bottom of Figure 14-a) according to the cartoon factor, whereby each of the immediate left and right surroundings is introduced according to a prescribed percentage of the visible FOV, as input from the user or code. This cartoon factor can range anywhere between 0 and 50% starting at the extreme respective left and right sides of the local FOV; note that the latter value (50%) acts as the limiting case for introducing the entire surrounding from the left and right into the visible FOV, which is similar to the “Peli” method. This aims to examine the effect of both compensating the image by introducing a contour of surrounding cells, as well as preserving (in the non-limiting cases) the visual region defined by quality photoreceptors in the local FOV.

This partial cartoon introduction is presented as a novel contribution in this dissertation. It is worthy to note that another variation of the partial cartoon to the “Peli” method involves eliminating all upper and lower layers of the surrounding image from the cartoon since their effect of enhancing vision is usually minimal and their presence in the cartoon serves to further distort the image rather than enhance it [43]. The presented cartoon in this technique assumes the same vertical proportion as the original surrounding. Realistically, it is only the horizontal dimension that matters the most in practical applications [43], and is thus the one investigated in this dissertation. Expansion of the partial cartoon concept to both the horizontal and vertical dimensions is considered further in the *Recommendations* section.

### Formulation

The aim of this section is to summarize and derive the convex warping equation, as described by Juday-Loshin (polar coordinate form) [26] and Amerijckx et al. (Cartesian form) [2] based on their papers, which are described in the *Literature Review*. The cartoon superimposition technique is not an analytical method, but its criteria are nonetheless treated in detail in the section on coding. As shown in Figure 15, an arbitrary pixel in the original image and after performing remapping (in the target image) as determined by "general retinal eccentricity formula" described by Weisel and Hubel [55], is governed by the following equations:

- $R = Kr^k$ (1)
- $\Theta = \theta$ (2)

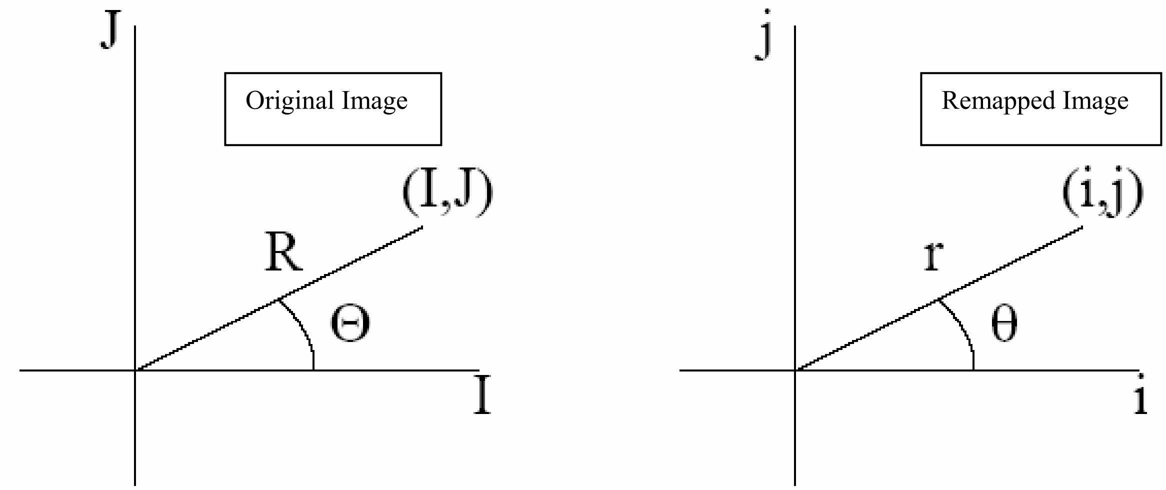


**Figure 15 Polar and Cartesian coordinate presentation of the location parameters for an arbitrary pixel in the original (left) and remapped (right) images [2, p. 381].**

On switching between polar into Cartesian coordinates it can be shown that:

- $$i = \frac{(I^2 + J^2)^{\frac{1-k}{k}}}{K^{\frac{1}{k}}} I \qquad (3)$$
- $$j = \frac{(I^2 + J^2)^{\frac{1-k}{k}}}{K^{\frac{1}{k}}} J \qquad (4),$$

where $(R, \Theta)$ and $(I, J)$ are the position of a pixel in the original image before remapping in polar and Cartesian coordinates, respectively, $(r, \theta)$ and $(i, j)$ are the position of a pixel in the target image after remapping in polar and Cartesian coordinates, respectively, $K$ is the factor of proportionality, and $k$ is the remapping factor.

The above equations offer the advantage that they are derived from the physical spatial distribution property describing "proportionality between the cortical distance, from the projection of the fovea toward projection along a given meridian, and the logarithm of the related retinal eccentricity" [2, p. 381]. The formula also preserves the conformality of the resulting pixels. Amerijckx et al. [2] suggest that this formula has a limitation since it considerably reduces the size of the target image during the remapping process (as compared to the original image); thus, they suggest a procedure that involves using two CCD camera views, instead of one. This drawback is handled in this dissertation, however, by choosing a "world" image having significantly high-resolution and by experimentally determining the appropriate remapping and proportionality factors for the remapped images.

### *Matlab*® *Code*

*Matlab*® was chosen as the programming medium for carrying out the different warping techniques in this dissertation. It contains various readily available, well-explained functions and commands for various image processing operations. Documentation is also available for various engineering, and medical applications, as explained by Semmlow [45] and Rafael et al. [48]. The *Matlab*® main script file "remap_main.m," as well as the functions (or subroutines) called from within this file are included in Appendix 1: Listing of the Image Remapping Code. Furthermore, the hierarchical structure of the code and functions is described in Figure 16.

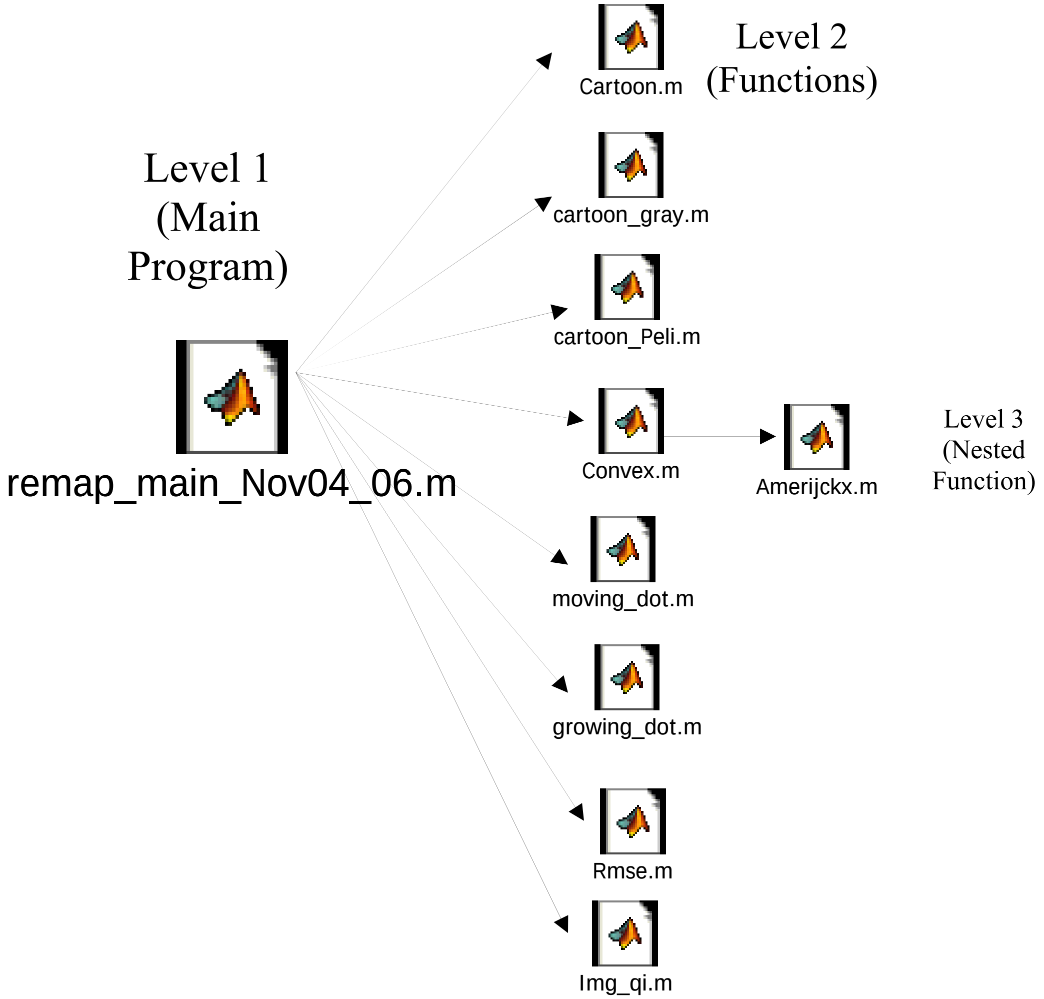


**Figure 16 Hierarchical structure of the main code (remap_main_Nov04_06.m), as well as 8 called functions, one which in turn calls a nested function, each designed to carry out a different image warping or image quality operation; note, the UIQI function (img_qi.m) has been adapted (with permission) from code developed by Wang et al. [52, 54].**

## Code Objectives and Layout

The objective of this program is to process any image according to the three warping methods described earlier, with user-specified warping parameters, as well as to display these images in both the static and live modes for movie mode evaluation. As the code is amply provided with major and minor documentation, the following sections take the reader through a sample run of the code, showing a general illustrated description of the self-prepared code, including sequence, menus, some important functions, warping-related operations, filtering, and applying the live schemes.

The user is first prompted with a main menu from which to choose if a new image is desired. The following options are available: the convex projection technique, the partial cartoon superimposition technique (including options either varying the cartoon or the cartoon gray shade), the cartoon superimposition (or “Peli”) technique, the movie mode, and a display of the processed images. A final menu option allows the display of the RMSE and UIQI image quality measures. On selecting a new image, the user is offered the flexibility of locating the image from anywhere on the pc (in bitmap format); “scenery2.bmp” is selected here for demonstration. The user is finally re-directed to the main menu, which is interrupted at an entry of ‘-1’ to end the program.

### Convex Projection Technique

On choosing to process the image using the convex projection technique, the user is prompted to input the warping parameters, the remapping (*k*, default: 1.15) and proportionality (*K*, default: 0.5) factors. The behavior of this technique to changes of these two unique convex remapping, *k* and *K,* variables is shown in the *Results* section. As shown in equations 3 and 4, the remapping factor (*k*) affects the “convexity” of the remapped image, or the distance ratio of the remapped pixel from the center of the image relative to its original location; the proportionality factor (*K*) depicts the final size of the original image in comparison to the original image. Note, further details are available in the papers by Juday et al. [26] and Amerijckx et al. [2].

The original image is then displayed to the user, providing the option of physically selecting a region of interest, which is facilitated by the zooming features of *Matlab*®. Subsequently, the convex projection function is called and the remapping equations (equation 3 and 4) are substituted for each pixel, resulting in the target image, which is then filtered using a pre-existing `fspecial('unsharp')` *Matlab*® kernel for enhancement. Distortions due to holes at image corners and at the center of the image (as shown in Figure 37) are also removed; these originate mainly during forward mapping, as described in the *Literature Review* section.

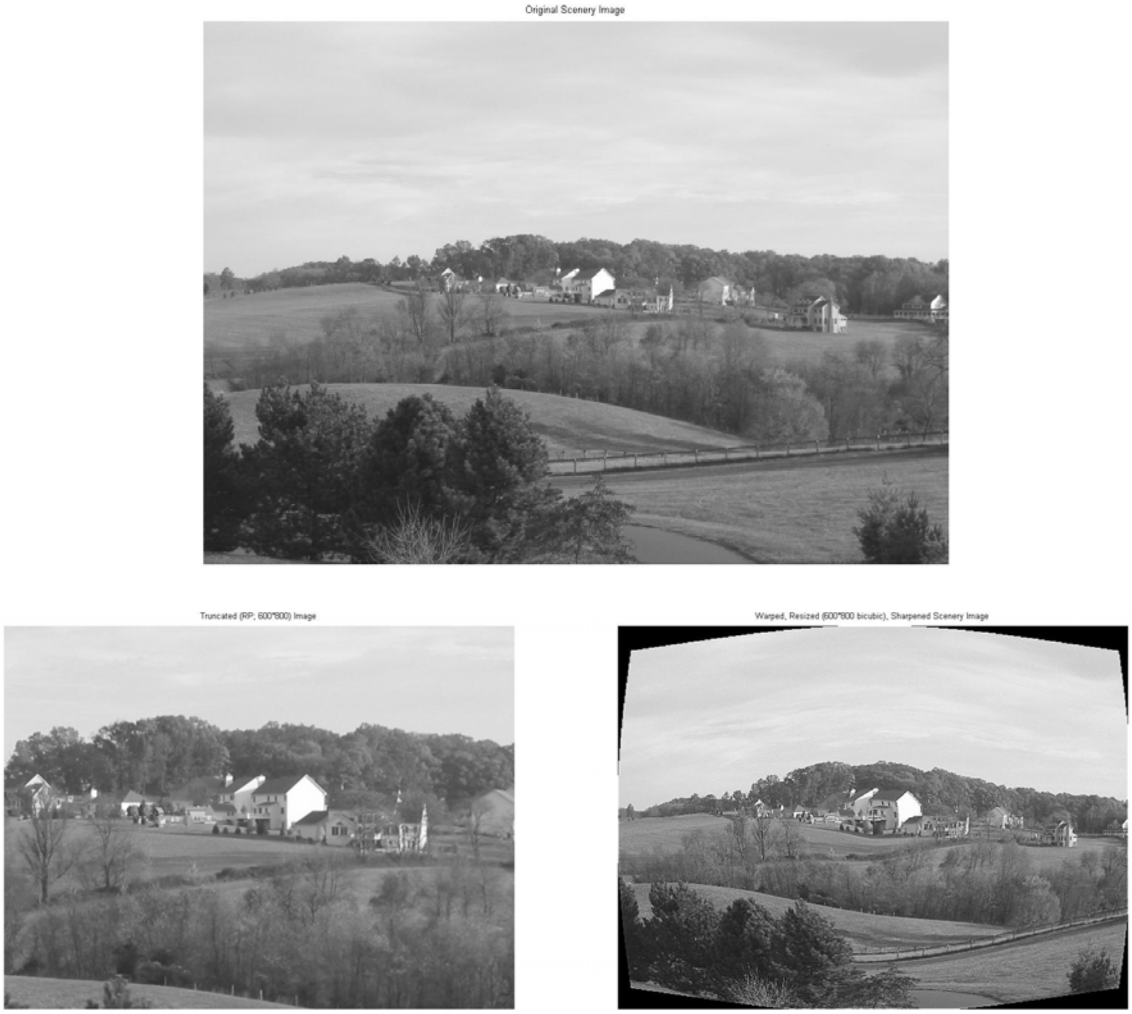
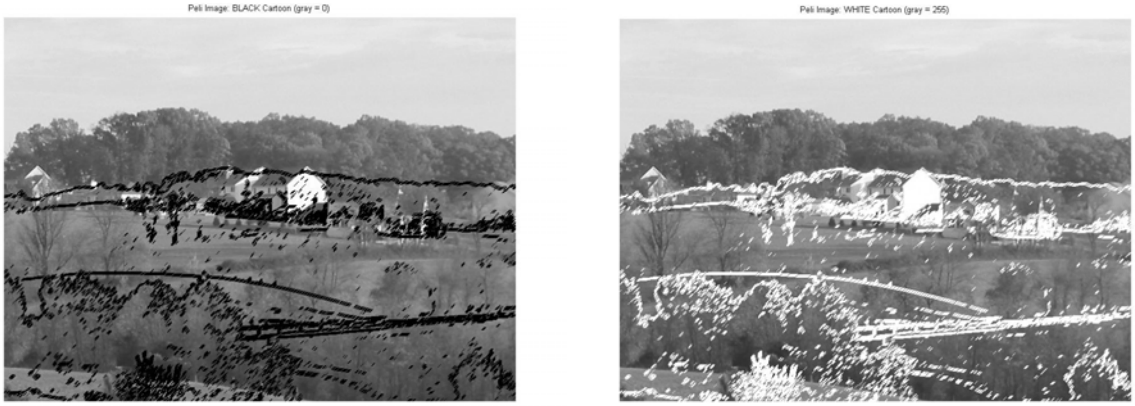

Original Scenery Image
Truncated (RP; 600*800) Image
Warped, Resized (600*800 bicubic), Sharpened Scenery Image

**Figure 17 *Matlab*® code demonstration for warping a scenery image using the convex projection method; top: base image (grayscale used for simplification); bottom left: truncated (or viewable) image, included for comparison; bottom right: warped image, resized to 600*800 pixels for standardization, with bicubic interpolation method used for higher calculation accuracy, and filtering applied to increase image sharpness.**

As demonstrated in Figure 17, the scenery image (top) is not all visible in the truncated area (bottom-left), while the remapping algorithm allows the entire scenery to be visible (bottom-right); in addition, the central area (white house in the middle of the image) appears with some distortion, and with a size comparable to that of the truncated image. An output is then presented for the magnification of the central area with respect to the image, before and after remapping. In this case an enlargement of 191% occurred, i.e. the white house is almost 3 times as large as its relative size in the original image.

### Cartoon (or "Peli") Superimposition Technique

Figure 18 demonstrates the application of the cartoon (or "Peli") superimposition technique to the same truncated image. Note how this results in overshadowing most of the truncated image. Black, white (the default color as demonstrated in Figure 9), gray, as well as a "mean" gray shade is possible, which involves the average of the entire grayscale luminance values of the entire image, which happens here to be 151.


Peli Image: BLACK Cartoon (gray = 0)

Peli Image: WHITE Cartoon (gray = 255)

**Figure 18 *Matlab*® code demonstration for applying the "Peli" (cartoon superimposition) technique for the image in Figure 17; left: using a black cartoon; right: using a white -default, as in the literature- cartoon.**

### Partial Cartoon Superimposition Technique

Figure 19 demonstrates the application of the partial cartoon superimposition technique to the same truncated image. The program allows varying the cartoon factor, including 25%, 30% (chosen here are default), 35%, 40%, 45%, and 50% (limiting case), as well as the same cartoon color variations as above, white, black (default), gray, and mean. Note the center of the image is clearly visible in all the demonstrated cases, while the immediate left and right surrounding are hinted by the superimposed cartoons.

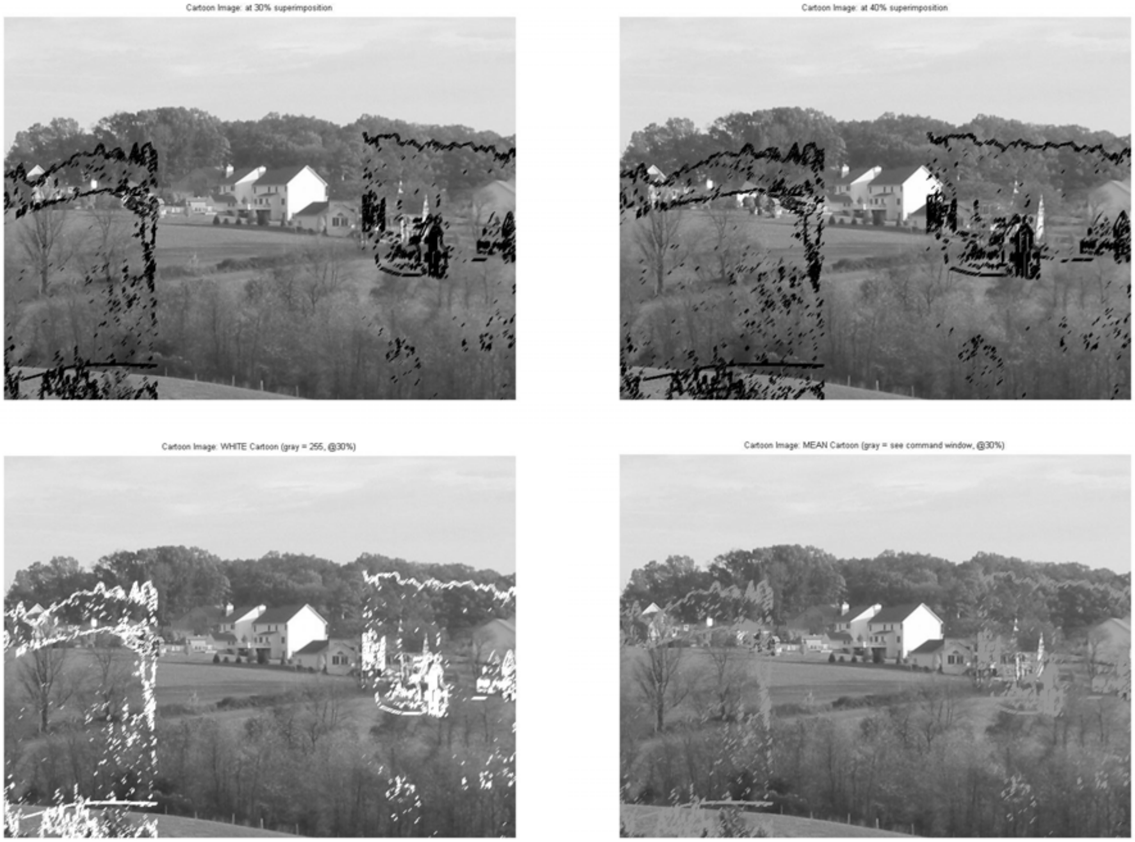

Cartoon Image: at 30% superimposition
Cartoon Image: at 40% superimposition
Cartoon Image: WHITE Cartoon (gray = 255, @30%)
Cartoon Image: MEAN Cartoon (gray = see command window, @30%)

**Figure 19 *Matlab*® code demonstration for applying the partial cartoon technique to the same truncated area (as in Figure 17) using different cartoon factors (upper left: 30%, upper right: 40%) as well as different gray shades (lower left: white, lower right: mean); note, in most cases, the respective, immediate left and right surrounding areas are small yet identifiable (set of trees and bushes on the left, and two houses on the right, etc.), while the center of the image is still clearly visible (unlike in Figure 18).**

## Movie Mode

The code provides a live or video feature (shown in Figure 20 and Figure 21), whereby the warped images (using either technique) are displayed in their static mode, while simulating an intrusion of an object into the field of view of the individual. The first feature is that of a moving dot (Figure 20), where a point (simulating a flying object, for instance) is introduced into the image from an arbitrary point on either side of the image and slowly moves across the image for about 20 seconds. This is used to simulate an intrusive object into the field of view, and for testing the required time it would require the subject to perceive the object.

The second video feature (shown in Figure 21) is that of an area that grows from an arbitrary location in the form of a widening square. This is used to simulate changes that occur within the image (e.g. a fire or an approaching object), in order to study the subject response time required to detect gradual changes or incidents.

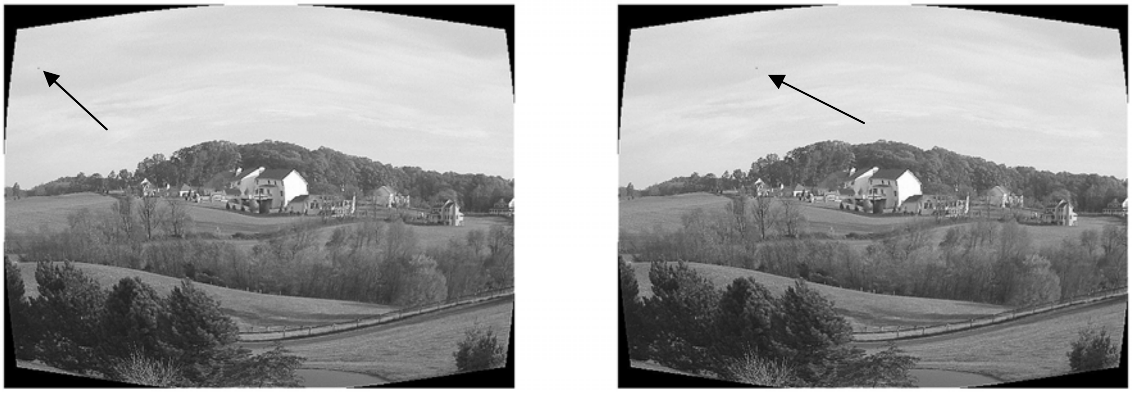


**Figure 20 *Matlab*® code demonstration for the "moving-dot" video mode, shown at two different stages.**

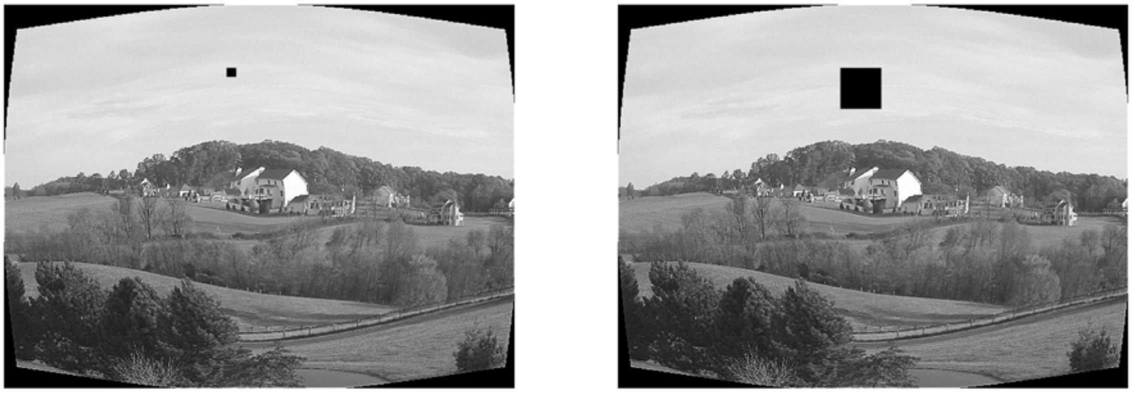


**Figure 21 *Matlab*® code demonstration for the "growing-square" video mode, shown at two different stages.**

### *Image Quality Evaluation*

The final objective task achieved by the code is evaluating the image quality of all image warps described above, compared to the original unmodified image, using the root mean square error, RMSE, and the "Universal Image Quality Index," UIQI.

As described in the *Literature Review* section, image quality is practically evaluated through a measure of the root mean square error, RMSE. The formula is given by:

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}\left(x_n - y_n\right)^2}{N}} \qquad (5),$$

where $x_n$ and $y_n$ are the luminance values of any pixel in the original and warped images, respectively, and *N* represents the total number of pixels in either image.

Like the RMSE, the UIQI index is described as another technique which measures image quality independent of viewing conditions. The *Matlab*® code for the UIQI (prepared by Wang) has been adapted and integrated as a called function. Note, this addition into the dissertation has been done with clear copyright permission from the authors (as indicated in the last section of Appendix 1: Listing of the Image Remapping Code) [54]. The UIQI is also claimed to be superior to the RMSE method as the terms of its equation respectively evaluate loss of correlation, luminance distortion, and contrast distortion in one convenient quantity, with values ranging from -1 to 1, where 1 denotes highest relative quality [52]. The equation is given by:

$$UIQI = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\overline{xy}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \qquad (6),$$

where, $\bar{x}$ and $\bar{y}$ are, respectively, the average luminance values of the original and warped images; $\sigma_x$ and $\sigma_y$ denote the standard deviation for the luminance values of the original and warped images, respectively; and $\sigma_{xy}$ denotes the standard deviation between the luminance values of the two images.

Note that although the “Peli”, partial cartoon, and convex methods are quite distinct warping techniques, the general applicability and image-viewing independence of the RMSE and UIQI measures allow image quality to be computed and compared in an unbiased fashion, especially since the overall size of the original and warped images is forced to be the same. The aim is to objectively optimize each viewing condition by finding the warped image with the lowest RMSE and highest UIQI values for each of the presented variables above.

### *Optimization Scheme*

The efficiency of a warped image, as described in the literature review, is governed by distortion (or image quality) and informational content. This dissertation aims to evaluate the three presented warping schemes according to these two parameters both objectively and subjectively. Qualitative, objective optimality of the image warp is achieved by calculating a tradeoff between minimal distortion (RMSE from equation 5) and highest informational content. The subjective (also qualitative) optimality is used to verify the practicality of the objective measurement; subject feedback is used to determine the maximum informational content versus distortion results, which are finally compared to their objective counterparts.

As described in the introduction, most of the quality vision lies in the macular photoreceptor region of the retina, as shown in Figure 1. Using a rough interpolation conversion, this corresponds to a measure of about 15 degrees on a **visual eccentricity** scale, or 7.5 degrees from either side of direct vision. Another very narrow “spike” of quality vision (restricted to rod cells only) appears further away at 20-degrees, having less visual significance [47]. Vision quality then quickly deteriorates from that point on, as the spatial density quickly falls at peripheral vision compared to central vision.

### *Survey and Experimentation*

The aim of this section is to describe the procedure used to carry out the various criteria of experimental investigation and optimization mentioned at the beginning of this chapter. As it is the primary objective of this dissertation to investigate and optimize the central viewing conditions (based on numerical and experimental data), it has been suggested that perceiving the image may change with different images and views. Five such grayscale base images are considered, as displayed in Figure 22, examining landscape, close-up, text, obstacle, and home scenes. Grayscale images are chosen to eliminate bias that may arise due to the presence of color in the subjective studies.

Permission to perform individual, voluntary, non-invasive, minimal risk surveying for the images presented in this dissertation has become available after completing the NIH (National Institutes of Health) "Human Participants Protection Education for Research Teams" online course [41]. In addition, approval has been obtained form the West Virginia University Office of Research Compliance. A copy of these certificates and approvals is provided in Appendix 2: Subject Testing Certificates.

The data collection survey, including also base and warped images, is shown in Appendix 3: Image Set 1, Base Images, and Appendix 4: Central Image Optimization Survey. Each image is shown for five to ten seconds (on an LCD screen) by which time the subject must respond to a measure of detail and distortion of the warped (scaled from 0 to 10) versus the control (truncated) image. The questions are based on a reading level of grade 8 (Flesch Kincaid reading level = 7.8, from *Microsoft Word*® *Readability Statistics*), in order to support administering the surveys to a large variety of intellects. Subjects are also prompted for their age and gender.

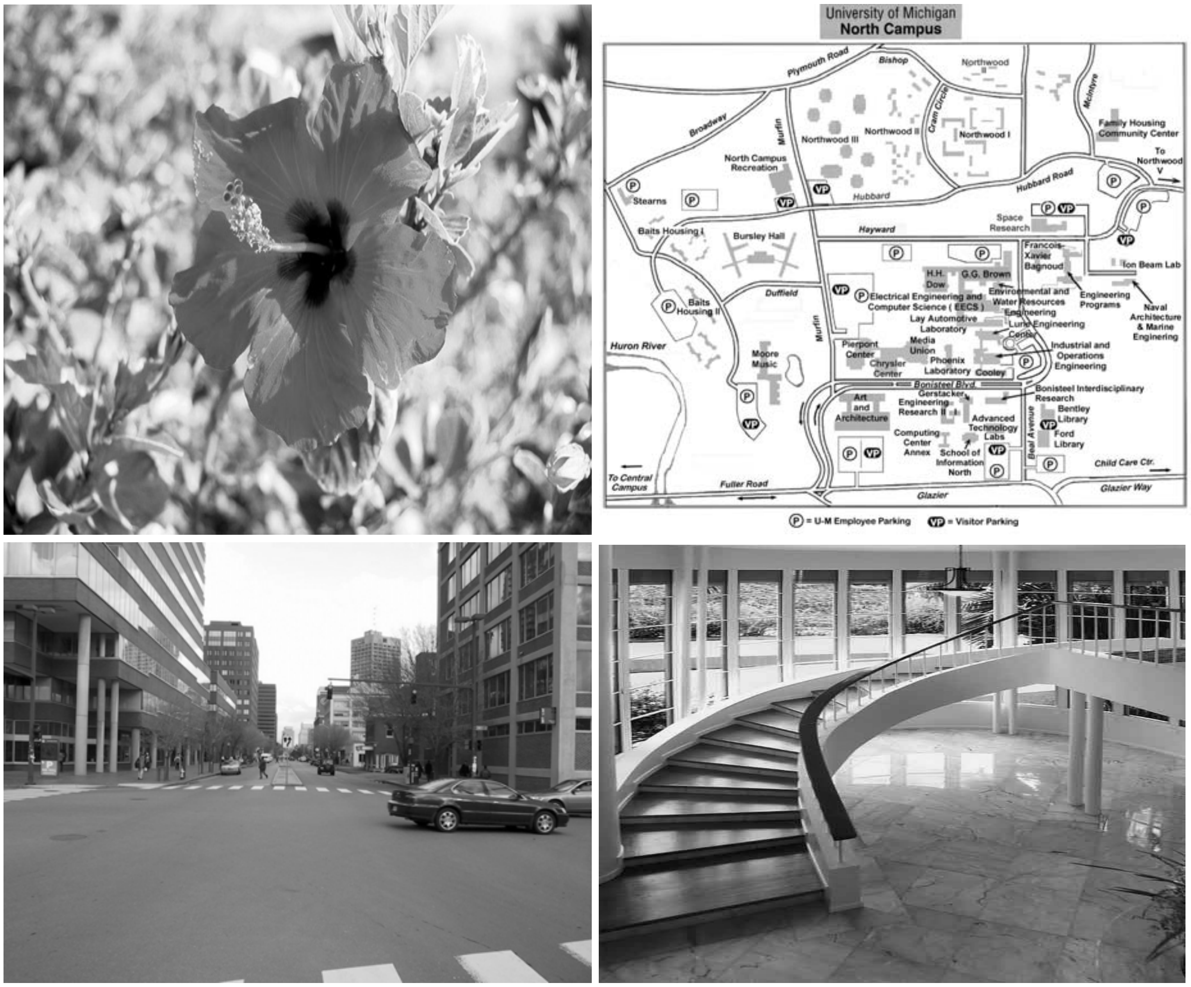


**Figure 22 Five sample, base images for experimentation: 1. scenery (Figure 17), 2. close-up (top-left), 3. text (top-right), 4. obstacle or hazard (bottom-left), and 5. home (bottom-right).**

In addition, the questions have been randomly ordered to prevent a predictability pattern in the answers. The survey also includes a feedback section, allowing subjects to suggest improvements. A pilot study was first conducted on 23 subjects to determine the feasibility, subject response, and overall efficiency of the "Central Vision Optimization Survey," which was then followed by an improved (main) survey applied on 115 subjects. The collected data was used to generate a correlation between each of the highlighted experimental criteria, aiming to optimize detail and minimizing distortion, as treated in the upcoming sections of the dissertation.

## ***Results and Discussion***

The objective and subjective results produced using the aforementioned analysis are summarized in the following sections. First, an overview of the various objective, subjective, and summarizing objective versus subjective data and plots are demonstrated and analyzed. This is followed by a display of the actual results, including plots and actual data. Finally, an insight into the data is shown, with hints to the final *Conclusions* section.

### Overview of Results

Table 2, Table 3, and Table 4 provide, respectively, a summary to the data and plots generated from implementing the different objective, subjective, and cross-comparative (objective/subjective) variables and criteria elaborated in the *Research Objectives* section. The aim is to present a specific-to-general demonstration for the results of each variable/criterion, in order to determine the optimum method (i.e. one with highest image quality) both objectively and subjectively. This is first achieved by examining the results of each individual variable (e.g. cartoon factor, gray shade, convex parameters, etc.), through finally ending with a general, integrative comparison of the three methods, the presented partial cartoon method versus the documented "Peli" and convex methods.

The results are tabulated and plotted (in the same figure), and in-turn discussed in the remainder of this section of the dissertation, building up primary conclusions as they start to unveil. The objective results are those produced by processing the photos in image set one (shown in Appendix 3: Image Set 1, Base Images) and image set two (Appendix 4: Central Image Optimization Survey) using the previously discussed image remapping *Matlab*® code. The data was then exported to *Microsoft Excel*® (after minimal post-processing), where it was plotted as data tables and bar charts representing measures of RMSE and UIQI for each evaluated criterion.

**Table 2 Table summarizing objective data tables and plots of the various data from the two image sets generated using *Matlab*®; plots are classified by *independent* variable or criterion under consideration; the dependent variables are the calculated RMSE and UIQI values shown in the listed figures; the last row pertains to the correlation coefficients between data generated using the three methods.**

| **Investigated Objective Variables or Comparison** |
|---|
| • **Partial Cartoon Method, Cartoon Factors (25-50%)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 24 Image Set 2: Figure 25 |
| • **Partial Cartoon Method, Gray Shades, (Black, White, Gray, and Mean)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 27 Image Set 2: Figure 28 |
| • **"Peli" Method, Gray Shades (Black, White, Gray, and Mean)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 30 Image Set 2: Figure 31 |
| • **Partial Cartoon versus "Peli" Methods, Gray Shades (Black and White)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 32 Image Set 2: Figure 33 |
| • **Convex Method, *k* and *K* factors**<br>**(*k*:1.15,*K*:.4; *k*:1.15,*K*:.7; *k*:1.15,*K*:1; *k*:1.1,*K*:.5; *k*:1.15,*K*:.5; *k*:1.2,*K*:.5)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 35 Image Set 2: Figure 36 |
| • **Partial Cartoon versus Convex Methods, Movie Mode (Growing Area)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 38 Image Set 2: Figure 39 |
| • **Partial Cartoon versus "Peli" versus Convex Methods (Defaults)**<br>RMSE and UQI Characteristics<br>Image Set 1: Figure 41 Image Set 2: Figure 42 |
| • **Partial Cartoon versus "Peli" versus Convex Methods**<br>Correlation Coefficients<br>Image Sets 1 and 2: Figure 43 |

The subjective results involve surveys for image set one photos (conducted on Sep 7, 2006), while the main survey (conducted Nov 9-15, 2006) involves image set two photos, as shown in Appendix 4: Central Image Optimization Survey. A closed-loop processing of the experimental data involved conducting the surveys, and manually recording the numerical individual subject feedback results onto *Microsoft Excel*®. Note that incomplete surveys were discarded (those missing than 50% of the entries), and it was ensured that each subject fully understood the contents, terms, and conditions of the survey, and that they both completed the survey willingly and filled the subject consent form.

As with image quality objective measures, detail and distortion experimental measures were then normalized (out of one) to enable a comparison to their objective counterparts. The average mean values and standard error (as described in the *Subjective Results* upcoming section) were computed for the 115 subjects in each data category (i.e. investigated image compensation technique), and finally plotted as data tables and bar graphs, in individual figures, for each specific-to-general assessed data variable and comparison, as with the objective results. The resulting data tables, corresponding bar graphs, and discussion of each objective, subjective, and comparative variable and criterion are demonstrated in the following sections, supplemented by various remapped images to physically illustrate the meaning of the results for a sample image.

## Objective (Analytical) Results

As shown in Table 2, the objective results are focused on evaluating the partial cartoon method in various criteria, including the optimum cartoon factor (Figure 24 and Figure 25), and best gray shade (Figure 27 and Figure 28), as indicated by the image quality measures –also tabulated in each of these figures– of RMSE and UIQI for the two image sets. It is worthy of noting first that in the partial cartoon method, only the horizontal portion of the left and right surrounding is visible, which implies that no image repetition occurs (inside the truncated image). It also implies the loss of only one dimension (as opposed to two dimensions in the

“Peli” method) of proportionality, where the vertical proportion of the truncated image is conserved. Furthermore, the upper and lower portions of the invisible (above and below the truncated image) are eliminated allowing for a larger, clearer cartoon, while the center of the image is mostly conserved.

The chosen cartoon factors range between 25 and 50%, with a 5% increment, large enough to cause a visible difference in the image, as illustrated in Figure 23. This achieves a compromise between the fact that when the cartoon is small (< 25%), the *surrounding* is less identifiable, and conversely when the cartoon is too large (40-50%), the *center* of the truncated image is less identifiable. Figure 24 and Figure 25 show the consistent increases in error (with a consistent image quality decrease) with increasing cartoon factor. As a result, the selected tradeoff (or default) cartoon is 30%, to clear as much as possible of the quality field of vision. Note, however, that verification is necessary for this with a subjective analysis, especially since data does not necessarily indicate which images will appear with a higher image quality.

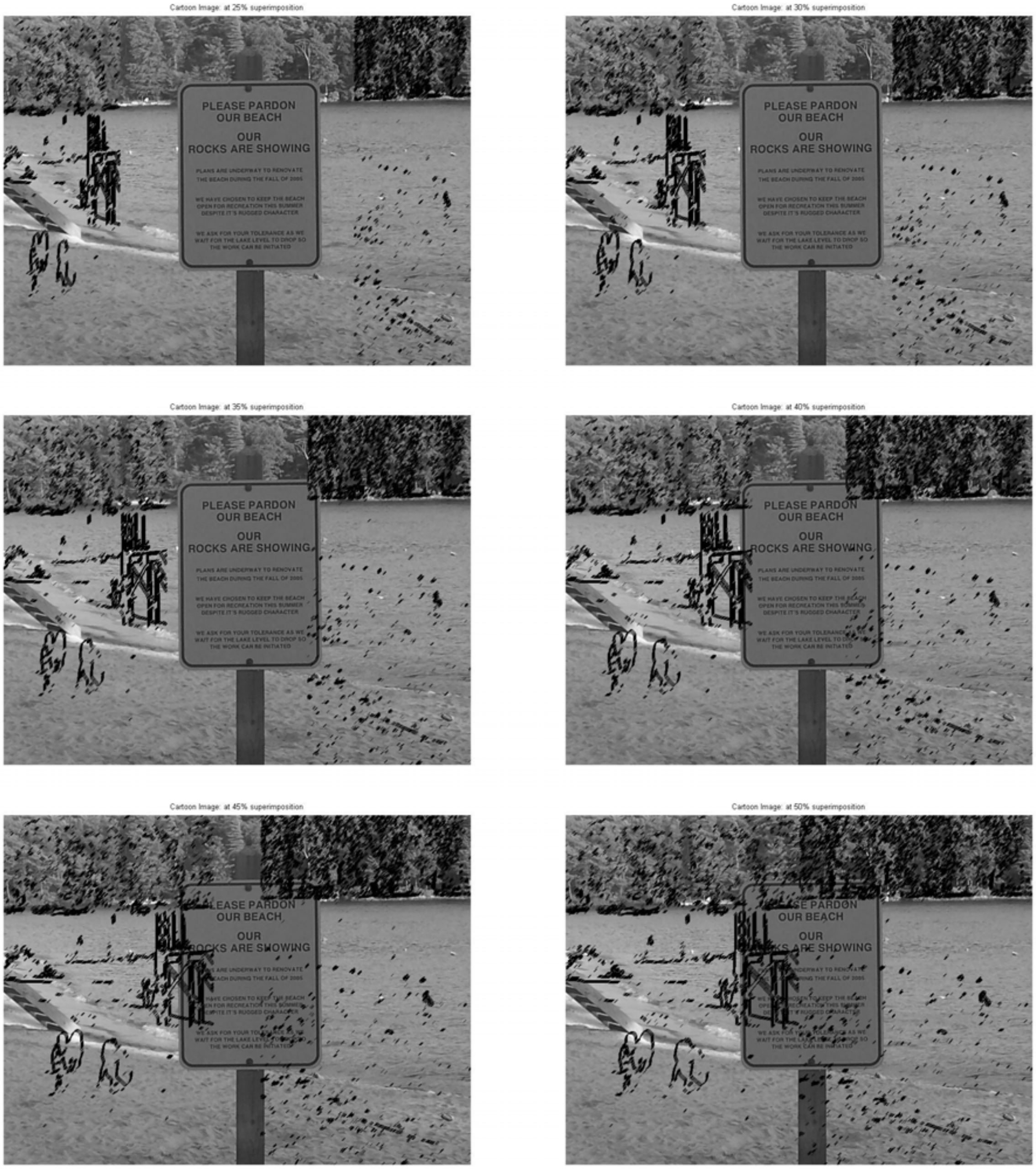
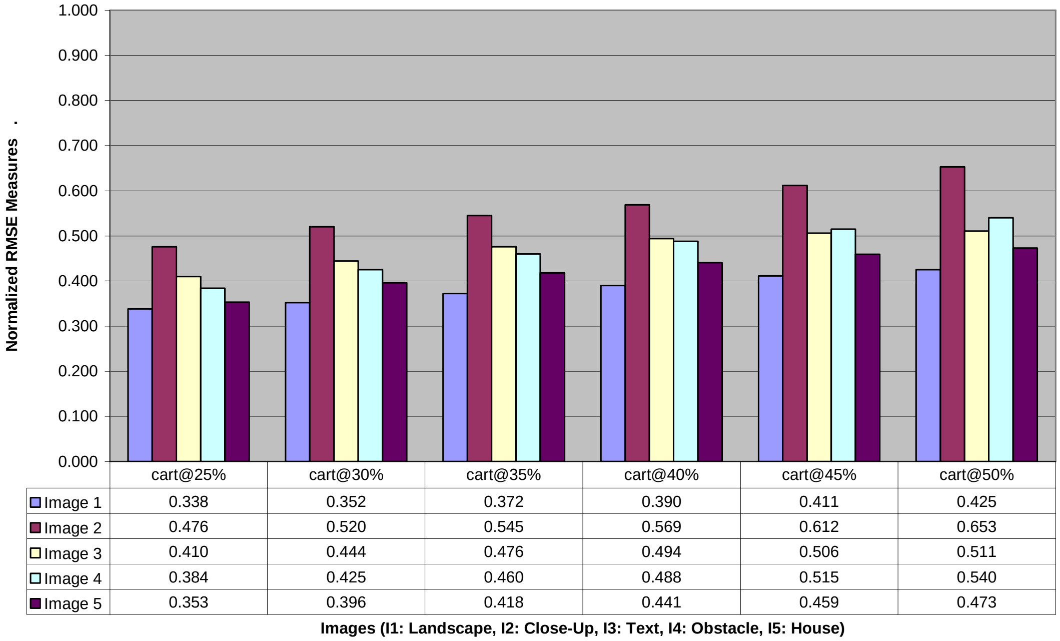

Cartoon Image: at 25% superimposition
Cartoon Image: at 30% superimposition
Cartoon Image: at 35% superimposition
Cartoon Image: at 40% superimposition
Cartoon Image: at 45% superimposition
Cartoon Image: at 50% superimposition
PLEASE PARDON
OUR BEACH
OUR
ROCKS ARE SHOWING

**Figure 23 Demonstration for applying the partial cartoon technique to a sample image (image 3 “text,” from image set 1) using different cartoon factors; top-left: 25% cartoon; top-right: 30% cartoon; mid-left: 35% cartoon; mid-right: 40% cartoon; bottom-left: 45% cartoon; bottom-right: 50% cartoon (limiting case); note how as the cartoon grows larger in size, its clarity increases, while the center of the image is gradually compromised.**

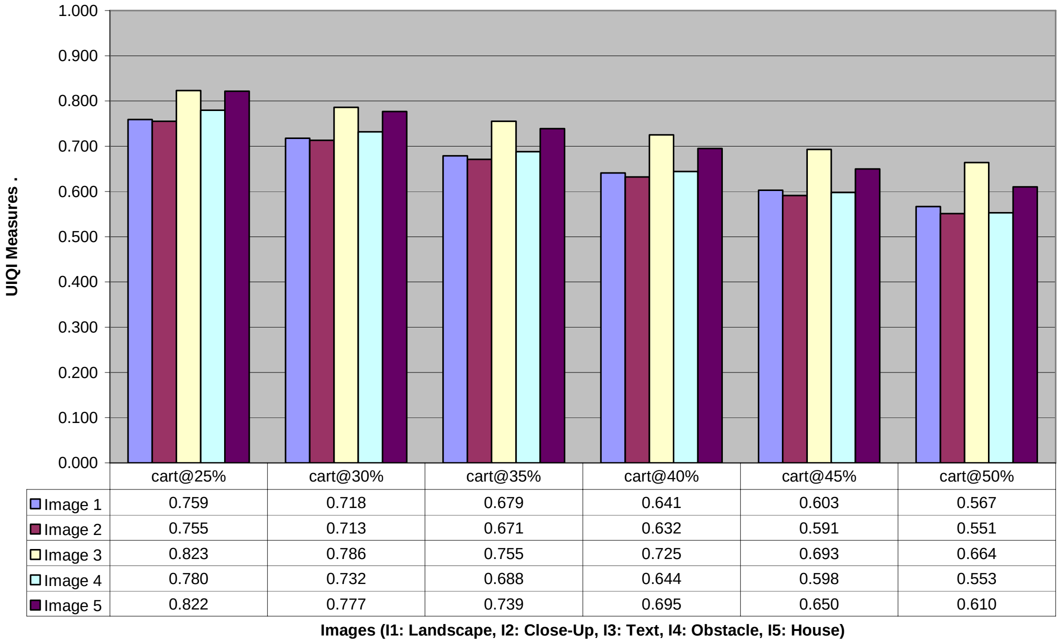
RMSE Characteristics of Partial Cartoon Method (25-50% Superimposition; Image Set 1)



| | cart@25% | cart@30% | cart@35% | cart@40% | cart@45% | cart@50% |
|---|---|---|---|---|---|---|
| Image 1 | 0.338 | 0.352 | 0.372 | 0.390 | 0.411 | 0.425 |
| Image 2 | 0.476 | 0.520 | 0.545 | 0.569 | 0.612 | 0.653 |
| Image 3 | 0.410 | 0.444 | 0.476 | 0.494 | 0.506 | 0.511 |
| Image 4 | 0.384 | 0.425 | 0.460 | 0.488 | 0.515 | 0.540 |
| Image 5 | 0.353 | 0.396 | 0.418 | 0.441 | 0.459 | 0.473 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)

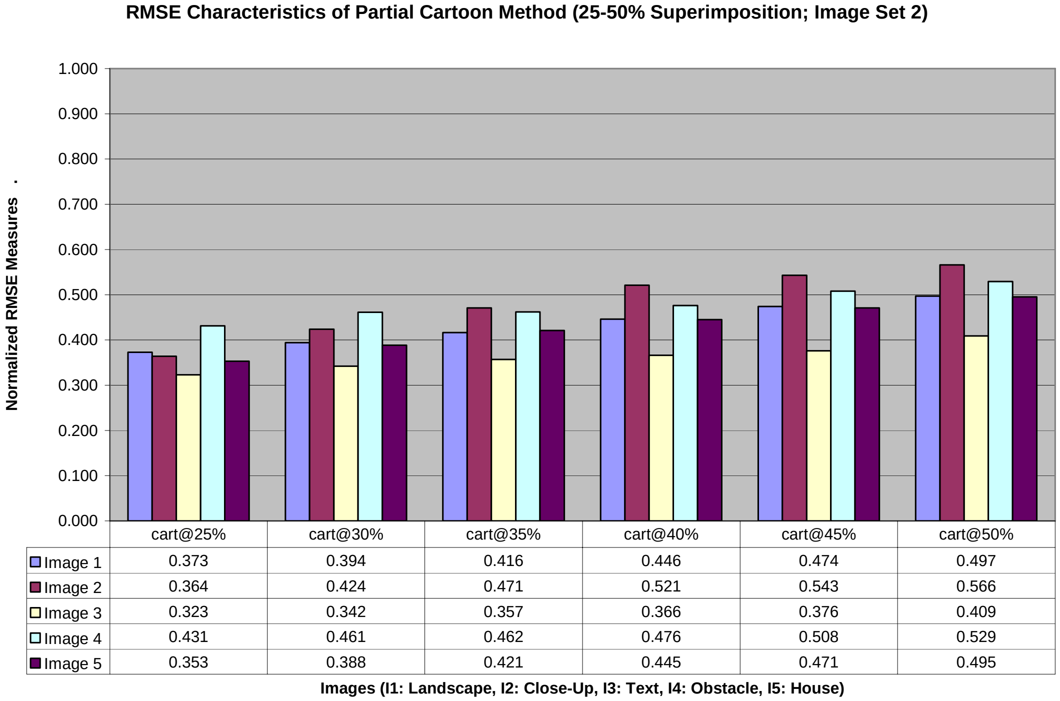
UIQI Characteristics of Partial Cartoon Method (25-50% Superimposition; Image Set 1)



| | cart@25% | cart@30% | cart@35% | cart@40% | cart@45% | cart@50% |
|---|---|---|---|---|---|---|
| Image 1 | 0.759 | 0.718 | 0.679 | 0.641 | 0.603 | 0.567 |
| Image 2 | 0.755 | 0.713 | 0.671 | 0.632 | 0.591 | 0.551 |
| Image 3 | 0.823 | 0.786 | 0.755 | 0.725 | 0.693 | 0.664 |
| Image 4 | 0.780 | 0.732 | 0.688 | 0.644 | 0.598 | 0.553 |
| Image 5 | 0.822 | 0.777 | 0.739 | 0.695 | 0.650 | 0.610 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)

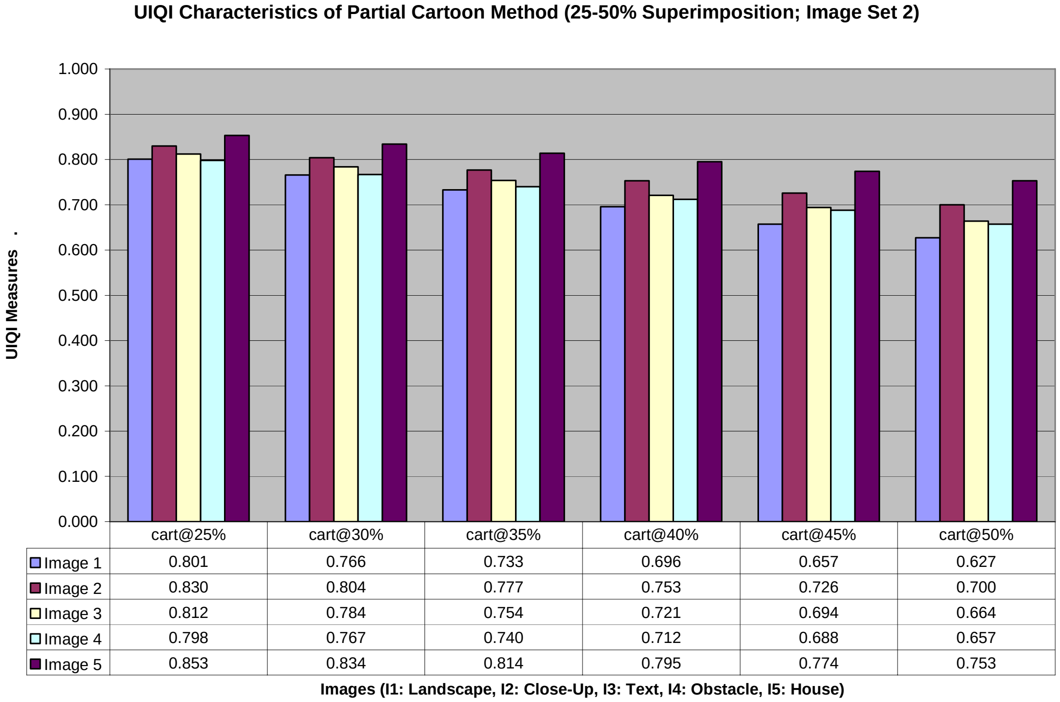
**Figure 24 Objective characteristics of the partial cartoon method for various cartoon factors (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**



RMSE Characteristics of Partial Cartoon Method (25-50% Superimposition; Image Set 2)

| | cart@25% | cart@30% | cart@35% | cart@40% | cart@45% | cart@50% |
|---|---|---|---|---|---|---|
| ▨ Image 1 | 0.373 | 0.394 | 0.416 | 0.446 | 0.474 | 0.497 |
| ■ Image 2 | 0.364 | 0.424 | 0.471 | 0.521 | 0.543 | 0.566 |
| □ Image 3 | 0.323 | 0.342 | 0.357 | 0.366 | 0.376 | 0.409 |
| □ Image 4 | 0.431 | 0.461 | 0.462 | 0.476 | 0.508 | 0.529 |
| ■ Image 5 | 0.353 | 0.388 | 0.421 | 0.445 | 0.471 | 0.495 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)



UIQI Characteristics of Partial Cartoon Method (25-50% Superimposition; Image Set 2)

| | cart@25% | cart@30% | cart@35% | cart@40% | cart@45% | cart@50% |
|---|---|---|---|---|---|---|
| ▨ Image 1 | 0.801 | 0.766 | 0.733 | 0.696 | 0.657 | 0.627 |
| ■ Image 2 | 0.830 | 0.804 | 0.777 | 0.753 | 0.726 | 0.700 |
| □ Image 3 | 0.812 | 0.784 | 0.754 | 0.721 | 0.694 | 0.664 |
| □ Image 4 | 0.798 | 0.767 | 0.740 | 0.712 | 0.688 | 0.657 |
| ■ Image 5 | 0.853 | 0.834 | 0.814 | 0.795 | 0.774 | 0.753 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)

**Figure 25 Objective characteristics of the partial cartoon method for various cartoon factors (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

Figure 27 and Figure 28 examine the partial cartoon method for various gray shades (at the default 30% cartoon factor) of the superimposed cartoon in image sets one and two, at black (i.e. the default, luminance value of 0), white (luminance of 255), gray (128), and image-mean values. The image means for the two image sets are shown in the respective figures. The white cartoon offers the highest contrast to image components (as indicated from several survey feedbacks), which is clear even in daylight conditions. However, this causes the cartoon to become dominant over regions of the truncated image, which also manifests itself in the shown objective results through highest RMSE (and lowest UIQI) values, compared to all other gray shades.

Cartoons with gray and “mean” shades have luminance values which appear relatively close, and the mean is farthest from gray in dark settings (78), as expected. In addition, both gray and mean cartoons offer similar, low contrast over the truncated image, and thus are hard to distinguish. However, both generally yield the lowest RMSE and highest UIQI for all images. Finally, the black cartoon offers a compromise between high image quality and contrast and for all images, where it remains visible for all daylight images and most portions of dark images, without dominating over the truncated image. The physical effect of varying cartoon gray shades for the partial cartoon method is also practically demonstrated in Figure 26.

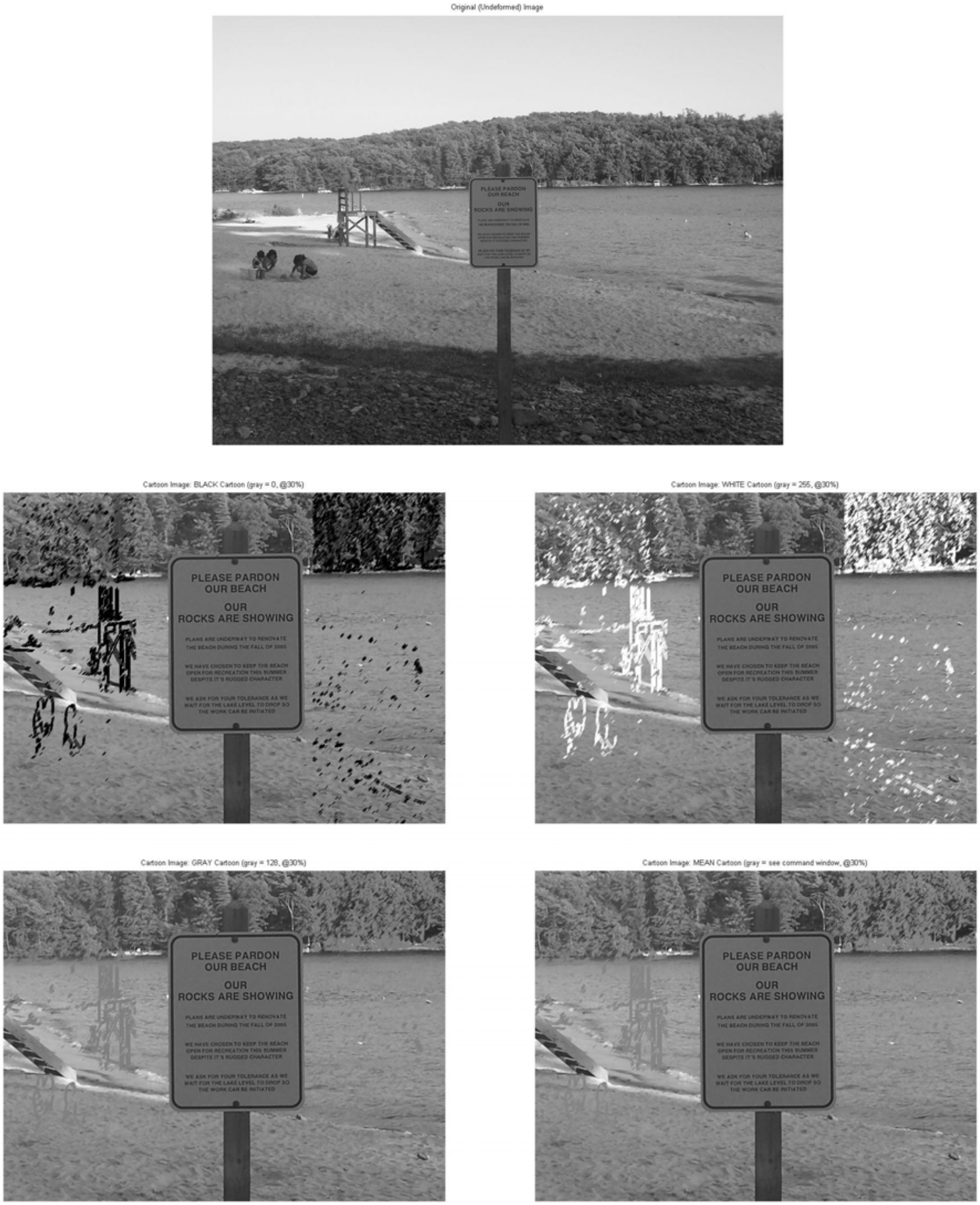
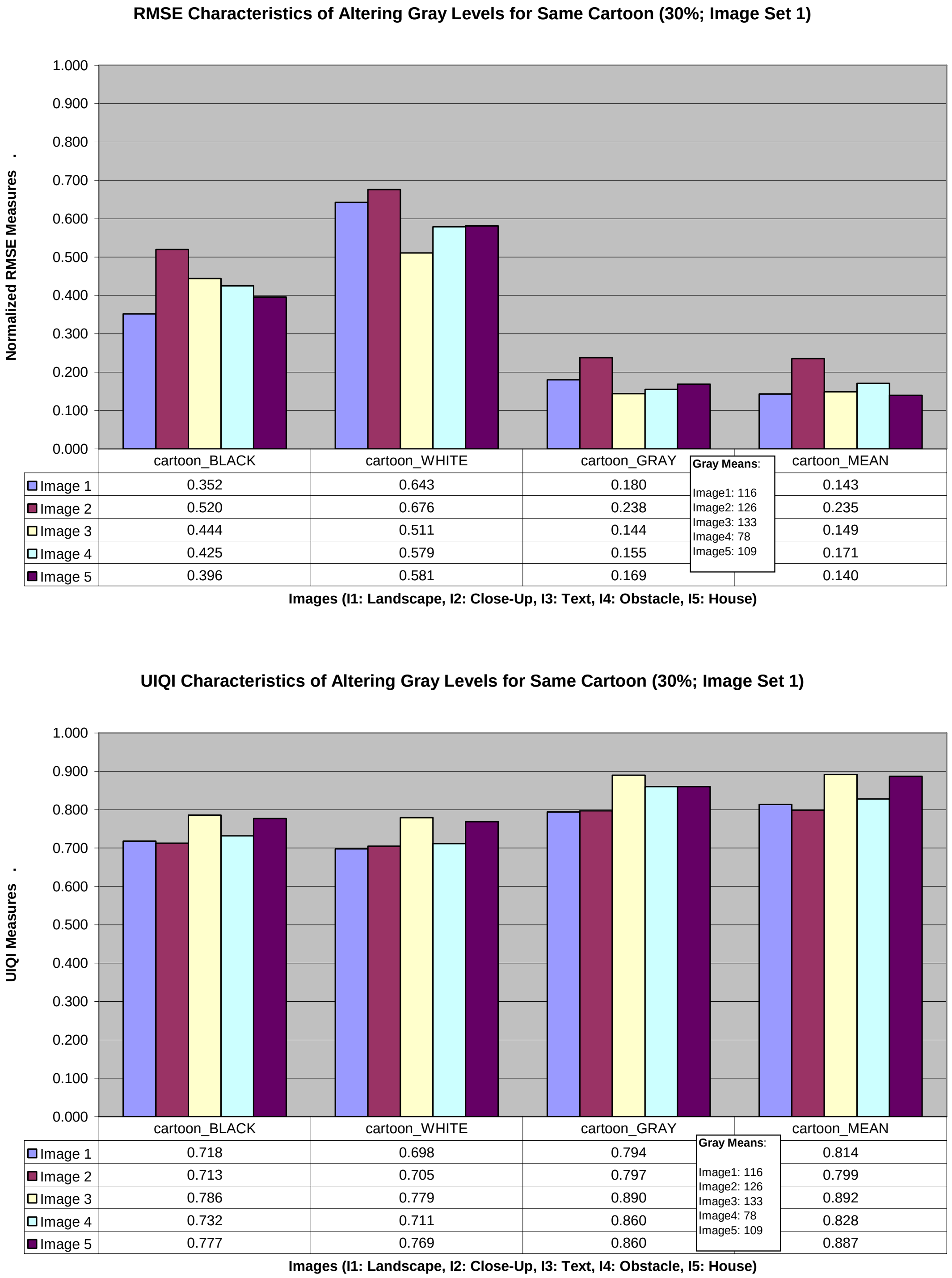

Original (Undeformed) Image
Cartoon Image: BLACK Cartoon (gray = 0, @30%)
Cartoon Image: WHITE Cartoon (gray = 255, @30%)
Cartoon Image: GRAY Cartoon (gray = 128, @30%)
Cartoon Image: MEAN Cartoon (gray = see command window, @30%)
PLEASE PARDON
OUR BEACH
OUR
ROCKS ARE SHOWING
PLANS ARE UNDERWAY TO RENOVATE
THE BEACH DURING THE FALL OF 2005
WE HAVE CHOSEN TO KEEP THE BEACH
OPEN FOR RECREATION THIS SUMMER
DESPITE IT'S RUGGED CHARACTER
WE ASK FOR YOUR TOLERANCE AS WE
WAIT FOR THE LAKE LEVEL TO DROP SO
THE WORK CAN BE INITIATED

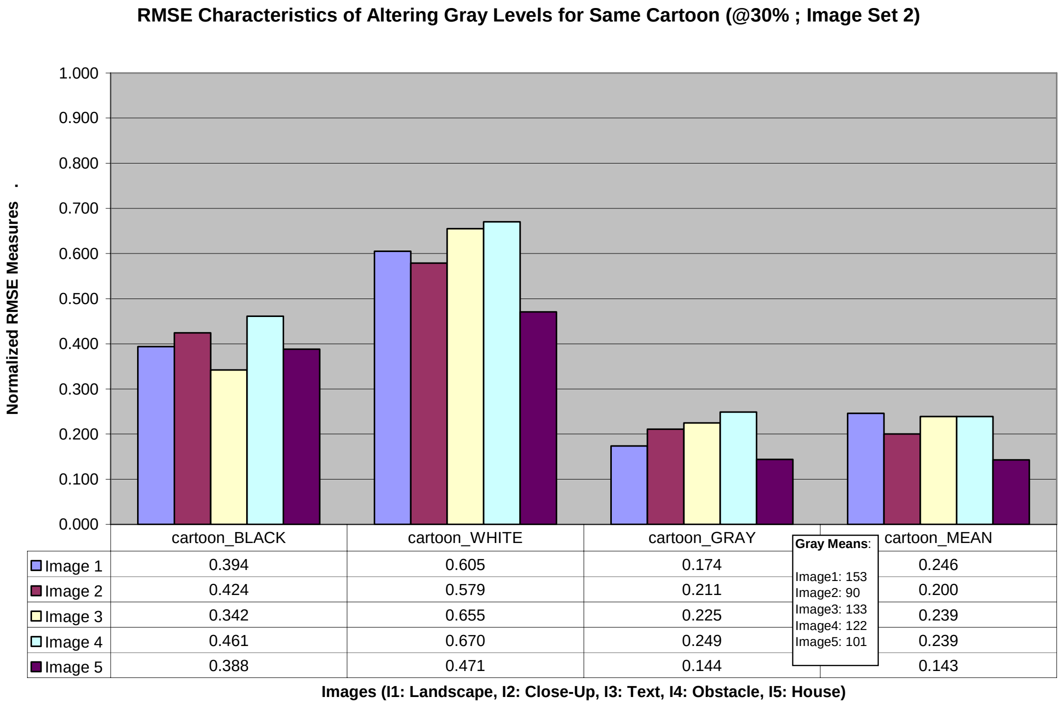
**Figure 26 Demonstration for applying the partial cartoon technique to a sample image (image 3 "text," from image set 1) using different gray shades; top: base image; mid-left: black cartoon (luminance: 0); mid-right: white cartoon (luminance: 255); bottom-left: gray cartoon (luminance: 128); bottom-right: mean cartoon (luminance: 133); note how the black and white cartoons provide higher contrast and more visible details.**



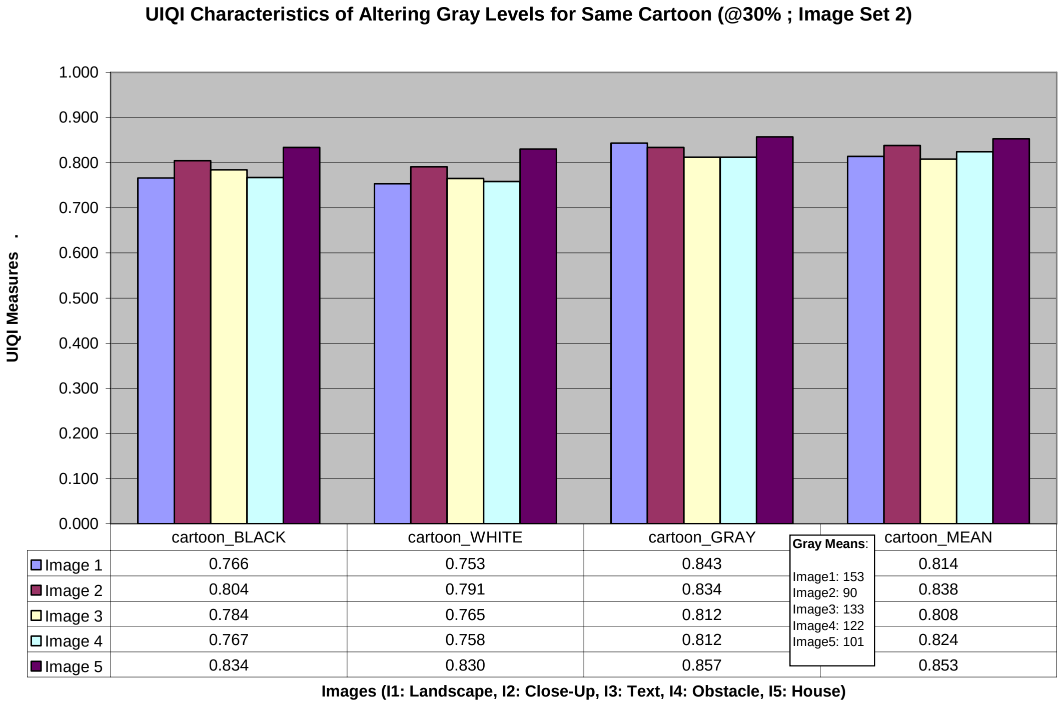
|  | cartoon_BLACK | cartoon_WHITE | cartoon_GRAY | cartoon_MEAN |
|---|---|---|---|---|
| Image 1 | 0.352 | 0.643 | 0.180 | 0.143 |
| Image 2 | 0.520 | 0.676 | 0.238 | 0.235 |
| Image 3 | 0.444 | 0.511 | 0.144 | 0.149 |
| Image 4 | 0.425 | 0.579 | 0.155 | 0.171 |
| Image 5 | 0.396 | 0.581 | 0.169 | 0.140 |

|  | cartoon_BLACK | cartoon_WHITE | cartoon_GRAY | cartoon_MEAN |
|---|---|---|---|---|
| Image 1 | 0.718 | 0.698 | 0.794 | 0.814 |
| Image 2 | 0.713 | 0.705 | 0.797 | 0.799 |
| Image 3 | 0.786 | 0.779 | 0.890 | 0.892 |
| Image 4 | 0.732 | 0.711 | 0.860 | 0.828 |
| Image 5 | 0.777 | 0.769 | 0.860 | 0.887 |

**Figure 27 Objective characteristics of the partial cartoon method for various gray shades (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**



| | cartoon_BLACK | cartoon_WHITE | cartoon_GRAY | cartoon_MEAN |
|---|---|---|---|---|
| Image 1 | 0.394 | 0.605 | 0.174 | 0.246 |
| Image 2 | 0.424 | 0.579 | 0.211 | 0.200 |
| Image 3 | 0.342 | 0.655 | 0.225 | 0.239 |
| Image 4 | 0.461 | 0.670 | 0.249 | 0.239 |
| Image 5 | 0.388 | 0.471 | 0.144 | 0.143 |



| | cartoon_BLACK | cartoon_WHITE | cartoon_GRAY | cartoon_MEAN |
|---|---|---|---|---|
| Image 1 | 0.766 | 0.753 | 0.843 | 0.814 |
| Image 2 | 0.804 | 0.791 | 0.834 | 0.838 |
| Image 3 | 0.784 | 0.765 | 0.812 | 0.808 |
| Image 4 | 0.767 | 0.758 | 0.812 | 0.824 |
| Image 5 | 0.834 | 0.830 | 0.857 | 0.853 |

**Figure 28 Objective characteristics of the partial cartoon method for various gray shades (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

The “Peli” method is evaluated through the same gray shades as above for image sets one and two, respectively, shown in Figure 30 and Figure 31. Here, the default method (as indicated in the literature and shown in Figure 9) is the white cartoon. It is stressed here that the method involves superimposing the entire image cartoon (including the truncated image) over the truncated image, and it thus not sensitive to the central image portion, possibly causing reduced or zero visibility. In addition, the cartoon is relatively small with a loss of proportionality versus the truncated image in both the horizontal and vertical dimensions, further distorting the image and requiring subject training, as noted in the literature.

The order of image quality for different gray shades is consistent with those listed with the partial cartoon where the highest are the mean and gray shades, then black, and finally the white cartoon. The exceptions to this exist in image set one, for the close-up and text scenes, where the UIQI for the black and white shades come within a 0.5% difference, which is accounted for by greater loss of contrast in the center of the image. As with the partial cartoon method, the black gray shade offers the best compromise between quality and visibility, while in all cases not being sensitive to the central image, especially with text images. This is illustrated in Figure 29.

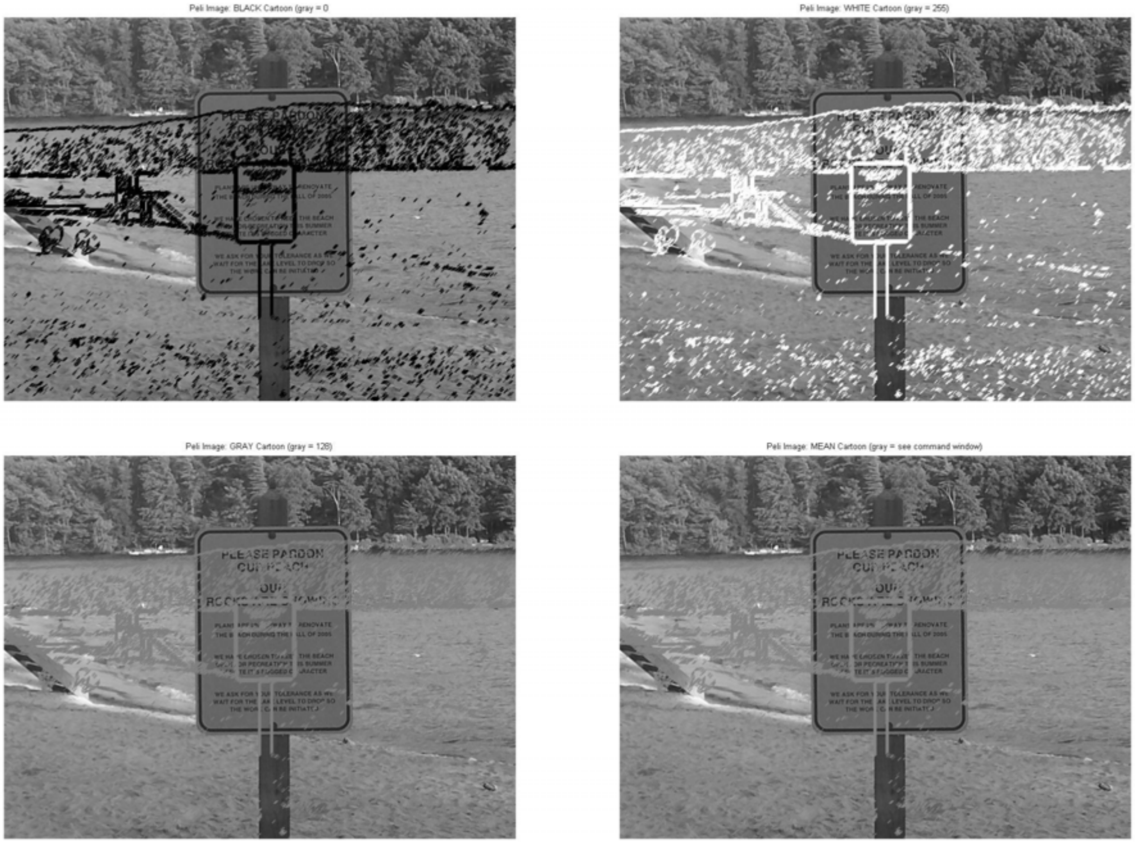
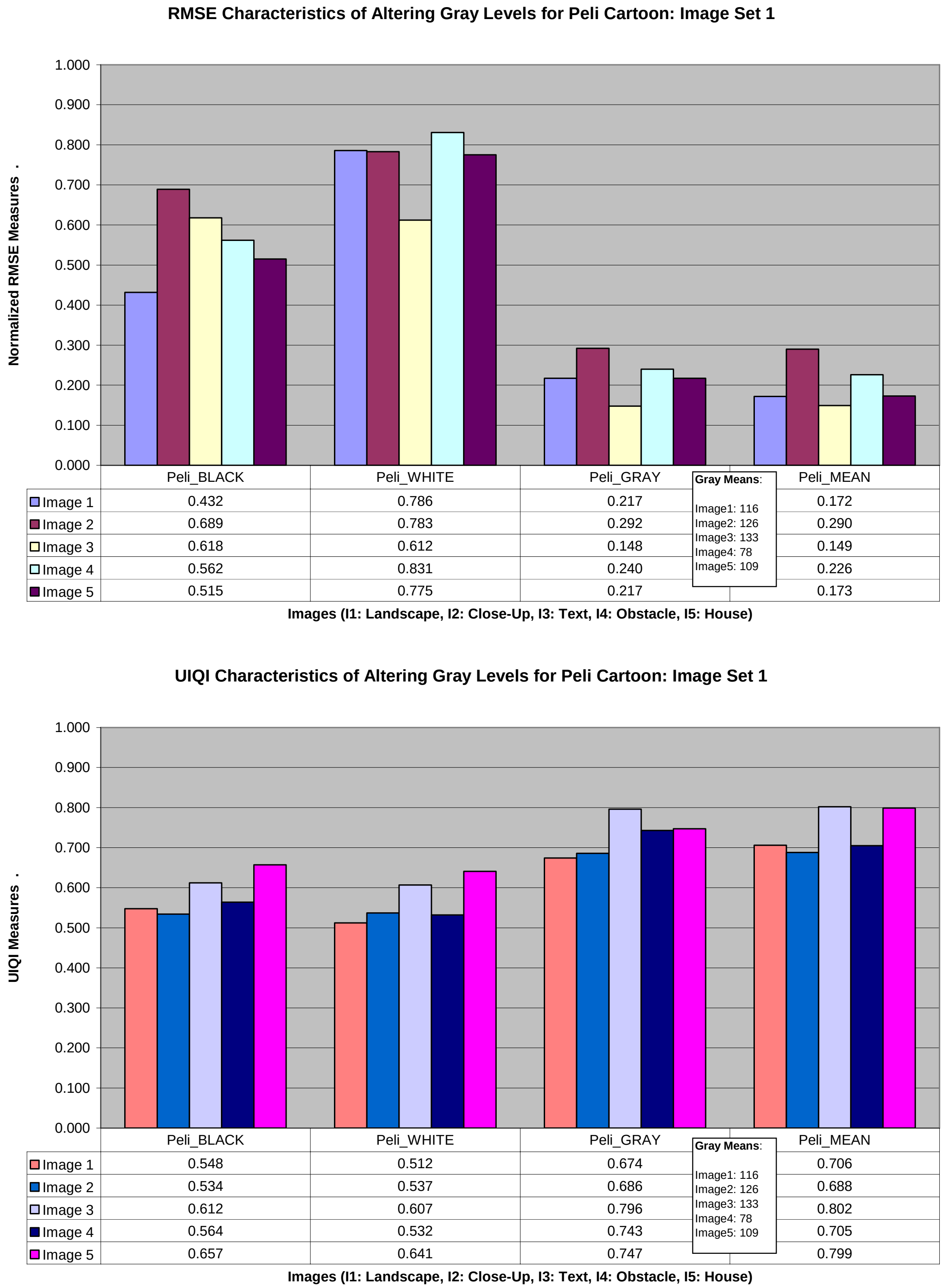

Peli Image: BLACK Cartoon (gray = 0
Peli Image: WHITE Cartoon (gray = 255)
Peli Image: GRAY Cartoon (gray = 128)
Peli Image: MEAN Cartoon (gray = see command window)

**Figure 29 Demonstration for applying the "Peli" superimposition technique to a sample image (image 3 "text," from image set 1) using different gray shades; top-left: black cartoon (luminance: 0); top-right: white cartoon (luminance: 255); bottom-left: gray cartoon (luminance: 128); bottom-right: mean cartoon (luminance: 133); note how in all cases, the sign is almost completely dominated by the cartoon.**



| | Peli_BLACK | Peli_WHITE | Peli_GRAY | Peli_MEAN |
|---|---|---|---|---|
| Image 1 | 0.432 | 0.786 | 0.217 | 0.172 |
| Image 2 | 0.689 | 0.783 | 0.292 | 0.290 |
| Image 3 | 0.618 | 0.612 | 0.148 | 0.149 |
| Image 4 | 0.562 | 0.831 | 0.240 | 0.226 |
| Image 5 | 0.515 | 0.775 | 0.217 | 0.173 |

Gray Means:

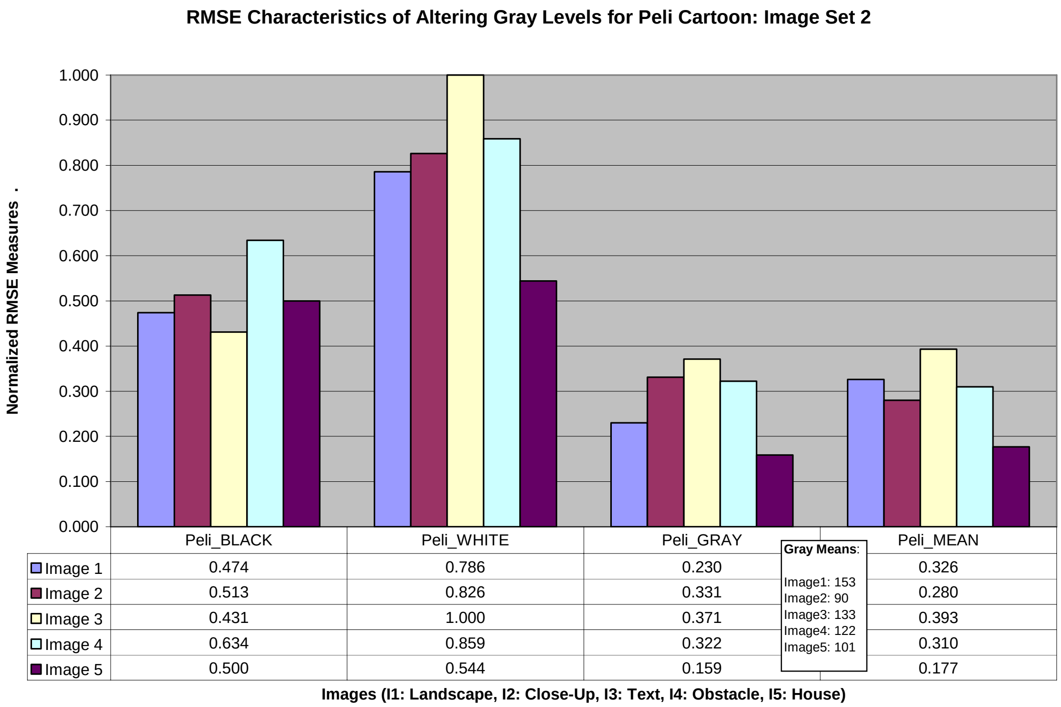
Image1: 116
Image2: 126
Image3: 133
Image4: 78
Image5: 109

| | Peli_BLACK | Peli_WHITE | Peli_GRAY | Peli_MEAN |
|---|---|---|---|---|
| Image 1 | 0.548 | 0.512 | 0.674 | 0.706 |
| Image 2 | 0.534 | 0.537 | 0.686 | 0.688 |
| Image 3 | 0.612 | 0.607 | 0.796 | 0.802 |
| Image 4 | 0.564 | 0.532 | 0.743 | 0.705 |
| Image 5 | 0.657 | 0.641 | 0.747 | 0.799 |

Gray Means:

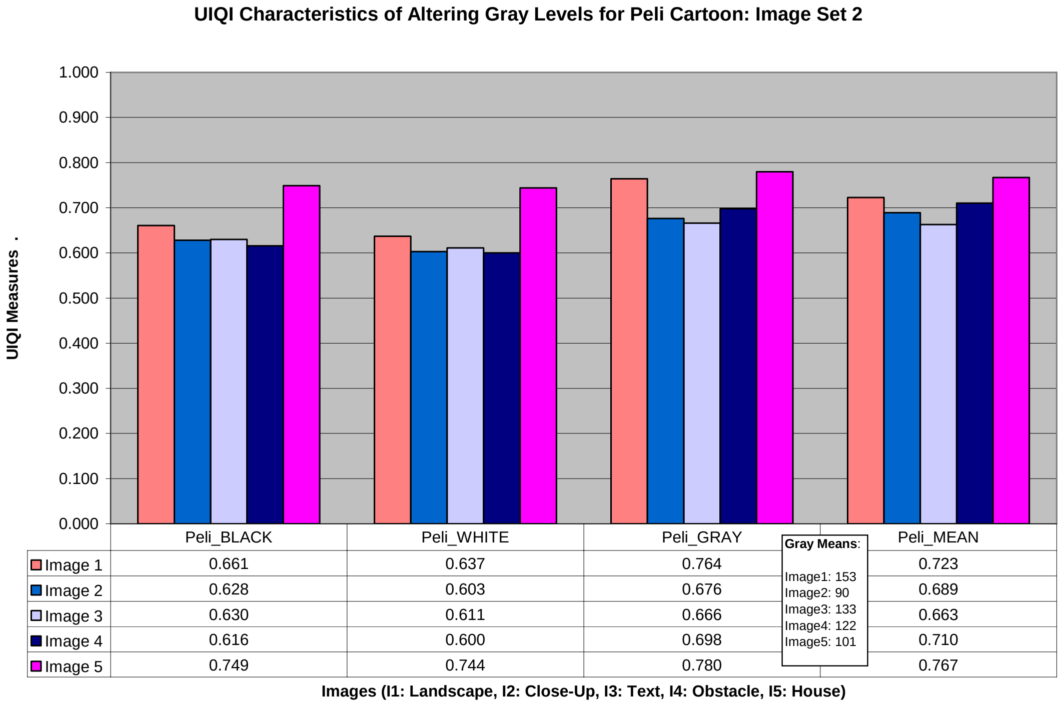
Image1: 116
Image2: 126
Image3: 133
Image4: 78
Image5: 109

**Figure 30 Objective characteristics of the “Peli” method for various gray shades (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**



RMSE Characteristics of Altering Gray Levels for Peli Cartoon: Image Set 2

| | Peli_BLACK | Peli_WHITE | Peli_GRAY | Peli_MEAN |
|---|---|---|---|---|
| Image 1 | 0.474 | 0.786 | 0.230 | 0.326 |
| Image 2 | 0.513 | 0.826 | 0.331 | 0.280 |
| Image 3 | 0.431 | 1.000 | 0.371 | 0.393 |
| Image 4 | 0.634 | 0.859 | 0.322 | 0.310 |
| Image 5 | 0.500 | 0.544 | 0.159 | 0.177 |



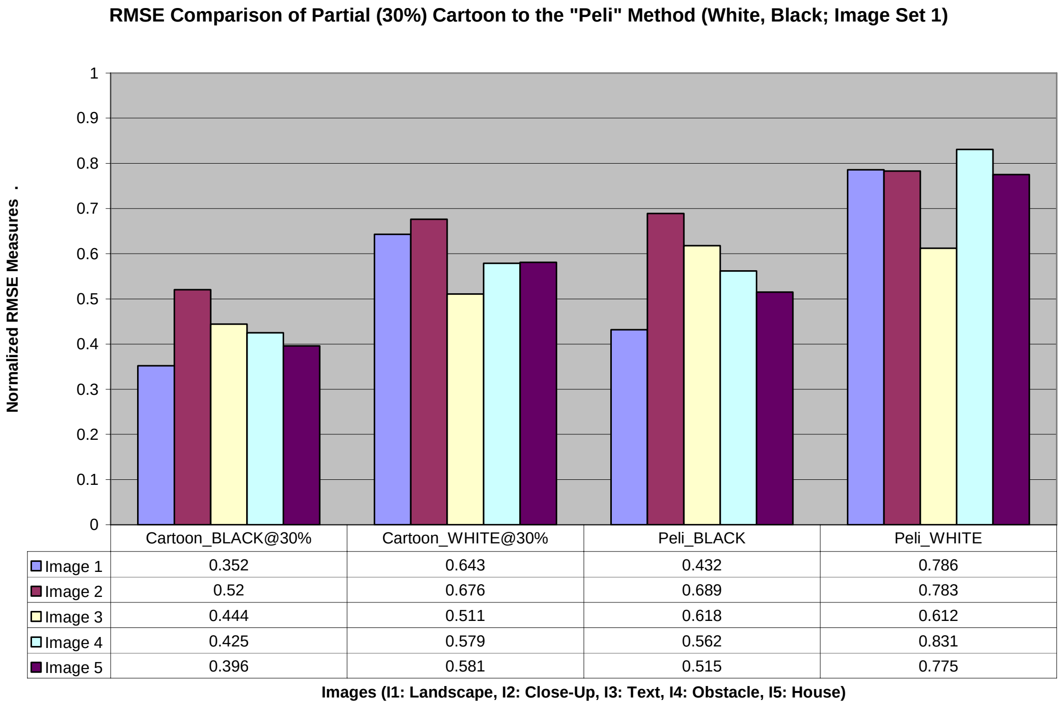
UIQI Characteristics of Altering Gray Levels for Peli Cartoon: Image Set 2

| | Peli_BLACK | Peli_WHITE | Peli_GRAY | Peli_MEAN |
|---|---|---|---|---|
| Image 1 | 0.661 | 0.637 | 0.764 | 0.723 |
| Image 2 | 0.628 | 0.603 | 0.676 | 0.689 |
| Image 3 | 0.630 | 0.611 | 0.666 | 0.663 |
| Image 4 | 0.616 | 0.600 | 0.698 | 0.710 |
| Image 5 | 0.749 | 0.744 | 0.780 | 0.767 |

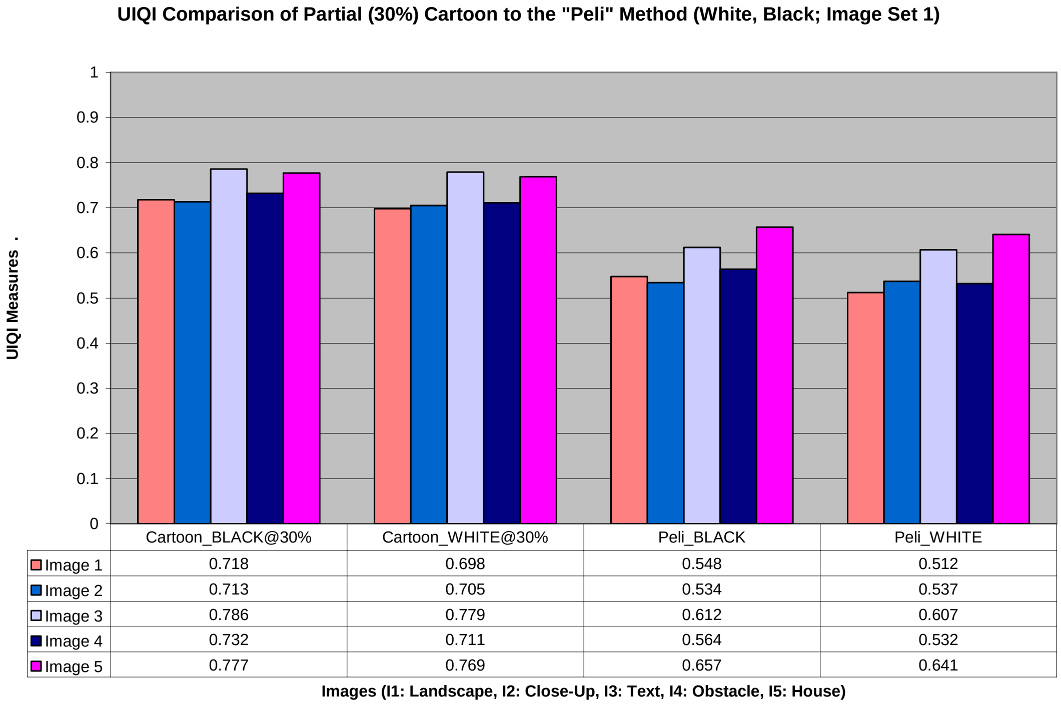
**Figure 31 Objective characteristics of the "Peli" method for various gray shades (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

A comparison of the partial cartoon and “Peli” methods through various gray shades in image sets one and two is shown in Figure 32 and Figure 33. Only the default black (for the partial cartoon) and white (for “Peli”) cartoons are considered, and mean and gray cartoons are discarded due to low visibility and low contrast, as shown in Figure 26 and Figure 29 above. Consistent image quality trends are visible in both plots, with minor differences between the RMSE and UIQI values.

It is noted that the RMSE characteristics ranked from lowest to highest include the black partial cartoon, black “Peli” cartoon, white partial cartoon, and white Peli cartoon (highest RMSE), while the UIQI characteristics ranked from top to bottom are the black then white partial cartoons, followed by the black and white “Peli” cartoons. Also note that UIQI variations among partial cartoon image sets is smaller (about 7%) than among Peli images (about 13%), i.e. almost double the partial cartoon method variability. This difference is justified by the UIQI measures being more sensitive to overall contrast differences than the RMSE. Overall, image quality results are mostly in favor of the partial cartoon compared to the Peli method. Specifically, the black, partial cartoon ranks first among all image quality measures.

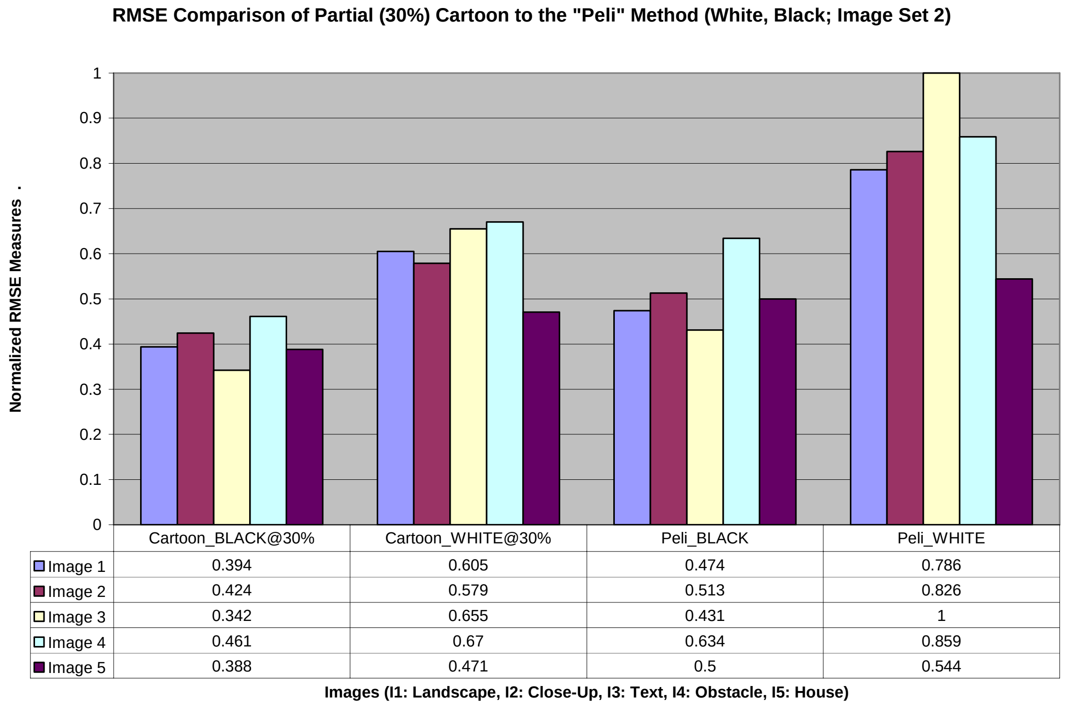


| | Cartoon_BLACK@30% | Cartoon_WHITE@30% | Peli_BLACK | Peli_WHITE |
|---|---|---|---|---|
| Image 1 | 0.352 | 0.643 | 0.432 | 0.786 |
| Image 2 | 0.52 | 0.676 | 0.689 | 0.783 |
| Image 3 | 0.444 | 0.511 | 0.618 | 0.612 |
| Image 4 | 0.425 | 0.579 | 0.562 | 0.831 |
| Image 5 | 0.396 | 0.581 | 0.515 | 0.775 |



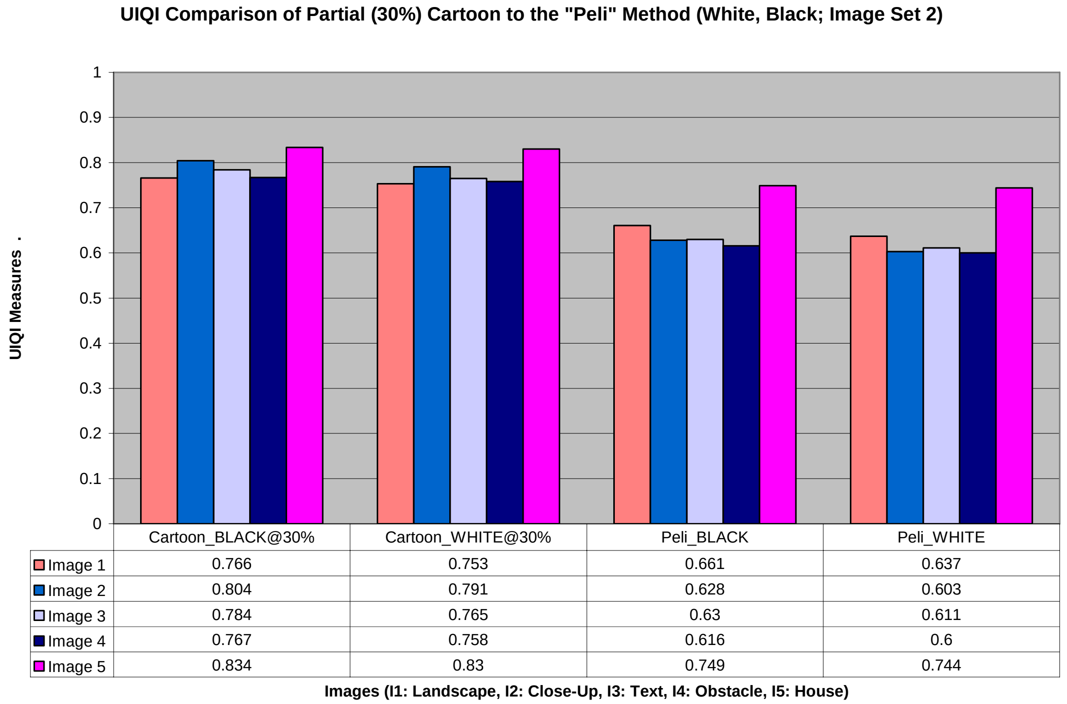
| | Cartoon_BLACK@30% | Cartoon_WHITE@30% | Peli_BLACK | Peli_WHITE |
|---|---|---|---|---|
| Image 1 | 0.718 | 0.698 | 0.548 | 0.512 |
| Image 2 | 0.713 | 0.705 | 0.534 | 0.537 |
| Image 3 | 0.786 | 0.779 | 0.612 | 0.607 |
| Image 4 | 0.732 | 0.711 | 0.564 | 0.532 |
| Image 5 | 0.777 | 0.769 | 0.657 | 0.641 |

**Figure 32 Objective comparison of the partial cartoon and "Peli" methods for various gray shades (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**



RMSE Comparison of Partial (30%) Cartoon to the "Peli" Method (White, Black; Image Set 2)

| | Cartoon_BLACK@30% | Cartoon_WHITE@30% | Peli_BLACK | Peli_WHITE |
|---|---|---|---|---|
| ■ Image 1 | 0.394 | 0.605 | 0.474 | 0.786 |
| ■ Image 2 | 0.424 | 0.579 | 0.513 | 0.826 |
| □ Image 3 | 0.342 | 0.655 | 0.431 | 1 |
| □ Image 4 | 0.461 | 0.67 | 0.634 | 0.859 |
| ■ Image 5 | 0.388 | 0.471 | 0.5 | 0.544 |



UIQI Comparison of Partial (30%) Cartoon to the "Peli" Method (White, Black; Image Set 2)

| | Cartoon_BLACK@30% | Cartoon_WHITE@30% | Peli_BLACK | Peli_WHITE |
|---|---|---|---|---|
| ■ Image 1 | 0.766 | 0.753 | 0.661 | 0.637 |
| ■ Image 2 | 0.804 | 0.791 | 0.628 | 0.603 |
| □ Image 3 | 0.784 | 0.765 | 0.63 | 0.611 |
| ■ Image 4 | 0.767 | 0.758 | 0.616 | 0.6 |
| ■ Image 5 | 0.834 | 0.83 | 0.749 | 0.744 |

**Figure 33 Objective comparison of the partial cartoon and "Peli" methods for various gray shades (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

Characteristics of the convex method for various remapping (*k*, the measure of the relative translation of each pixel from the origin) and proportionality (*K*, the measure of the relative image size) factors are displayed in Figure 35 and Figure 36, respectively, for image sets one and two. The purpose of this analysis is to determine optimum convex parameters for this method between measured quality and visible clarity (which is also assessed subjectively). The convex method, as previously mentioned, is involved with completely replacing the truncated image with a convex or "fish-eye" compensated image. The testing is based on fixing *k* for various *K* factors, and vice versa as shown in the 3-D figures. Note that the convex method, in general, requires large computational time[1], which significantly reduces the resulting resolution, and the corresponding final image size. This is practically illustrated in Figure 34.

It is observed that increasing *K* (up to 1) reduces the size of the resulting image. Thus, when resized to the truncated image size (600*800), distortion significantly increases. On the other hand, increasing *k* furthers remapped points from the image origin, which apparently increases the convexity, as well as the distortion of the warped image. Deviation from default values is illustrated in Figure 37. It is noted that the selected range of warping parameters has been limited to 1.10 to 1.20 for *k* and 0.4 to 1.0 for *K*. For values below this range (for either parameter), significant discontinuities in the form of holes at the image center appear in the warped image, while a significant decrease in resolution is the result of increasing *k* and *K* beyond the selected ranges.

[1] Average *Matlab*® computational times are 5.7, 3.0, and 2.2 sec for the convex, "Peli," and partial cartoon methods.

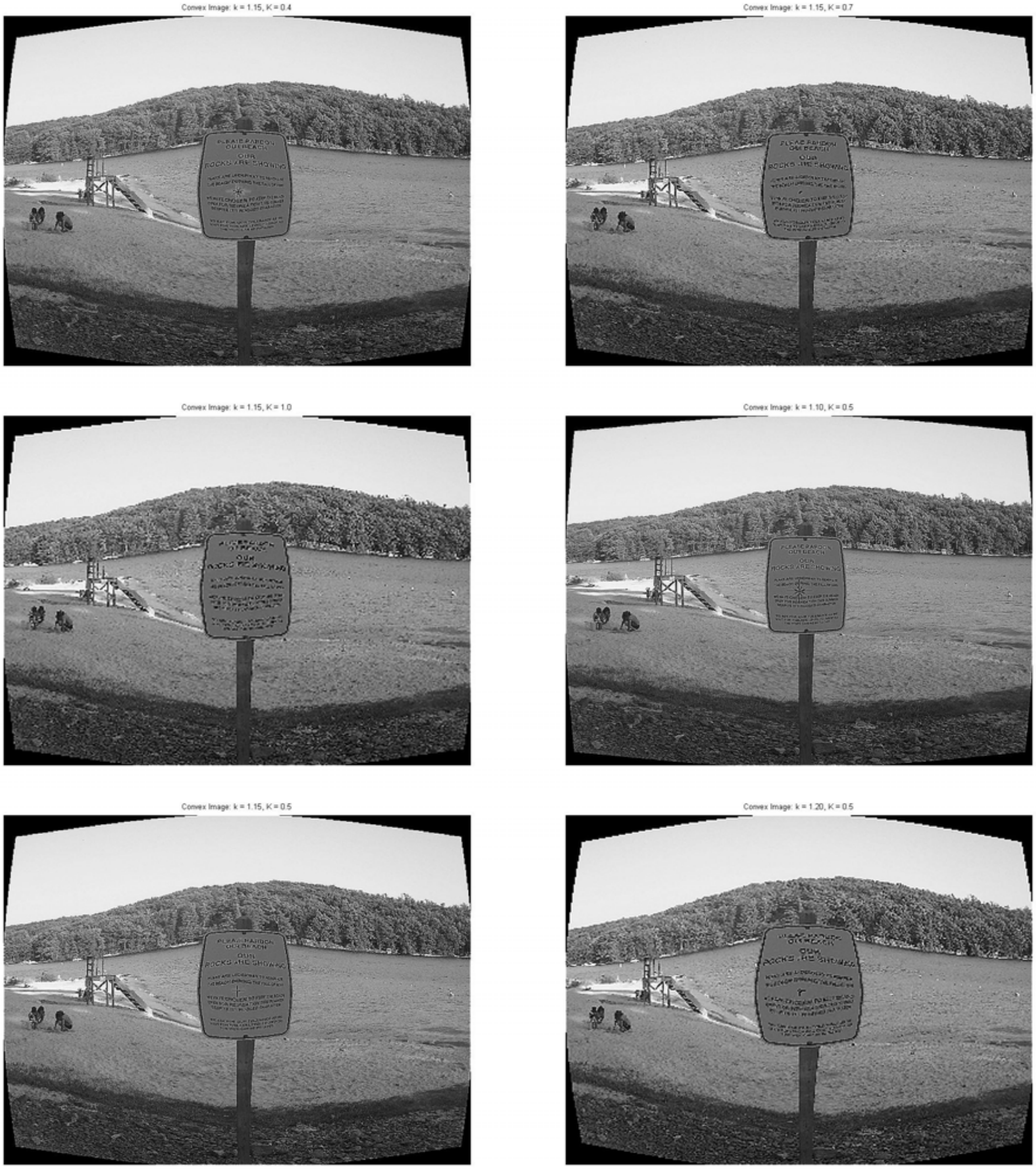
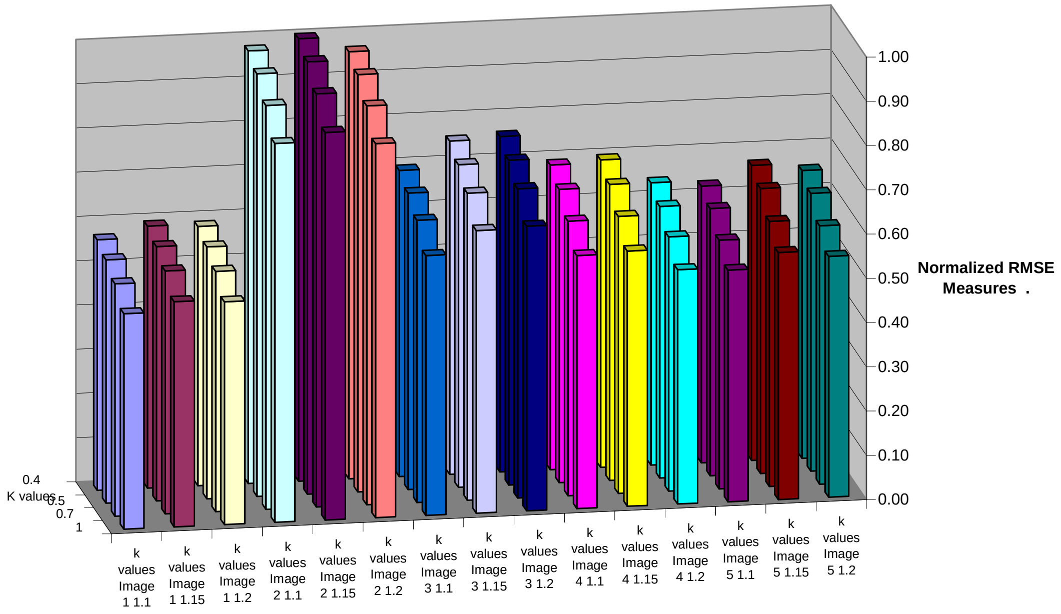

Convex Image: k = 1.15, K = 0.4
Convex Image: k = 1.15, K = 0.7
Convex Image: k = 1.15, K = 1.0
Convex Image: k = 1.10, K = 0.5
Convex Image: k = 1.15, K = 0.5
Convex Image: k = 1.20, K = 0.5

**Figure 34 Demonstration for applying the convex or "fish-eye" technique to a sample image (image 3 "text," from image set 1) using different *k* (remapping) and *K* (proportionality) parameters; the first three images involve varying *K* (0.4, 0.7, 1.0) at the fixed, default *k* (1.15), and the last three involve varying *k* (1.10, 1.15, 1.20) at the fixed default *K* (0.5);.note the compromise between increased convexity (*k*) and increased image distortion, as well the decreased resolution due to resizing reduced-size images for various *K* factors.**

RMSE Characteristics of Convex Method (*k* : remapping, *K* : proportionality); Image Set 1



UIQI Characteristics of Convex Method (*k* : remapping, *K* : proportionality); Image Set 1

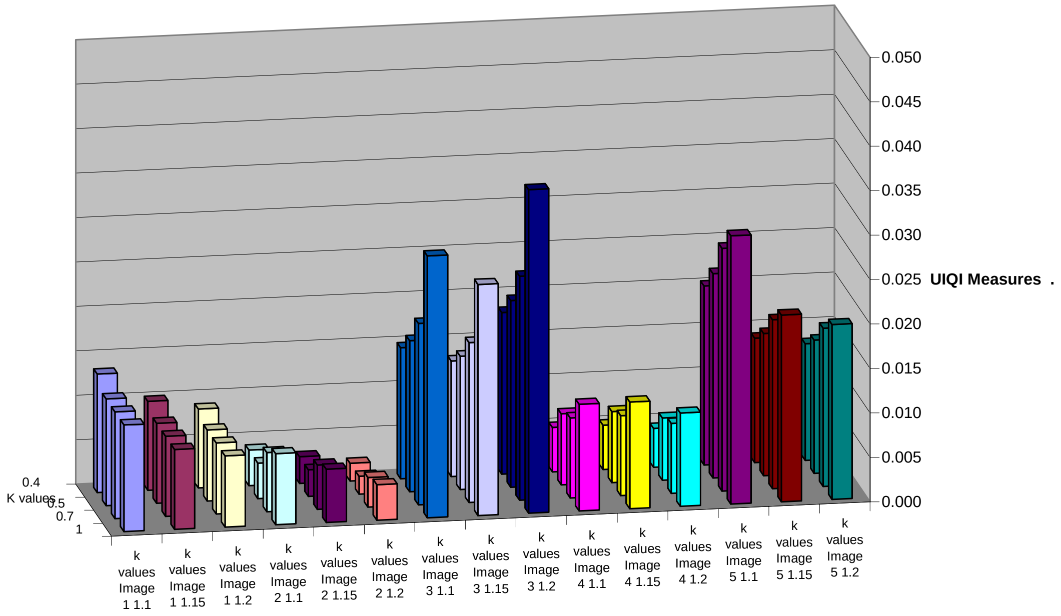


**Figure 35 Objective characteristics of the convex method for various remapping (*k*) and proportionality (*K*) factors (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

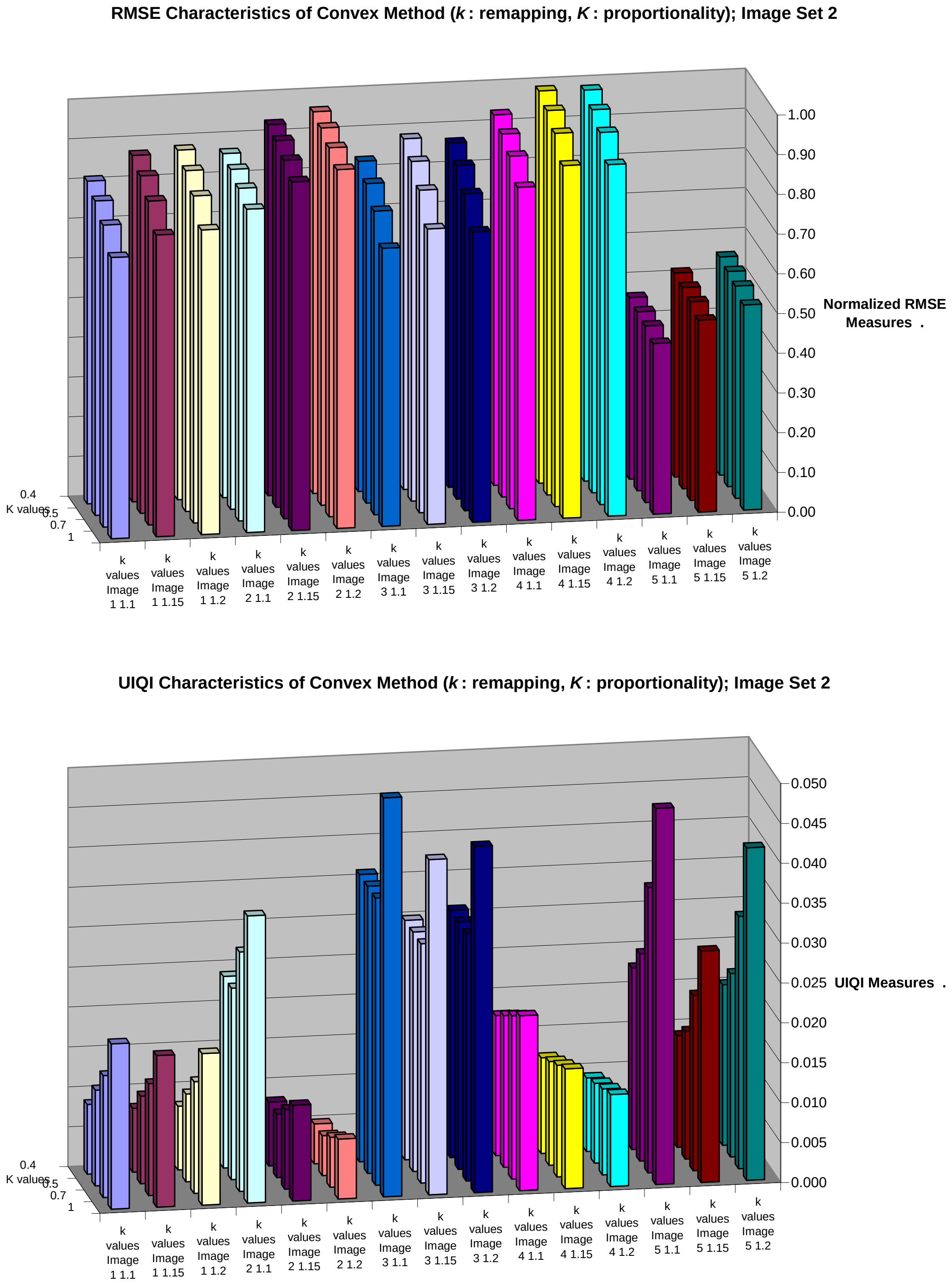


**Figure 36 Objective characteristics of the convex method for various remapping (*k*) and proportionality (*K*) factors (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

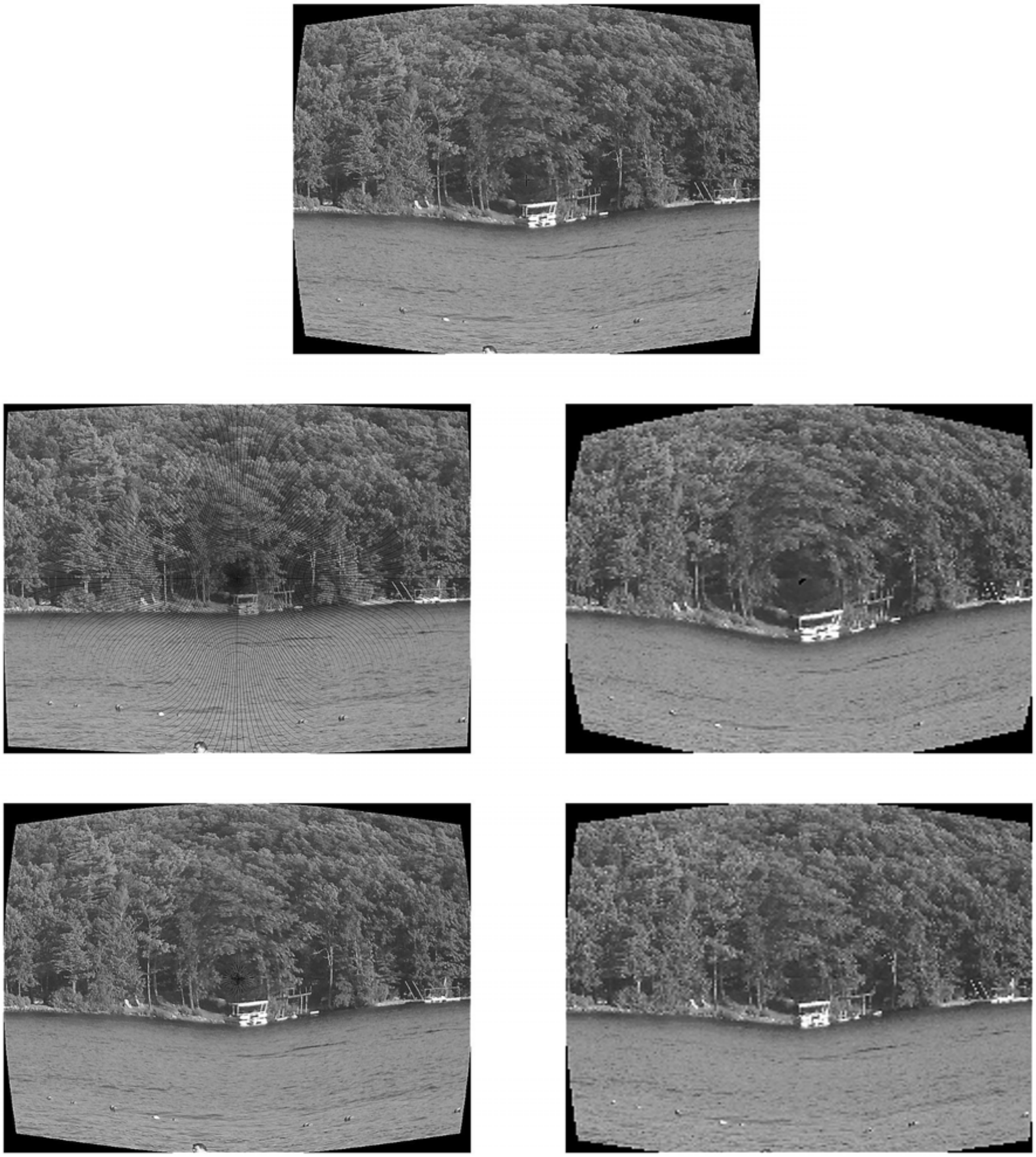
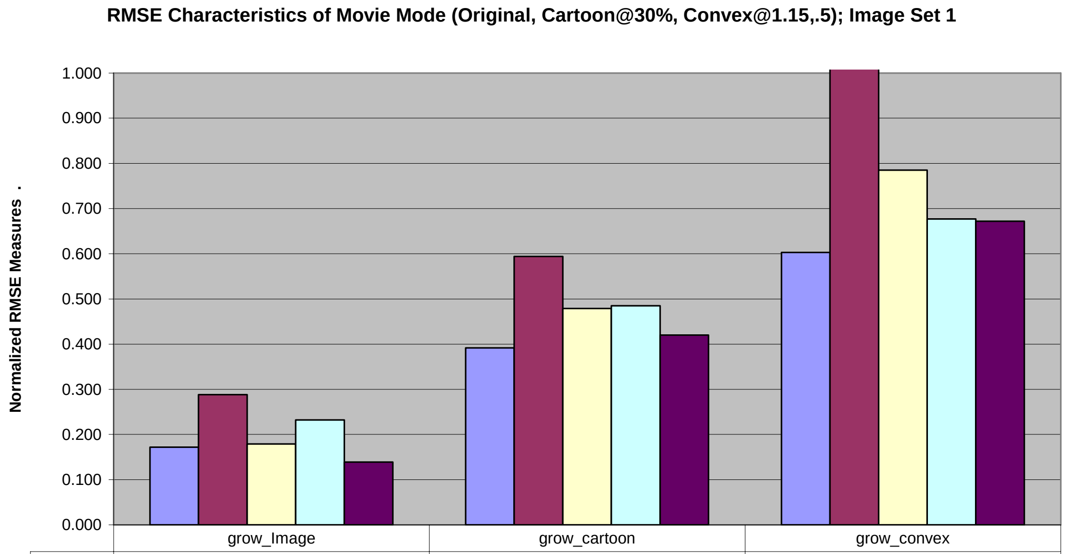


**Figure 37 Demonstration for applying the convex technique to a sample image (image 1 “landscape,” from image set 1) while deviating *k* (remapping) and *K* (proportionality) parameters significantly enough to be eliminated from both the objective and subjective analysis; top: default truncated image at *k* = 1.15, *K* = 0.5; mid-left: *k* = 1.05, *K* = 0.5; mid-right: *k* = 1.25, *K* = 0.5; bottom-left: *k* = 1.15, *K* = 0.3; bottom-right: *k* = 1.15, *K* = 1.5;.note the increase in discontinuities or decrease in resolution due to this deviation.**

The minimum RMSE occurs at $k$ = 1.15, and $K$ = 1.0, which is accounted for by minimal number of pixels generated in the warped image. The previous values are chosen along with $k$ = 1.15, $K$ = 0.5 (default values) for further comparison in the subjective analysis. Also note that UIQI values are very low, which is accounted for by the many discontinuities and high distortion in the images, documented by the authors [2,26] with regards to this method of image warping.

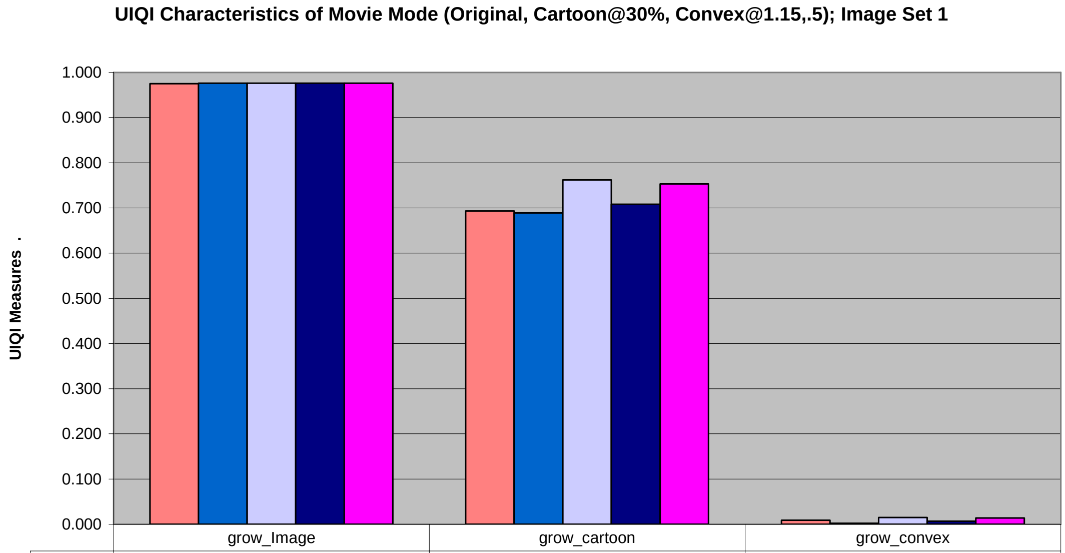
Comparison of the movie modes in the form of a growing area in the image for the partial cartoon and convex methods is shown in Figure 38 and Figure 39, respectively, for image sets one and two. The purpose of this technique is to determine the movie quality for the original image compared to the partial cartoon and convex methods, where the control is the original image. Note that the moving dot could not be assessed objectively, so this method has been deferred to the subjective analysis. Also, for evaluation purposes, the growing dot has been assessed at the end of the movie process, i.e. at maximum distortion. Defaults used with the partial cartoon method are the 30%, black cartoon, and a convex method with $k$ = 1.10, $K$ = 0.5.

On inspection of the comparative image quality measures, the RMSE results show consistency with previous results, where the cartoon method is superior (almost 1.5-2 times higher) than the convex method, while the original image is the overall highest, as predicted. The UIQI data indicates that the cartoon method is also higher (reaching up to 0.75 out of a possible maximum of 1) and consistent (a range of approximately 6% for all images), the convex image results are again almost zero, while the control values are almost one, the latter being interpreted as a good control sign. The same conclusion is drawn here as with all previous objective results, i.e. that the cartoon method is a superior one, even in the movie mode.



| | grow_Image | grow_cartoon | grow_convex |
|---|---|---|---|
| ■ Image 1 | 0.172 | 0.392 | 0.603 |
| ■ Image 2 | 0.288 | 0.594 | 1.011 |
| □ Image 3 | 0.179 | 0.479 | 0.785 |
| □ Image 4 | 0.232 | 0.485 | 0.677 |
| ■ Image 5 | 0.139 | 0.420 | 0.672 |

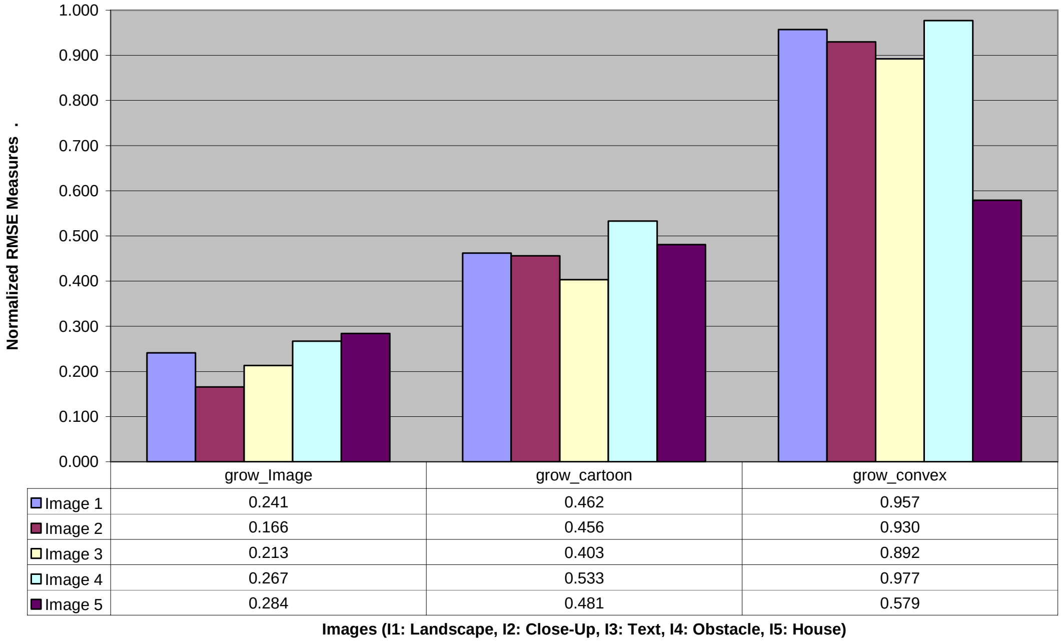
Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)



| | grow_Image | grow_cartoon | grow_convex |
|---|---|---|---|
| ■ Image 1 | 0.975 | 0.693 | 0.009 |
| ■ Image 2 | 0.976 | 0.689 | 0.002 |
| □ Image 3 | 0.976 | 0.762 | 0.015 |
| ■ Image 4 | 0.976 | 0.708 | 0.007 |
| ■ Image 5 | 0.976 | 0.753 | 0.014 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)

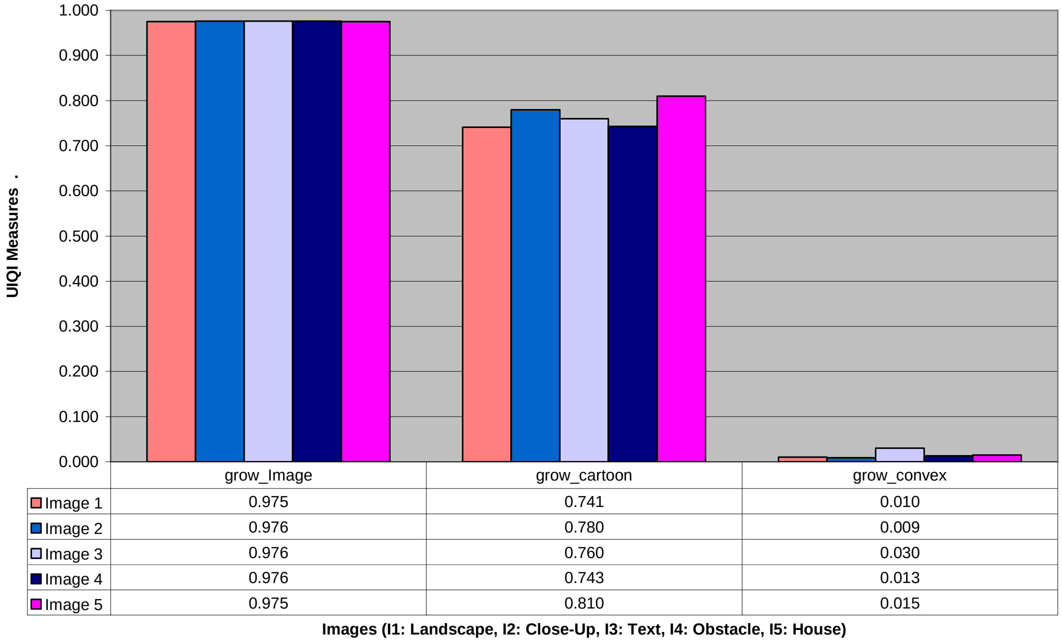
**Figure 38 Objective characteristics of the movie modes for the partial cartoon and convex methods (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**



RMSE Characteristics of Movie Mode
(Original, Cartoon@30%, Convex@k:1.15,K:.5); Image Set 2

| | grow_Image | grow_cartoon | grow_convex |
|---|---|---|---|
| Image 1 | 0.241 | 0.462 | 0.957 |
| Image 2 | 0.166 | 0.456 | 0.930 |
| Image 3 | 0.213 | 0.403 | 0.892 |
| Image 4 | 0.267 | 0.533 | 0.977 |
| Image 5 | 0.284 | 0.481 | 0.579 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)



UIQI Characteristics of Movie Mode
(Original, Cartoon@30%, Convex@k:1.15,K:.5); Image Set 2

| | grow_Image | grow_cartoon | grow_convex |
|---|---|---|---|
| Image 1 | 0.975 | 0.741 | 0.010 |
| Image 2 | 0.976 | 0.780 | 0.009 |
| Image 3 | 0.976 | 0.760 | 0.030 |
| Image 4 | 0.976 | 0.743 | 0.013 |
| Image 5 | 0.975 | 0.810 | 0.015 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)

**Figure 39 Objective characteristics of the movie modes for the partial cartoon and convex methods (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

A comprehensive comparison of the partial cartoon, “Peli,” and convex methods for image sets one and two is respectively plotted and tabulated in Figure 41 and Figure 42, and practically illustrated in Figure 40. The aim here is to determine the best objective method after determining the local default parameters among each of these three methods, and the extent of superiority of each to the other. The defaults of the three methods are black cartoon at 30% superimposition for the partial cartoon method, white cartoon (as shown in the literature [7]) for the “Peli” method, and $k = 1.15$, $K = 0.5$ parameters for the convex method.

The plots indicate once again that the partial cartoon method offers the lowest normalized RMSE (as low as about 0.34), and highest UIQI (as high as 0.83), which is consistent for all images. At second place, the convex and “Peli” method compete in an inconsistent manner. A conclusion can be drawn for objective results that the partial cartoon is the method with highest image quality measures, yet verification is needed to determine if this will be consistent with subjective (or perception) results.

Figure 43 demonstrates the overall consistency, or correlation (***r***) of the shown results. It is shown that ***r*** is statistically significant (i.e. repeatable within 95% certainty for the acquired data points) across all images, as well as between RMSE and UIQI values, although here a negative correlation exists, as well as between image sets (almost 1). The lowest ***r*** values exist for convex images (-0.37), which remains statistically significant (based on degrees of freedom [6]). Note, however, no correlation is possible with regards to mean shades or image setting (***r***: -0.01), which implies that results are independent of the nature of the images themselves, also supporting the choice of RMSE and UIQI measures which are independent of image properties.

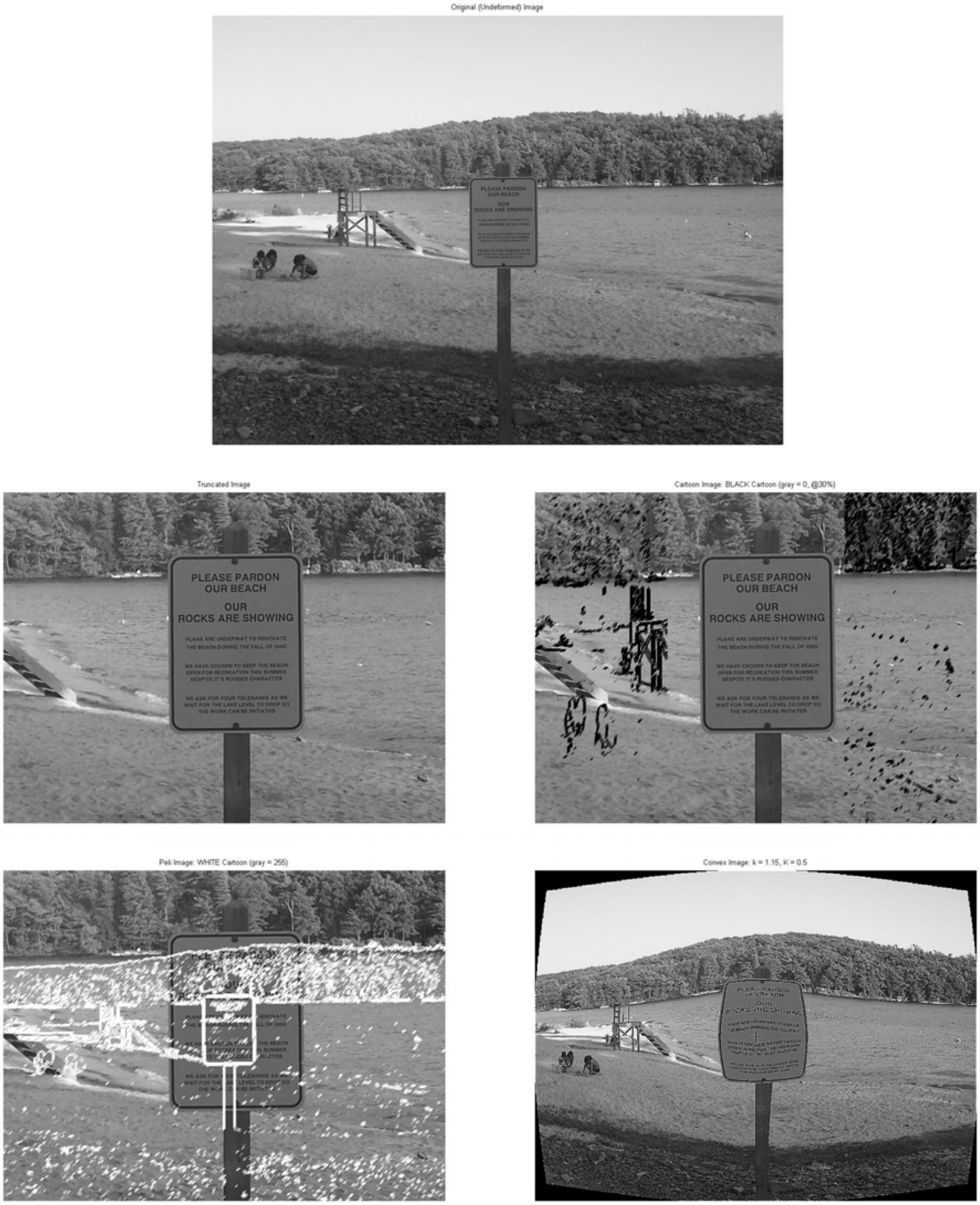
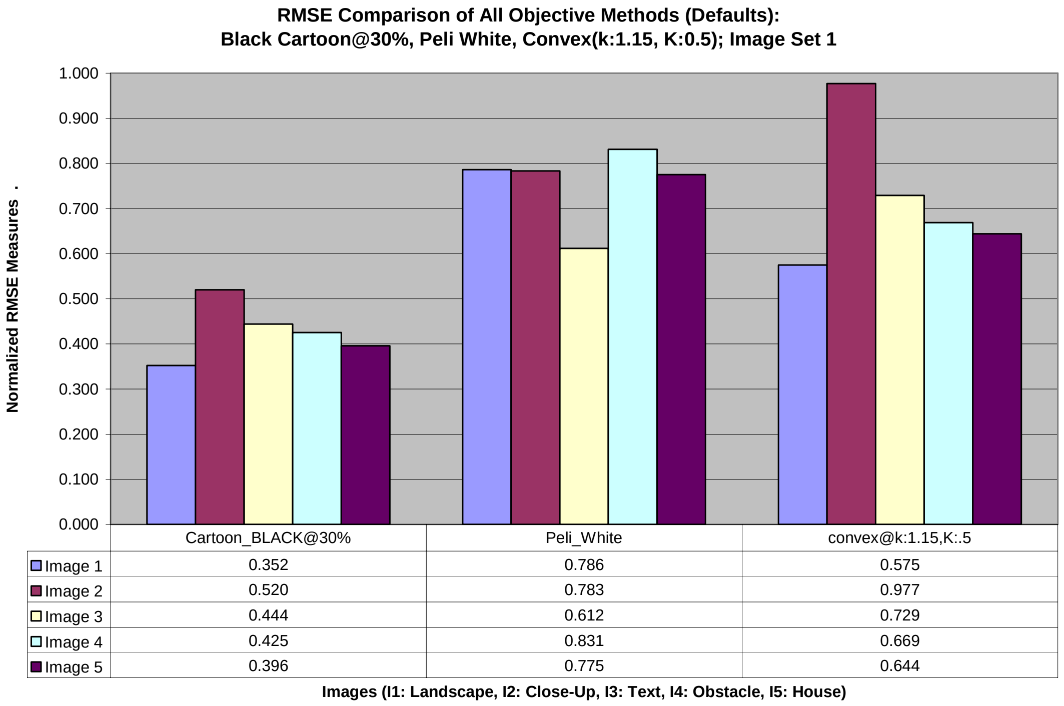
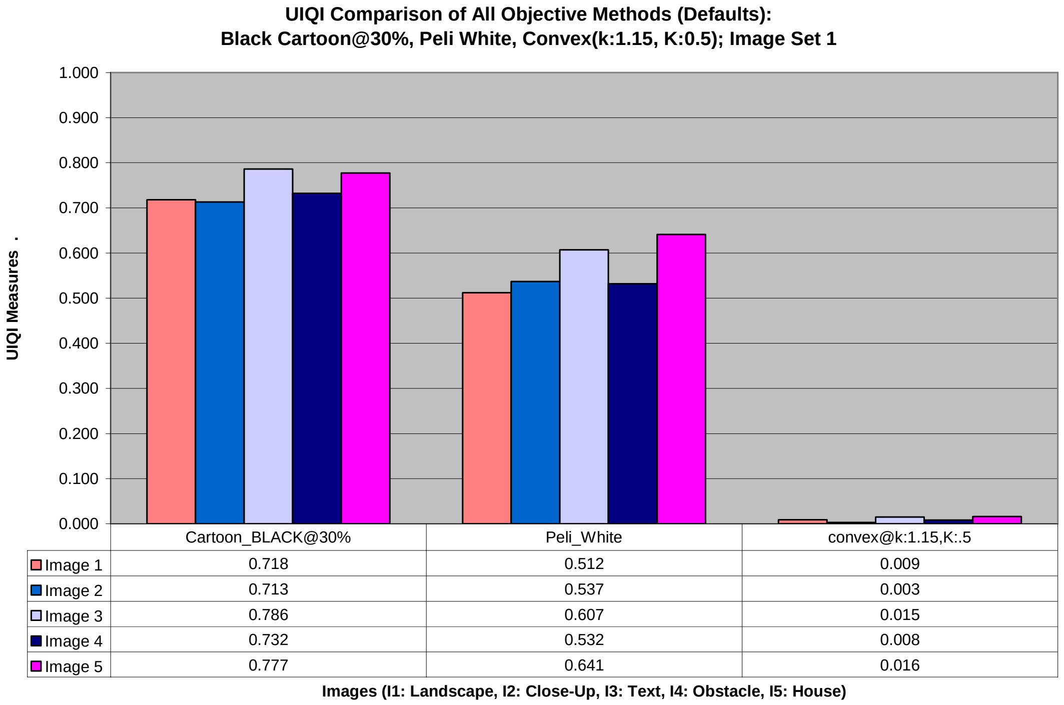

Original (Undeformed) Image
Truncated Image
Cartoon Image: BLACK Cartoon (gray = 0, @30%)
Peli Image: WHITE Cartoon (gray = 255)
Convex Image: k = 1.15, K = 0.5
PLEASE PARDON
OUR BEACH
OUR
ROCKS ARE SHOWING
PLANS ARE UNDERWAY TO RENOVATE
THE BEACH DURING THE FALL OF 2005
WE HAVE CHOSEN TO KEEP THE BEACH
OPEN FOR RECREATION THIS SUMMER
DESPITE IT'S RUGGED CHARACTER
WE ASK FOR YOUR TOLERANCE AS WE
WAIT FOR THE LAKE LEVEL TO DROP SO
THE WORK CAN BE INITIATED

**Figure 40 Comparison of the partial cartoon, "Peli," and convex methods for a sample image (image 3 "text," from image set 1) using remapping defaults of each method; top: base image; mid-left: truncated image; mid-right: partial cartoon method (30%, black); bottom-left: "Peli" method (white); bottom-right: convex method (*k*: 1.15, *K*: 0.5); both the text in the sign *and* the surroundings are clear *only* in the partial cartoon image.**

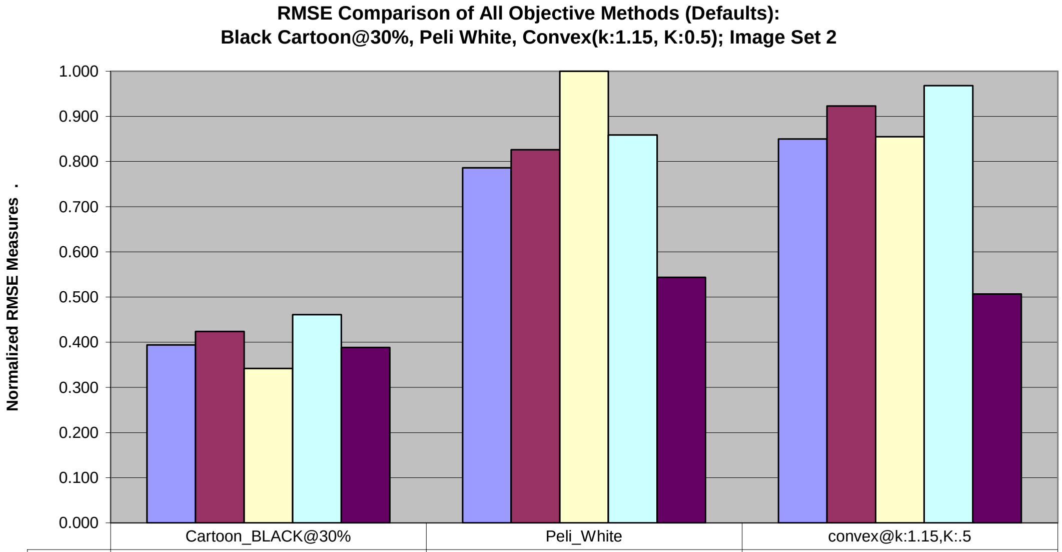

RMSE Comparison of All Objective Methods (Defaults): Black Cartoon@30%, Peli White, Convex(k:1.15, K:0.5); Image Set 1

| | Cartoon_BLACK@30% | Peli_White | convex@k:1.15,K:.5 |
|---|---|---|---|
| Image 1 | 0.352 | 0.786 | 0.575 |
| Image 2 | 0.520 | 0.783 | 0.977 |
| Image 3 | 0.444 | 0.612 | 0.729 |
| Image 4 | 0.425 | 0.831 | 0.669 |
| Image 5 | 0.396 | 0.775 | 0.644 |

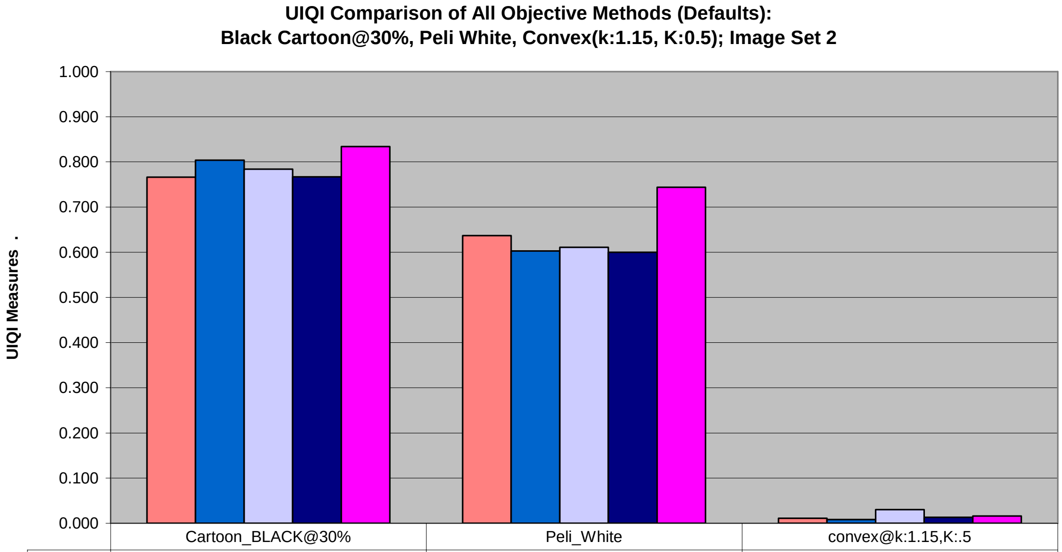

UIQI Comparison of All Objective Methods (Defaults): Black Cartoon@30%, Peli White, Convex(k:1.15, K:0.5); Image Set 1

| | Cartoon_BLACK@30% | Peli_White | convex@k:1.15,K:.5 |
|---|---|---|---|
| Image 1 | 0.718 | 0.512 | 0.009 |
| Image 2 | 0.713 | 0.537 | 0.003 |
| Image 3 | 0.786 | 0.607 | 0.015 |
| Image 4 | 0.732 | 0.532 | 0.008 |
| Image 5 | 0.777 | 0.641 | 0.016 |

**Figure 41 Objective comparison of the partial cartoon, "Peli," and convex methods (image set 1); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**



| | Cartoon_BLACK@30% | Peli_White | convex@k:1.15,K:.5 |
|---|---|---|---|
| ▨ Image 1 | 0.394 | 0.786 | 0.850 |
| ▨ Image 2 | 0.424 | 0.826 | 0.923 |
| □ Image 3 | 0.342 | 1.000 | 0.855 |
| □ Image 4 | 0.461 | 0.859 | 0.968 |
| ■ Image 5 | 0.388 | 0.544 | 0.507 |



| | Cartoon_BLACK@30% | Peli_White | convex@k:1.15,K:.5 |
|---|---|---|---|
| ▨ Image 1 | 0.766 | 0.637 | 0.011 |
| ■ Image 2 | 0.804 | 0.603 | 0.008 |
| □ Image 3 | 0.784 | 0.611 | 0.030 |
| ■ Image 4 | 0.767 | 0.600 | 0.013 |
| ■ Image 5 | 0.834 | 0.744 | 0.016 |

**Figure 42 Objective comparison of the partial cartoon, "Peli," and convex methods (image set 2); top: data and plot for normalized RMSE values; bottom: data and plot for UIQI values.**

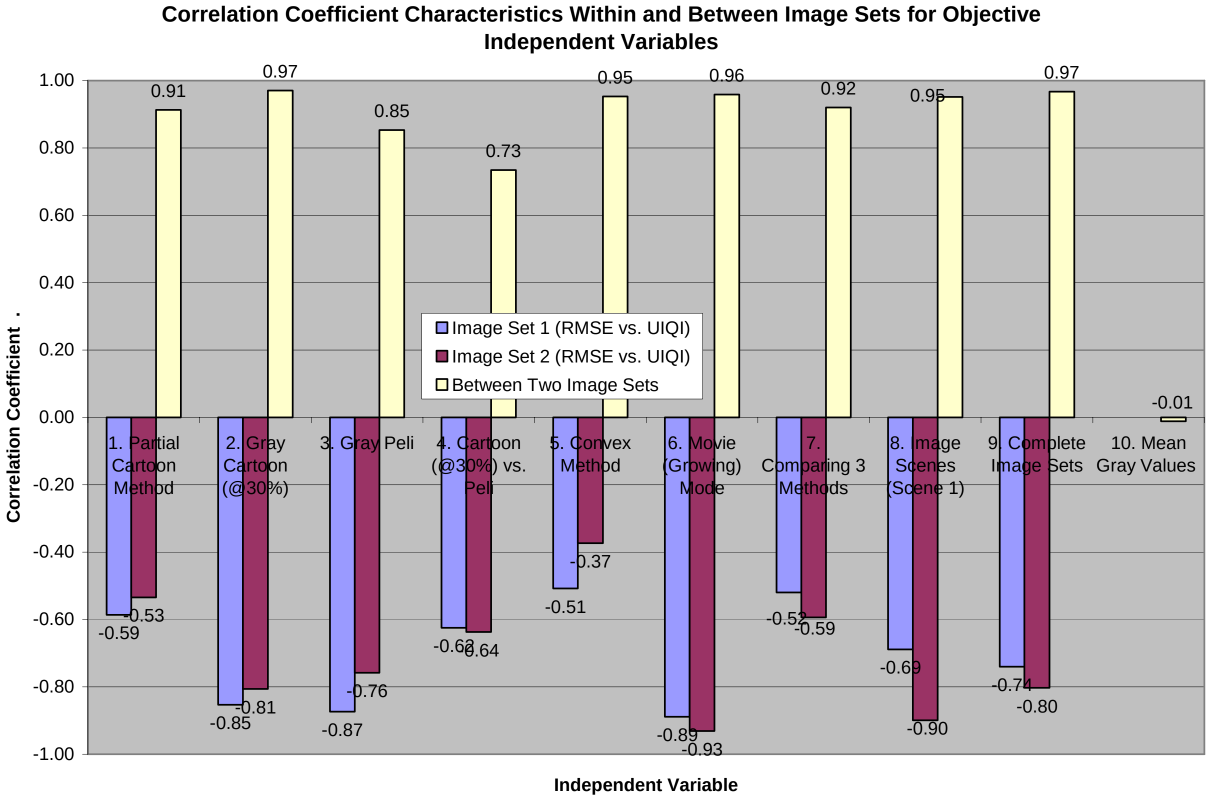


**Figure 43 Correlation coefficients for all analyzed objective variables (image set 1 and 2); note, all coefficients achieve statistical significance, except those between mean gray values of image sets.**

## Summary of Objective Results

Finally, here are some noteworthy conclusions for the above objective results. The partial cartoon method (at prescribed defaults) yields the highest image quality results in all study categories. The "Peli" and convex methods compete for second place. The results are all precise, varying between the lowest at ***r*** = -0.37, for the convex method, to a highest of ***r*** = 0.97. Objective results, in addition, do not depend on image setting or illumination. They do, however, depend on the image content and quality measures. Finally, objective results require subjective verification, to see how realistic they are, since human perception is expected to be different than analytical data.

## Subjective (Experimental) Results

The surveys were conducted mainly on students -with one exception- in the age group of 17-22, as shown in Figure 44. Data entries from two subjects were eliminated since they had completed less than half of the survey, while missing data for surveys where more than half the entries were completed were compensated using mean values from the remaining surveys.

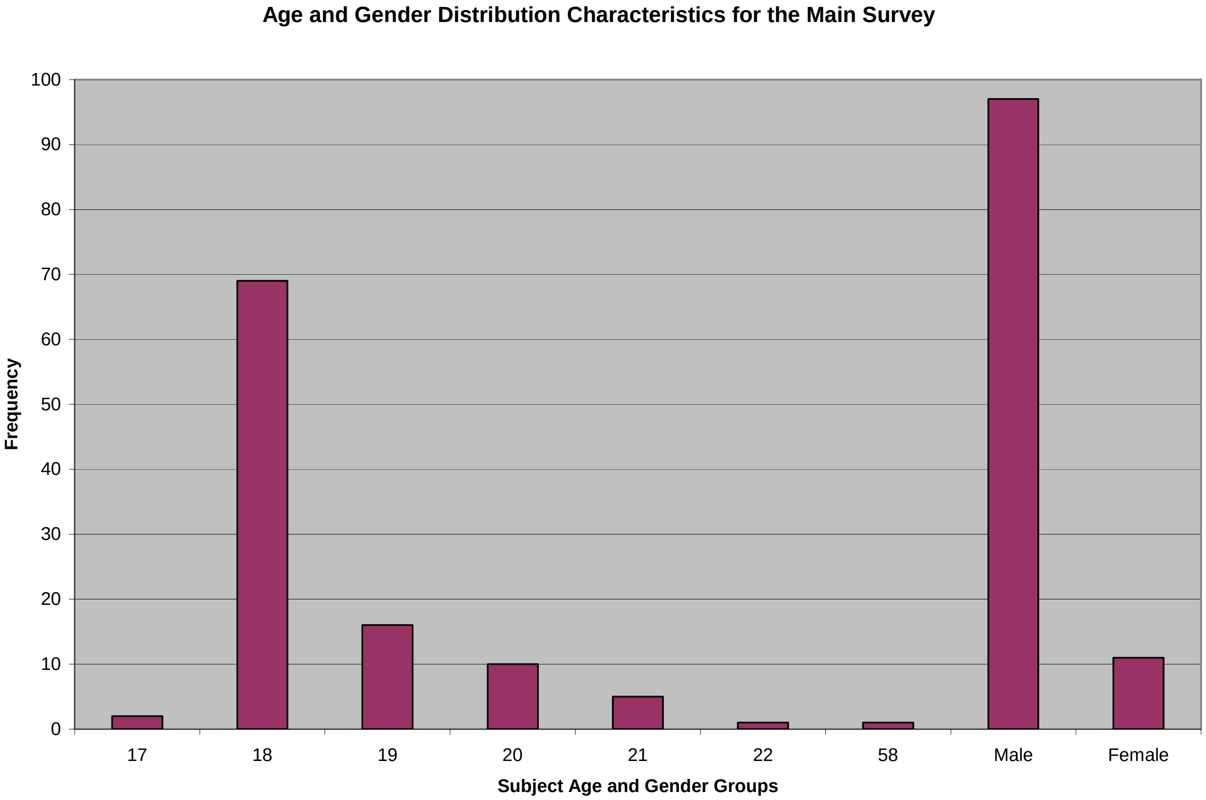


**Figure 44 Subjective characteristics of age and gender frequency distributions for participants in the main survey; note, results are very similar to those of the "pilot study."**

The subjective results are divided into normalized detail and distortion measures for images shown in the pilot and the subsequent main surveys. An overview of the subjective tabulated data and plots is shown in Table 3. The displayed results are based on mean values of subject responses, each accompanied by a measure of the standard error. The standard error (SE) is evaluated as the standard deviation divided by the square root of the subject size [6], which is

23 for the pilot study and 115 for the main survey. As the SE values are always positive, they are plotted above the detail and distortion bars for the following subjective results.

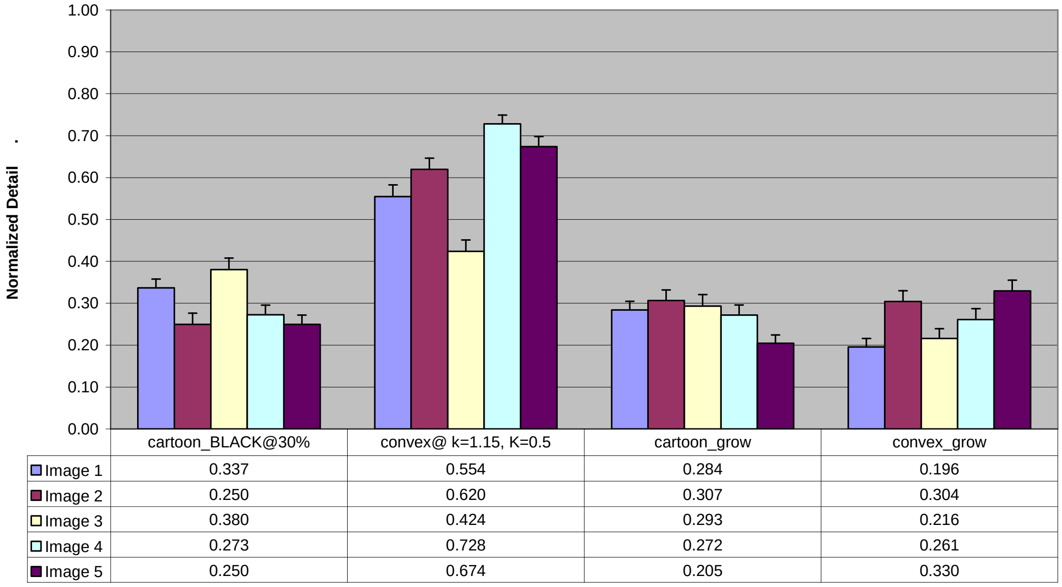
**Table 3 Table summarizing subjective data tables and plots of the various data from the two image sets generated experimentally (image set 1 for the pilot study, and image set 2 for the main survey); plots are classified by *independent* variable or criterion under consideration; the dependent variables are the recorded feedback detail and distortion values shown in the listed figures.**

| **Investigated Subjective Variables or Comparison** |
|---|
| • **Age and Gender Characteristics, Main Survey (Nov 8-15, 06):**<br>Frequency<br>Image Set 2: Figure 44 |
| • **Partial Cartoon versus Convex Methods, Pilot Study (Sep 07, 06),**<br>**Defaults (still mode), and Movie Mode (Growing Area):**<br>Detail and Distortion Characteristics (with standard error)<br>Image Set 1: Figure 45 |
| • **Partial Cartoon Method, Main Survey (Nov 8-15, 06)**<br>**Cartoon Factors (30%, 40%), Gray Shades (Black and White):**<br>Detail and Distortion Characteristics (with standard error)<br>Image Set 2: Figure 46 |
| • **Partial Cartoon versus "Peli" Methods, Main Survey (Nov 8-15, 06)**<br>**Gray Shades (Black and White):**<br>Detail and Distortion Characteristics (with standard error)<br>Image Set 2: Figure 47 |
| • **Partial Cartoon versus Convex Methods, Main Survey (Nov 8-15, 06)**<br>**Movie Mode (Moving Dot, Growing Area):**<br>Detail and Distortion Characteristics (with standard error)<br>Image Set 2: Figure 48 |
| • **Partial Cartoon versus "Peli" versus Convex Methods, Main Survey (Nov 8-15, 06)**<br>**Defaults:**<br>Detail and Distortion Characteristics (with standard error)<br>Image Set 2: Figure 49 |

The pilot study results shown in Figure 45 are only intended to provide a general evaluation to the quality of the survey itself, manifested through the subject feedback to the elements of the survey, as well as preliminary comparisons between the partial cartoon and convex methods in both the still and movie modes. As a result, the “Peli” method is not introduced until the main survey due to its relative resemblance to the partial cartoon method.

Figure 45 also indicates that the partial cartoon method has the lowest distortion value (0.40) for the text scene, almost half that recorded for the convex method (0.78). However, the partial cartoon method has lower detail measures than the convex method for all images. In addition, the distortion measures for only the text scene are significantly lower than the convex method (as low as a third for the house image). Finally, the cartoon and convex methods compete for the growing area movie mode among the different images as indicated by fluctuating detail and distortion values in Figure 45. The general conclusion from the pilot survey is that the convex method (peak detail of 0.73, and lowest distortion of 0.32) appears better appreciated by the subjects than the cartoon method (peak detail of only 0.38), with the exception of the text scene (distortion of only 0.40), as noted above.

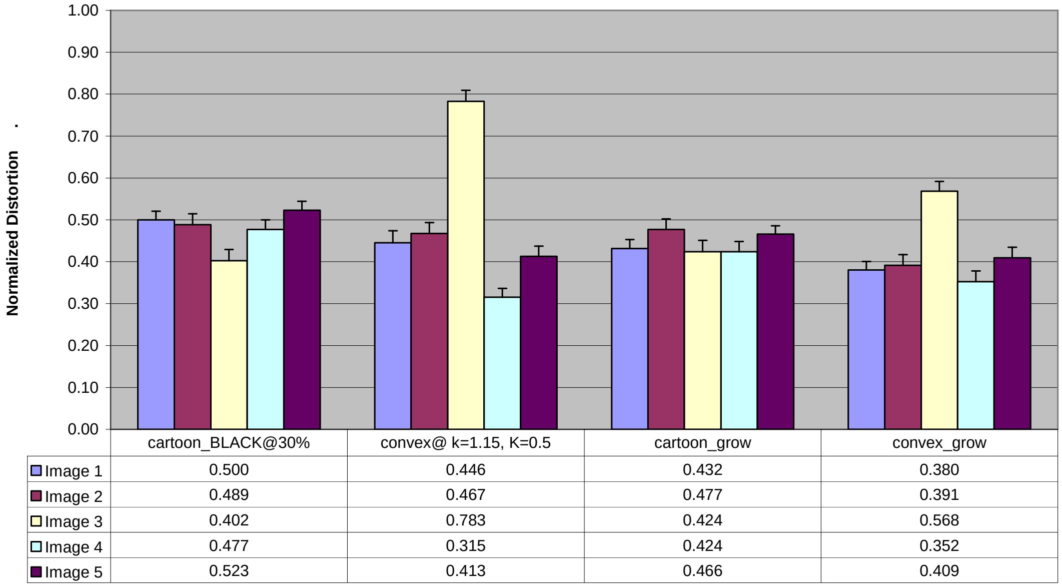
**Subjective Detail Comparison for Pilot Study Variables:**
**Cartoon Defaults, Convex Defaults, Moving Dot: Cartoon, Convex (Pilot Survey; Image Set 1)**



| | cartoon_BLACK@30% | convex@ k=1.15, K=0.5 | cartoon_grow | convex_grow |
|---|---|---|---|---|
| Image 1 | 0.337 | 0.554 | 0.284 | 0.196 |
| Image 2 | 0.250 | 0.620 | 0.307 | 0.304 |
| Image 3 | 0.380 | 0.424 | 0.293 | 0.216 |
| Image 4 | 0.273 | 0.728 | 0.272 | 0.261 |
| Image 5 | 0.250 | 0.674 | 0.205 | 0.330 |

**Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House**

**Subjective Distortion Comparison for Pilot Study Variables:**
**Cartoon Defaults, Convex Defaults, Moving Dot: Cartoon, Convex (Pilot Survey; Image Set 1)**



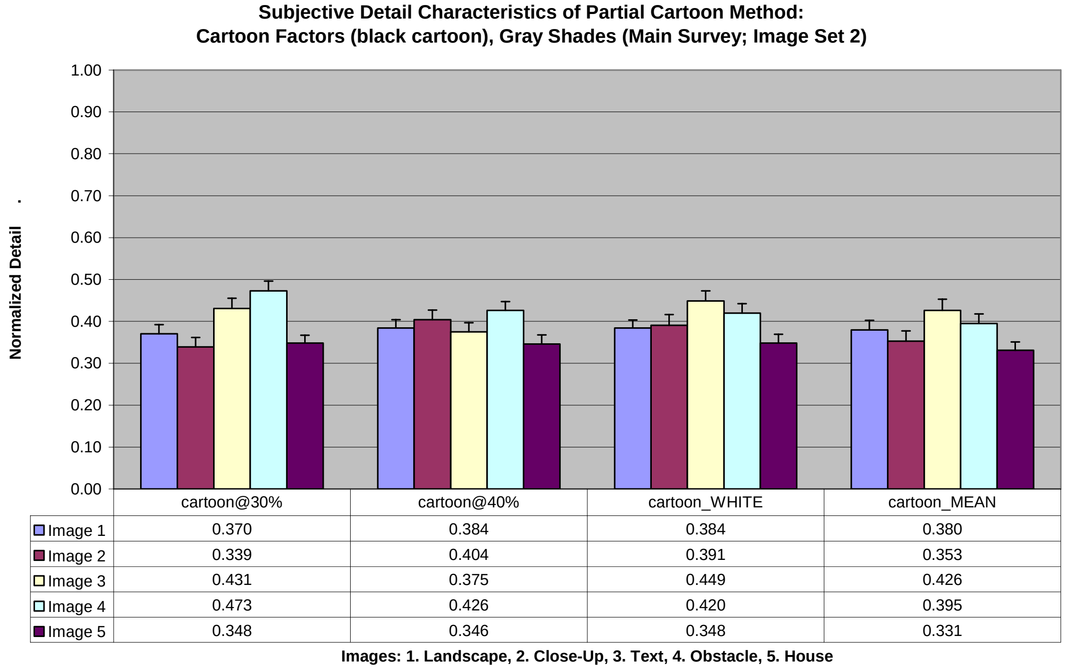
| | cartoon_BLACK@30% | convex@ k=1.15, K=0.5 | cartoon_grow | convex_grow |
|---|---|---|---|---|
| Image 1 | 0.500 | 0.446 | 0.432 | 0.380 |
| Image 2 | 0.489 | 0.467 | 0.477 | 0.391 |
| Image 3 | 0.402 | 0.783 | 0.424 | 0.568 |
| Image 4 | 0.477 | 0.315 | 0.424 | 0.352 |
| Image 5 | 0.523 | 0.413 | 0.466 | 0.409 |

**Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House**

**Figure 45 Summarized subjective characteristics (with standard error) of the pilot survey; top: data and plot for normalized detail values; bottom: data and plot for normalized distortion values.**

Regarding the main survey, the partial cartoon method is analyzed for different cartoon factors (30 and 40%) as well as different gray shades (black and white) in Figure 46. The results indicate that the black partial cartoon (at 30%) scores the highest detail (as high as 0.47 in the obstacle image), except in 1 image, while scoring the lowest distortion among most images. This is in exclusion of the mean cartoon due to its offering very little contrast from the surrounding image, as indicated from survey evaluation responses, making it hard to identify. In addition, the white partial cartoon has much higher relative distortion measures.

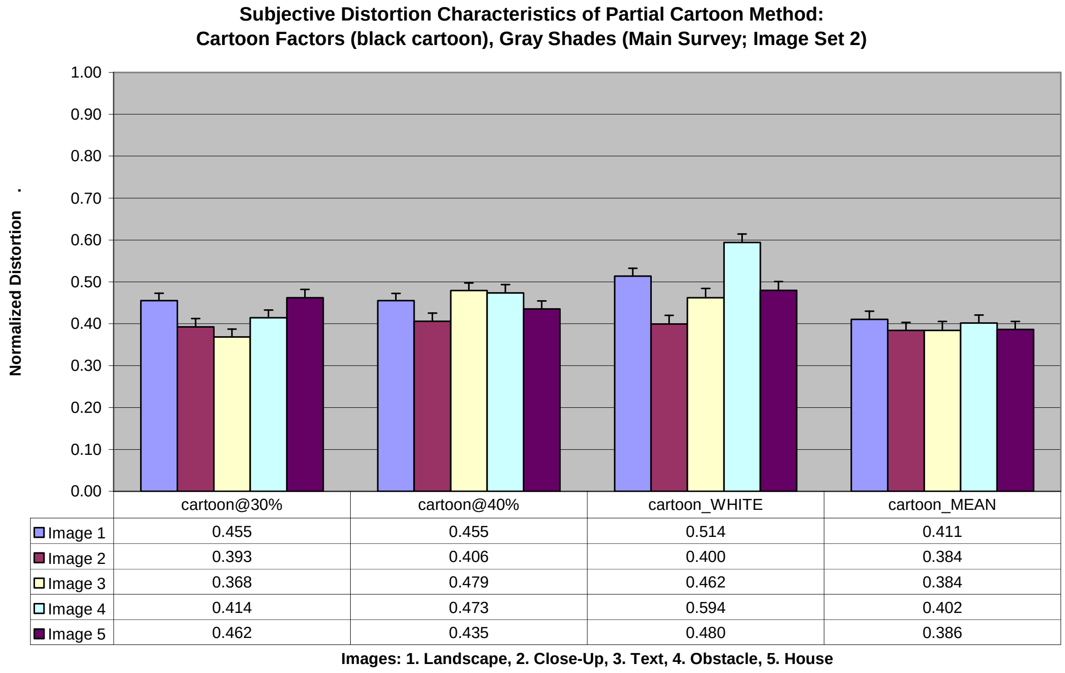
A comparison of the partial cartoon (at 30%) versus the “Peli” method for white and black gray shades is shown in Figure 47. Although the “Peli” method (especially for the white gray shade) has a relatively higher detail in some images, it exhibits much higher overall distortion (ranging 0.44 to 0.74). This shows that the partial cartoon method is generally favored over the “Peli” method, justified by the preserved center of the image. This is in exception to the house image, with respective partial cartoon and “Peli” respective detail values of 0.35 versus 0.59 and distortion values of a close margin of 0.46 versus 0.49. This is justified by there being low contrast between the black cartoon shade and the remainder of the image in dark environments.



Subjective Detail Characteristics of Partial Cartoon Method:
Cartoon Factors (black cartoon), Gray Shades (Main Survey; Image Set 2)

| | cartoon@30% | cartoon@40% | cartoon_WHITE | cartoon_MEAN |
|---|---|---|---|---|
| Image 1 | 0.370 | 0.384 | 0.384 | 0.380 |
| Image 2 | 0.339 | 0.404 | 0.391 | 0.353 |
| Image 3 | 0.431 | 0.375 | 0.449 | 0.426 |
| Image 4 | 0.473 | 0.426 | 0.420 | 0.395 |
| Image 5 | 0.348 | 0.346 | 0.348 | 0.331 |

Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House

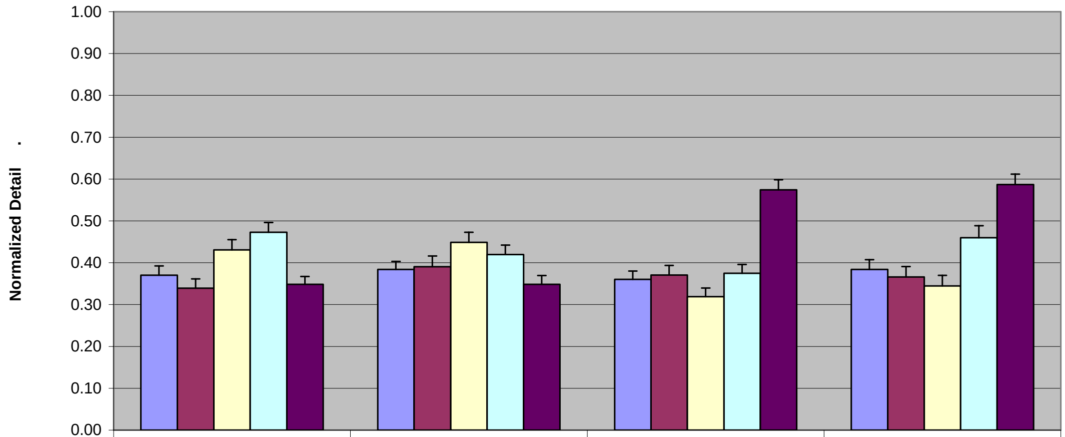


Subjective Distortion Characteristics of Partial Cartoon Method:
Cartoon Factors (black cartoon), Gray Shades (Main Survey; Image Set 2)

| | cartoon@30% | cartoon@40% | cartoon_WHITE | cartoon_MEAN |
|---|---|---|---|---|
| Image 1 | 0.455 | 0.455 | 0.514 | 0.411 |
| Image 2 | 0.393 | 0.406 | 0.400 | 0.384 |
| Image 3 | 0.368 | 0.479 | 0.462 | 0.384 |
| Image 4 | 0.414 | 0.473 | 0.594 | 0.402 |
| Image 5 | 0.462 | 0.435 | 0.480 | 0.386 |

Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House

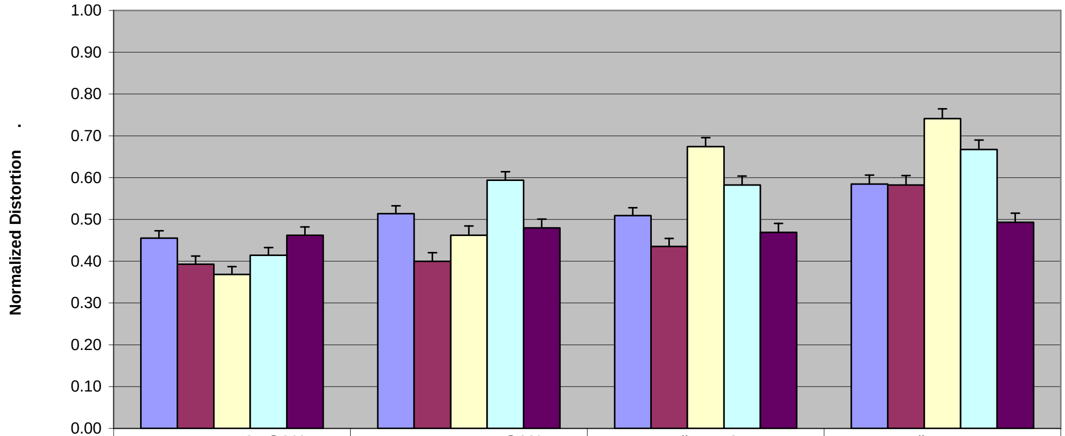
**Figure 46 Subjective characteristics (with standard error) of the partial cartoon method (image set 2); top: data and plot for normalized detail values; bottom: data and plot for normalized distortion values.**



Subjective Detail Comparison of Partial Cartoon and "Peli" Methods:
Gray Shades (Main Survey; Image Set 2)

| | cartoon_BLACK@30% | cartoon_WHITE@30% | Peli_BLACK | Peli_WHITE |
|---|---|---|---|---|
| 🟦 Image 1 | 0.370 | 0.384 | 0.360 | 0.384 |
| 🟥 Image 2 | 0.339 | 0.391 | 0.371 | 0.366 |
| 🟨 Image 3 | 0.431 | 0.449 | 0.319 | 0.345 |
| ⬜ Image 4 | 0.473 | 0.420 | 0.375 | 0.460 |
| 🟪 Image 5 | 0.348 | 0.348 | 0.574 | 0.587 |

Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House



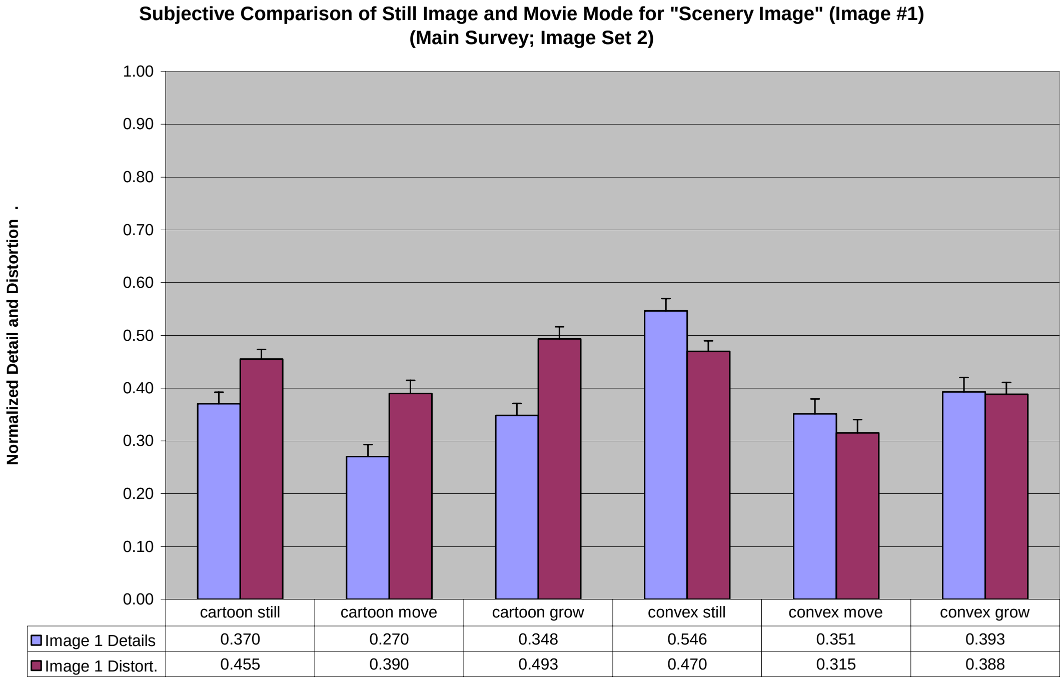
Subjective Distortion Comparison of Partial Cartoon and "Peli" Methods:
Gray Shades (Main Survey; Image Set 2)

| | cartoon_BLACK@30% | cartoon_WHITE@30% | Peli_BLACK | Peli_WHITE |
|---|---|---|---|---|
| 🟦 Image 1 | 0.455 | 0.514 | 0.509 | 0.585 |
| 🟥 Image 2 | 0.393 | 0.400 | 0.435 | 0.583 |
| 🟨 Image 3 | 0.368 | 0.462 | 0.674 | 0.741 |
| ⬜ Image 4 | 0.414 | 0.594 | 0.583 | 0.667 |
| 🟪 Image 5 | 0.462 | 0.480 | 0.469 | 0.493 |

Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House

**Figure 47 Subjective comparison (with standard error) of the partial cartoon versus "Peli" methods (image set 2); top: data and plot for normalized detail values; bottom: data and plot for normalized distortion values.**

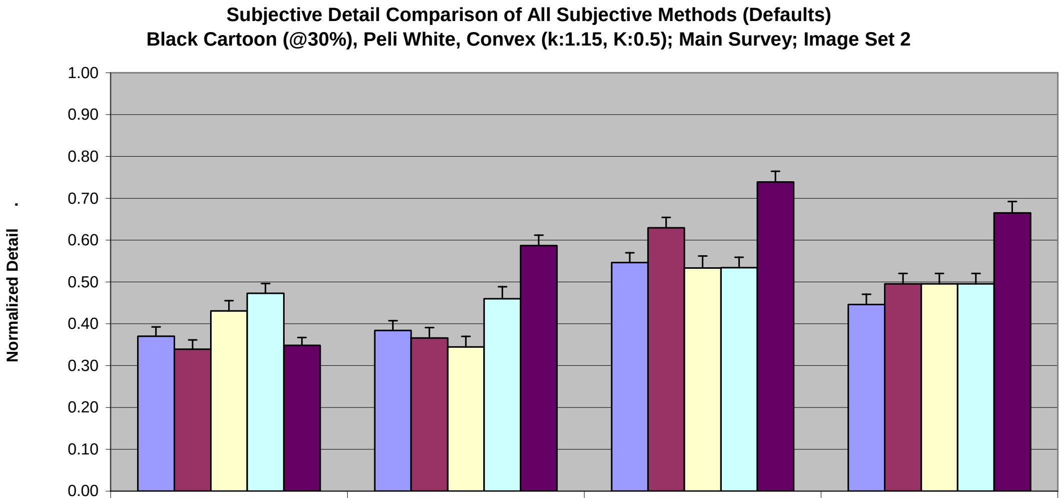
An assessment of the still versus movie modes (in both the moving dot and growing area versions), conducted solely for image number 1 is shown in Figure 48. This is applied to both the partial cartoon and convex methods. Although the partial cartoon method has a slightly lower recorded image distortion value (0.46) than its convex counterpart (0.47) in the still mode, the convex method is somewhat higher in both movie modes (maximum cartoon versus convex detail values of 0.35 and 0.39, and minimum distortion values of 0.39 and 0.32, respectively).



| | cartoon still | cartoon move | cartoon grow | convex still | convex move | convex grow |
|---|---|---|---|---|---|---|
| ■ Image 1 Details | 0.370 | 0.270 | 0.348 | 0.546 | 0.351 | 0.393 |
| ■ Image 1 Distort. | 0.455 | 0.390 | 0.493 | 0.470 | 0.315 | 0.388 |

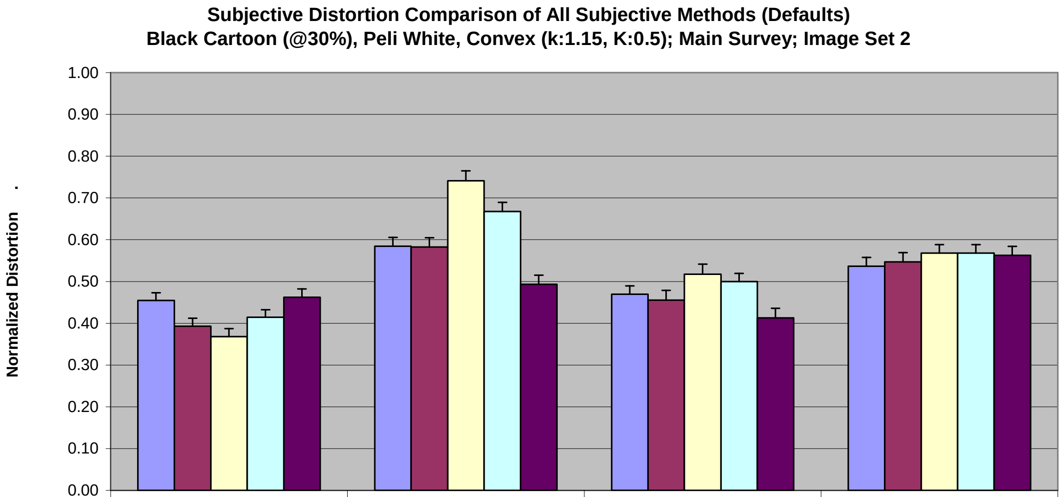
**Figure 48 Subjective characteristics (with standard error) of the movie modes for the partial cartoon and convex methods (image set 2), including normalized detail and normalized distortion values.**

Finally, a general comparison of the partial cartoon versus “Peli” versus convex methods is displayed in Figure 49. This indicates that the partial cartoon method (at black gray shade and 30% defaults) has the lowest distortion in most images (lowest: 0.39), except the dark scene. The convex technique, however, provide the highest detail measures for all images (highest: 0.74).



| | cart@30% | Peli_white | convex@ k=1.15, K=0.5 | convex@ k=1.15, K=1.0 |
|---|---|---|---|---|
| ■ Image 1 | 0.370 | 0.384 | 0.546 | 0.446 |
| ■ Image 2 | 0.339 | 0.366 | 0.629 | 0.496 |
| □ Image 3 | 0.431 | 0.345 | 0.533 | 0.495 |
| □ Image 4 | 0.473 | 0.460 | 0.534 | 0.495 |
| ■ Image 5 | 0.348 | 0.587 | 0.739 | 0.665 |

**Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House**



| | cart@30% | Peli_white | convex@ k=1.15, K=0.5 | convex@ k=1.15, K=1.0 |
|---|---|---|---|---|
| ■ Image 1 | 0.455 | 0.585 | 0.470 | 0.537 |
| ■ Image 2 | 0.393 | 0.583 | 0.455 | 0.547 |
| □ Image 3 | 0.368 | 0.741 | 0.518 | 0.568 |
| □ Image 4 | 0.414 | 0.667 | 0.500 | 0.568 |
| ■ Image 5 | 0.462 | 0.493 | 0.413 | 0.563 |

**Images: 1. Landscape, 2. Close-Up, 3. Text, 4. Obstacle, 5. House**

**Figure 49 Subjective comparison of the partial cartoon, "Peli," and convex methods (image set 2); top: data and plot for normalized detail values; bottom: data and plot for normalized distortion values.**

## Summary of Subjective Results

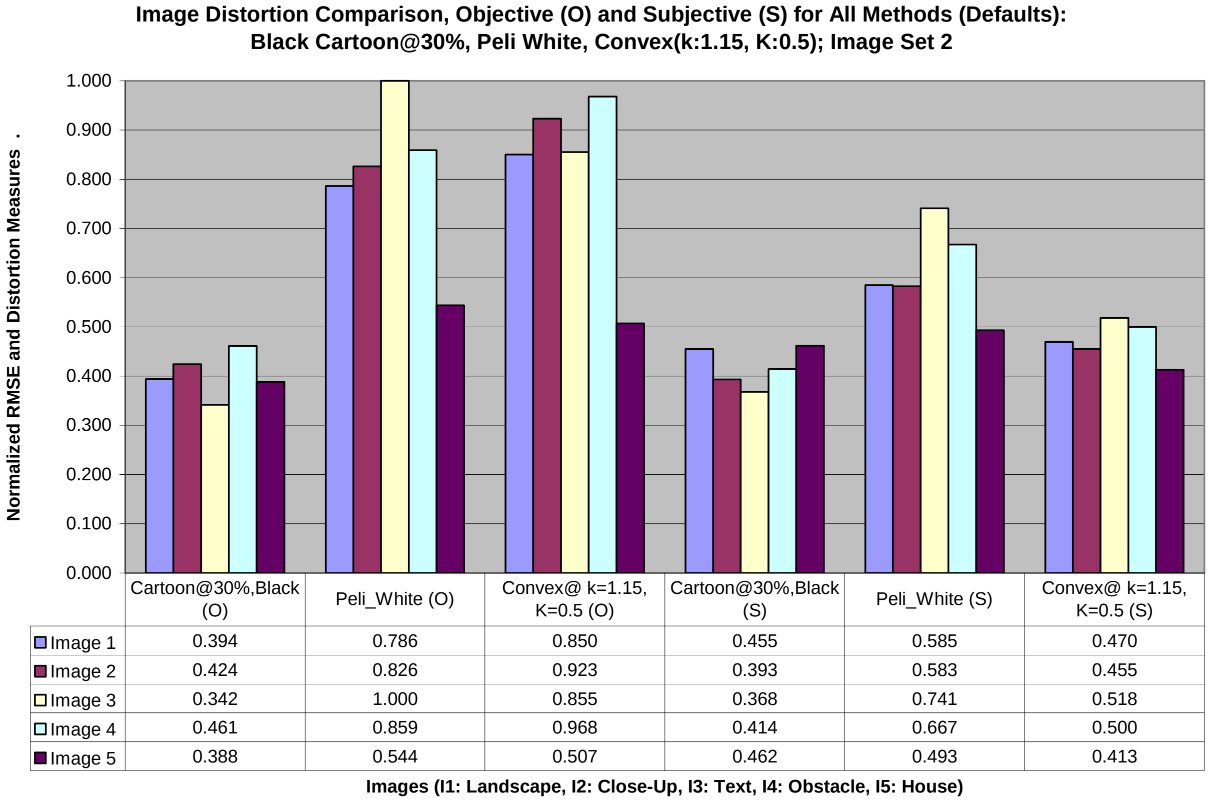
In summary, the partial cartoon method (at default criteria of a 30% superimposed black cartoon) has the lowest distortion measures in most images compared to all other subjective techniques (at a lowest of 0.37). The exception to this appears with the home and other images with an average luminance of less than 128. This is justified by the low contrast between the cartoon and the truncated image, consequently deeming it unsuitable for this particular application. Furthermore, the convex method is most superior for all image details measures, in both the still and movie modes (at a highest of 0.74). Therefore, there does not, in general, seem to be direct correlation between image detail and distortion data, and it remains to compare overall objective and subjective results, as provided in the following section.

**Table 4 Table summarizing cross-comparative objective and subjective data tables and plots generated both objectively and subjectively for all three methods (partial cartoon, "Peli" and convex methods) for image set 2; the dependent variables are the calculated image quality or distortion values shown in the listed figures.**

| **Investigated Objective and Subjective Comparisons** |
|---|
| • **Partial Cartoon versus "Peli" versus Convex Methods, Defaults**<br>Distortion/Error Characteristics;<br>Image Set 2: Figure 50 |
| • **Partial Cartoon versus "Peli" versus Convex Methods, Defaults**<br>Detail/Quality Characteristics;<br>Image Set 2: Figure 51 |

## Objective versus Subjective Results

A summary and comparison of the objective and subjective results (at established default criteria in each image compensation method) is demonstrated in Table 4. The RMSE and distortion measures in Figure 50 are used to indicate image error or discontinuities in each image, while UIQI and detail are regarded as general measures of extent of image quality in Figure 51.



| | Cartoon@30%,Black (O) | Peli_White (O) | Convex@ k=1.15, K=0.5 (O) | Cartoon@30%,Black (S) | Peli_White (S) | Convex@ k=1.15, K=0.5 (S) |
|---|---|---|---|---|---|---|
| □ Image 1 | 0.394 | 0.786 | 0.850 | 0.455 | 0.585 | 0.470 |
| ■ Image 2 | 0.424 | 0.826 | 0.923 | 0.393 | 0.583 | 0.455 |
| □ Image 3 | 0.342 | 1.000 | 0.855 | 0.368 | 0.741 | 0.518 |
| □ Image 4 | 0.461 | 0.859 | 0.968 | 0.414 | 0.667 | 0.500 |
| ■ Image 5 | 0.388 | 0.544 | 0.507 | 0.462 | 0.493 | 0.413 |

**Figure 50 Subjective and objective distortion/error comparison of the partial cartoon, “Peli,” and convex methods at default warping factors (image set 2); note how the partial cartoon method has the least overall error characteristics in both approaches.**

The results in Figure 50 and Figure 51 show that the partial cartoon method is superior in objective measures of RMSE and UIQI for all techniques, and for all images to the “Peli” and convex techniques. This is demonstrated at a grand UIQI high of 0.83, and grand low RMSE of 0.34 (for the text image). However, the figures show that these results are only consistent in part with the subjective analysis. In terms of distortion, the partial cartoon method indeed achieves the lowest subjective values in most images, reaching as low as 0.37 for the text image, which matches with the objective results shown above. The exception to this exists in the dark environment, where the convex method has lower values (0.41 versus 0.46 for the partial cartoon method).

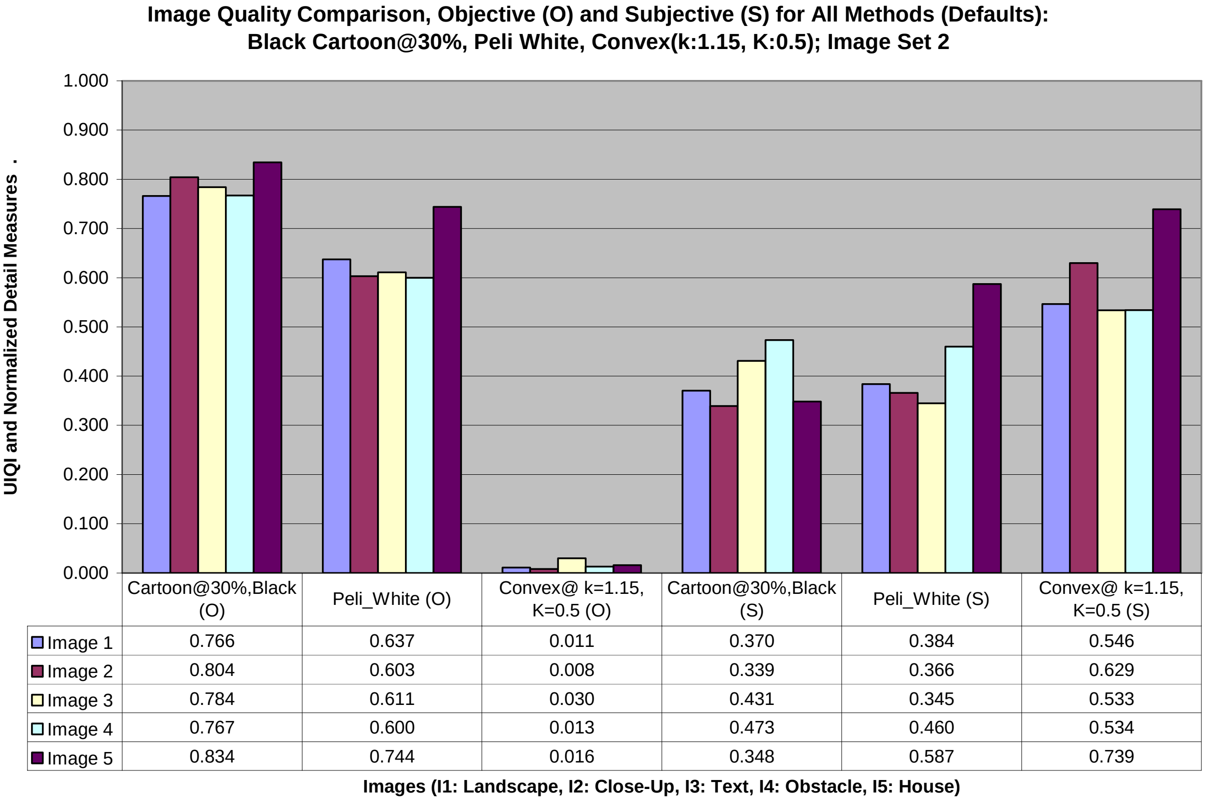


Image Quality Comparison, Objective (O) and Subjective (S) for All Methods (Defaults):
Black Cartoon@30%, Peli White, Convex(k:1.15, K:0.5); Image Set 2

| | Cartoon@30%,Black (O) | Peli_White (O) | Convex@ k=1.15, K=0.5 (O) | Cartoon@30%,Black (S) | Peli_White (S) | Convex@ k=1.15, K=0.5 (S) |
|---|---|---|---|---|---|---|
| ■ Image 1 | 0.766 | 0.637 | 0.011 | 0.370 | 0.384 | 0.546 |
| ■ Image 2 | 0.804 | 0.603 | 0.008 | 0.339 | 0.366 | 0.629 |
| □ Image 3 | 0.784 | 0.611 | 0.030 | 0.431 | 0.345 | 0.533 |
| □ Image 4 | 0.767 | 0.600 | 0.013 | 0.473 | 0.460 | 0.534 |
| ■ Image 5 | 0.834 | 0.744 | 0.016 | 0.348 | 0.587 | 0.739 |

Images (I1: Landscape, I2: Close-Up, I3: Text, I4: Obstacle, I5: House)

**Figure 51 Subjective and objective image detail/quality comparison of the partial cartoon, "Peli," and convex methods at default warping factors (image set 2); note how unlike the previous figure, the partial cartoon method is superior objectively, and to only some of the "Peli" subjective image scenes.**

On the other hand, image detail (subjective) measures are less than that of the convex method in all images. This is demonstrated, for instance, by an all high detail measure of 0.73 for the home image (convex method) versus a local high of 0.47 for partial cartoon method applied to the obstacle image. This is also true for the two subjectively evaluated movie (moving dot and growing area) modes, as presented in the last section.

These results dictate the mention of some preliminary conclusions. Image details and distortion measures are not correlated, due to inconsistency with their objective counterparts.

This is expected and is justified by the fact that a detailed image is not necessarily less distorted, and vice versa.

An alternative explanation is that the objective and subjective measures do not seem to be measuring the same corresponding criteria, which possibly implies that the RMSE and UIQI are physical, yet not “real” measures of image perception. In other words, although the cartoon method has a higher “objective” quality, the image detail does not imply (necessarily) a superior perception of quality for humans.

# Conclusions and Recommendations

Research in this dissertation can be summarized as one which involves the presentation of a new central image compensation technique, the “partial cartoon method,” motivated by both the quality photoreceptor distribution at the human visual center, as well as a previous cartoon superimposition or “Peli” technique. In addition, the presented method has been compared theoretically, objectively, and subjectively versus the documented “Peli” and convex methods, based on previous literature of the latter two methods.

The partial cartoon technique digitally involves introducing the immediate right and left of the surrounding (invisible) image to its respective location in the truncated image, at a custom “cartoon factor” (ranging from 0 to 50%), and a specified gray shade (white, black, gray, or image-mean) values. The optimum evaluated technique parameters (or method defaults) were generated as a black cartoon at 30% superimposition.

The presented method has been shown to be superior in all objective (analytical) measures (including the RMSE and UIQI well-documented image quality methods) to both the “Peli” and convex methods. However, in evaluating the technique subjectively for image detail and distortion characteristics, through a series of surveys conducted on a combined total of about 135 subjects (mostly students), the partial cartoon method was shown only to be superior in the majority of image distortion measures. This also comes in qualitative agreement to the rank order (among the three methods) determined by analytical results, with regards to least image error (RMSE) or distortion results for the partial cartoon method.

Subjectively, the technique was not superior in any relative image detail measures, justified as being due to the nature of the method which only provides a “hint” of the invisible surrounding, while not distorting the central portion of the image, rather than providing complete details. Furthermore, the technique was shown not to be subjectively suitable for an indoor, home, or low illumination environment. It is, however, best suited for daylight, outdoor conditions, especially for such scenes as signs, obstacles, or mixed layout environments. Potential applications of the partial cartoon method, therefore, may include areas such as low-vision, surveillance, robotics, and gaming. These results are summarized in Table 5.

**Table 5 Summary of research objectives and results, including image compensation methods presented in this dissertation, and various theoretical, analytical, and experimental variables and parameters; this is a follow-up to Table 1 presented in the *Introduction*.**

| Comparison | Convex Method | "Peli" Method | Partial Cartoon Method |
|---|---|---|---|
| ***Basis/Grounds for Technology*** | Equation related to photoreceptor distribution [2,47] | Not located in literature (targeted for low-vision [7,42]) | Motivated by clearing central image |
| ***Objective Results*** | Lowest UIQI | Second place, overall | Highest UIQI, Lowest RMSE (refer to Figure 41, Figure 42) |
| ***Subjective Results*** | Highest Detail (highest distortion) | Second place, overall | Lowest Distortion (refer to Figure 49) |
| ***Customizability/ Variability*** | Varying *k* (remapping factor, refer to Figure 34) | Varying Gray Shade (refer to Figure 29) | Varying Cartoon Level, Gray Shade (refer to Figure 23, Figure 26) |
| ***Applications*** | Not appropriate for low-vision applications [28] | Low-vision (in motion) [7,42] | Low-vision, surveillance, gaming (mixed-layout, daylight conditions) [9,10] |

Further testing is recommended to verify the subjective efficiency of the partial cartoon method in its present form, including a larger, more diverse subject group, also including different races, and possibly people with varying vision conditions, including the variation with and without vision correction. Electronic testing is also proposed to further increase the subject (population) size, and the speed and automation at which data can be collected and analyzed.

Rather than introducing a moving object to the still image, more dynamic features should be integrated to verify the effect of a moving base image on subjective responses.

A training of subjects, as mentioned in related literature is also proposed, where subjects are to be shown the images or movies warped with the partial cartoon method over separate periods of time, while noting the effect of time on appreciating adaptation to the irregularities and discontinuities of the method. This can also be used to further assess the variation of objective versus subjective results as subjects learn to adapt to the compensated images.

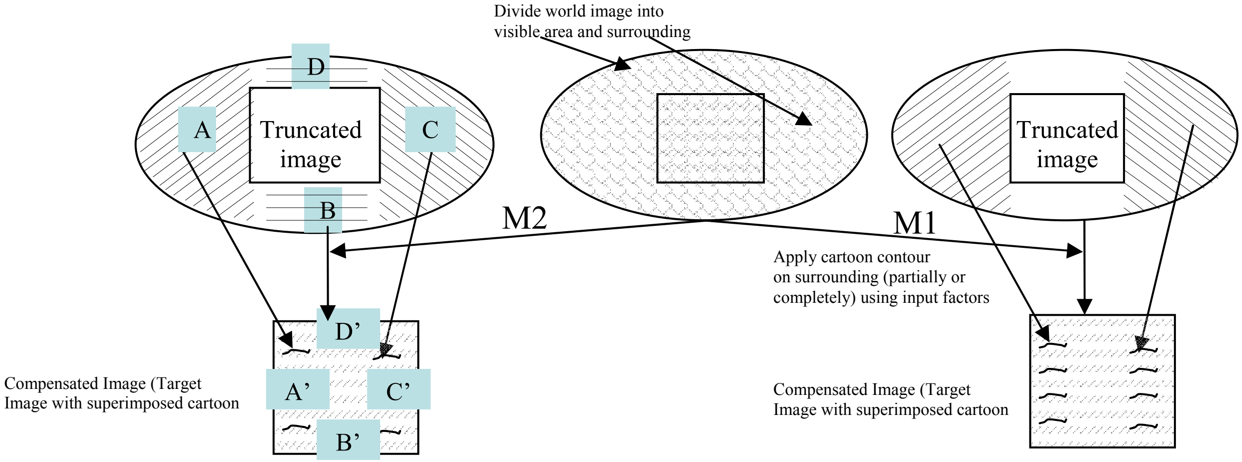

Divide world image into visible area and surrounding
D
A
Truncated image
C
B
M2
M1
Truncated image
Apply cartoon contour on surrounding (partially or completely) using input factors
D'
A'
C'
B'
Compensated Image (Target Image with superimposed cartoon
Compensated Image (Target Image with superimposed cartoon

**Figure 52 Two recommended methods for expanded research of the partial cartoon superimposition strategy; right: M1 strategy involving introducing the complete left and right (as opposed to the immediate left and right) invisible area to the truncated area; left: M2 strategy involving introducing the complete invisible image to the corresponding locations in the truncated image, while preserving as much of the central image in the truncated image undistorted.**

Finally, two theoretical variations of the demonstrated partial cartoon method are proposed for evaluation, as shown in Figure 52. The first involves the superimposition of the *entire* horizontal surrounding, rather than the *immediate* left and right, using the same partial cartoon parameters. This may serve in applications where there is higher significance to the vertical (upper or lower) field of view. Another potential, more sophisticated technique involves superimposing the entire invisible image (like the "Peli" method) in respective locations in the truncated image, while the center is still clear (i.e. without a cartoon superimposition). Relative comparison would, subsequently, provide potential alternatives to the currently used image compensation techniques, which as quoted from the literature [58], are in great need of further research to which this dissertation has aimed to achieve, at least partially.

# Bibliography

1. **Agrawal, M**. (2002). *Three Dimensional Reconstruction from Multiple Images*. Ph.D. Dissertation. University of Maryland College Park.

2. **Amerijckx, C, Legat, JD, and Trullemans, C**. (1999). "Design and Implementation of a Remapping Algorithm for Visual Prosthesis." *Proceedings of the 12th International Conference on Vision Interface*. 380-385

3. **Avcıbaş, I, Sankur, B, and Sayood, K**. (2002). "Statistical Evaluation of Image Quality Measures." *Journal of Electronic Imaging* **11**: 206–223.

4. **Beghdadi, A., and Pesquet-Popescu, B**. (2003). "A New Image Distortion Measure Based on Wavelet Decomposition." *Seventh International Symposium on Signal Processing and its Applications. Proceedings*. **1**: 485- 488.

5. **Black, HS**. (1953). *Modulation Theory*. Van Nostrand: Princeton.

6. **Bluman, AG**. (1998). *Elementary Statistics. A Step by Step Approach*. Third Ed. McGraw: Boston. (Chapter 7).

7. **Bowers, AR, Luo, G, Rensing, NM, and Peli, E**. (2004). "Evaluation of a Prototype Minified Augmented-View Device for Patients with Impaired Night Vision." *Ophthal Physiol Opt* **24**: 296–312.

8. **Ellenberger, SL, and Young, IT**. (2000). "Microscope Image Acquisition." Eds. Baldock, R, and Graham, J. *Image Processing and Analysis*. Oxford UP: Oxford. 1-36.

9. **El-Sherbeeny, AM, Odom, JV, and Smith, JE**. (2005). "A Non-Invasive Image Compensation Technique for Retinitis Pigmentosa Patients." Abstract in: *Developments in Vision Enhancement Technology and Their Evaluation*. West Virginia University Eye Institute.

10. **El-Sherbeeny, AM, Odom, JV, and Smith, JE**. (2006). "Spatial Density Distribution as a Basis for Image Compensation." *Journal of Modern Optics* **53**: 1241-1244.

11. **Evans, OD, and Kim, Y**. (1998). "Efficient Implementation of Image Warping on a Multimedia Processor." *Real-Time Imaging* **4**: 417-428.

12. **Fisher, TE, and Juday, RD**. (1988). "Programmable Video Remapper." *Digital and Optical Shape Representation and Pattern Recognition. SPIE Proceedings*. Ed. Juday, RD. **938**: 122-128.

13. **Fiume, E, Fournier, A, and Canale, V**. (1987). "Conformal Texture Mapping." *Eurographics Proceedings*. 53-64.

14. **Frederick, C, and Schwartz, EL**. (1990). "Conformal Image Warping." *IEEE Computer Graphics and Applications*. **10**: 54-61.

15. **Gao, P**. (1998). *Automatic Image Morphing Using Work Minimization*. Ph.D. Dissertation. Brigham Young University.

16. **Georgiev, T**. (2004). *Analytic Warping*. Patent Number 6,734,851. Accessed online at: http://www.uspto.gov/.

17. **Glasbey, CA, and Mardia, KV**. (1998). “A Review of Image-Warping Methods.” *Journal of Applied Statistics* **25**: 155-171.

18. **Greggain, L, Goel, J, and Beyer, R**. (1997). *Digital Image Warping System*. Patent Number 5,594,676. Accessed online at: http://www.uspto.gov/.

19. **Gu, X, and Yau, S**. (2003). “Global Conformal Surface Parameterization.” *Eurographics Symposium on Geometry Processing*. **43**: 127:137.

20. **Hamza, RM**. (2005). *Methods, Data, and Systems to Warp an Image*. Patent Number 6,934,422. Accessed online at: http://www.uspto.gov/.

21. **Heckbert, PS**. (1989). *Fundamentals of Texture Mapping and Image Warping*. Master’s Thesis. University of California, Berkeley, CA.

22. **Ho, JS, Loshin, DS, Barton, RS, and Juday, RD**. (1995). “Testing of Remapping for Reading Enhancement for Patients with Central Visual Field Losses.” *Visual Information Processing IV. SPIE Proceedings*. Eds. Huck, FO; and Juday, RD. **2488**: 417-424.

23. **Hofsetz, C, Ng, K, Chen, G, McGuinness, P, Max, N, and Liu, Y**. (2004). “Image-Based Rendering of Range Data with Estimated Depth Uncertainty.” *IEEE Computer Society* **24**: 34-41.

24. **Jackway, PT, and Deriche, M**. (1996). “Scale-Space Properties of the Multiscale Morphological Dilation-Erosion.” *IEEE Trans. Patterns Anal. Mach. Intell.* **18:** 38-51.

25. **Jin, M, Wang, Y, Gu, X, and Yau, S**. (2005). “Optimal Global Conformal Surface Parameterization for Visualization.” *Communications in Information and Systems* **4**: 117-134.

26. **Juday, RD, and Loshin, DS**. (1988). “Some Examples of Image Warping for Low Vision Prosthesis.” *Digital and Optical Shape Representation and Pattern Recognition. SPIE Proceedings*. Ed. Juday, RD. **938**:163-168.

27. **Juday, RD, Barton, RS, Johnson, CD, and Loshin, DS**. (1994). “Conformal and Other Image Warpings for Reading with Field Defect.” *Visual Information Processing III*. Eds. Huck, FO; and Juday, RD. **2239**: 92-102.

28. **Juday, RD, and Loshin, DS**. (1989). “Quasi-Conformal Remapping for Compensation of Human Visual Field Defects – Advances in Image Remapping for Human Field Defects.” *Optical Pattern Recognition. SPIE Proceedings.* **1053**: 124-130.

29. **Lee, SS**. (1994). *Digital Image Warping*. Master’s Thesis. University of Alberta, Canada.

30. **Lewis, A, Garcia, R, and Zhaoping, L**. (2003). “The Distribution of Visual Objects on the Retina: Connecting Eye Movements and Cone Distributions.” *Journal of Vision* **3**: 893-905.

31. **Lie, OY, and Toet, A**. (1998). “Applications of Digital Image Warping in Surveillance and Navigation.” *Displays* **19**: 133-139.

32. **Lohmeyer, MS, Burt, PJ, and van der Wal, GS**. (2000). *Quality Image Warper*. Patent Number 6,061,477. Accessed online at: http://www.uspto.gov/.

33. **Loshin, DS, and Juday, RD**. (1989). “The programmable Remapper: Clinical Applications for Patients with Field Defects.” *Optometry and Vision Science* **66**: 389-395.

34. **Loshin, DS, Smith, AT, and Juday, RD**. (1992). "Human Low Vision Image Warping - Channel Matching Considerations." *Visual Information Processing. SPIE Proceedings*. **1705**: 304-313.

35. **Loshin, DS, Wensveen, J, Juday, RD; and Barton, RS**. (1993). "Design of a Reading Test for Low Vision Image Warping." *Visual Information Processing II. SPIE Proceedings*. **1961**: 67-72.

36. **Luo, G, and Peli, E**. (2004). "Kinematics of Visual Search by Tunnel-Vision Patients with Augmented-Vision See-Through HMD." *SID 04 Digest, Digest of Technical Papers, Society for Information Display*. 1578-1581.

37. **Massof, RW, and Dagnelie, D**. (1996). "Toward an Artificial Eye." *IEEE Spectrum* **33**: 20-29.

38. **Masters, T**. (1993). *Signal and Image Processing with Neural Networks: A C++ Sourcebook*. Wiley: New York.

39. **Mathews, JH, and Howell, RW**. (1996). *Complex Analysis for Mathematics and Engineering*. Wm. C. Brown: Dubuque.

40. **Miller, JM, Demer, JL, Poukens, V, Pavlovski, DS, Nguyen, HN, and Rossi, EA**. (2003). "Extraocular Connective Tissue Architecture." *Journal of Vision* **3**: 240-251.

41. **Online Clinical Trials Education Courses**. National Cancer Institute. US National Institute of Health. Available online at: http://cme.cancer.gov/clinicaltrials/learning/humanparticipant-protections.asp?action=register.

42. **Peli, E**. (2003). *Wide-Band Image Enhancement*. Patent Number 6,611,618. Accessed online at: http://www.uspto.gov/.

43. **Peli, E**. June 4, 2005. *Personal Interview*.

44. **Popescu, V, Lastra, A, and Eyles, J**. (2004). *Methods and Apparatus for Rendering Images using 3D Warping Techniques*. Patent Number 6,756,993. Accessed online at: http://www.uspto.gov/.

45. **Rafael, CG, Richard, EW, and Steven, LE**. (2004). "Digital Image Processing using MATLAB." Prentice: Upper Saddle River. (Chapter 3).

46. **Raskar, R, van Baar, J, Beardsley, P, Willwacher, T, Rao, S, and Forlines, C**. (2003). "iLamps: Geometrically Aware and Self-Configuring Projectors." *ACM Transactions on Graphics (TOG). Proceedings*. **22**: 809–818.

47. **Rodieck, RW**. (1998). *The First Steps in Seeing*. Sinauer: Sunderland. (Chapter 9).

48. **Semmlow, JL**. (2004). "Biosignal and Biomedical Image Processing: MATLAB-Based Applications." Marcel Dekker: New York. (Chapter 11).

49. **Silverbrook, K, and Lapstun, P**. (2003). *Digital Image Warping System*. Patent Number 6,636,216. Accessed online at: http://www.uspto.gov/.

50. **Teo, P, and Heeger, D**. (1994). "Perceptual Image Distortion" *First IEEE International Conference on Image Processing* **2**: 982-986.

51. **Wang, L, Gu, X, Mueller, K, and Yau, S**. (2005). "Uniform Texture Synthesis and Texture Mapping Using Global Parameterization." *The Visual Computer* **21**: 801–810.

52. **Wang, Z, and Bovik, AC**. (2002). “A Universal Image Quality Index.” *IEEE Signal Processing Letters* **9**: 81-84.

53. **Wang, Z, and Simoncelli, EP**. (2005). “An Adaptive Linear System Framework for Image Distortion Analysis.” *Proc. 12th IEEE Intl. Conf. Image Processing*. **III**: 1160-1163.

54. **Wang, Z**. *Demo Images and Free Software for “A Universal Image Quality Index.”* Online at: http://www.cns.nyu.edu/~zwang/files/research/quality_index/demo.html. Accessed: Nov 18, 2006.

55. **Wiesel, TN, and Hubel, DH**. (1977). “Functional Architecture of Macaque Monkey Visual Cortex.” *Ferrier Lecture* **198**: 1-57.

56. **Wolberg, G**. (1990). *Digital Image Warping*. IEEE Computer Society P: Los Alamitos.

57. **Wolberg, G**. (1990). *Separable Image Warping: Implications and Techniques*. Ph.D. Dissertation. Columbia University.

58. **Wolffsohn, JS, and Peterson, RC**. (2003). “A Review of Current Knowledge on Electronic Vision Enhancement Systems for the Visually Impaired.” *Ophthal. Physiol. Opt*. **23**: 35-42.

59. **Wu, E, and Zheng, X**. (2003). “Composition of Novel Views through an Efficient Image Warping.” *The Visual Computer* **19**: 319–328.

60. **Yuille, AL, Ferraro, M, and Zhang, T**. (1998). “Image Warping for Shape Recovery and Recognition.” *Computer Vision and Image Understanding* **72**: 1-22.

# Appendix

## *Appendix 1: Listing of the Image Remapping Code*

1. This is the main program. It contains the main menu and calls all the other functions as shown in Figure 16. This file is called “`remap_main_Nov04_06.m`,” indicating the last date when the file was edited.

```
% prepared by Ahmed El-Sherbeeny, April 27, 2006
% updated Nov 04, 2006
% file to warp using central image magnification technique

%close all existing image screens, clear filename, and clear memory
clear, clc, close all
format short, format compact
select = 0; % for menu selection (to keep the program active in while loop)
m = 0; % this is a counter for the number of images viewed

%%%%%%%%%%%%%%%%%%%%%%%%% Main Menu %%%%%%%%%%%%%%%%%%%%%%%%%
while select ~= -1
    % Show Main Menu and Prompt User
    fprintf('\n\n\n\n\n')

disp('***********************************************************************
***')
    disp('Welcome to Sherbo''s Central Image Magnification and Optimization
Software')

disp('***********************************************************************
***')
    select = menu('Please choose from one of the following options: ', ...
        '1. Choose A New Image', ...
        '2. Fish-Eye or Convex Warping', ...
        '3. Cartoon Warping: variable percentage superimposition', ...
        '4. Cartoon Warping: variable gray shade', ...
        '5. Cartoon Warping: complete (Peli) superimposition', ...
        '6. Moving/Growing Dots', ...
        '7. Display Warped Images (Do Choices 1-5 first)', ...
        '8. Compare Image Quality Results (RMSE and UIQI)', ...
        'Quit');  % case 9
    clc

    % Match selection to one of available options
    switch select
        case 1,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % I. Retrieve, save, and determine image properties
            close all
            filename = '';
            % get file from pc (uses c drive as default location) and loads
image file
```

```
            % both existing filename and pathname are stored
            while (isempty(filename))
                [filename, pathname] = uigetfile('C:\*.bmp', 'Load Image
file');
            end

            % increment number of selected images
            m = m + 1;

            % both existing filename and pathname are now stored
            filename = fullfile(pathname,filename);

            % store image as I1
            I1 = imread(filename);

            % store image info (e.g. width and height)
            info = imfinfo(filename);

            % determine width, height, and origin
            width = info.Width;
            height = info.Height;

            % determine the location of the origin (center of the image)
            orig_h = fix(height/2);
            orig_w = fix(width/2);
            clc

        case 2,
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % II. Convex (Fish-Eye) Technique

            close all

            %%%%% assign each quadrant of the original image into individual
images
            Quad1 = I1(1: orig_h, orig_w: width); %quadrant 1 (top right; go
ccw)
            Quad2 = I1(1: orig_h, 1: orig_w);     %quadrant 2 (top left)
            Quad3 = I1(orig_h: height, 1: orig_w);%quadrant 3 (bottom left)
            Quad4 = I1(orig_h: height, orig_w: width); %quadrant 4 (bottom
right)
            % note, it's only neccessary that rows match for Q1 & Q2; Q3 & Q4
            % also, columns must match for Q2 & Q3; Q1 & Q4

            %%%%% input magnification parameters
            convex_parameters = menu('Convex Warping Parameters: ', ...
                '1. remapping = 1.15, proportionality = 0.4', ... % k = 1.15,
K = 0.4 (fix k)
                '2. remapping = 1.15, proportionality = 0.7', ... % k = 1.15,
K = 0.7 (fix k)
                '3. remapping = 1.15, proportionality = 1.0', ... % k = 1.15,
K = 1.0 (fix k)
                '4. remapping = 1.10, proportionality = 0.5', ... % k = 1.10,
K = 0.5 (fix K)
```

```
                '5. remapping = 1.15, proportionality = 0.5', ... % k = 1.15,
K = 0.5 (fix K)
                '6. remapping = 1.20, proportionality = 0.5', ... % k = 1.20,
K = 0.5 (fix K)
                'ALL (for image quality calculations)');
            switch convex_parameters
                case 1, % k = 1.15, K = 0.4 (fix k at default value 1.15 -1)
                    k = 1.15; % set remapping factor to 1.15
                    K = 0.4; % set proportionality factor to 0.4
                case 2, % k = 1.15, K = 0.7 (fix k-2)
                    k = 1.15; % set remapping factor to 1.15
                    K = 0.7; % set proportionality factor to 0.7
                case 3, % k = 1.15, K = 1.0 (fix k-3)
                    k = 1.15; % set remapping factor to 1.15
                    K = 1.0; % set proportionality factor to 1.0
                case 4, % k = 1.10, K = 0.5 (fix K at default value of 0.5 -
1)
                    k = 1.10; % set remapping factor to 1.10
                    K = 0.5; % set proportionality factor to 0.5
                case 5, % k = 1.15, K = 0.5 (fix K-2)
                    k = 1.15; % set remapping factor to 1.15
                    K = 0.5; % set proportionality factor to 0.5
                case 6, % k = 1.20, K = 0.5 (fix K-3)
                    k = 1.20; % set remapping factor to 1.20
                    K = 0.5; % set proportionality factor to 0.5
            end

            %%%%% Apply convex technique using given warping parameters;
recalls functions: convex() and Amerijckx()
            if convex_parameters ~= 7 % so long as user has not chosen to
calculate all at once
                I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
            end

            %%%%% Determine the image quality parameters (RMSE and UIQI) for
selected convex image vs. original (undeformed) image
            % first recall the original image, resize it to convex size, and
scale values between 0-255
            I1_600_800 = imresize(I1, [600 800], 'bicubic'); % adopted resize
method, matching convex method;

            % setup the original (reference) image pixels: convert to scale
(out of 1) then expand to 0-255 range
            I1_orig_double = 255.0 * im2double(I1_600_800);

            % determine the Root-Mean-Square Error (RMSE) and Universal Image
Quality Index (UIQI) for each convex image vs. original image
            switch convex_parameters % go back and match parameters to
created image
                case 1, % k = 1.15, K = 0.4 (fix k at default value 1.15 -1)
                    % first, rename the convex image according to its
parameters
                    I_convex_115_04 = I_warp_stand_bicubic_sharp;
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_04(m) = RMSE(I_convex_115_04,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_04(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_04); %recall function img_qi.m

                case 2, % k = 1.15, K = 0.7 (fix k-2)
                    % first, rename the convex image according to its
parameters
                    I_convex_115_07 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_07(m) = RMSE(I_convex_115_07,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_07(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_07); %recall function img_qi.m

                case 3, % k = 1.15, K = 1.0 (fix k-3)
                    % first, rename the convex image according to its
parameters
                    I_convex_115_10 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_10(m) = RMSE(I_convex_115_10,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_10(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_10); %recall function img_qi.m

                case 4, % k = 1.10, K = 0.5 (fix K at default value of 0.5 -
1)
                    % first, rename the convex image according to its
parameters
                    I_convex_110_05 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_110_05(m) = RMSE(I_convex_110_05,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_110_05(m), quality_map] =
img_qi(I1_orig_double, I_convex_110_05); %recall function img_qi.m
```

```
                case 5, % k = 1.15, K = 0.5 (fix K-2)
                    % first, rename the convex image according to its
parameters
                    I_convex_115_05 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_05(m) = RMSE(I_convex_115_05,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_05(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_05); %recall function img_qi.m

                case 6, % k = 1.20, K = 0.5 (fix K-3)
                    % first, rename the convex image according to its
parameters
                    I_convex_120_05 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_120_05(m) = RMSE(I_convex_120_05,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_120_05(m), quality_map] =
img_qi(I1_orig_double, I_convex_120_05); %recall function img_qi.m

                case 7, % all cases of k and K above (1-6) combined for
calculation purposes
                    %%%%% 1. k = 1.15, K = 0.4 (fix k at default value 1.15 -
1)
                    % set convex parameters
                    k = 1.15; % set remapping factor to 1.15
                    K = 0.4; % set proportionality factor to 0.4
                    % Apply convex technique using these parameters;
                    I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
                    % rename the convex image according to its parameters
                    I_convex_115_04 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_04(m) = RMSE(I_convex_115_04,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_04(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_04); %recall function img_qi.m

                    %%%%% 2. k = 1.15, K = 0.7 (fix k-2)
                    % set convex parameters
                    k = 1.15; % set remapping factor to 1.15
                    K = 0.7; % set proportionality factor to 0.7
                    % Apply convex technique using these parameters;
                    I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
                    % rename the convex image according to its parameters
                    I_convex_115_07 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_07(m) = RMSE(I_convex_115_07,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_07(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_07); %recall function img_qi.m

                    %%%%% 3. k = 1.15, K = 1.0 (fix k-3)
                    % set convex parameters
                    k = 1.15; % set remapping factor to 1.15
                    K = 1.0; % set proportionality factor to 1.0
                    % Apply convex technique using these parameters;
                    I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
                    % rename the convex image according to its parameters
                    I_convex_115_10 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_10(m) = RMSE(I_convex_115_10,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_10(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_10); %recall function img_qi.m

                    %%%%% 4. k = 1.10, K = 0.5 (fix K at default value of 0.5
-1)
                    % set convex parameters
                    k = 1.10; % set remapping factor to 1.10
                    K = 0.5; % set proportionality factor to 0.5
                    % Apply convex technique using these parameters;
                    I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
                    % rename the convex image according to its parameters
                    I_convex_110_05 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_110_05(m) = RMSE(I_convex_110_05,
I1_orig_double);
```

```
                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_110_05(m), quality_map] =
img_qi(I1_orig_double, I_convex_110_05); %recall function img_qi.m

                    %%%%% 5. k = 1.15, K = 0.5 (fix K-2)
                    % set convex parameters
                    k = 1.15; % set remapping factor to 1.15
                    K = 0.5; % set proportionality factor to 0.5
                    % Apply convex technique using these parameters;
                    I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
                    % rename the convex image according to its parameters
                    I_convex_115_05 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_115_05(m) = RMSE(I_convex_115_05,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_115_05(m), quality_map] =
img_qi(I1_orig_double, I_convex_115_05); %recall function img_qi.m

                    %%%%% 6. k = 1.20, K = 0.5 (fix K-3)
                    % set convex parameters
                    k = 1.20; % set remapping factor to 1.20
                    K = 0.5; % set proportionality factor to 0.5
                    % Apply convex technique using these parameters;
                    I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2,
Quad3, Quad4);
                    % rename the convex image according to its parameters
                    I_convex_120_05 = I_warp_stand_bicubic_sharp;

                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_convex_120_05(m) = RMSE(I_convex_120_05,
I1_orig_double);

                    % Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_convex_120_05(m), quality_map] =
img_qi(I1_orig_double, I_convex_120_05); %recall function img_qi.m
            end

            %%%%% Show Convex Image and Prompt User to Select Area of Concern
            % create a copy of the original image
            if convex_parameters ~= 7
                I1_copy = I1;
                % Prompt user to highlight portion of image (with mouse)
required for magnification
                imshow(uint8(I1)), title('Original (Undeformed) Image'); %
first show the original image and prompt selection
```

```
                area_orig_select = menu('What would you like to do?','1.
Select Area of Concern','2. Continue');
                close % close the current image
                clc
                if area_orig_select == 1
                    area = imcrop(I1_copy);
                    close % close the current figure
                    % find area of selected region
                    [area_h area_w] = size(area);
                    % find selected area ratio to total initial area
                    area_ratio_before = (area_h * area_w) / (width * height);
                end

                % Prompt user to find corresponding selected area in warped
image
                imshow(uint8(I_warp_stand_bicubic_sharp)), title('Convex
Image'); % now show the warped image and prompt selection
                area_warp_select = menu('What would you like to do?','1.
Select Same Area After Warping','2. Continue');
                close % close the current image
                if area_warp_select == 1
                    I_warp_copy = uint8(I_warp_stand_bicubic_sharp);
                    area_warp = imcrop(uint8(I_warp_copy));
                    close
                    % find warped selected area
                    [warp_h warp_w] = size(area_warp);
                    % find selected area ratio to total final (standardized)
area
                    area_ratio_after = (warp_h * warp_w) / (600 * 800);
                    % calculate relative increase of selected region size
w.r.t. total area
                    percent_increase = abs(area_ratio_before -
area_ratio_after) / area_ratio_before * 100;
                    % print %ge increase in size

fprintf('\n\n****************************************************************
**********\n')
                    fprintf('relative size of selected region w.r.t. to
visible image increased by: %.0f %%\n', percent_increase);

fprintf('********************************************************************
******\n')
                end
            end

        case 3,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % This section is designed to implement the cartoon (partial)
            % superimposition with variable superimposition percentages
            % the horizontal (1-D) surrounding here has been selected to
            % the left and right of the truncated image
            close all

            %%%%% Create Truncated Image
            % truncated image consists of the original 600*800 central image
portion
            I_trunc = I1(orig_h - 299: orig_h + 300, orig_w - 399: orig_w +
400);

            %%%% Determine first the size of the desired superimposed cartoon
            cartoon_choice = menu('Please choose desired superimposed
relative cartoon size', ...
                '25%', '30%', '35%', '40%', '45%', '50%', 'ALL (for image
quality calculations)');
            switch cartoon_choice
                case 1 % 25% cartoon superimposition
                    cartoon_size = 0.25;
                    %%%%% recall function cartoon (input cartoon size,
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_25 = I_trunc_Peli;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli), title('Cartoon Image: at
25% superimposition')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_25_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_25(m) = RMSE(I1_cartoon_25_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_25(m), quality_map] = img_qi(I1_double,
I1_cartoon_25_double);

                case 2 % 30% cartoon superimposition
                    cartoon_size = 0.3;
                    %%%%% recall function cartoon (input cartoon size,
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_30 = I_trunc_Peli;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli), title('Cartoon Image: at
30% superimposition')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_30_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_30(m) = RMSE(I1_cartoon_30_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_30(m), quality_map] = img_qi(I1_double,
I1_cartoon_30_double);

                case 3 % 35% cartoon superimposition
                    cartoon_size = 0.35;
                    %%%%% recall function cartoon (input cartoon size,
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_35 = I_trunc_Peli;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli), title('Cartoon Image: at
35% superimposition')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_35_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_35(m) = RMSE(I1_cartoon_35_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_35(m), quality_map] = img_qi(I1_double,
I1_cartoon_35_double);

                case 4 % 40% cartoon superimposition
                    cartoon_size = 0.4;
                    %%%%% recall function cartoon
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_40 = I_trunc_Peli;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli), title('Cartoon Image: at
40% superimposition')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_40_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
```

```
                    RMSE_cartoon_40(m) = RMSE(I1_cartoon_40_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_40(m), quality_map] = img_qi(I1_double,
I1_cartoon_40_double);

                case 5 % 45% cartoon superimposition
                    cartoon_size = 0.45;
                    %%%%% recall function cartoon (input cartoon size,
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_45 = I_trunc_Peli;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli), title('Cartoon Image: at
45% superimposition')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_45_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_45(m) = RMSE(I1_cartoon_45_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_45(m), quality_map] = img_qi(I1_double,
I1_cartoon_45_double);

                case 6 % 50% cartoon superimposition (i.e. the limiting, or
Peli, case)
                    cartoon_size = 0.5;
                    %%%%% recall function cartoon
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_50 = I_trunc_Peli;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli), title('Cartoon Image: at
50% superimposition')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_50_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
```

```
                    RMSE_cartoon_50(m) = RMSE(I1_cartoon_50_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_50(m), quality_map] = img_qi(I1_double,
I1_cartoon_50_double);

                case 7 % 25%, 30%, 35%, 40%, 45%, and 50% cartoon
superimposition calculations for RMSE and UIQI

                    % 25%
                    cartoon_size = 0.25;
                    %%%%% recall function cartoon (input cartoon size,
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_25 = I_trunc_Peli;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_25_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_25(m) = RMSE(I1_cartoon_25_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_25(m), quality_map] = img_qi(I1_double,
I1_cartoon_25_double);

                    % 30%
                    cartoon_size = 0.3;
                    %%%%% recall function cartoon (input cartoon size,
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_30 = I_trunc_Peli;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_30_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_30(m) = RMSE(I1_cartoon_30_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
```

```
                    [UIQI_cartoon_30(m), quality_map] = img_qi(I1_double, 
I1_cartoon_30_double);

                    % 35%
                    cartoon_size = 0.35;
                    %%%%% recall function cartoon (input cartoon size, 
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_35 = I_trunc_Peli;

                    %%%%% Determine the RMSE between the current cartoon 
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image: 
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_35_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original, 
returning value of RMSE)
                    RMSE_cartoon_35(m) = RMSE(I1_cartoon_35_double, 
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index) 
between same two images as above
                    [UIQI_cartoon_35(m), quality_map] = img_qi(I1_double, 
I1_cartoon_35_double);

                    % 40%
                    cartoon_size = 0.4;
                    %%%%% recall function cartoon
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_40 = I_trunc_Peli;

                    %%%%% Determine the RMSE between the current cartoon 
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image: 
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_40_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original, 
returning value of RMSE)
                    RMSE_cartoon_40(m) = RMSE(I1_cartoon_40_double, 
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index) 
between same two images as above
                    [UIQI_cartoon_40(m), quality_map] = img_qi(I1_double, 
I1_cartoon_40_double);

                    % 45%
                    cartoon_size = 0.45;
                    %%%%% recall function cartoon (input cartoon size, 
original and truncated images, returning cartoon image
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_45 = I_trunc_Peli;
                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_45_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_45(m) = RMSE(I1_cartoon_45_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_45(m), quality_map] = img_qi(I1_double,
I1_cartoon_45_double);

                    % 50%
                    cartoon_size = 0.5;
                    %%%%% recall function cartoon
                    I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);
                    I_cartoon_50 = I_trunc_Peli;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_50_double = 255.0 * im2double(I_trunc_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_50(m) = RMSE(I1_cartoon_50_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_50(m), quality_map] = img_qi(I1_double,
I1_cartoon_50_double);
            end

            %%%%% display the truncated image (for comparison)
            figure, imshow(I_trunc), title('Truncated image: no
superimposition')


        case 4,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % This section is designed to implement the cartoon
superimposition
            % with variable shades of gray, namely:
            % a) black (gray = 0), b) white (gray = 255), c) gray (gray =
128),
            % d) mean (gray = image average), and e) all the above
            %
            % assumptions: the horizontal (1-D) surrounding here has been
selected to
            % the left and right of the truncated image,
            % also the 30% superimposed cartoon has been selected as the
            % default size
            close all

            %%%%% Create Truncated Image
            % truncated image consists of the original 600*800 central image
portion
            I_trunc = I1(orig_h - 299: orig_h + 300, orig_w - 399: orig_w +
400);

            %%%% Determine first the shade of gray (between 0-255) of the
superimposed cartoon
            gray_choice = menu('Please choose desired shade of gray (0-255)
for the superimposed cartoon', ...
                '1. BLACK (0)', '2. WHITE (255)', '3. GRAY (128)', '4. MEAN
(calculated image average)', '5. ALL (for image quality calculations)');
            switch gray_choice
                case 1 % BLACK cartoon (gray_level = 0), note this is the
default choice (i.e. matching previous work)
                    gray_level = 0;
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_g0 = I_trunc_Peli_gray;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli_gray), title('Cartoon Image:
at 30% superimposition, BLACK Cartoon (gray = 0)')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_g0_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_g0(m) = RMSE(I1_cartoon_s30_g0_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_g0(m), quality_map] = img_qi(I1_double,
I1_cartoon_s30_g0_double);

                case 2 % WHITE cartoon (gray_level = 255): corresponds to
Peli's shade of gray for superimposed cartoon
                    gray_level = 255;
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
```

```
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_g255 = I_trunc_Peli_gray;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli_gray), title('Cartoon Image:
at 30% superimposition, WHITE Cartoon (gray = 255)')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_g255_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_g255(m) =
RMSE(I1_cartoon_s30_g255_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_g255(m), quality_map] =
img_qi(I1_double, I1_cartoon_s30_g255_double);

                case 3 % GRAY cartoon (gray_level = 128)
                    gray_level = 128;
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_g128 = I_trunc_Peli_gray;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli_gray), title('Cartoon Image:
at 30% superimposition, GRAY Cartoon (gray = 128)')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_g128_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_g128(m) =
RMSE(I1_cartoon_s30_g128_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_g128(m), quality_map] =
img_qi(I1_double, I1_cartoon_s30_g128_double);

                case 4 % MEAN cartoon (gray_level = image average)
```

```
                    gray_level = round(mean(mean(I1))); %note, this is the
mean of the complete, original image
                    gray_mean(m) = gray_level % store the value of the gray
mean level for the current image
                    % Also, this method has not been proposed in any
                    % previous image compensation research
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_gmean = I_trunc_Peli_gray;

                    %%%%% display the truncated image
                    figure, imshow(I_trunc_Peli_gray), title('Cartoon Image:
at 30% superimposition, MEAN Cartoon (gray = image average)')

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_gmean_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_gmean(m) =
RMSE(I1_cartoon_s30_gmean_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_gmean(m), quality_map] =
img_qi(I1_double, I1_cartoon_s30_gmean_double);

                case 5 % BLACK (0), WHITE (255), GRAY (128), and MEAN
(calculated image average),
                    % (at 30% cartoon superimposition) RMSE and UIQI
calculations

                    % 1. BLACK cartoon (gray_level = 0), note this is the
default choice (i.e. matching previous work)
                    gray_level = 0;
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_g0 = I_trunc_Peli_gray;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_g0_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
```

```
                    RMSE_cartoon_s30_g0(m) = RMSE(I1_cartoon_s30_g0_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_g0(m), quality_map] = img_qi(I1_double,
I1_cartoon_s30_g0_double);

                    % 2. WHITE cartoon (gray_level = 255): corresponds to
Peli's shade of gray for superimposed cartoon
                    gray_level = 255;
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_g255 = I_trunc_Peli_gray;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_g255_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_g255(m) =
RMSE(I1_cartoon_s30_g255_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_g255(m), quality_map] =
img_qi(I1_double, I1_cartoon_s30_g255_double);

                    % 3. GRAY cartoon (gray_level = 128)
                    gray_level = 128;
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_g128 = I_trunc_Peli_gray;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_g128_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_g128(m) =
RMSE(I1_cartoon_s30_g128_double, I1_double);
```

```
                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_g128(m), quality_map] =
img_qi(I1_double, I1_cartoon_s30_g128_double);

                    % 4. MEAN cartoon (gray_level = image average)
                    gray_level = round(mean(mean(I1))); %note, this is the
mean of the complete, original image
                    gray_mean(m) = gray_level; % store the value of the gray
mean level for the current image
                    % Also, this method has not been proposed in any
                    % previous image compensation research
                    % (in the encountered literature)
                    %%%%% print the mean gray level for the current image on
command window
                        fprintf('\n\n\n***** average gray level for current
image = %i *****\n\n\n', gray_mean(m))
                    %%%%% recall function cartoon_gray (input: gray level,
original and truncated images; returning cartoon image)
                    I_trunc_Peli_gray = cartoon_gray(gray_level, I1,
I_trunc);
                    I_cartoon_s30_gmean = I_trunc_Peli_gray;

                    %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                    % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
                    I1_cartoon_s30_gmean_double = 255.0 *
im2double(I_trunc_Peli_gray);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_cartoon_s30_gmean(m) =
RMSE(I1_cartoon_s30_gmean_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_cartoon_s30_gmean(m), quality_map] =
img_qi(I1_double, I1_cartoon_s30_gmean_double);
            end

            %%%%% display the truncated image (for comparison)
            figure, imshow(I_trunc), title('Truncated image: no
superimposition')

        case 5,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % This section is designed to implement the COMPLETE (PELI)
superimposition
            % with variable shades of gray, namely:
            % a) black (gray = 0), b) white (gray = 255), c) gray (gray =
128),
            % d) mean (gray = image average), and e) all the above
            %
            % The aim is to compare the Peli technique to the other
            % proposed cartoon methods, precisely with 1. the horizontal only
            % (variable percentage) superimposition (vs. Peli's complete
image superimposition) and 2. with the
            % variable gray shades (white, gray, black, and mean shades)
            % vs. Peli white only cartoon
            close all

            %%%%% Create Truncated Image
            % truncated image consists of the original 600*800 central image
portion
            I_trunc = I1(orig_h - 299: orig_h + 300, orig_w - 399: orig_w +
400);

            %%%% Determine first the shade of gray (between 0-255) of the
superimposed cartoon
            Peli_choice = menu('Please choose desired shade of gray (0-255)
for the Peli (complete superimposed) Cartoon ', ...
                '1. BLACK (0)', '2. WHITE (255)', '3. GRAY (128)', '4. MEAN
(calculated image average)', '5. ALL (for image quality calculations)');
            switch Peli_choice
                case 1 % BLACK Peli cartoon (gray_level = 0), note this is
the default choice (i.e. matching previous work)
                    gray_level = 0;
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_g0 = I_Peli;

                    %%%%% display the Peli image
                    figure, imshow(I_Peli), title('Peli (complete
superimposed Cartoon), BLACK Cartoon (gray = 0)')

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_g0_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_g0(m) = RMSE(I1_Peli_g0_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_g0(m), quality_map] = img_qi(I1_double,
I1_Peli_g0_double);

                case 2 % WHITE Peli cartoon (gray_level = 255): EXACT PELI
TECHNIQUE (including gray shade and size and nature of cartoon)
                    gray_level = 255;
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_g255 = I_Peli;
                    %%%%% display the Peli image
                    figure, imshow(I_Peli), title('Peli (complete
superimposed Cartoon), WHITE Cartoon (gray = 255)')

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_g255_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_g255(m) = RMSE(I1_Peli_g255_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_g255(m), quality_map] = img_qi(I1_double,
I1_Peli_g255_double);

                case 3 % GRAY Peli cartoon (gray_level = 128)
                    gray_level = 128;
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_g128 = I_Peli;

                    %%%%% display the Peli image
                    figure, imshow(I_Peli), title('Peli (complete
superimposed Cartoon), GRAY Cartoon (gray = 128)')

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_g128_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_g128(m) = RMSE(I1_Peli_g128_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_g128(m), quality_map] = img_qi(I1_double,
I1_Peli_g128_double);

                case 4 % MEAN Peli cartoon (gray_level = image average)
                    gray_level = round(mean(mean(I1))); %note, this is the
mean of the complete, original image
                    gray_mean(m) = gray_level % store the value of the gray
mean level for the current image
                    % Also, this method has not been proposed in any
                    % previous image compensation research
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_gmean = I_Peli;

                    %%%%% display the Peli image
                    figure, imshow(I_Peli), title('Peli (complete
superimposed Cartoon), MEAN Cartoon (gray = image average)')

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_gmean_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_gmean(m) = RMSE(I1_Peli_gmean_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_gmean(m), quality_map] = img_qi(I1_double,
I1_Peli_gmean_double);

                case 5 % BLACK (0), WHITE (255), GRAY (128), and MEAN Peli
cartoon (calculated image average),
                    % RMSE and UIQI calculations

                    % 1. BLACK Peli cartoon (gray_level = 0), note this is
the default choice (i.e. matching previous work)
                    gray_level = 0;
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_g0 = I_Peli;

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_g0_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_g0(m) = RMSE(I1_Peli_g0_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_g0(m), quality_map] = img_qi(I1_double,
I1_Peli_g0_double);

                    % 2. WHITE Peli cartoon (gray_level = 255): EXACT PELI
TECHNIQUE (including gray shade and size and nature of cartoon)
                    gray_level = 255;
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_g255 = I_Peli;

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_g255_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_g255(m) = RMSE(I1_Peli_g255_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_g255(m), quality_map] = img_qi(I1_double,
I1_Peli_g255_double);

                    % 3. GRAY Peli cartoon (gray_level = 128)
                    gray_level = 128;
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_g128 = I_Peli;

                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_g128_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_g128(m) = RMSE(I1_Peli_g128_double, I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_g128(m), quality_map] = img_qi(I1_double,
I1_Peli_g128_double);

                    % 4. MEAN Peli cartoon (gray_level = image average)
                    gray_level = round(mean(mean(I1))); %note, this is the
mean of the complete, original image
                    gray_mean(m) = gray_level; % store the value of the gray
mean level for the current image
                    % Also, this method has not been proposed in any
                    % previous image compensation research
                    % (in the encountered literature)
                    %%%%% print the mean gray level for the current image on
command window
                        fprintf('\n\n\n***** average gray level for current
image = %i *****\n\n\n', gray_mean(m))
                    %%%%% recall function cartoon_Peli (input: gray level,
original and truncated images; returning Peli image)
                    I_Peli = cartoon_Peli(gray_level, I1, I_trunc);
                    I_Peli_gmean = I_Peli;
                    %%%%% Determine the RMSE between the current Peli cartoon
image and original image (I_trunc)
                    % setup the Peli and truncated (reference) image: convert
to scale (out of 1) then expand to 0-255 range
                    I1_Peli_gmean_double = 255.0 * im2double(I_Peli);
                    I1_double = 255.0 * im2double(I_trunc);
                    % recall function RMSE (takes warped image and original,
returning value of RMSE)
                    RMSE_Peli_gmean(m) = RMSE(I1_Peli_gmean_double,
I1_double);

                    %%%%% Determine the UIQI (Universal Image Quality Index)
between same two images as above
                    [UIQI_Peli_gmean(m), quality_map] = img_qi(I1_double,
I1_Peli_gmean_double);

            end

            %%%%% display the truncated image (for comparison)
            figure, imshow(I_trunc), title('Truncated image: no
superimposition')


        case 6,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Dynamic mode
            % This section is concerned with showing images (natural and
            % warped) along with moving and growing dot schemes
            % it recalls the functions: moving_dot and growing_dot
            % Also, it is concerned with calculating RMSE and UIQI for
            % convex and cartoon warping techniques in the dynamic modes
            % However, this is possible only for the growing area

            close all
            % allow repeating the menu (e.g. for comparing various dynamic
modes)
            kick_out = 0; % variable testing if user wants to repeat

            while kick_out ~= -1
                dynamic = menu('Dynamic Mode. Please Select:', ...
                '1. Truncated Image: Moving Dot', ...
                '2. Truncated Image: Growing Dot', ...
                '3. Convex Image: Moving Dot', ...
                '4. Convex Image: Growing Dot', ...
                '5. Cartoon Image: Moving Dot', ...
                '6. Cartoon Image: Growing Dot', ...
                'None. Go back to the main menu');

                switch dynamic
                    case 1,
                        I_move_dot = moving_dot(I_trunc);
                    case 2,
                        I_grow_dot = growing_dot(I_trunc);
                        %%%%% find the RMSE and UIQI for I_trunc after
growing dot complete
                        %%%%% Determine the RMSE between the I_trunc growing
dot and original image (I_trunc)
                        % setup the truncated (reference) image: convert to
scale (out of 1) then expand to 0-255 range
                        I1_double = 255.0 * im2double(I_trunc);
                        I1_grow_double = 255.0 * im2double(I_grow_dot);
                        % recall function RMSE (takes warped image and
original, returning value of RMSE)
                        RMSE_I1_grow(m) = RMSE(I1_grow_double, I1_double);

                        %%%%% Determine the UIQI (Universal Image Quality
Index) between same two images as above
                        [UIQI_I1_grow(m), quality_map] = img_qi(I1_double,
I1_grow_double); %recall function img_qi.m

                    case 3,
                        I_move_dot = moving_dot(I_convex_115_05); % moving
dot on convex
                    case 4,
                        I_grow_dot_convex = growing_dot(I_convex_115_05); %
growing dot on convex
                        %%%%% find the RMSE and UIQI for I_convex image after
growing dot complete
                        %%%%% Determine the RMSE between the convex growing
dot and original image (I1)
                        % first recall the original image, resize it to
convex size, and scale values between 0-255
                        I1_600_800 = imresize(I1, [600 800], 'bicubic'); %
adopted resize method, matching convex method;
                        % setup the original (reference) image pixels:
convert to scale (out of 1) then expand to 0-255 range
                        I1_orig_double = 255.0 * im2double(I1_600_800);
                        % recall function RMSE (takes warped image and
original, returning value of RMSE)
                        RMSE_convex_grow(m) = RMSE(I_grow_dot_convex,
I1_orig_double);

                        %%%%% Determine the UIQI (Universal Image Quality
Index) between same two images as above
                        [UIQI_convex_grow(m), quality_map] =
img_qi(I1_orig_double, I_grow_dot_convex); %recall function img_qi.m

                    case 5,
                        I_move_dot = moving_dot(I_cartoon_30); % calls 30%
cartoon image
                    case 6,
                        I_grow_dot_cartoon = growing_dot(I_cartoon_30); %
calls 30% cartoon image
                        %%%%% find the RMSE and UIQI for I_cartoon_30 image
after growing dot complete
                        %%%%% Determine the RMSE between the current cartoon
image and original image (I_trunc)
                        % setup the cartoon and truncated (reference) image:
convert to scale (out of 1) then expand to 0-255 range
```

```
                        I1_double = 255.0 * im2double(I_trunc);
                        I1_grow_cartoon_double = 255.0 *
im2double(I_grow_dot_cartoon);
                        % recall function RMSE (takes warped image and
original, returning value of RMSE)
                        RMSE_cartoon_grow(m) = RMSE(I1_grow_cartoon_double,
I1_double);

                        %%%%% Determine the UIQI (Universal Image Quality
Index) between same two images as above
                        [UIQI_cartoon_grow(m), quality_map] =
img_qi(I1_double, I1_grow_cartoon_double);
                    case 7, % Go back to the main menu
                        kick_out = -1;
                end
            end


        case 7,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % III. DISPLAY IMAGES: original, and warped image(s) vs.
truncated image (for comparison)
            close all

            % allow displaying multiple images
            kick_out = 0; % variable testing if user wants to display more
images or not

            while kick_out ~= -1

                % prompt user with menu to determine desired image(s) to be
displayed
                show_image = menu('Which image would you like to display?',
...
                    '1.  Original (Undeformed) Image', ...
                    '2.  Truncated (Restricted FOV) Image', ...
                    '3.  Convex Image: k = 1.15, K = 0.4', ...
                    '4.  Convex Image: k = 1.15, K = 0.7', ...
                    '5.  Convex Image: k = 1.15, K = 1.0', ...
                    '6.  Convex Image: k = 1.10, K = 0.5', ...
                    '7.  Convex Image: k = 1.15, K = 0.5', ...
                    '8.  Convex Image: k = 1.20, K = 0.5', ...
                    '9.  Cartoon Image: at 25% superimposition', ...
                    '10. Cartoon Image: at 30% superimposition', ...
                    '11. Cartoon Image: at 35% superimposition', ...
                    '12. Cartoon Image: at 40% superimposition', ...
                    '13. Cartoon Image: at 45% superimposition', ...
                    '14. Cartoon Image: at 50% superimposition', ...
                    '15. Cartoon Image: BLACK Cartoon (gray = 0, @30%)', ...
                    '16. Cartoon Image: WHITE Cartoon (gray = 255, @30%)',
...
                    '17. Cartoon Image: GRAY Cartoon (gray = 128, @30%)', ...
                    '18. Cartoon Image: MEAN Cartoon (gray = average, @30%)',
...
                    '19. Peli Image: BLACK Cartoon (gray = 0)', ...
                    '20. Peli Image: WHITE Cartoon (gray = 255)', ...
                    '21. Peli Image: GRAY Cartoon (gray = 128)', ...
                    '22. Peli Image: MEAN Cartoon (gray = average)', ...
                    'None. Go back to the main menu');

                % show desired image (note, previously chosen images will
remain open)
                switch show_image
                    case 1, % original image
                        figure, imshow(uint8(I1)), title('Original
(Undeformed) Image')
                    case 2, % truncated image
                        figure, imshow(uint8(I_trunc)), title('Truncated
Image')
                    case 3, % convex image: k = 1.15, K = 0.4
                        figure, imshow(uint8(I_convex_115_04)), title('Convex
Image: k = 1.15, K = 0.4')
                    case 4, % convex image: k = 1.15, K = 0.7
                        figure, imshow(uint8(I_convex_115_07)), title('Convex
Image: k = 1.15, K = 0.7')
                    case 5, % convex image: k = 1.15, K = 1.0
                        figure, imshow(uint8(I_convex_115_10)), title('Convex
Image: k = 1.15, K = 1.0')
                    case 6, % convex image: k = 1.10, K = 0.5
                        figure, imshow(uint8(I_convex_110_05)), title('Convex
Image: k = 1.10, K = 0.5')
                    case 7, % convex image: k = 1.15, K = 0.5
                        figure, imshow(uint8(I_convex_115_05)), title('Convex
Image: k = 1.15, K = 0.5')
                    case 8, % convex image: k = 1.20, K = 0.5
                        figure, imshow(uint8(I_convex_120_05)), title('Convex
Image: k = 1.20, K = 0.5')
                    case 9, % Cartoon Image: at 25% superimposition
                        figure, imshow(I_cartoon_25), title('Cartoon Image:
at 25% superimposition')
                    case 10, % Cartoon Image: at 30% superimposition
                        figure, imshow(I_cartoon_30), title('Cartoon Image:
at 30% superimposition')
                    case 11, % Cartoon Image: at 35% superimposition
                        figure, imshow(I_cartoon_35), title('Cartoon Image:
at 35% superimposition')
                    case 12, % Cartoon Image: at 40% superimposition
                        figure, imshow(I_cartoon_40), title('Cartoon Image:
at 40% superimposition')
                    case 13, % Cartoon Image: at 45% superimposition
                        figure, imshow(I_cartoon_45), title('Cartoon Image:
at 45% superimposition')
                    case 14, % Cartoon Image: at 50% superimposition
                        figure, imshow(I_cartoon_50), title('Cartoon Image:
at 50% superimposition')
                    case 15, % Cartoon Image: BLACK Cartoon (gray = 0, @30%)
                        figure, imshow(I_cartoon_s30_g0), title('Cartoon
Image: BLACK Cartoon (gray = 0, @30%)')
                    case 16, % Cartoon Image: WHITE Cartoon (gray = 255,
@30%)
                        figure, imshow(I_cartoon_s30_g255), title('Cartoon
Image: WHITE Cartoon (gray = 255, @30%)')
```

```
                    case 17, % Cartoon Image: GRAY Cartoon (gray = 128, @30%)
                        figure, imshow(I_cartoon_s30_g128), title('Cartoon
Image: GRAY Cartoon (gray = 128, @30%)')
                    case 18, % Cartoon Image: MEAN Cartoon (gray = see
command window, @30%)
                        figure, imshow(I_cartoon_s30_gmean), title('Cartoon
Image: MEAN Cartoon (gray = see command window, @30%)')
                        %print the mean gray level for the current image on
command window
                        fprintf('\n\n\n***** average gray level for current
image = %i *****\n\n\n', gray_mean(m))
                    case 19, % Peli Image: BLACK Cartoon (gray = 0)
                        figure, imshow(I_Peli_g0), title('Peli Image: BLACK
Cartoon (gray = 0)')
                    case 20, % Peli Image: WHITE Cartoon (gray = 255)
                        figure, imshow(I_Peli_g255), title('Peli Image: WHITE
Cartoon (gray = 255)')
                    case 21, % Peli Image: GRAY Cartoon (gray = 128)
                        figure, imshow(I_Peli_g128), title('Peli Image: GRAY
Cartoon (gray = 128)')
                    case 22, % Peli Image: MEAN Cartoon (gray = see command
window)
                        figure, imshow(I_Peli_gmean), title('Peli Image: MEAN
Cartoon (gray = see command window)')
                        %print the mean gray level for the current image on
command window
                        fprintf('\n\n\n***** average gray level for current
image = %i *****\n\n\n', gray_mean(m))
                    case 23, % Go back to the main menu
                        kick_out = -1;
                end
            end

        case 8,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % IV. DISPLAY OBJECTIVE IMAGE QUALITY RESULTS for convex and
cartoon images:
            % 1. RMSE (Root-Mean-Square Error)
            % 2. UIQI (Universal Image Quality Index)
            % In addition, the gray mean level is displayed
            close all
            clc

disp('************************************************************************
***')
            disp('OBJECTIVE IMAGE QUALITY RESULTS for warped (convex and
cartoon) images')

disp('************************************************************************
***')
            %%%%% show RMSE values (columns) 1. for the cartoon (25-50%)
images,
            %%%%% 2. for the gray cartoon images:
            %%%%%   black (0), white (255), gray (128), and mean
{gray_mean(m)}
            %%%%% 3. for the Peli cartoons (same gray shades as above):
```

```
            %%%%%   black (0), white (255), gray (128), and mean
{gray_mean(m)}
            %%%%% 4. convex (differenet k's and K's) warped images, and
            %%%%% 5. select movie (grow) images

            disp(' ')
            disp(' ')
            disp('1. Root Mean Square Error (RMSE) values for the cartoon
(25-50%), gray cartoons (0, 255, gray, mean), Peli cartoons(0, 255, gray,
mean), convex warped images, and selected movie images')

disp('**********************************************************************
****************************************************************************
************************************')
            disp(' ')
            % 1. print RMSE values for cartoon images (with different
percentages)

fprintf('Image#\tcartoon@25%%\tcartoon@30%%\tcartoon@35%%\tcartoon@40%%\tcart
oon@45%%\tcartoon@50%%\n');
            table_RMSE_cartoon = [  (1:m)'      RMSE_cartoon_25'
RMSE_cartoon_30'    RMSE_cartoon_35'    ...
                RMSE_cartoon_40'    RMSE_cartoon_45'    RMSE_cartoon_50']
            % 2. print RMSE values for cartoon images (with different gray
shades)

fprintf('Image#\tcartoon@BLACK\tcartoon@WHITE\tcartoon@GRAY\tcartoon@MEAN\n')
;
            table_RMSE_cartoon_gray = [  (1:m)'      RMSE_cartoon_s30_g0'
RMSE_cartoon_s30_g255'    RMSE_cartoon_s30_g128'    RMSE_cartoon_s30_gmean']
            % 3. print RMSE values for Peli images (with different gray
shades)

fprintf('Image#\tPeli@BLACK\tPeli@WHITE\tPeli@GRAY\tPeli@MEAN\n');
            table_RMSE_Peli = [  (1:m)'      RMSE_Peli_g0'    RMSE_Peli_g255'
RMSE_Peli_g128'    RMSE_Peli_gmean']
            % 4. print RMSE values for convex images

fprintf('Image#\tconvex@k=1.15,K=.4\tconvex@k=1.15,K=.7\tconvex@k=1.15,K=1\tc
onvex@k=1.1,K=.5\tconvex@k=1.15,K=.5\tconvex@1.2,K=.5\n');
            table_RMSE_convex = [  (1:m)'      RMSE_convex_115_04'
RMSE_convex_115_07'    RMSE_convex_115_10'    RMSE_convex_110_05'    ...
                RMSE_convex_115_05'    RMSE_convex_120_05']
            % 5. print RMSE values for movie images (i.e. after movie is
complete)
            fprintf('Image#\ttrunc_grow\tcartoon_grow\tconvex_grow\n');
            table_RMSE_grow = [  (1:m)'      RMSE_I1_grow'
RMSE_cartoon_grow'    RMSE_convex_grow'    ]
            % find the global maximum RMSE (the largest RMSE value in all
images)
            max_RMSE1 = max(max(max(table_RMSE_cartoon, table_RMSE_convex)));
% find the maximum RMSE in cartoon images vs. convex images
            max_RMSE2 = max(max(table_RMSE_cartoon_gray)); % find maximum
RMSE in gray cartoon images
            max_RMSE3 = max(max(table_RMSE_Peli)); % find maximum RMSE in
Peli cartoon images
```

```
            max_RMSE4 = max(max(table_RMSE_grow)); %find the maximum grow
RMSE
            max_RMSE = max([max_RMSE1, max_RMSE2, max_RMSE3, max_RMSE3]); %
find the global maximum RMSE
            %% normalize all RMSE errors to be out of 1 (relative to the
global maximum RMSE calculated above)
            % first normalize RMSE errors for both convex and cartoon tables
(since they both have the same number of columns)
            for i = 1 : m   % i.e. across all the images
                for j = 2 : size(table_RMSE_cartoon,2)  % across all columns
except first (which is index number)
                    table_RMSE_cartoon_norm(i,j) = table_RMSE_cartoon(i,j) /
max_RMSE;  %normalize all cartoon RMSE errors
                    table_RMSE_convex_norm(i,j) = table_RMSE_convex(i,j) /
max_RMSE;  %normalize all convex RMSE errors
                end
            end
            % now, normalize RMSE errors for both cartoon_gray and Peli
tables (since they both have the same number of columns)
            for i = 1 : m   % i.e. across all the images
                for j = 2 : size(table_RMSE_cartoon_gray,2)  % across all
columns except first (which is an index number)
                    table_RMSE_cartoon_gray_norm(i,j) =
table_RMSE_cartoon_gray(i,j) / max_RMSE;  %normalize all gray cartoon RMSE
errors
                    table_RMSE_Peli_norm(i,j) = table_RMSE_Peli(i,j) /
max_RMSE;  %normalize all Peli cartoon images RMSE errors
                end
            end
            % finally, normalize RMSE errors for the growing dot table
            for i = 1 : m   % i.e. across all the images
                for j = 2 : size(table_RMSE_grow,2)  % across all columns
except first
                    table_RMSE_grow_norm(i,j) = table_RMSE_grow(i,j) /
max_RMSE;  %normalize all grow RMSE errors
                end
            end
            % set first column values in each table to the corresponding
image number
            table_RMSE_cartoon_norm(:, 1) = 1:m; % since column 1 is still
all zeros
            table_RMSE_cartoon_gray_norm(:, 1) = 1:m; % since column 1 is
still all zeros
            table_RMSE_Peli_norm(:, 1) = 1:m; % since column 1 is still all
zeros
            table_RMSE_convex_norm(:, 1) = 1:m; % since column 1 is still all
zeros
            table_RMSE_grow_norm(:, 1) = 1:m; % since column 1 is still all
zeros

            % display the RMSE table of results
            fprintf('Combined, Normalized RMSE values from tables shown
above\n');
            for i = 1 : m % i.e. for every image

fprintf('%i\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f
\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n',
...
                    i, table_RMSE_cartoon_norm(i,2),
table_RMSE_cartoon_norm(i,3), table_RMSE_cartoon_norm(i,4), ...
                    table_RMSE_cartoon_norm(i,5),
table_RMSE_cartoon_norm(i,6), table_RMSE_cartoon_norm(i,7), ...
                    table_RMSE_cartoon_gray_norm(i,2),
table_RMSE_cartoon_gray_norm(i,3), ...
                    table_RMSE_cartoon_gray_norm(i,4),
table_RMSE_cartoon_gray_norm(i,5), ...
                    table_RMSE_Peli_norm(i,2), table_RMSE_Peli_norm(i,3), ...
                    table_RMSE_Peli_norm(i,4), table_RMSE_Peli_norm(i,5), ...
                    table_RMSE_convex_norm(i,2), table_RMSE_convex_norm(i,3),
table_RMSE_convex_norm(i,4), ...
                    table_RMSE_convex_norm(i,5), table_RMSE_convex_norm(i,6),
table_RMSE_convex_norm(i,7), ...
                    table_RMSE_grow_norm(i,2), table_RMSE_grow_norm(i,3),
table_RMSE_grow_norm(i,4)')
            end

            %%%%% show UIQI values (columns) for the 1. cartoon (25-50%)
images,
            %%%%% 2. gray cartoon (0, 255, 128, mean) images,
            %%%%% 3. Peli cartoon (0, 255, 128, mean) images, and
            %%%%% 4. convex warped images,
            %%%%% 5. select movie (grow) images
            disp(' ')
            disp(' ')
            disp('2. Universal Image Quality Index (UIQI) values for the
cartoons (25-50%), gray cartoons (0, 255, 128, mean), Peli cartoons (0, 255,
128, mean), convex warped images, and select movie images')

disp('************************************************************************
**************************************************')
            disp(' ')

fprintf('Image#\tcart@25%%\tcart@30%%\tcart35%%\tcart40%%\tcart@45%%\tcart@50
%%\tgBLACK\tgWHITE\tgGRAY\tgMEAN\tP_BLACK\tP_WHITE\tP_GRAY\tP_MEAN\tconv@1.15
,.4\tconv@1.15,.7\tconv@1.15,1\tconv@1.1,.5\tconve@1.15,.5\tconv@1.2,.5\tgrow
_I\tgrow_cart\tgrow_conv\n');
            disp(' ')
            % collect all the UIQI values in one table
            table_UIQI = [  (1:m)'      UIQI_cartoon_25'    UIQI_cartoon_30'
UIQI_cartoon_35'    ...
                UIQI_cartoon_40'    UIQI_cartoon_45'    UIQI_cartoon_50'
...
                UIQI_cartoon_s30_g0',    UIQI_cartoon_s30_g255', ...
                UIQI_cartoon_s30_g128',  UIQI_cartoon_s30_gmean', ...
                UIQI_Peli_g0',    UIQI_Peli_g255', ...
                UIQI_Peli_g128',  UIQI_Peli_gmean', ...
                UIQI_convex_115_04'      UIQI_convex_115_07'
UIQI_convex_115_10'     ...
                UIQI_convex_110_05'      UIQI_convex_115_05'
UIQI_convex_120_05'...
                UIQI_I1_grow'    UIQI_cartoon_grow'    UIQI_convex_grow'];
            % display the UIQI table of results
            for i = 1 : m % i.e. for every image

fprintf('%i\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f
\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n',
...
                    i, table_UIQI(i,2), table_UIQI(i,3), table_UIQI(i,4),...
                    table_UIQI(i,5), table_UIQI(i,6), table_UIQI(i,7),...
                    table_UIQI(i,8), table_UIQI(i,9), table_UIQI(i,10), ...
                    table_UIQI(i,11), table_UIQI(i,12), table_UIQI(i,13),...
                    table_UIQI(i,14), table_UIQI(i,15), table_UIQI(i,16), ...
                    table_UIQI(i,17), table_UIQI(i,18), table_UIQI(i,19), ...
                    table_UIQI(i,20), table_UIQI(i,21), table_UIQI(i,22), ...
                    table_UIQI(i,23), table_UIQI(i,24));
            end

            % save RMSE and UIQI results to .mat file using the following
command:
            % save RMSE_UIQI_results table_RMSE_norm table_UIQI

            %%%%% display the vector carrying the calculated mean values of
each selected image
            fprintf('Image#\tAverage Gray Level\n')
            fprintf('******\t******************\n\n')
            for i = 1 : m % i.e. for every image
                fprintf('%i\t%i\n', i, gray_mean(i))
            end


        case 9, %this kicks the user out of the program
            select = -1;
    end
end
close all
disp('thanks for using Sherbo''s Central Image Magnification and Optimization
Software')
```

2. Below is a listing of all the functions recalled from within the main `remap_main.m` program.

   a. This function is named "`convex.m`" which performs the convex or "fish-eye" image compensation.

```
function I_warp_stand_bicubic_sharp = convex(k, K, Quad1, Quad2, Quad3,
Quad4);

% this function is designed to input convex parameters and 4 quadrants of
% original image, and to return the convex image

%%%%% Warp each quarter of the image, and put them together
%%%%% Quadrant 1: flip up-down, magnify, then flip up-down again
% flip Quadrant one up-down to resemble Quadrant 4
Quad1 = flipud(Quad1);
% perform magnification (recall function Americjkx)
Quad1 = Amerijckx(Quad1, k, K);
% flip Quadrant 1 back to first quadrant
Quad1 = flipud(Quad1);


%%%%% Quadrant 2: flip up-down, then left-right; magnify, ...
% then flip up-down, then left-right back
% flip up-down, then left-right (onto Quad 4)
Quad2 = flipud(fliplr(Quad2));
% recall magnification function
Quad2 = Amerijckx(Quad2, k, K);
% then flip up-down, then left-right back
Quad2 = flipud(fliplr(Quad2));


%%%%% Quadrant 3: flip left-right; magnify, the flip back to left-right
% flip left-right (onto Quad 4)
Quad3 = fliplr(Quad3);
% recall magnification function
Quad3 = Amerijckx(Quad3, k, K);
% then flip up-down, then left-right back
Quad3 = fliplr(Quad3);

%%%%% Quadrant 4: magnify (no flipping required)
% recall magnification function
Quad4 = Amerijckx(Quad4, k, K);

%%%%% Merge Images and filter: Merge all four quadrants onto one image, then
initial filter and display image
% merge all four warped images
% then filter by removing 'holes' created by forward mapping at the corners
%    (creating 'zero' pixel regions); this involves removing a small pixel
%    bounday from each quadrant
I_warp = [Quad2(2:end-5, 2:end-5) Quad1(2:end-5, 2:end-5) ; Quad3(2:end-5,
2:end-5) Quad4(2:end-5, 2:end-5)];
```

```
%%%%% Image size Standardization: Resize the warped image to the standard
600*800 using the bicubic method
%    (to minimize distrotions as opposed to the linear default method)
%I_warp_stand_nearest = imresize(I_warp, [600 800]); % included for
comparison
%I_warp_stand_bilinear = imresize(I_warp, [600 800], 'bilinear'); % included
for comparison
I_warp_stand_bicubic = imresize(I_warp, [600 800], 'bicubic'); % adopted
resize method

%%%%% Image Filtering (Secondary): filter the image to increase sharpness
h = fspecial('unsharp'); %this is the special sharpness kernel (Matlab)
I_warp_stand_bicubic_sharp = imfilter(I_warp_stand_bicubic,h);

% the convex warped image (I_warp_stand_bicubic_sharp) is now returned to
% the main program
```

b. This sub-function (or nested function) is named “`Amerijckx.m`” after the authors who developed the convex equation into Cartesian coordinates.

```
% the purpose of this function is to perform central magnification
% using the central magnification scheme proposed by Juday-Loshin and
% Americjkx
function quad = Amerijckx(quad, k, K)

% first determine and assign the dimensions of the quadrant
Quad_size = size(quad);
height_Q1 = Quad_size(1);
width_Q1 = Quad_size(2);

% perform quadrant magnification
R = [];
%K = 0.4;  %0.4;
%k = 1.15;
for i = 1: height_Q1
    for j = 1: width_Q1
        if (i==1 & j==1)
            C=1;
        else
            C= (((i)^2+(j)^2)^((1-k)/k))/(K^(1/k)); % magn. eqn.

        end

        I=(ceil(C*i)+1);
        J=(ceil(C*j)+1);
        R(I,J)=quad(i,j);
    end
end

% return the deformed quadrant
quad = R;
```

c. The next function is used to generate the partial cartoon superimposition technique for various cartoon sizes (black cartoon): “`cartoon.m`”

```
function I_trunc_Peli = cartoon(cartoon_size, I1, I_trunc);

%%%%% first determine size of columns to left of truncated image
cols_surround = fix((size(I1, 2) - size(I_trunc, 2) + 1)/2);

%%%%% determine only the HORIZONTAL area surrounding left and right
(invisible) portions of the image
rows_top = fix((size(I1, 1) - size(I_trunc, 1) + 1)/2); % determine size of
row above/below truncated image
% determine area to the left of the truncated area
I1_left = I1(rows_top+1: rows_top + size(I_trunc,1), 1:cols_surround);
% determine area to the right of the truncated area
I1_right = I1(rows_top+1: rows_top + size(I_trunc,1), end-
cols_surround+1:end);
    % delete: show left and right areas
    % figure, imshow(uint8(I1_left)), figure, imshow(uint8(I1_right))

    % delete: make sure all areas can be reassembled back to original trunc
area
    % I1_putback = [I1_left I_trunc I1_right];
    % figure, imshow(uint8(I1_putback))

%%%%% resize the left area to a defined percentage of the visible left area
(chosen here as 30%)
Peli_left = imresize(I1_left, [size(I1_left,1) fix(cartoon_size *
size(I_trunc, 2))], 'bicubic');
% Apply the thresholding/edge filtering technique (using 'Sobel' method)
WEIGHT = edge(Peli_left,'sobel');
se1 = strel('disk',0);
se2 = strel('line',6, 45); % structural element construction: to edit the
appearance of the filtered image
%(i.e. in terms of the weight and thickness of the line, etc.); for more
thickness increase the 6 to 7, 8, etc.
WEIGHT = ~imdilate(WEIGHT,[se1 se2]); %dilates the image according to the
these constructed paramters
    % delete: display weighted/thresholded image
    % figure; imshow(WEIGHT); title('Weight Array');
    % assigning weighted image as Peli_left_edge
Peli_left_edge = WEIGHT;

%%%%% Repeat the above procedure to resize the right area to a defined
percentage of the visible right area
Peli_right = imresize(I1_right, [size(I1_left,1) fix(cartoon_size *
size(I_trunc, 2))], 'bicubic');
WEIGHT = edge(Peli_right,'sobel');
se1 = strel('disk',0);
se2 = strel('line',6, 45); % structural element construction: to edit the
appearance of the filtered image
%(i.e. in terms of the weight and thickness of the line, etc.); for more
thickness increase the 6 to 7, 8, etc.
```

```
WEIGHT = ~imdilate(WEIGHT,[se1 se2]); %dilates the image according to the
these constructed paramters
    % delete: display weighted/thresholded image
    % figure; imshow(WEIGHT); title('Weight Array');
Peli_right_edge = WEIGHT;

%%%%% Make a copy of the truncated image (where the Peli image is to be
shown)
I_trunc_Peli = I_trunc;
% Overlay the left edge filter image over the truncated image
for i = 1 : size(Peli_left_edge,1)
    for j = 1 : size(Peli_left_edge,2)
        if Peli_left_edge(i,j) == 0
            I_trunc_Peli(i,j) = 0;
        end
    end
end
    % delete: show overlay of left portion on turncated image
    % figure, imshow(uint8(I_trunc_Peli))
    % Make a copy of the equivalent edge portion of the truncated image
I_trunc_Peli_right = I_trunc_Peli(:, size(I_trunc, 2) -
size(Peli_left_edge,2) + 1 : size(I_trunc, 2));
% repeat process for overlay of the right edge filter image over the
truncated image
for i = 1:size(Peli_right_edge,1)
    for j = 1 : size(Peli_right_edge,2)
        if Peli_right_edge(i,j) == 0
            I_trunc_Peli_right(i,j) = 0;
        end
    end
end

%%%%% finally combine the right edge-filtered portion to the truncated image
I_trunc_Peli(:, size(I_trunc, 2) - size(Peli_left_edge,2) + 1 : size(I_trunc,
2)) = I_trunc_Peli_right;

% now the cartoon-superimposed image with the specified cartoon parameter
will be returned to the main program
```

d. The next function is used to generate the partial cartoon superimposition technique for various gray shades (30% superimposition): “`cartoon_gray.m`”

```
function I_trunc_Peli = cartoon_gray(gray, I1, I_trunc);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% aim of this function is to produce cartoon superimposition within
% truncated image, having various levels of gray (at fixed superimposition
% percentage (set here to 30%)
% inputs to the function are: 1. levels of gray:
% either black (0), white (255), gray (128), or mean (image average)
% 2. current image, and 3. 600*800 truncated image at center of image
% output from the function is the truncated image with the superimposed
cartoon
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%% first determine size of columns to left of truncated image
cols_surround = fix((size(I1, 2) - size(I_trunc, 2) + 1)/2);

%%%%% determine only the HORIZONTAL area surrounding left and right
(invisible) portions of the image
rows_top = fix((size(I1, 1) - size(I_trunc, 1) + 1)/2); % determine size of
row above/below truncated image
% determine area to the left of the truncated area
I1_left = I1(rows_top+1: rows_top + size(I_trunc,1), 1:cols_surround);
% determine area to the right of the truncated area
I1_right = I1(rows_top+1: rows_top + size(I_trunc,1), end-
cols_surround+1:end);
    % delete: show left and right areas
    % figure, imshow(uint8(I1_left)), figure, imshow(uint8(I1_right))

    % delete: make sure all areas can be reassembled back to original trunc
area
    % I1_putback = [I1_left I_trunc I1_right];
    % figure, imshow(uint8(I1_putback))

%%%%% resize the left area to a defined percentage of the visible left area
(chosen here as 30%)
cartoon_size = 0.30;    %set cartoon size here to 30%
Peli_left = imresize(I1_left, [size(I1_left,1) fix(cartoon_size *
size(I_trunc, 2))], 'bicubic');
% Apply the thresholding/edge filtering technique (using 'Sobel' method)
WEIGHT = edge(Peli_left,'sobel');
se1 = strel('disk',0);
se2 = strel('line',6, 45); % structural element construction: to edit the
appearance of the filtered image
%(i.e. in terms of the weight and thickness of the line, etc.); for more
thickness increase the 6 to 7, 8, etc.
WEIGHT = ~imdilate(WEIGHT,[se1 se2]); %dilates the image according to the
these constructed paramters
    % delete: display weighted/thresholded image
    % figure; imshow(WEIGHT); title('Weight Array');
    % assigning weighted image as Peli_left_edge
```

```
Peli_left_edge = WEIGHT;

%%%%% Repeat the above procedure to resize the right area to a defined
percentage of the visible right area
Peli_right = imresize(I1_right, [size(I1_left,1) fix(cartoon_size *
size(I_trunc, 2))], 'bicubic');
WEIGHT = edge(Peli_right,'sobel');
se1 = strel('disk',0);
se2 = strel('line',6, 45); % structural element construction: to edit the
appearance of the filtered image
%(i.e. in terms of the weight and thickness of the line, etc.); for more
thickness increase the 6 to 7, 8, etc.
WEIGHT = ~imdilate(WEIGHT,[se1 se2]); %dilates the image according to the
these constructed paramters
    % delete: display weighted/thresholded image
    % figure; imshow(WEIGHT); title('Weight Array');
Peli_right_edge = WEIGHT;

%%%%% Make a copy of the truncated image (where the Peli image is to be
shown)
I_trunc_Peli = I_trunc;
% Overlay the left edge filter image over the truncated image
for i = 1 : size(Peli_left_edge,1)
    for j = 1 : size(Peli_left_edge,2)
        if Peli_left_edge(i,j) == 0
            I_trunc_Peli(i,j) = gray;
        end
    end
end
    % delete: show overlay of left portion on turncated image
    % figure, imshow(uint8(I_trunc_Peli))
    % Make a copy of the equivalent edge portion of the truncated image
I_trunc_Peli_right = I_trunc_Peli(:, size(I_trunc, 2) -
size(Peli_left_edge,2) + 1 : size(I_trunc, 2));
% repeat process for overlay of the right edge filter image over the
truncated image
for i = 1:size(Peli_right_edge,1)
    for j = 1 : size(Peli_right_edge,2)
        if Peli_right_edge(i,j) == 0
            I_trunc_Peli_right(i,j) = gray;
        end
    end
end

%%%%% finally combine the right edge-filtered portion to the truncated image
I_trunc_Peli(:, size(I_trunc, 2) - size(Peli_left_edge,2) + 1 : size(I_trunc,
2)) = I_trunc_Peli_right;

% now the cartoon-superimposed image with the specified cartoon parameter
will be returned to the main program
```

e. The next function is used to generate the “Peli” cartoon superimposition technique for various gray shades: “`cartoon_Peli.m`”

```
function I_trunc_Peli = cartoon_gray(gray, I1, I_trunc);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% aim of this function is to produce cartoon superimposition within
% truncated image, having various levels of gray,
% most importantly accroding to the Peli (complete) superimposition
% standards (where entire image is overlayed onto the truncated image)
% inputs to the function are: 1. levels of gray:
% either black (0), white (255), gray (128), or mean (image average)
% 2. current image, and 3. 600*800 truncated image at center of image
% output from the function is the truncated image with the superimposed
cartoon
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%% first resize the original image to the size of tghe truncated image
% make a copy of the original image
I1_resize = I1;
% resize the image exactly to the row and column measures of the truncated
image
I1_resize = imresize(I1_resize, [size(I_trunc, 1) size(I_trunc, 2)],
'bicubic');

%%%%%% Apply the thresholding/edge filtering technique (using 'Sobel' method)
WEIGHT = edge(I1_resize,'sobel');
se1 = strel('disk',0);
se2 = strel('line',6, 45); % structural element construction: to edit the
appearance of the filtered image
%(i.e. in terms of the weight and thickness of the line, etc.); for more
thickness increase the 6 to 7, 8, etc.
WEIGHT = ~imdilate(WEIGHT,[se1 se2]); %dilates the image according to the
these constructed paramters
    % delete: display weighted/thresholded image
    % figure; imshow(WEIGHT); title('Weight Array');
    % assigning weighted image as Peli_left_edge
I1_cartoon = WEIGHT; %assign the cartoon to a new image

%%%%% Make a copy of the truncated image (where the Peli image is to be
shown)
I_trunc_Peli = I_trunc;
% Overlay the left edge filter image over the truncated image
for i = 1 : size(I1_cartoon,1)
    for j = 1 : size(I1_cartoon,2)
        if I1_cartoon(i,j) == 0
            I_trunc_Peli(i,j) = gray;
        end
    end
end

% now the complete cartoon-superimposed (Peli) image with the specified gray
shade will be returned to the main program
```

f. The next function is used to generate the moving dot (dynamic) technique: “moving_dot.m”

```
% routine to produce moving point across the image
function I_move_point = moving_dot(I)

% define the target image and coordinates of moving point
i = 100; % row 100 will be used as the row in which the point moves
n = 200; % the point will move n number of times
I_move_point = I; % defines target image and makes a copy into a blank image
imshow(uint8(I)) % first display motion image before processing
for j = 1:n
    I_move_point = I; %make a new copy of the target image (to delete
previous point)
    I_move_point(i, j) = 0; % set new point to zero, and ...
    I_move_point(i+1, j) = 0; % set the surrounding points (right, below,
diagonal right)...
    I_move_point(i, j+1) = 0; % also to zero (for larger size moving point)
    I_move_point(i+1, j+1) = 0;
    imshow(uint8(I_move_point)) %display the current image
    F(j) = getframe(gcf); %capture the frame into array F
end
```

g. The next function is used to generate the growing area (dynamic) technique: “growing_dot.m”

```
% routine to produce growing area in the image
function I_grow_point = growing_dot(I)

% define the target image and coordinates of where area grows
orig_grow_i  = 100; orig_grow_j = 350; % location where dot starts to grow
n = 100; % the number of times the area will grow
I_grow_point = I; % defines target image and makes a copy into a blank image
imshow(uint8(I)) % first display motion image before processing
for times = 0:n %starts from zero since it starts first with only one point
    I_grow_point = I; %make a new copy of the target image (to delete
previous point)
    I_grow_point(orig_grow_i: orig_grow_i + times, orig_grow_j: orig_grow_j +
times) = 0;
        % set origin and neighboring points to zero (grows bigger)
    imshow(uint8(I_grow_point)) %display the current image
    F(times+1) = getframe(gcf); %capture the frame into array F
end
```

h. The next function is used to calculate the RMSE (Root Mean Square Error) between a base image and a warped image: “RMSE.m”

```
function RMSE_I = RMSE(I_warp, I_orig)

%%% this function determines the RMSE between the warped and original image
%%% input to it is warped and original image, returning value of RMSE
% note, make sure incoming image pixel values are of format double

%%% first, check and correct intensities less than zero or greater than 255
for i = 1:600
    for j = 1:800
        if I_warp(i,j) < 0
            I_warp(i,j) = 0;
        end
        if I_orig(i,j) < 0
            I_orig(i,j) = 0;
        end
        if I_warp(i,j) > 255
            I_warp(i,j) = 255;
        end
        if I_orig(i,j) > 255
            I_orig(i,j) = 255;
        end
    end
end

%%% calculate the RMSE (Root-Mean-Square Error)
% calculate the sum of the squares of the errors
RMSE_sum = 0; % reset the sum first to zero
for i = 1:600
    for j = 1:800
        RMSE_sum = RMSE_sum + (I_warp(i,j) - I_orig(i,j))^2;
    end
end

% now complete calculations: divide sum by # of pixels and take square root
RMSE_I = sqrt(RMSE_sum / (600*800));

%%% RMSE value is now returned and added to the RMSE vector for this type of
warp
```

i. The next function is used to calculate the UIQI (Universal Image Quality Index) between a base image and a warped image: “img_qi.m.” Note, this function has been adapted with copyright permission (as listed below) from the original authors.

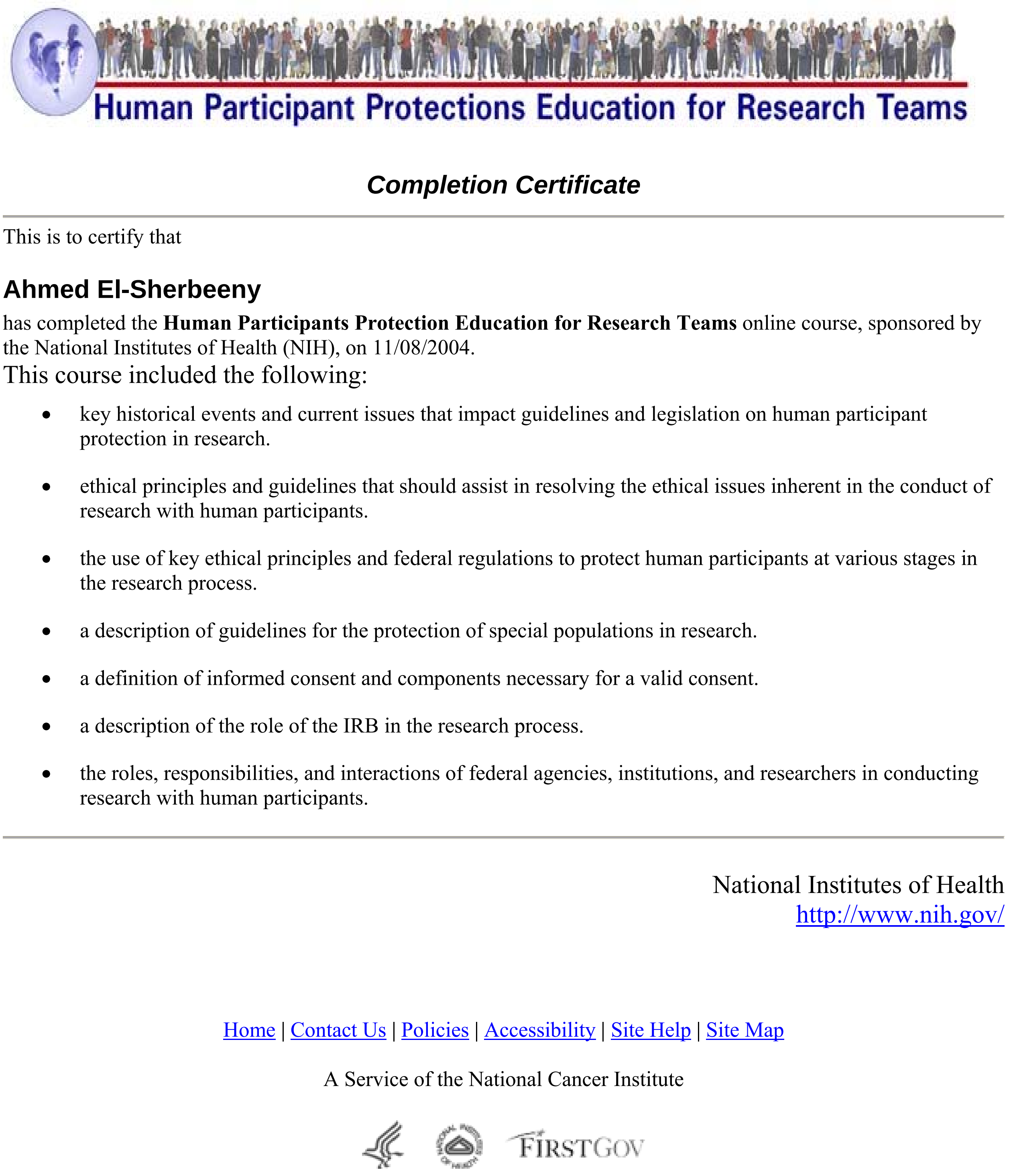
```
function [quality, quality_map] = img_qi(img1, img2, block_size)

%========================================================================
%
%Copyright (c) 2001 The University of Texas at Austin
%All Rights Reserved.
%
%This program is free software; you can redistribute it and/or modify
%it under the terms of the GNU General Public License as published by
%the Free Software Foundation; either version 2 of the License, or
%(at your option) any later version.
%
%This program is distributed in the hope that it will be useful,
%but WITHOUT ANY WARRANTY; without even the implied warranty of
%MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
%GNU General Public License for more details.
%
%The GNU Public License is available in the file LICENSE, or you
%can write to the Free Software Foundation, Inc., 59 Temple Place -
%Suite 330, Boston, MA 02111-1307, USA, or you can find it on the
%World Wide Web at http://www.fsf.org.
%
%Author  : Zhou Wang
%Version : 1.0
%
%The authors are with the Laboratory for Image and Video Engineering
%(LIVE), Department of Electrical and Computer Engineering, The
%University of Texas at Austin, Austin, TX.
%
%Kindly report any suggestions or corrections to zwang@ece.utexas.edu
%
%Acknowledgement:
%The author would like to thank Mr. Umesh Rajashekar, the Matlab master
%in our lab, for spending his precious time and giving his kind help
%on writing this program. Without his help, this program would not
%achieve its current efficiency.
%
%========================================================================
%
%This is an efficient implementation of the algorithm for calculating
%the universal image quality index proposed by Zhou Wang and Alan C.
%Bovik. Please refer to the paper "A Universal Image Quality Index"
%by Zhou Wang and Alan C. Bovik, published in IEEE Signal Processing
%Letters, 2001. In order to run this function, you must have Matlab's
%Image Processing Toobox.
%
%Input : an original image and a test image of the same size
%Output: (1) an overall quality index of the test image, with a value
%            range of [-1, 1].
%        (2) a quality map of the test image. The map has a smaller
%            size than the input images. The actual size is
%            img_size - BLOCK_SIZE + 1.
%
%Usage:
%
%1. Load the original and the test images into two matrices
%   (say img1 and img2)
%
%2. Run this function in one of the two ways:
%
%   % Choice 1 (suggested):
%   [qi qi_map] = img_qi(img1, img2);
%
%   % Choice 2:
%   [qi qi_map] = img_qi(img1, img2, BLOCK_SIZE);
%
%   The default BLOCK_SIZE is 8 (Choice 1). Otherwise, you can specify
%   it by yourself (Choice 2).
%
%3. See the results:
%
%   qi                    %Gives the over quality index.
%   imshow((qi_map+1)/2)  %Shows the quality map as an image.
%
%========================================================================

if (nargin == 1 | nargin > 3)
   quality = -Inf;
   quality_map = -1*ones(size(img1));
   return;
end

if (size(img1) ~= size(img2))
   quality = -Inf;
   quality_map = -1*ones(size(img1));
   return;
end

if (nargin == 2)
   block_size = 8;
end

N = block_size.^2;
sum2_filter = ones(block_size);

img1_sq   = img1.*img1;
img2_sq   = img2.*img2;
img12 = img1.*img2;

img1_sum   = filter2(sum2_filter, img1, 'valid');
img2_sum   = filter2(sum2_filter, img2, 'valid');
img1_sq_sum = filter2(sum2_filter, img1_sq, 'valid');
img2_sq_sum = filter2(sum2_filter, img2_sq, 'valid');
img12_sum = filter2(sum2_filter, img12, 'valid');
```

```
img12_sum_mul = img1_sum.*img2_sum;
img12_sq_sum_mul = img1_sum.*img1_sum + img2_sum.*img2_sum;
numerator = 4*(N*img12_sum - img12_sum_mul).*img12_sum_mul;
denominator1 = N*(img1_sq_sum + img2_sq_sum) - img12_sq_sum_mul;
denominator = denominator1.*img12_sq_sum_mul;

quality_map = ones(size(denominator));
index = (denominator1 == 0) & (img12_sq_sum_mul ~= 0);
quality_map(index) = 2*img12_sum_mul(index)./img12_sq_sum_mul(index);
index = (denominator ~= 0);
quality_map(index) = numerator(index)./denominator(index);

quality = mean2(quality_map);
```

## *Appendix 2: Subject Testing Certificates*

The following pages show records of completion of research certificates and approvals obtained specifically at the aim of conducting subjective experiments.

1. Completion certificate for: “Human Participants Protection Education for Research Teams” online course obtained from the National Institutes of Health (NIH) on Nov 08, 2004.
2. Approval issued by the West Virginia University Office of Research Compliance for conducting the “Central Vision Optimization Survey,” issued August 4, 2006.
3. Approval issued by the West Virginia University Office of Research Compliance for the “Consent and Information” form collected from each volunteer taking the “Central Vision Optimization Survey” survey, issued August 4, 2006.



Human Participant Protections Education for Research Teams

## *Completion Certificate*

This is to certify that

### Ahmed El-Sherbeeny

has completed the **Human Participants Protection Education for Research Teams** online course, sponsored by the National Institutes of Health (NIH), on 11/08/2004.

This course included the following:

- key historical events and current issues that impact guidelines and legislation on human participant protection in research.
- ethical principles and guidelines that should assist in resolving the ethical issues inherent in the conduct of research with human participants.
- the use of key ethical principles and federal regulations to protect human participants at various stages in the research process.
- a description of guidelines for the protection of special populations in research.
- a definition of informed consent and components necessary for a valid consent.
- a description of the role of the IRB in the research process.
- the roles, responsibilities, and interactions of federal agencies, institutions, and researchers in conducting research with human participants.

National Institutes of Health
http://www.nih.gov/

Home | Contact Us | Policies | Accessibility | Site Help | Site Map

A Service of the National Cancer Institute

FIRSTGOV

WestVirginiaUniversity
Office of Research Compliance

**DATE:** August 4, 2006

This research will be monitored for re-approval annually.
**APPROVAL PERIOD: August 4, 2006 to August 3, 2007**

**IRB #:** 17000

**TO:** Ahmed El-Sherbeeny
James Smith

**TITLE:** Central Vision Optimization Survey

**AGENCY:** N/A

The Institutional Review Board (IRB) for the Protection of Human Research Subjects, has approved the project above. Approval was based on the descriptive material and procedures you submitted for review. Should any changes in your project be necessary, **prior approval must be obtained from the IRB.**

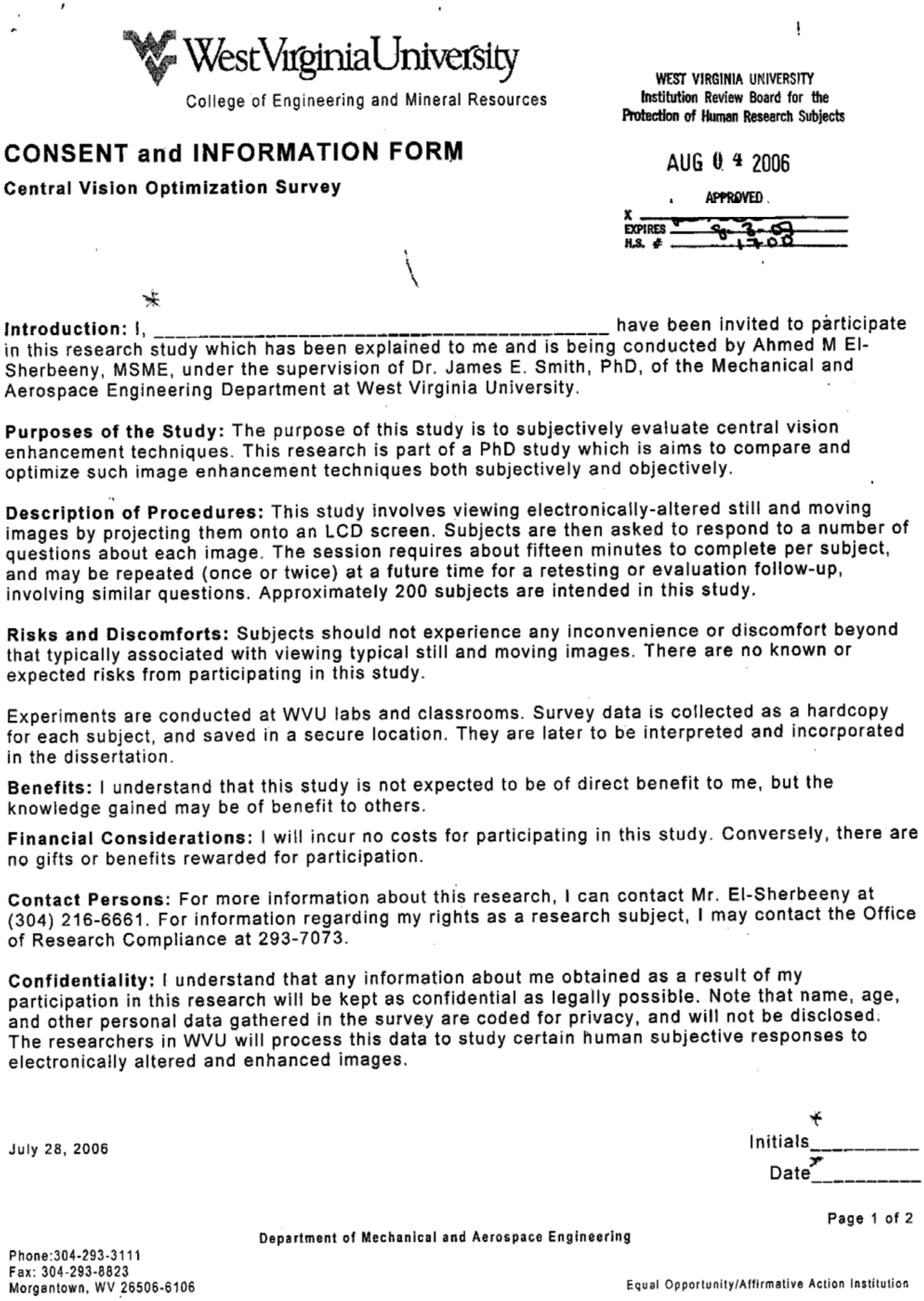
The following have been approved for the above study:

| | | | |
|---|---|---|---|
| Consent Form | _X_Yes __No | Consent Form Waiver | __Yes _X_No |
| Assent Form | ___Yes _X_No | Recruitment Ad | __Yes _X_No |
| PHI Authorization | ___Yes _X_No | PHI Waiver | __Yes _X_No |

All of the above forms must have a current IRB approval stamp. It is the responsibility of the investigator to submit all requests for revisions to the Office of Research Compliance for the IRB's approval.

Lilo A. Ast
Senior Program Coordinator for
Research Compliance

LAA/clg



WestVirginiaUniversity

College of Engineering and Mineral Resources

WEST VIRGINIA UNIVERSITY
Institution Review Board for the
Protection of Human Research Subjects

AUG 04 2006

APPROVED

X ____________
EXPIRES 8-3-07
H.S. # 1700

# CONSENT and INFORMATION FORM

**Central Vision Optimization Survey**

**Introduction:** I, ________________________________________ have been invited to participate in this research study which has been explained to me and is being conducted by Ahmed M El-Sherbeeny, MSME, under the supervision of Dr. James E. Smith, PhD, of the Mechanical and Aerospace Engineering Department at West Virginia University.

**Purposes of the Study:** The purpose of this study is to subjectively evaluate central vision enhancement techniques. This research is part of a PhD study which is aims to compare and optimize such image enhancement techniques both subjectively and objectively.

**Description of Procedures:** This study involves viewing electronically-altered still and moving images by projecting them onto an LCD screen. Subjects are then asked to respond to a number of questions about each image. The session requires about fifteen minutes to complete per subject, and may be repeated (once or twice) at a future time for a retesting or evaluation follow-up, involving similar questions. Approximately 200 subjects are intended in this study.

**Risks and Discomforts:** Subjects should not experience any inconvenience or discomfort beyond that typically associated with viewing typical still and moving images. There are no known or expected risks from participating in this study.

Experiments are conducted at WVU labs and classrooms. Survey data is collected as a hardcopy for each subject, and saved in a secure location. They are later to be interpreted and incorporated in the dissertation.

**Benefits:** I understand that this study is not expected to be of direct benefit to me, but the knowledge gained may be of benefit to others.

**Financial Considerations:** I will incur no costs for participating in this study. Conversely, there are no gifts or benefits rewarded for participation.

**Contact Persons:** For more information about this research, I can contact Mr. El-Sherbeeny at (304) 216-6661. For information regarding my rights as a research subject, I may contact the Office of Research Compliance at 293-7073.

**Confidentiality:** I understand that any information about me obtained as a result of my participation in this research will be kept as confidential as legally possible. Note that name, age, and other personal data gathered in the survey are coded for privacy, and will not be disclosed. The researchers in WVU will process this data to study certain human subjective responses to electronically altered and enhanced images.

July 28, 2006

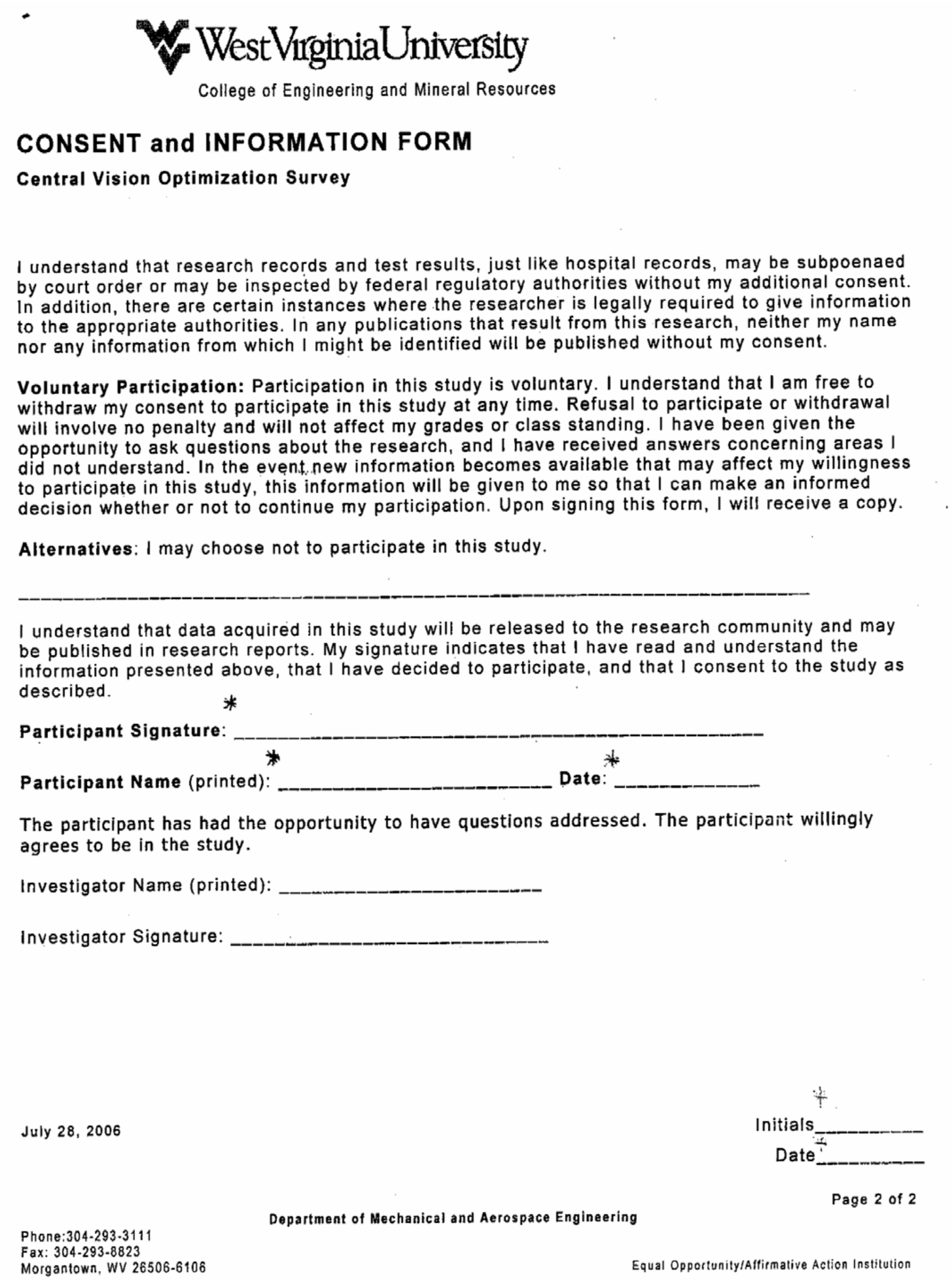
Initials__________
Date__________

Page 1 of 2

Department of Mechanical and Aerospace Engineering

Phone:304-293-3111
Fax: 304-293-8823
Morgantown, WV 26506-6106

Equal Opportunity/Affirmative Action Institution



WestVirginiaUniversity

College of Engineering and Mineral Resources

## CONSENT and INFORMATION FORM

**Central Vision Optimization Survey**

I understand that research records and test results, just like hospital records, may be subpoenaed by court order or may be inspected by federal regulatory authorities without my additional consent. In addition, there are certain instances where the researcher is legally required to give information to the appropriate authorities. In any publications that result from this research, neither my name nor any information from which I might be identified will be published without my consent.

**Voluntary Participation:** Participation in this study is voluntary. I understand that I am free to withdraw my consent to participate in this study at any time. Refusal to participate or withdrawal will involve no penalty and will not affect my grades or class standing. I have been given the opportunity to ask questions about the research, and I have received answers concerning areas I did not understand. In the event new information becomes available that may affect my willingness to participate in this study, this information will be given to me so that I can make an informed decision whether or not to continue my participation. Upon signing this form, I will receive a copy.

**Alternatives**: I may choose not to participate in this study.

I understand that data acquired in this study will be released to the research community and may be published in research reports. My signature indicates that I have read and understand the information presented above, that I have decided to participate, and that I consent to the study as described.

*

**Participant Signature**: ____________________________________________

*                                                                  *

**Participant Name** (printed): _________________________ **Date**: ____________

The participant has had the opportunity to have questions addressed. The participant willingly agrees to be in the study.

Investigator Name (printed): ________________________

Investigator Signature: ____________________________

July 28, 2006

Initials__________

Date__________

Page 2 of 2

Department of Mechanical and Aerospace Engineering

Phone:304-293-3111
Fax: 304-293-8823
Morgantown, WV 26506-6106

Equal Opportunity/Affirmative Action Institution

***Appendix 3: Image Set 1, Base Images***

## Image Set 1 (Base Images; Deep Creek Lake, MD Summer 2006)

a. Landscape

b. Close-Up

c. Text

d. Obstacle

e. Home

## *Appendix 4: Central Image Optimization Survey*

The following pages show a full display of the main “Central Image Optimization Survey” as well as the corresponding images shown during the survey.

1. Complete listing of the approved “Central Image Optimization Survey” (updated Nov 8, 2006). The survey is divided into three sections, “I. Subjects Information,” “II. Image Feedback” and “III. Survey Evaluation” sections.

2. Complete image set 2 used for the conducted (main) survey, including control (base) and –randomly ordered– warped images of the:

   a. Scenery Image

   b. Close-up Image

   c. Text Image

   d. Obstacle Image

   e. House Image

# Central Image Optimization Survey (Updated Nov 8, 2006)
## Conducted by: Ahmed M. EL-Sherbeeny, MSME

**Date:**

Thanks for participating in this survey, which aims to collect information that will be used in a PhD dissertation on optics. The goal is to find out if we can make an image clearer by making its central portion larger than the rest of the image. First, you will be asked some personal information. Then you will be shown five images, each with a different warping scheme, then asked to evaluate each image according to two parameters: how much more of the image has become visible after the warping (as compared to a control image) and the significance of the distortion as a result of this warping.

You will answer a total of about 50 questions, taking up about 15-20 minutes of your time. Note, you may be recalled in the near future for a follow-up survey, to determine the effect of time on your feedback. Please also take the time to help us improve this survey by completing the "Survey Evaluation" section at the end of the survey.

## Part I. Subject Information

Although we honor your privacy, we want to determine some general personal information about you and whether you have taken this experiment before or not.

1. **Name code**. Please provide your initials followed by the last 4 digits of your student number (if applicable): □□□-□□□□
2. Please provide your **age:** □

| | | |
|---|---|---|
| 3. What is your gender? | Male | Female |
| | □ | □ |

## Part II. Image Feedback

On the following pages, you will be shown five images, each of a different scene in black and white. Please look at the image closely for a few seconds. You will then be shown the same image undergoing different "warps" (i.e. or changes) using different techniques; you will be shown each of these warps, and then we would like you to carefully take your time in answering each of the questions that follow. These pertain to two aspects for each of the images: how much more (i.e. detail) of the image can you identify (as compared to the original image), and how much more distorted is this warped image (compared to the original image).

**Image 1: Scenery.** First examine a general landscape image

Look at the following control image, then look at the following images/movies and respond to the following questions (about 10 seconds each). Mark the closest answer with a **check**.

***1.*** *Peli white*

a. How much more **detail** can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more **distorted** is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***2.*** *cartoon mean (@30%)*

a. How much more **detail** can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more **distorted** is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***3.*** *cartoon black (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***4.*** *convex k:1.15 K:1.10*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***5.*** *convex k:1.15 K:0.05*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***6.*** *Peli black*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***Cont. Image 1: Scenery***

***7.*** *cartoon white (at 30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***8.*** *cartoon black (@40%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***9.*** *convex moving dot (k:1.15, K:0.05)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***10.*** *convex growing dot (k:1.15, K:0.05)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***11.*** *cartoon growing dot (@35%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***12.*** *cartoon moving dot (@35%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

**Image 2: Close-up.**

Look at the following control image, then look at the following images/movies and respond to the following questions (about 10 seconds each). Mark the closest answer with a **check**.

***1.*** *cartoon white (at 30%)*

a. How much more **detail** can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more **distorted** is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***2.*** *Peli black*

a. How much more **detail** can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more **distorted** is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***3.*** *convex k:1.15 K:0.05*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***4.*** *convex k:1.15 K:1.10*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***5.*** *cartoon black (@40%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***6.*** *cartoon mean (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***Cont. Image 2: Close-up.***

***7.*** *Peli white*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***8.*** *cartoon black (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

**Image 3: Text.**

Look at the following control image, then look at the following images/movies and respond to the following questions (about 10 seconds each). Mark the closest answer with a **check**.

***1.*** *Peli white*

a. How much more **detail** can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more **distorted** is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***2.*** *cartoon black (@30%)*

a. How much more **detail** can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more **distorted** is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***3.*** *cartoon black (@40%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***4.*** *Peli black*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***5.*** *cartoon mean (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***6.*** *cartoon white (at 30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference Minor Details Some Details Significant Details Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion Minor Distortion Some Distortion Significant Distortion Max Distortion

***Cont. Image 3: Text.***

***7.*** *convex k:1.15 K:1.10*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***8.*** *convex k:1.15 K:0.05*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

**Image 4: Obstacle.**

Look at the following control image, then look at the following images/movies and respond to the following questions (about 10 seconds each). Mark the closest answer with a **check**.

***1.*** *convex k:1.15 K:1.10*

a. How much more **detail** can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more **distorted** is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***2.*** *cartoon black (@30%)*

a. How much more **detail** can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more **distorted** is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***3.*** *convex k:1.15 K:0.05*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***4.*** *Peli white*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***5.*** *cartoon white (at 30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***6.*** *Peli black*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***Cont. Image 4: Obstacle.***

***7.*** *cartoon black (@40%)*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***8.*** *cartoon mean (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

**Image 5: Home.**

Look at the following control image, then look at the following images/movies and respond to the following questions (about 10 seconds each). Mark the closest answer with a **check**.

***1.*** *cartoon black (@40%)*

a. How much more **detail** can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more **distorted** is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***2.*** *cartoon white (at 30%)*

a. How much more **detail** can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more **distorted** is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***3.*** *cartoon mean (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***4.*** *Peli white*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***5.*** *cartoon black (@30%)*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***6.*** *Peli black*

a. How much more detail can you identify in this image (as compared to the control image)?

| | | | | |
|---|---|---|---|---|
| No Difference | Minor Details | Some Details | Significant Details | Max Details |

b. How much more distorted is this image (as compared to the control image) ?

| | | | | |
|---|---|---|---|---|
| No Distortion | Minor Distortion | Some Distortion | Significant Distortion | Max Distortion |

***Cont. Image 5: Home.***

***7.*** *convex k:1.15 K:0.05*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference    Minor Details    Some Details    Significant Details    Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion    Minor Distortion    Some Distortion    Significant Distortion    Max Distortion

***8.*** *convex k:1.15 K:1.10*

a. How much more detail can you identify in this image (as compared to the control image)?

No Difference    Minor Details    Some Details    Significant Details    Max Details

b. How much more distorted is this image (as compared to the control image) ?

No Distortion    Minor Distortion    Some Distortion    Significant Distortion    Max Distortion

## Part III. Survey Evaluation

Your opinion and feedback is very valuable to the success of this research. Please take a few more minutes to tell us what you think of this survey and what should be done to improve the quality and results of this research.

1. What did you **like** most about this survey (if applicable)?

2. What did you **dislike** most about this survey (if applicable)?

3. What change(s) or **improvement**(s) can you suggest for this survey (if applicable)?

Thank you!

# Image Set 2 (Base Images, NOT DISPLAYED), WVU Evansdale Campus (Fall 2006)

a. Landscape

b. Close-Up

c. Text

Original (Undeformed) Image
National Research Center for Coal & Energy

d. Obstacle

e. Home

# Image Set 2:
# Truncated –Control– Images (DISPLAYED)

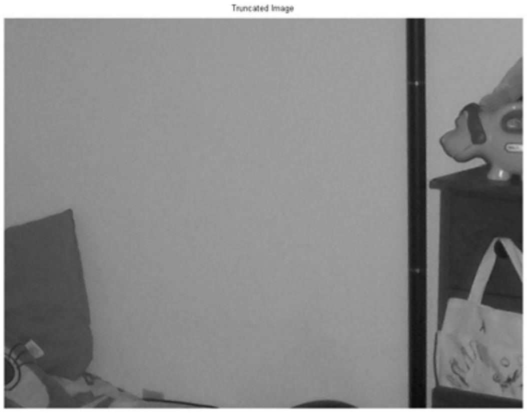
a. Landscape

b. Close-Up

c. Text

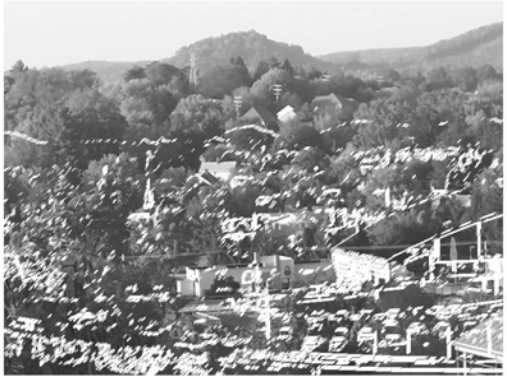
d. Obstacle

e. Home

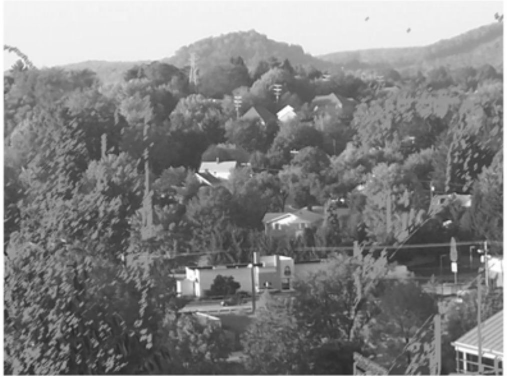


# a. Image 1 (Scenery Image): Warped Images

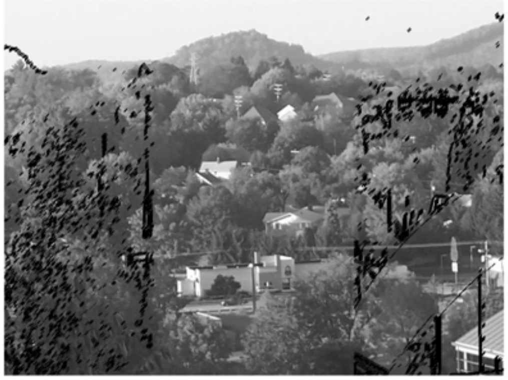
Peli Image: WHITE Cartoon (gray = 255)



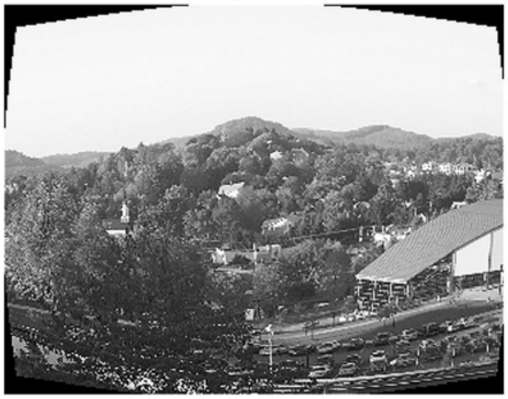
Cartoon Image: MEAN Cartoon (gray = see command window, @30%)



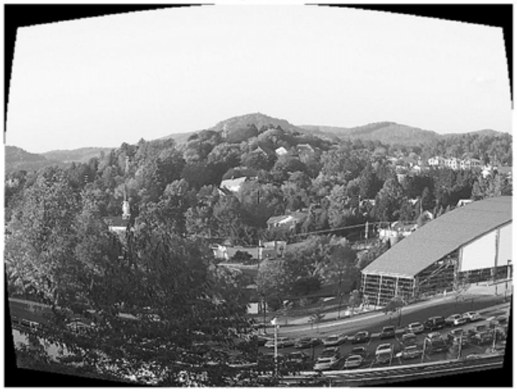
Cartoon Image: at 30% superimposition



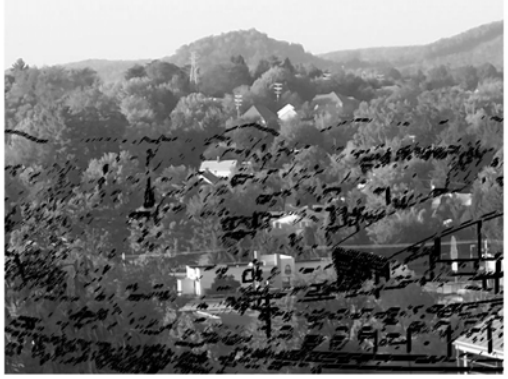
Convex Image: k = 1.15, K = 1.0



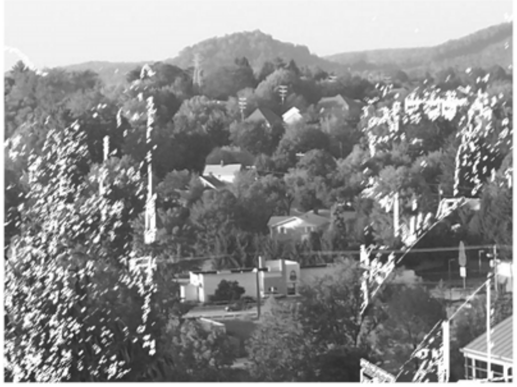
Convex Image: k = 1.15, K = 0.5



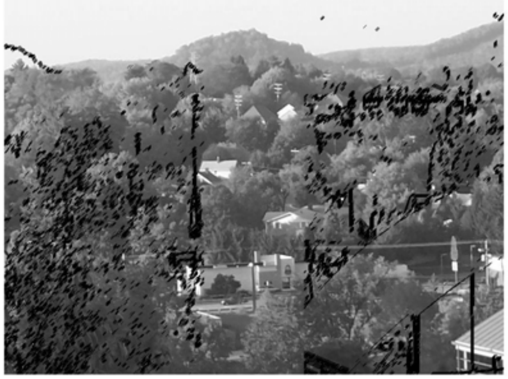
Peli Image: BLACK Cartoon (gray = 0)



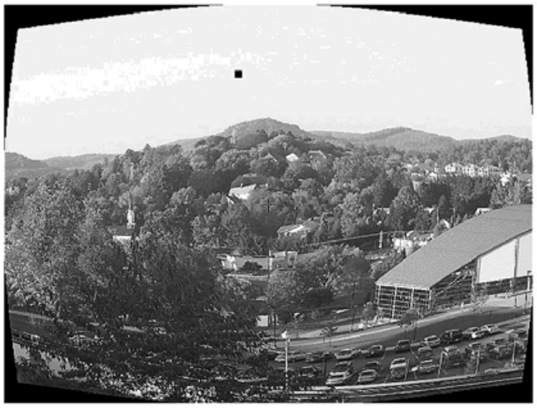
Cartoon Image: WHITE Cartoon (gray = 255, @30%)



Cartoon Image: at 40% superimposition



## a. Image 1 (Scenery Image), Continued: Movie Images (animated .gif files)



# b. Image 2 (Close-Up Image): Warped Images

Cartoon Image: WHITE Cartoon (gray = 255,, @30%)

Peli Image: BLACK Cartoon (gray = 0)

Convex Image: k = 1.15, K = 0.5

Convex Image: k = 1.15, K = 1.0

Cartoon Image: at 40% superimposition

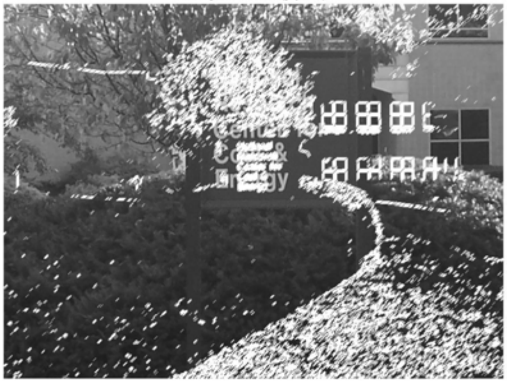
Cartoon Image: MEAN Cartoon (gray = see command window, @30%)

Peli Image: WHITE Cartoon (gray = 255)

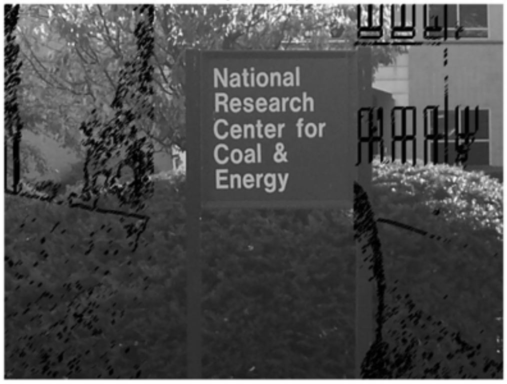
Cartoon Image: at 30% superimposition

# c. Image 3 (Text Image): Warped Images

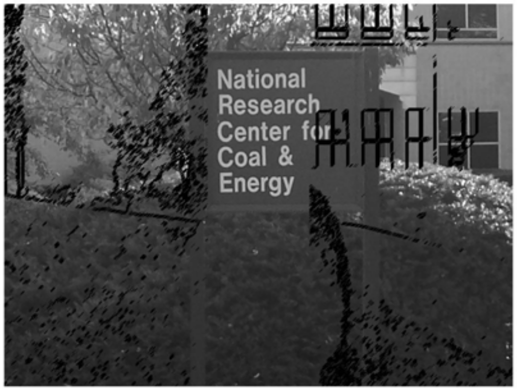
Peli Image: WHITE Cartoon (gray = 255)



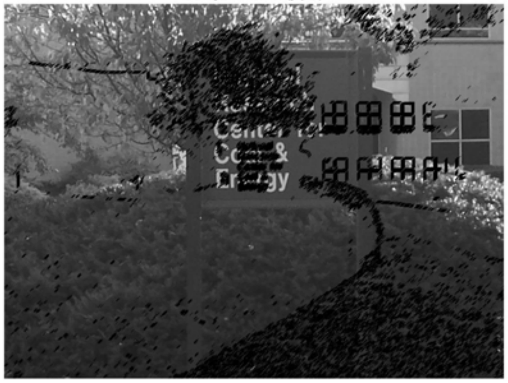
Cartoon Image: at 30% superimposition



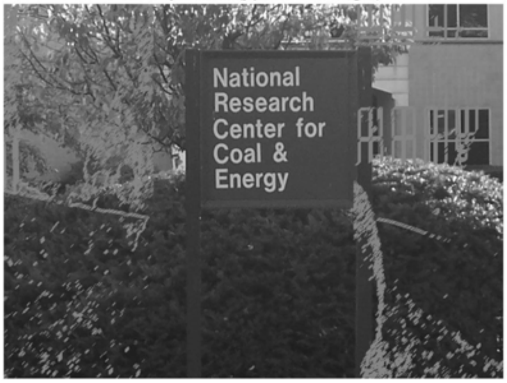
Cartoon Image: at 40% superimposition



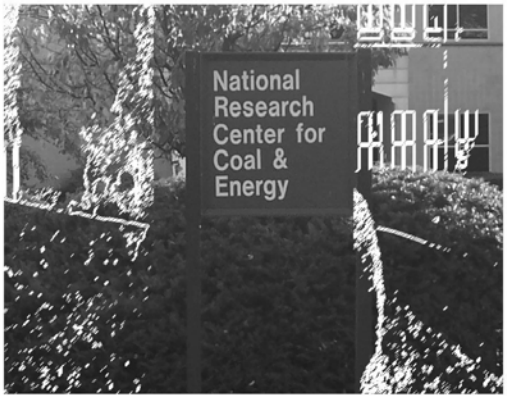
Peli Image: BLACK Cartoon (gray = 0)



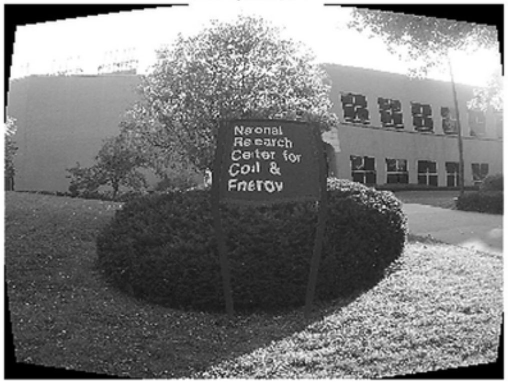
Cartoon Image: MEAN Cartoon (gray = see command window, @30%)



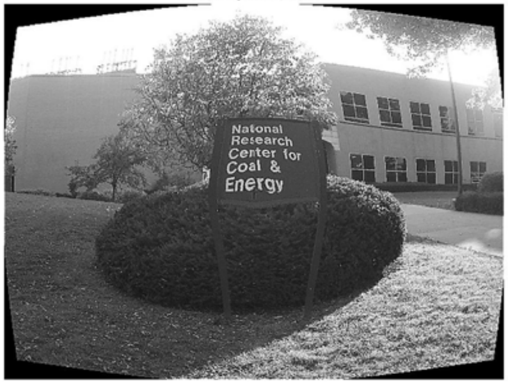
Cartoon Image: WHITE Cartoon (gray = 255, @30%)



Convex Image: k = 1.15, K = 1.0

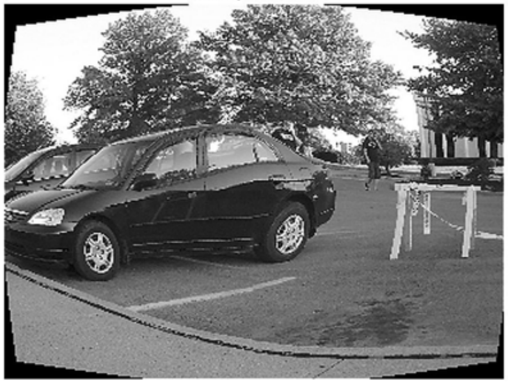


Convex Image: k = 1.15, K = 0.5

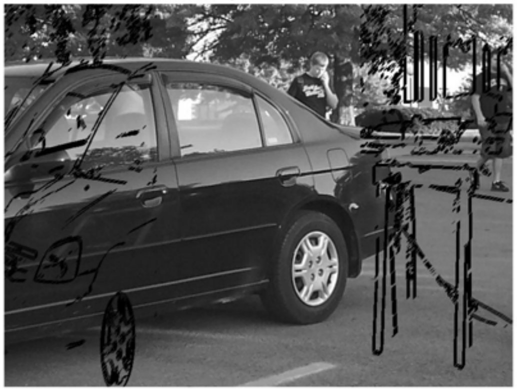


## d. Image 4 (Obstacle Image): Warped Images

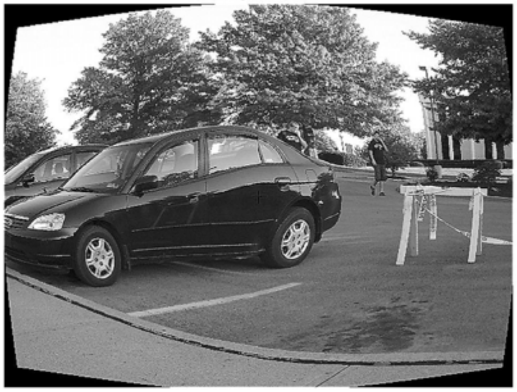
Convex Image: k = 1.15, K = 1.0



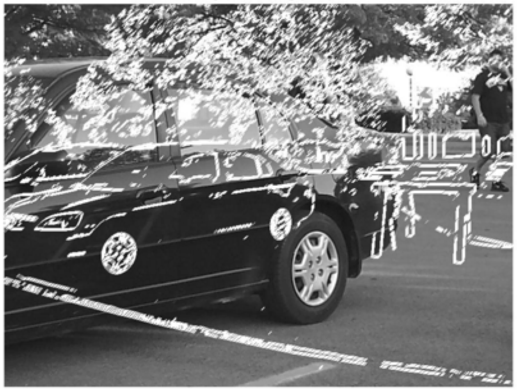
Cartoon Image: at 30% superimposition



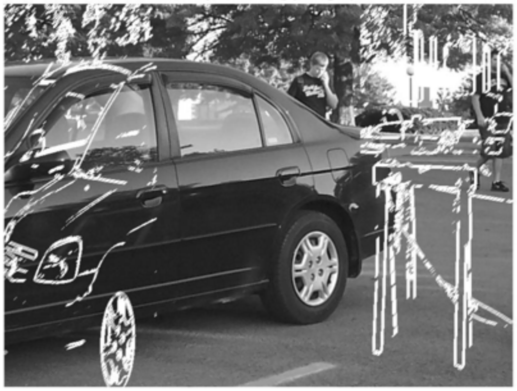
Convex Image: k = 1.15, K = 0.5



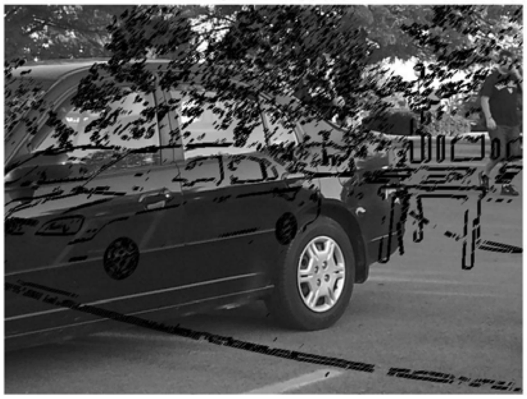
Peli Image: WHITE Cartoon (gray = 255)



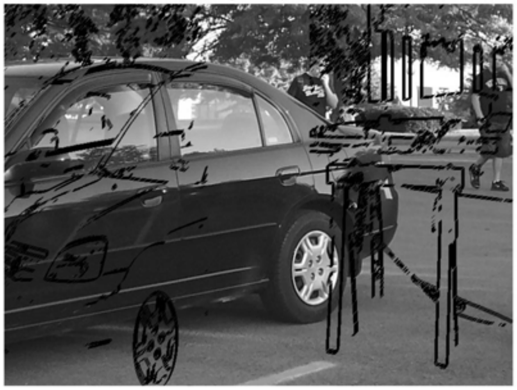
Cartoon Image: WHITE Cartoon (gray = 255, @30%)



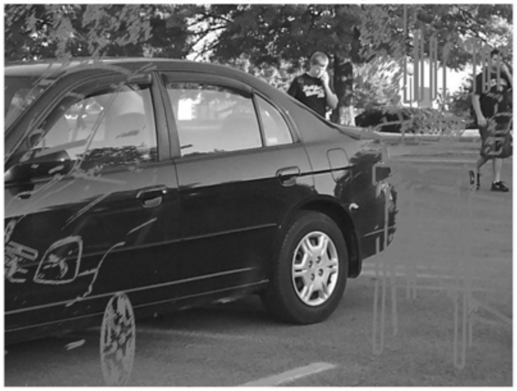
Peli Image: BLACK Cartoon (gray = 0)



Cartoon Image: at 40% superimposition

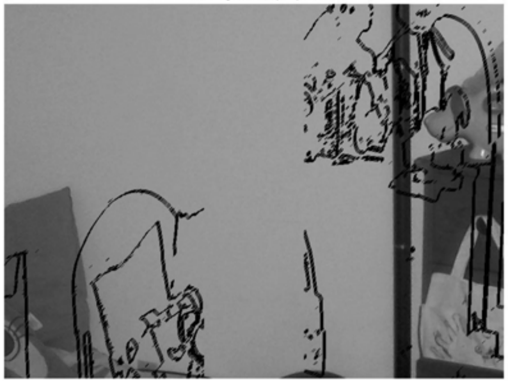


Cartoon Image: MEAN Cartoon (gray = see command window, @30%)

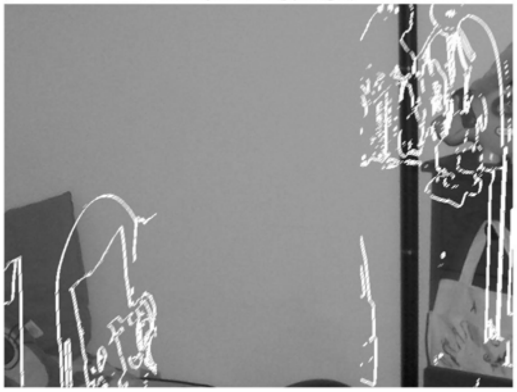


## e. Image 5 (Home Image): Warped Images

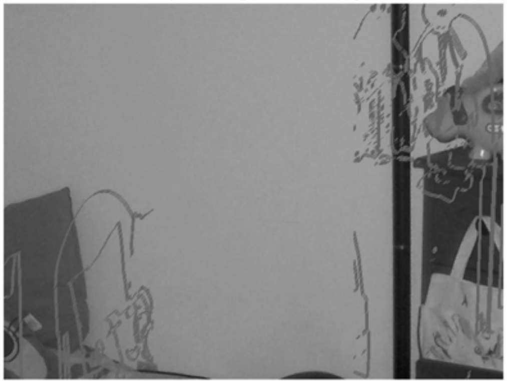
Cartoon Image: at 40% superimposition



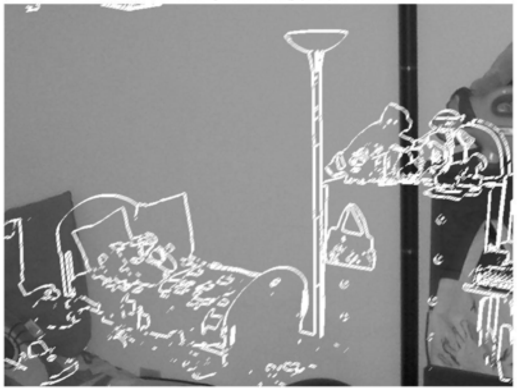
Cartoon Image: WHITE Cartoon (gray = 255, @30%)



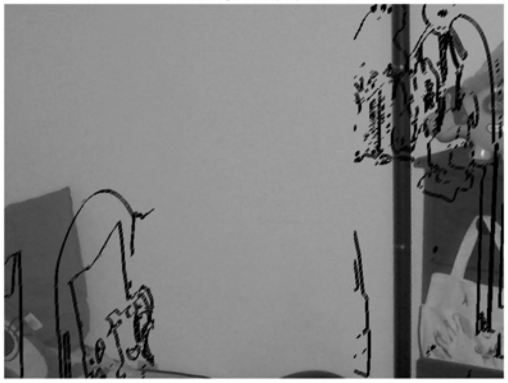
Cartoon Image: MEAN Cartoon (gray = see command window, @30%)



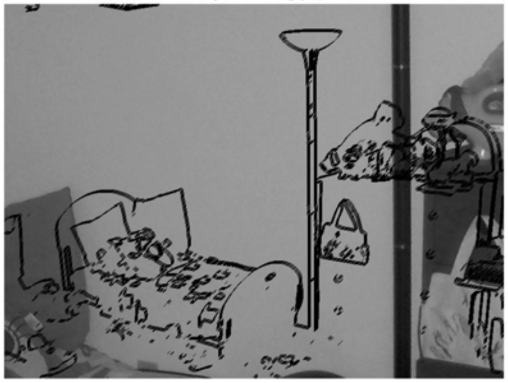
Peli Image: WHITE Cartoon (gray = 255)



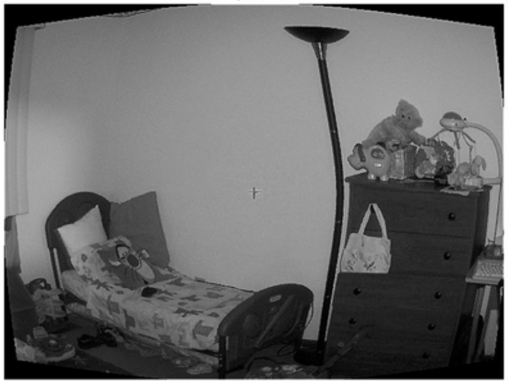
Cartoon Image: at 30% superimposition



Peli Image: BLACK Cartoon (gray = 0)



Convex Image: k = 1.15, K = 0.5



Convex Image: k = 1.15, K = 1.0